ABSTRACT

HOLODNAK, JOHN T. Topics in Randomized Algorithms for Numerical Linear Algebra. (Under the direction of Ilse Ipsen.)

In this dissertation, we present results for three topics in randomized algorithms. Each topic is related to random sampling.

We begin by studying a randomized algorithm for matrix multiplication that randomly samples outer products. We show that if a set of deterministic conditions is satisfied, then the algorithm can compute the exact product. In addition, we show probabilistic bounds on the two norm relative error of the algorithm.

In the second part, we discuss the sensitivity of leverage scores to perturbations. Leverage scores are scalar quantities that give a notion of importance to the rows of a matrix. They are used as sampling probabilities in many randomized algorithms. We show bounds on the difference between the leverage scores of a matrix and a perturbation of the matrix.

In the last part, we approximate functions over an active subspace of parameters. To identify the active subspace, we apply an algorithm that relies on a random sampling scheme. We show bounds on the accuracy of the active subspace identification algorithm and construct an approximation to a function with 3556 parameters using a ten-dimensional active subspace.

Topics in Randomized Algorithms for Numerical Linear Algebra

by
John T. Holodnak

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Applied Mathematics

Raleigh, North Carolina

2015

APPROVED BY:

Petros Drineas	Tim Kelley
Ralph Smith	Ilse Ipsen Chair of Advisory Committee

DEDICATION

Mom and Dad
Andy, Ellen, and Eric
Emma and Sophie

BIOGRAPHY

The author was born in 1987 and grew up in Eagleville, Ohio. He went to elementary and high school in the nearby town of Jefferson. While in school, he played the piano, trumpet, and French horn, and was a member of the high school golf and model United Nations debate teams. He graduated from high school in 2006 and began college at Ohio Northern University, where he majored in mathematics. At Ohio Northern, he developed an appreciation for advanced mathematics and decided to attend graduate school. He was accepted into the graduate mathematics program at North Carolina State University in 2010 and completed his degree in 2015.

At the time of writing, the author continues to enjoy playing the piano, though he no longer likes debate, and holds out hope that a Cleveland sports team will one day win something. He likes reading the same books multiple times, playing strategy-based games, and trying to do yoga.

"It's the job that's never started as takes longest to finish."
J.R.R. Tolkien, *The Lord of the Rings*

ACKNOWLEDGEMENTS

The work contained in this thesis would not have been possible without the support of a large number of people. The most prominent are listed below. Without a doubt, however, some have been unintentionally excluded.

- I thank my advisor, Ilse Ipsen, for the countless hours she spent correcting my disjointed work. Without her constantly pushing me to produce the best work of which I was capable, the quality of this thesis would be greatly diminished.
- I thank my committee members, Ralph, Petros, Tim, and Blanton, all of whom provided advice and encouragement.
- I thank my friends Lydia, Alan, Mike, Corbin, and Anastasia (who are listed in random order) for never letting me take myself too seriously and for tolerating all my peculiarities.
- I thank my friends and officemates, past and present, Dan, Nick, Emily, Shira, Kristina, Mary, and Kayla, for their encouragement and for listening to my endless complaints.
- I thank my undergraduate advisor, Mihai Caragiu, for introducing me to math beyond calculus, making sure I was sent off to conferences and research programs, and having far more confidence in me than I ever had in myself.
- I thank the rest of the Ohio Northern University math and stat faculty, especially
  - Rachel Rader - for listening to endless concerns, worries, and problems, both real and imaginary,
  - Don Hunt - for high expectations and his great sense of humor,
  - Ron Johns - for teaching me numerical mathematics,
  - Harold Putt - for teaching me how to write proofs,
  - Sandy Schroeder - for escorting me (along with Rachel Rader) to and from various conferences, far from rural Ohio.
- I thank Rick Havens for giving me a high school math education worthy of a city much bigger than Jefferson, Ohio.
- Finally, I thank my family, for 27 years of continuous support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Computations on matrices are performed in a wide range of disciplines. For example, numerical analysts may want to solve a system of linear equations or factor a matrix; statisticians may want to compute a covariance matrix (via matrix multiplication) or perform principal component analysis on a dataset; graph theorists may want to compute the eigenvalues and eigenvectors of an adjacency matrix.

If the matrix on which these computations are performed is large (making the computations time consuming), then an approximation to the exact solution may be desirable. Recently, randomized algorithms have been used to approximate the solution to matrix computations including matrix multiplication [36, 86], least squares [43], the column subset selection problem [15], the CUR decomposition [41], and the Nyström approximation [96]. Very broadly, randomized algorithms for matrix computations seek to choose either a small number of columns (rows) or linear combinations of columns (rows) and then perform the computation on the smaller matrix. Random sampling algorithms choose columns (rows) according to some probability distribution while random projection algorithms choose linear combinations of columns (rows) through multiplication by special matrices. For surveys of randomized algorithms in general, see [57] or [78].

In this dissertation, we study three topics that either use or are directly related to random sampling. In the following, we give brief overviews of each topic. More detailed introductions to each topic appear at the beginning of each chapter.

In Chapter 2, we study an algorithm by Drineas, Kannan, and Mahoney [36] that approximates the Gram matrix $\mathbf{A}\mathbf{A}^T$ by randomly sampling outer products. We show deterministic conditions under which $\mathbf{A}\mathbf{A}^T$ can be computed exactly by the approximation algorithm. In particular, we show that for matrices with rank one, it is possible for the algorithm to compute the exact product with just one sample. We also show tighter probabilistic bounds on the relative error in the two-norm. The bounds are tighter than previous results, because they have smaller

constants and do not depend on the size of the matrix $\mathbf{A}$. Experimentally, we investigate two types of sampling probabilities and show that one, based on the columns norms of $\mathbf{A}$, produces smaller average errors for a variety of test matrices. Finally, we use our bounds on matrix multiplication to bound the smallest singular value of a matrix consisting of columns sampled from a matrix with orthonormal rows.

Chapter 3 focuses on the sensitivity of leverage scores to perturbations. Leverage scores are quantities associated with the rows of a matrix and are used as sampling probabilities in randomized algorithms that approximate low rank factorizations [40], CUR decompositions [41], the column subset selection problem [15], Nyström approximations [96], least squares problems [38], and matrix completion [16]. We produce bounds on the relative difference between the leverage scores of a matrix $\mathbf{A}$ and a perturbation $\mathbf{A} + \mathbf{\Delta A}$, when the leverage scores are computed by a QR decomposition. The bounds recognize that individual leverage scores are sensitive to the condition number, the total mass of the perturbation, the perturbation to specific rows, and the magnitude of the leverage score. Experiments show that the bounds capture the qualitative behavior of the actual perturbation errors.

In Chapter 4, we examine an algorithm by Constantine, Dow, and Wang [28] that approximates functions depending on random parameters over an "active subspace" of their parameter space. The construction of the active subspace involves a random sampling scheme that evaluates the gradient of the function at points sampled randomly according to distribution of the random parameters. We provide a tighter bound on the number of samples necessary to accurately approximate the active subspace. In addition, we extend an existing test problem, which defines a function with a one-dimensional active subspace, to create test problems that define functions with active subspaces of any dimension. We describe three criteria for measuring the error of functions constructed over active subspaces. Finally, for a ten-dimensional version of our test problem, we find an approximation to the function and demonstrate, using our error measures, that it approximates the original function with relative accuracy.

Chapter 2

Gram Matrix Approximation

2.1 Introduction

Given a real matrix $\mathbf{A} = \begin{pmatrix} A_1 & \dots & A_n \end{pmatrix}$ with n columns A_j, can one approximate the Gram matrix $\mathbf{A}\mathbf{A}^T$ from just a *few* columns? We answer this question by presenting deterministic conditions for the exact[1] computation of $\mathbf{A}\mathbf{A}^T$ from a few columns, and probabilistic error bounds for approximations.

Our motivation (Section 2.1.1) is followed by an overview of the results (Section 2.1.2), and a literature survey (Section 2.1.3). Those not familiar with established notation can find a review in Section 2.1.4.

2.1.1 Motivation

The objective is the analysis of a randomized algorithm for approximating $\mathbf{A}\mathbf{A}^T$. Specifically, it is a Monte Carlo algorithm for sampling outer products and represents a special case of the ground breaking work on randomized matrix multiplication by Drineas, Kannan, and Mahoney [35, 36].

The basic idea is to represent $\mathbf{A}\mathbf{A}^T$ as a sum of outer products of columns,

$$\mathbf{A}\mathbf{A}^T = A_1 A_1^T + \cdots + A_n A_n^T.$$

The Monte Carlo algorithm [35, 36], when provided with a user-specified positive integer c, samples c columns $A_{t_1}, \ldots, A_{t_c}$ according to probabilities p_j, $1 \le j \le n$, and then approximates $\mathbf{A}\mathbf{A}^T$ by a weighted sum of c outer products

$$\mathbf{X} = w_1 A_{t_1} A_{t_1}^T + \cdots + w_c A_{t_c} A_{t_c}^T.$$

[1] We assume infinite precision, and no round off errors.

The weights are set to $w_j = 1/(cp_{t_j})$ so that $\mathbf{X}$ is an unbiased estimator, $\mathbb{E}[\mathbf{X}] = \mathbf{A}\mathbf{A}^T$. Intuitively, one would expect the algorithm to do well for matrices of low rank.

The intuition is based on the singular value decomposition. Given left singular vectors U_j associated with the $k \equiv \mathsf{rank}(\mathbf{A})$ non-zero singular values σ_j of $\mathbf{A}$, one can represent $\mathbf{A}\mathbf{A}^T$ as a sum of k outer products,

$$\mathbf{A}\mathbf{A}^T = \sigma_1^2 U_1 U_1^T + \cdots + \sigma_k^2 U_k U_k^T.$$

Hence for matrices $\mathbf{A}$ of low rank, a few left singular vectors and singular values suffice to reproduce $\mathbf{A}\mathbf{A}^T$ exactly. Thus, if $\mathbf{A}$ has columns that "resemble" its left singular vectors, the Monte Carlo algorithm should have a chance to perform well.

2.1.2 Contributions and Overview

We sketch the main contributions of this chapter. All proofs are relegated to Section 2.7.

Deterministic conditions for exact computation (Section 2.2)

To calibrate the potential of the Monte-Carlo algorithm [35, 36] and establish connections to existing work in linear algebra, we first derive deterministic conditions that characterize when $\mathbf{A}\mathbf{A}^T$ can be computed *exactly* from a few columns of $\mathbf{A}$. Specifically:

- We present necessary and sufficient conditions (Theorem 2.2) for computing $\mathbf{A}\mathbf{A}^T$ exactly from $c \geq \mathsf{rank}(\mathbf{A})$ columns $A_{t_1}, \ldots, A_{t_c}$ of $\mathbf{A}$,

  $$\mathbf{A}\mathbf{A}^T = w_1 A_{t_1} A_{t_1}^T + \cdots + w_c A_{t_c} A_{t_c}^T.$$

  The conditions and weights w_j depend on the right singular vector matrix $\mathbf{V}$ associated with the non-zero singular values of $\mathbf{A}$.

- For matrices with $\mathsf{rank}(\mathbf{A}) = 1$, this is always possible (Corollary 2.1).

- In the special case where $c = \mathsf{rank}(\mathbf{A})$ (Theorem 2.3), the weights are equal to inverse leverage scores, $w_j = 1/\|\mathbf{V}^T e_{t_j}\|_2^2$. However, they do not necessarily correspond to the largest leverage scores.

Sampling probabilities for the Monte-Carlo algorithm (Section 2.3)

Given an approximation $\mathbf{X}$ from the Monte-Carlo algorithm [35, 36], we are interested in the two-norm relative error due to randomization, $\|\mathbf{X} - \mathbf{A}\mathbf{A}^T\|_2 / \|\mathbf{A}\mathbf{A}^T\|_2$. Numerical experiments compare two types of sampling probabilities:

- "Optimal" probabilities $p_j^{opt} = \|A_j\|_2^2/\|\mathbf{A}\|_F^2$ [36], and
- Leverage score probabilities $p_j^{lev} = \|\mathbf{V}^T e_j\|_2^2/k$ [12, 14].

The experiments illustrate that sampling columns of $\mathbf{X}$ with the "optimal" probabilities produces a smaller error than sampling with leverage score probabilities. This was not obvious a priori, because the "optimal" probabilities are designed to minimize the expected value of the Frobenius norm absolute error, $\mathbb{E}[\|\mathbf{X} - \mathbf{A}\mathbf{A}^T\|_F^2]$. Furthermore, corresponding probabilites p_j^{opt} and p_j^{lev} can differ by orders of magnitude.

For matrices $\mathbf{A}$ of rank one though, we show (Theorem 2.4) that the probabilities are identical, $p_j^{opt} = p_j^{lev}$ for $1 \le j \le n$, and that the Monte Carlo algorithm always produces the exact result, $\mathbf{X} = \mathbf{A}\mathbf{A}^T$, when it samples with these probabilities.

Probabilistic bounds (Sections 2.4 and 2.5)

We present probabilistic bounds for $\|\mathbf{X} - \mathbf{A}\mathbf{A}^T\|_2/\|\mathbf{A}\mathbf{A}^T\|_2$ when the Monte-Carlo algorithm samples with two types of sampling probabilities.

- Sampling with "nearly optimal" probabilities $p_j^{\beta} \ge \beta\, p_j^{opt}$, where $\beta \le 1$ (Theorems 2.5 and 2.6). We show that

$$\|\mathbf{X} - \mathbf{A}\mathbf{A}^T\|_2/\|\mathbf{A}\mathbf{A}^T\|_2 \le \epsilon \quad \text{with probability at least } 1-\delta,$$

  provided the number of sampled columns is at least

$$c \ge c_0(\epsilon)\, \frac{\ln(\rho(\mathbf{A})/\delta)}{\beta\epsilon^2}\mathsf{sr}(\mathbf{A}), \qquad \text{where} \quad 2 \le c_0(\epsilon) \le 2.7.$$

  Here $\rho(\mathbf{A}) = \mathsf{rank}(\mathbf{A})$ or $\rho(\mathbf{A}) = 4\,\mathsf{sr}(\mathbf{A})$, where $\mathsf{sr}(\mathbf{A})$ is the stable rank of $\mathbf{A}$. The bound containing $\mathsf{rank}(\mathbf{A})$ is tighter for matrices with $\mathsf{rank}(\mathbf{A}) \le 4\,\mathsf{sr}(\mathbf{A})$.

  Note that the amount of sampling depends on the rank or the stable rank, but not on the dimensions of $\mathbf{A}$. Numerical experiments (Section 2.4.4) illustrate that the bounds are informative, even for stringent success probabilities and matrices of small dimension.

- Sampling with leverage score probabilities p_j^{lev} (Theorem 2.7). The bound corroborates the numerical experiments in Section 2.3.2, but is not as tight as the bounds for "nearly optimal" probabilities, since it depends only on $\mathsf{rank}(\mathbf{A})$, and $\mathsf{rank}(\mathbf{A}) \ge \mathsf{sr}(\mathbf{A})$.

Singular value bounds (Section 2.6)

Given a $m \times n$ matrix $\mathbf{Q}$ with orthonormal rows, $\mathbf{Q}\mathbf{Q}^T = \mathbf{I}_m$, the Monte-Carlo algorithm computes $\mathbf{QS}$ by sampling $c \ge m$ columns from $\mathbf{Q}$ with the "optimal" probabilities. The

goal is to derive a positive lower bound for the smallest singular value $\sigma_m(\mathbf{QS})$, as well as an upper bound for the two-norm condition number with respect to left inversion $\kappa(\mathbf{QS}) \equiv \sigma_1(\mathbf{QS})/\sigma_m(\mathbf{QS})$.

Surprisingly, Theorem 2.5 leads to bounds (Theorems 2.8 and 2.10) that are not always as tight as the ones below. These bounds are based on a Chernoff inequality and represent a slight improvement over existing results.

- Bound for the smallest singular value (Theorem 2.9). We show that

$$\sigma_m(\mathbf{QS}) \geq \sqrt{1-\epsilon} \quad \text{with probability at least } 1-\delta,$$

  provided the number of sampled columns is at least

$$c \geq c_1(\epsilon)\, m\, \frac{\ln(m/\delta)}{\epsilon^2}, \qquad \text{where} \quad 1 \leq c_1(\epsilon) \leq 2.$$

- Condition number bound (Theorem 2.11). We show that

$$\kappa(\mathbf{QS}) \leq \frac{\sqrt{1+\epsilon}}{\sqrt{1-\epsilon}} \quad \text{with probability at least } 1-\delta,$$

  provided the number of sampled columns is at least

$$c \geq c_2(\epsilon)\, m\, \frac{\ln(2m/\delta)}{\epsilon^2}, \qquad \text{where} \quad 2 \leq c_2(\epsilon) \leq 2.6.$$

In addition, we derive corresponding bounds for uniform sampling with and without replacement (Theorems 2.9 and 2.11).

2.1.3 Literature Review

We review bounds for the relative error due to randomization of general Gram matrix approximations $\mathbf{AA}^T$, and also for the smallest singular value and condition number of sampled matrices $\mathbf{QS}$ when $\mathbf{Q}$ has orthonormal rows.

In addition to [35, 36], several other randomized matrix multiplication algorithms have been proposed [9, 25, 26, 73, 81, 86]. Sarlós's algorithms [86] are based on matrix transformations. Cohen and Lewis [25, 26] approximate large elements of a matrix product with a random walk algorithm. The algorithm by Belabbas and Wolfe [9] is related to the Monte Carlo algorithm [35, 36], but with different sampling methods and weights. A second algorithm by Drineas et al. [36] relies on matrix sparsification, and a third algorithm [35] estimates each matrix element independently. Pagh [81] targets sparse matrices, while Liberty [73] estimates the Gram matrix

Table 2.1: Frobenius-norm error due to randomization: Lower bounds on the number c of sampled columns in $\mathbf{X}$, so that $\|\mathbf{X} - \mathbf{A}\mathbf{A}^T\|_F / \|\mathbf{A}\mathbf{A}^T\|_F \leq \epsilon$ with probability at least $1 - \delta$. The second column specifies the sampling strategy: "opt" for sampling with "optimal" probabilities, and "u-wor" for uniform sampling without replacement. The last two bounds are special cases of bounds for general matrix products $\mathbf{AB}$.

Bound for # samples	Sampling	Reference
$\frac{(1+\sqrt{8\ln(1/\delta)})^2}{\epsilon^2} \frac{\|A\|_F^4}{\|AA^T\|_F^2}$	opt	[36, Theorem 2]
$\frac{1}{\epsilon^2\delta} \frac{\|A\|_F^4}{\|AA^T\|_F^2}$	opt	[46, Lemma 1], [47, Lemma 2]
$\frac{n^2}{(n-1)\delta\epsilon^2} \frac{\sum_{j=1}^n \|A_j\|_2^4}{\|AA^T\|_F^2}$	u-wor	[35, Lemma 7]
$\frac{36n\ln(1/\delta)}{\epsilon^2} \frac{\sum_{j=1}^n \|A_i\|_2^4}{\|AA^T\|_F^2}$	u-wor	[13, Lemma 4.13], [50, Lemma 4.3]

$\mathbf{A}\mathbf{A}^T$ by iteratively removing "unimportant" columns from $\mathbf{A}$.

Eriksson-Bique et al. [44] derive an importance sampling strategy that minimizes the variance of the inner products computed by the Monte Carlo method. Madrid, Guerra, and Rojas [74] present experimental comparisons of different sampling strategies for specific classes of matrices.

Excellent surveys of randomized matrix algorithms in general are given by Halko, Martinsson, and Tropp [57], and by Mahoney [78].

Gram matrix approximations

We review existing bounds for the error due to randomization of the Monte Carlo algorithm [35, 36] for approximating $\mathbf{A}\mathbf{A}^T$, where $\mathbf{A}$ is a real $m \times n$ matrix. Relative error bounds $\|\mathbf{X} - \mathbf{A}\mathbf{A}^T\| / \|\mathbf{A}\mathbf{A}^T\|$ in the Frobenius norm and the two-norm are summarized in Tables 2.1 and 2.2.

Table 2.1 shows probabilistic lower bounds for the number of sampled columns so that the Frobenius norm relative error $\left\|\mathbf{X} - \mathbf{A}\mathbf{A}^T\right\|_F / \left\|\mathbf{A}\mathbf{A}^T\right\|_F \leq \epsilon$. Not listed is a bound for uniform sampling without replacement [71, Corollary 1], because it cannot easily be converted to the format of the other bounds, and a bound for a greedy sampling strategy [9, p. 5].

Table 2.2 shows probabilistic lower bounds for the number of sampled columns so that the two-norm relative error $\left\|\mathbf{X} - \mathbf{A}\mathbf{A}^T\right\|_2 / \left\|\mathbf{A}\mathbf{A}^T\right\|_2 \leq \epsilon$. These bounds imply, roughly, that the number of sampled columns should be at least $\Omega(\mathsf{sr}(\mathbf{A}) \ln(\mathsf{sr}(\mathbf{A}))$ or $\Omega(\mathsf{sr}(\mathbf{A}) \ln(m))$.

Table 2.2: Two-norm error due to randomization, for sampling with "optimal" probabilities: Lower bounds on the number c of sampled columns in $\mathbf{X}$, so that $\|\mathbf{X} - \mathbf{A}\mathbf{A}^T\|_2 / \|\mathbf{A}\mathbf{A}^T\|_2 \leq \epsilon$ with probability at least $1-\delta$ for all bounds but the first. The first bound contains an unspecified constant C and holds with probability at least $1 - 2\exp(\tilde{C}/\delta)$, where $\tilde{C}$ is another unspecified constant (our ϵ corresponds to $\epsilon^2/2$ in [84, Theorem 1.1]). The penultimate bound is a special case of a bound for general matrix products $\mathbf{AB}$, while the last bound applies only to matrices with orthonormal rows.

Bound for # samples	Reference
$C\frac{\mathsf{sr}(A)}{\epsilon^2\delta}\ln(\mathsf{sr}(A)/(\epsilon^2\delta))$	[84, Theorems 1.1 and 3.1, and their proofs]
$\frac{4\mathsf{sr}(A)}{\epsilon^2}\ln(2m/\delta)$	[76, Theorem 17], [75, Theorem 20]
$\frac{96\mathsf{sr}(A)}{\epsilon^2}\ln\left(\frac{96\mathsf{sr}(A)}{\epsilon^2\sqrt{\delta}}\right)$	[43, Theorem 4]
$\frac{20\mathsf{sr}(A)}{\epsilon^2}\ln(16\mathsf{sr}(A)/\delta)$	[77, Theorem 3.1], [103, Theorem 2.1]
$\frac{21(1+\mathsf{sr}(A))}{4\epsilon^2}\ln(4\mathsf{sr}(A)/\delta)$	[65, Example 4.3]
$\frac{8m}{\epsilon^2}\ln(m/\delta)$	[89, Theorem 3.9]

Table 2.3: Smallest singular value of a matrix $\mathbf{QS}$ whose columns are sampled from a $m \times n$ matrix $\mathbf{Q}$ with orthonormal rows: Lower bounds on the number c of sampled columns, so that $\sigma_m(\mathbf{QS}) \geq \sqrt{1-\epsilon}$ with probability at least $1-\delta$. The second column specifies the sampling strategy: "opt" for sampling with "optimal" probabilities, "u-wr" for uniform sampling with replacement, and "u-wor" for uniform sampling without replacement.

Bound for # samples	Sampling	Reference
$\frac{6n\mu}{\epsilon^2}\ln(m/\delta)$	u-wor	[13, Lemma 4.3]
$\frac{4m}{\epsilon^2}\ln(2m/\delta)$	opt	[11, Lemma 13]
$\frac{3n\mu}{\epsilon^2}\ln(m/\delta)$	u-wr, u-wor	[66, Corollary 4.2]
$\frac{8n\mu}{3\epsilon^2}\ln(m/\delta)$	u-wr	[13, Lemma 4.4]
$\frac{2n\mu}{\epsilon^2}\ln(m/\delta)$	u-wor	[49, Lemma 1]

Singular value bounds

We review existing bounds for the smallest singular value of a sampled matrix $\mathbf{QS}$, where $\mathbf{Q}$ is $m \times n$ with orthonormal rows.

Table 2.3 shows probabilistic lower bounds for the number of sampled columns so that the smallest singular value $\sigma_m(\mathbf{QS}) \geq \sqrt{1-\epsilon}$. All bounds but one contain the coherence μ. Not listed is a bound [43, Lemma 4] that requires specific choices of ϵ, δ, and μ.

Condition number bounds

We are aware of only two existing bounds for the two-norm condition number $\kappa(\mathbf{QS})$ of a matrix $\mathbf{QS}$ whose columns are sampled from a $m \times n$ matrix $\mathbf{Q}$ with orthonormal rows. The first bound [2, Theorem 3.2] lacks explicit constants, while the second one [66, Corollary 4.2] applies to uniform sampling with and without replacement. It ensures $\kappa(\mathbf{QS}) \leq \frac{\sqrt{1+\epsilon}}{\sqrt{1-\epsilon}}$ with probability at least $1-\delta$, provided the number of sampled columns in $\mathbf{QS}$ is at least $c \geq 3\, n\mu\, \ln(2m/\delta)/\epsilon^2$.

Relation to subset selection

The Monte Carlo algorithm selects outer products from $\mathbf{AA}^T$, which is equivalent to selecting columns from $\mathbf{A}$, hence it can be viewed as a form of randomized column subset selection.

The traditional deterministic subset selection methods select exactly the required number of columns, by means of rank-revealing QR decompositions or SVDs [18, 51, 52, 56, 63]. In contrast, more recent methods are motivated by applications to graph sparsification [7, 6, 89]. They oversample columns from a matrix $\mathbf{Q}$ with orthonormal rows, by relying on a *barrier sampling* strategy[2]. The accuracy of the selected columns $\mathbf{QS}$ is determined by bounding the *reconstruction error*, which views $(\mathbf{QS})\,(\mathbf{QS})^T$ as an approximation to $\mathbf{QQ}^T = I$ [7, Theorem 3.1], [6, Theorem 3.1], [89, Theorem 3.2].

Boutsidis [11] extends this work to general Gram matrices $\mathbf{AA}^T$. Following [52], he selects columns from the right singular vector matrix $\mathbf{V}^T$ of $\mathbf{A}$, and applies barrier sampling simultaneously to the dominant and subdominant subspaces of $\mathbf{V}^T$.

In terms of randomized algorithms for subset selection, the two-stage algorithm by Boutsidis et al. [14] samples columns in the first stage, and performs a deterministic subset selection on the sampled columns in the second stage. Other approaches include volume sampling [46, 47], and CUR decompositions [42].

[2]The name comes about as follows: Adding a column q to $\mathbf{QS}$ amounts to a rank-one update qq^T for the Gram matrix $(\mathbf{QS})\,(\mathbf{QS})^T$. The eigenvalues of this matrix, due to interlacing, form "barriers" for the eigenvalues of the updated matrix $(\mathbf{QS})\,(\mathbf{QS})^T + qq^T$.

Leverage scores

In the late seventies, statisticians introduced leverage scores for outlier detection in regression problems [23, 60, 100]. More recently, Drineas, Mahoney et al. have pioneered the use of leverage scores for importance sampling in randomized algorithms, such as CUR decompositions [42], least squares problems [39], and column subset selection [14], see also the perspectives on statistical leverage [78, §6]. Fast approximation algorithms are being designed to make the computation of leverage scores more affordable [37, 72, 75].

2.1.4 Notation

All matrices are real. Matrices that can have more than one column are indicated in bold face, and column vectors and scalars in italics. The columns of the $m \times n$ matrix $\mathbf{A}$ are denoted by $\mathbf{A} = \begin{pmatrix} A_1 & \dots & A_n \end{pmatrix}$. The $n \times n$ identity matrix is $\mathbf{I}_n \equiv \begin{pmatrix} e_1 & \dots & e_n \end{pmatrix}$, whose columns are the canonical vectors e_j.

The thin Singular Value Decomposition (SVD) of a $m \times n$ matrix $\mathbf{A}$ with $\mathsf{rank}(\mathbf{A}) = k$ is $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where the $m \times k$ matrix $\mathbf{U}$ and the $n \times k$ matrix $\mathbf{V}$ have orthonormal columns, $\mathbf{U}^T\mathbf{U} = \mathbf{I}_k = \mathbf{V}^T\mathbf{V}$, and the $k \times k$ diagonal matrix of singular values is $\mathbf{\Sigma} = \mathsf{diag}\begin{pmatrix} \sigma_1 & \dots & \sigma_k \end{pmatrix}$, with $\sigma_1 \geq \cdots \geq \sigma_k > 0$. The Moore-Penrose inverse of $\mathbf{A}$ is $\mathbf{A}^\dagger \equiv \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^T$. The unique symmetric positive semi-definite square root of a symmetric positive semi-definite matrix $\mathbf{W}$ is denoted by $\mathbf{W}^{1/2}$.

The norms in this chapter are the two-norm $\|\mathbf{A}\|_2 \equiv \sigma_1$, and the Frobenius norm

$$\|\mathbf{A}\|_F \equiv \sqrt{\sum_{j=1}^{n} \|A_j\|_2^2} = \sqrt{\sigma_1^2 + \cdots + \sigma_k^2}.$$

The *stable rank* of a non-zero matrix $\mathbf{A}$ is $\mathsf{sr}(\mathbf{A}) \equiv \|\mathbf{A}\|_F^2 / \|\mathbf{A}\|_2^2$, where $1 \leq \mathsf{sr}(\mathbf{A}) \leq \mathsf{rank}(\mathbf{A})$.

Given a $m \times n$ matrix $\mathbf{Q} = \begin{pmatrix} Q_1 & \dots & Q_n \end{pmatrix}$ with orthonormal rows, $\mathbf{Q}\mathbf{Q}^T = \mathbf{I}_m$, the *two-norm condition number* with regard to left inversion is $\kappa(\mathbf{Q}) \equiv \sigma_1(\mathbf{Q})/\sigma_m(\mathbf{Q})$; the *leverage scores* [39, 42, 78] are the squared columns norms $\|Q_j\|_2^2$, $1 \leq j \leq m$; and the *coherence* [2, 17] is the largest leverage score,

$$\mu \equiv \max_{1 \leq j \leq m} \|Q_j\|_2^2.$$

The expected value of a scalar or a matrix-valued random random variable $\mathbf{X}$ is $\mathbb{E}[\mathbf{X}]$; and the probability of an event $\mathcal{X}$ is $\mathbb{P}[\mathcal{X}]$.

2.2 Deterministic conditions for exact computation

To gauge the potential of the Monte Carlo algorithm, and to establish a connection to existing work in linear algebra, we first consider the best case: The *exact* computation of $\mathbf{A}\mathbf{A}^T$ from a few columns. That is: Given c not necessarily distinct columns $A_{t_1}, \ldots, A_{t_c}$, under which conditions is $w_1 A_{t_1} A_{t_1}^T + \cdots + w_c A_{t_c} A_{t_c}^T = \mathbf{A}\mathbf{A}^T$?

Since a column can be selected more than once, and therefore the selected columns may not form a submatrix of $\mathbf{A}$, we express the c selected columns as $\mathbf{AS}$, where $\mathbf{S}$ is a $n \times c$ sampling matrix with

$$\mathbf{S} = \begin{pmatrix} e_{t_1} & \ldots & e_{t_c} \end{pmatrix}, \qquad 1 \leq t_1 \leq \ldots \leq t_c \leq n.$$

Then one can write

$$w_1 A_{t_1} A_{t_1}^T + \cdots + w_c A_{t_c} A_{t_c}^T = (\mathbf{AS})\mathbf{W}(\mathbf{AS})^T,$$

where $\mathbf{W} = \mathsf{diag}\begin{pmatrix} w_1 & \ldots & w_c \end{pmatrix}$ is diagonal weighting matrix. We answer two questions in this section:

1. Given a set of c columns $\mathbf{AS}$ of $\mathbf{A}$, when is $\mathbf{A}\mathbf{A}^T = (\mathbf{AS})\,\mathbf{W}\,(\mathbf{AS})^T$ *without any constraints* on $\mathbf{W}$? The answer is an expression for a matrix $\mathbf{W}$ with minimal Frobenius norm (Section 2.2.1).

2. Given a set of c columns $\mathbf{AS}$ of $\mathbf{A}$, what are necessary and sufficient conditions under which $(\mathbf{AS})\mathbf{W}(\mathbf{AS})^T = \mathbf{A}\mathbf{A}^T$ for a *diagonal matrix* $\mathbf{W}$? The answer depends on the right singular vector matrix of $\mathbf{A}$ (Section 2.2.2).

2.2.1 Optimal approximation (no constraints on W)

For a given set of c columns $\mathbf{AS}$ of $\mathbf{A}$, we determine a matrix $\mathbf{W}$ of minimal Frobenius norm that minimizes the absolute error of $(\mathbf{AS})\mathbf{W}(\mathbf{AS})^T$ in the Frobenius norm.

The following is a special case of [45, Theorem 2.1], without any constraints on the number of columns in $\mathbf{AS}$. The idea is to represent $\mathbf{AS}$ in terms of the thin SVD of $\mathbf{A}$ as $\mathbf{AS} = \mathbf{U}\mathbf{\Sigma}(\mathbf{V}^T\mathbf{S})$.

Theorem 2.1. *Given c columns $\mathbf{AS}$ of $\mathbf{A}$, not necessarily distinct, the unique solution of*

$$\min_{\mathbf{W}} \|\mathbf{A}\mathbf{A}^T - (\mathbf{AS})\mathbf{W}(\mathbf{AS})^T\|_F$$

with minimal Frobenius norm is $\mathbf{W}_{opt} = (\mathbf{AS})^\dagger \, \mathbf{A}\mathbf{A}^T \, ((\mathbf{AS})^\dagger)^T$.

If, in addition, $\mathsf{rank}(\mathbf{AS}) = \mathsf{rank}(\mathbf{A})$, then

$$(\mathbf{AS})\mathbf{W}_{opt}(\mathbf{AS})^T = \mathbf{A}\mathbf{A}^T \qquad \textit{and} \qquad \mathbf{W}_{opt} = (\mathbf{V}^T\mathbf{S})^\dagger((\mathbf{V}^T\mathbf{S})^\dagger)^T.$$

If also $c = \mathsf{rank}(\mathbf{AS}) = \mathsf{rank}(\mathbf{A})$*, then*

$$(\mathbf{AS})\mathbf{W}_{opt}(\mathbf{AS})^T = \mathbf{AA}^T \qquad \textit{and} \qquad \mathbf{W}_{opt} = (\mathbf{V}^T\mathbf{S})^{-1}(\mathbf{V}^T\mathbf{S})^{-T}.$$

Proof. See Section 2.7.1. □

Theorem 2.1 implies that if $\mathbf{AS}$ has maximal rank, then the solution $\mathbf{W}_{opt}$ of minimal Frobenius norm depends only on the right singular vector matrix of $\mathbf{A}$ and in particular only on those columns $\mathbf{V}^T\mathbf{S}$ that correspond to the columns in $\mathbf{AS}$.

2.2.2 Exact computation with outer products (diagonal W)

We present necessary and sufficient conditions under which $(\mathbf{AS})\mathbf{W}(\mathbf{AS})^T = \mathbf{AA}^T$ for a non-negative diagonal matrix $\mathbf{W}$, that is $w_1 A_{t_1} A_{t_1}^T + \cdots + w_c A_{t_c} A_{t_c}^T = \mathbf{AA}^T$.

Theorem 2.2. *Let* $\mathbf{A}$ *be a* $m \times n$ *matrix, and let* $c \geq k \equiv \mathsf{rank}(\mathbf{A})$*. Then*

$$\sum_{j=1}^{c} w_j \, A_{t_j} A_{t_j}^T = \mathbf{AA}^T$$

for weights $w_j \geq 0$*, if and only if the* $c \times k$ *matrix* $\mathbf{V}^T \begin{pmatrix} \sqrt{w_1}\, e_{t_1} & \cdots & \sqrt{w_c}\, e_{t_c} \end{pmatrix}$ *has orthonormal rows.*

Proof. See Section 2.7.2. □

Remark 2.1 (Comparison with barrier sampling method)**.** *Our results differ from those in [7, 6, 89] in that we present conditions for* $\mathbf{A}$ *and the weights for exact computation of* $\mathbf{AA}^T$*, while [7, 6, 89] present an algorithm that can produce an arbitrarily good approximation for any matrix* $\mathbf{A}$*.*

If $\mathbf{A}$ has rank one, then any c non-zero columns of $\mathbf{A}$ will do for representing $\mathbf{AA}^T$, and explicit expressions for the weights can be derived.

Corollary 2.1. *If* $\mathsf{rank}(\mathbf{A}) = 1$ *then for any* c *columns* $A_{t_j} \neq 0$*,*

$$\sum_{j=1}^{c} w_j \, A_{t_j} A_{t_j}^T = \mathbf{AA}^T \qquad \textit{where} \quad w_j = \frac{1}{c\,\|\mathbf{V}^T e_{t_j}\|_2^2} = \frac{\|\mathbf{A}\|_F^2}{\|A_{t_j}\|_2^2}, \quad 1 \leq j \leq c.$$

Proof. See Section 2.7.3. □

Hence, in the special case of rank-one matrices, the weights are inverse leverage scores of $\mathbf{V}^T$ as well as inverse normalized column norms of $\mathbf{A}$. Furthermore, in the special case $c = 1$,

Corollary 2.1 implies that any non-zero column of $\mathbf{A}$ can be chosen. In particular, choosing the column A_l of largest norm yields a weight $w_1 = 1/\|\mathbf{V}^T e_l\|_2^2$ of minimal value, where $\|\mathbf{V}^T e_l\|_2^2$ is the coherence of $\mathbf{V}^T$.

In the following, we look at Theorem 2.2 in more detail, and distinguish the two cases when the number of selected columns is greater than $\mathsf{rank}(\mathbf{A})$, and when it is equal to $\mathsf{rank}(\mathbf{A})$.

Number of selected columns greater than $\mathsf{rank}(\mathbf{A})$

We illustrate the conditions of Theorem 2.2 when $c > \mathsf{rank}(\mathbf{A})$. In this case, indices do not necessarily have to be distinct, and a column can occur repeatedly.

Example 2.1. *Let*

$$\mathbf{V}^T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

so that $\mathsf{rank}(\mathbf{A}) = 2$. *Also let* $c = 3$, *and select the first column twice,* $t_1 = t_2 = 1$ *and* $t_3 = 2$, *so that*

$$\mathbf{V}^T \begin{pmatrix} e_1 & e_1 & e_2 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

The weights $w_1 = w_2 = 1/2$ *and* $w_3 = 1$ *give a matrix*

$$\mathbf{V}^T \begin{pmatrix} 2^{-1/2} e_1 & 2^{-1/2} e_1 & e_2 \end{pmatrix} = \begin{pmatrix} 2^{-1/2} & 2^{-1/2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

with orthonormal rows. Thus, an exact representation does not require distinct indices.

However, although the above weights yield an exact representation, the corresponding weight matrix does not have minimal Frobenius norm.

Remark 2.2 (Connection to Theorem 2.1)**.** *If* $c > k \equiv \mathsf{rank}(\mathbf{A})$ *in Theorem 2.2, then no diagonal weight matrix* $\mathbf{W} = \mathsf{diag}\begin{pmatrix} w_1 & \dots & w_c \end{pmatrix}$ *can be a minimal norm solution* $\mathbf{W}_{opt}$ *in Theorem 2.1.*

To see this, note that for $c > k$, *the columns* $A_{t_1}, \dots, A_{t_c}$ *are linearly dependent. Hence the* $c \times c$ *minimal Frobenius norm solution* $\mathbf{W}_{opt}$ *has rank equal to* $k < c$. *If* $\mathbf{W}_{opt}$ *were to be diagonal, it could have only* k *non-zero diagonal elements, hence the number of outer products would be* $k < c$, *a contradiction.*

To illustrate this, let

$$\mathbf{V}^T = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

so that $\mathsf{rank}(\mathbf{A}) = 2$*. Also, let* $c = 3$*, and select columns* $t_1 = 1$*,* $t_2 = 2$ *and* $t_3 = 3$*, so that*

$$\mathbf{V}^T\mathbf{S} \equiv \mathbf{V}^T \begin{pmatrix} e_1 & e_2 & e_3 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

Theorem 2.1 implies that the solution with minimal Frobenius norm is

$$\mathbf{W}_{opt} = (\mathbf{V}^T\mathbf{S})^\dagger((\mathbf{V}\mathbf{S}^T)^\dagger) = \begin{pmatrix} 1/2 & 0 & 1/2 \\ 0 & 2 & 0 \\ 1/2 & 0 & 1/2 \end{pmatrix},$$

which is not diagonal.

However $\mathbf{W} = \mathsf{diag} \begin{pmatrix} 1 & 2 & 1 \end{pmatrix}$ *is also a solution since* $\mathbf{V}^T\mathbf{S}\mathbf{W}^{1/2}$ *has orthonormal rows. But* $\mathbf{W}$ *does not have minimal Frobenius norm since* $\|\mathbf{W}\|_F^2 = 6$*, while* $\|\mathbf{W}_{opt}\|_F^2 = 5$*.*

Number of selected columns equal to $\mathsf{rank}(\mathbf{A})$

If $c = \mathsf{rank}(\mathbf{A})$, then no column of $\mathbf{A}$ can be selected more than once, hence the selected columns form a submatrix of $\mathbf{A}$. In this case Theorem 2.2 can be strengthened: As for the rank-one case in Corollary 2.1, an explicit expression for the weights in terms of leverage scores can be derived.

Theorem 2.3. *Let* $\mathbf{A}$ *be a* $m \times n$ *matrix with* $k \equiv \mathsf{rank}(\mathbf{A})$*. In addition to the conclusions of Theorem 2.2 the following also holds: If*

$$\mathbf{V}^T \begin{pmatrix} \sqrt{w_1}e_{t_1} & \cdots & \sqrt{w_k}e_{t_k} \end{pmatrix}$$

has orthonormal rows, then it is an orthogonal matrix, and $w_j = 1/\|\mathbf{V}^T e_{t_j}\|_2^2$*,* $1 \leq j \leq k$*.*

Proof. See Section 2.7.4. □

Note that the columns selected from $\mathbf{V}^T$ do not necessarily correspond to the largest leverage scores. The following example illustrates that the conditions in Theorem 2.3 are non-trivial.

Example 2.2. *In Theorem 2.3 it is not always possible to find k columns from $\mathbf{V}^T$ that yield an orthogonal matrix.*

For instance, let

$$\mathbf{V}^T = \begin{pmatrix} 1/2 & 1/2 & 1/2 & 1/2 \\ -1/\sqrt{14} & -2/\sqrt{14} & 3/\sqrt{14} & 0 \end{pmatrix},$$

and $c = \mathsf{rank}(\mathbf{V}) = 2$. Since no two columns of $\mathbf{V}^T$ are orthogonal, no two columns can be scaled to be orthonormal. Thus no 2×2 matrix submatrix of $\mathbf{V}^T$ can give rise to an orthogonal matrix.

However, for $c = 3$ it is possible to construct a 2×3 matrix with orthonormal rows. Selecting columns $t_1 = 1$, $t_2 = 2$ and $t_3 = 3$ from $\mathbf{V}^T$, and weights $w_1 = \sqrt{5/2}$, $w_2 = \sqrt{2/5}$ and $w_3 = \sqrt{11/10}$ yields a matrix

$$\mathbf{V}^T \begin{pmatrix} \sqrt{\frac{5}{2}}e_1 & \sqrt{\frac{2}{5}}e_2 & \sqrt{\frac{11}{10}}e_3 \end{pmatrix} = \begin{pmatrix} \sqrt{\frac{5}{8}} & \sqrt{\frac{1}{10}} & \sqrt{\frac{11}{40}} \\ -\sqrt{\frac{5}{28}} & -\sqrt{\frac{4}{35}} & \sqrt{\frac{99}{140}} \end{pmatrix}$$

that has orthonormal rows.

Remark 2.3 (Connection to Theorem 2.1)**.** *In Theorem 2.3 the condition $c = k$ implies that the $k \times k$ matrix*

$$\mathbf{V}^T \begin{pmatrix} e_{t_1} & \dots & e_{t_k} \end{pmatrix} = \mathbf{V}^T\mathbf{S}$$

is non-singular. From Theorem 2.1 follows that $\mathbf{W}_{opt} = (\mathbf{V}^T\mathbf{S})^{-1}(\mathbf{V}^T\mathbf{S})^{-T}$ is the unique minimal Frobenius norm solution for $\mathbf{A}\mathbf{A}^T = (\mathbf{A}\mathbf{S})\mathbf{W}(\mathbf{A}\mathbf{S})^T$.

If, in addition, the rows of $\mathbf{V}^T\mathbf{S}\mathbf{W}_{opt}^{1/2}$ are orthonormal, then the minimal norm solution $\mathbf{W}_{opt}$ is a diagonal matrix,

$$\mathbf{W}_{opt} = (\mathbf{V}^T\mathbf{S})^{-1}(\mathbf{V}^T\mathbf{S})^{-T} = \mathsf{diag}\begin{pmatrix} \frac{1}{\|\mathbf{V}^T e_{t_1}\|_2^2} & \dots & \frac{1}{\|\mathbf{V}^T e_{t_k}\|_2^2} \end{pmatrix}.$$

2.3 Monte Carlo algorithm for Gram Matrix Approximation

We review the randomized algorithm to approximate the Gram matrix (Section 2.3.1); and discuss and compare two different types of sampling probabilities (Section 2.3.2).

2.3.1 The algorithm

The randomized algorithm for approximating $\mathbf{A}\mathbf{A}^T$, presented as Algorithm 1, is a special case of the BasicMatrixMultiplication Algorithm [36, Figure 2] which samples according to the

Exactly(c) algorithm [43, Algorithm 3], that is, independently and with replacement. This means a column can be sampled more than once.

A conceptual version of the randomized algorithm is presented as Algorithm 1. Given a user-specified number of samples c, and a set of probabilities p_j, this version assembles columns of the sampling matrix $\mathbf{S}$, then applies $\mathbf{S}$ to $\mathbf{A}$, and finally computes the product

$$\mathbf{X} = (\mathbf{AS})\,(\mathbf{AS})^T = \sum_{j=1}^{c} \frac{1}{cp_{t_j}}\, A_{t_j} A_{t_j}^T.$$

The choice of weights $1/(cp_{t_j})$ makes $\mathbf{X}$ an unbiased estimator, $\mathbb{E}[\mathbf{X}] = \mathbf{AA}^T$ [36, Lemma 3].

Algorithm 1 Conceptual version of randomized matrix multiplication [36, 43]

Input: $m \times n$ matrix $\mathbf{A}$, number of samples $1 \le c \le n$
Probabilities p_j, $1 \le j \le n$, with $p_j \ge 0$ and $\sum_{j=1}^{n} p_j = 1$

Output: Approximation $\mathbf{X} = (\mathbf{AS})\,(\mathbf{AS})^T$ where $\mathbf{S}$ is $n \times c$ with $\mathbb{E}[\mathbf{S}\,\mathbf{S}^T] = \mathbf{I}_n$

$\mathbf{S} = \mathbf{0}_{n\times c}$
for $j = 1 : c$ **do**
  Sample t_j from $\{1, \ldots, n\}$ with probability p_{t_j}
  independently and with replacement
  $S_j = e_{t_j}/\sqrt{cp_{t_j}}$
end for
$\mathbf{X} = (\mathbf{AS})\,(\mathbf{AS})^T$

Discounting the cost of sampling, Algorithm 1 requires $\mathcal{O}(m^2 c)$ flops to compute an approximation to $\mathbf{AA}^T$. Note that Algorithm 1 allows zero probabilities. Since an index corresponding to $p_j = 0$ can never be selected, division by zero does not occur in the computation of $\mathbf{S}$. Implementations of sampling with replacement are discussed in [44, Section 2.1]. For matrices of small dimension, one can simply use the Matlab function `randsample`.

2.3.2 Sampling probabilities

We consider two types of probabilities, the "optimal" probabilities from [36] (Section 2.3.2), and leverage score probabilities (Section 2.3.2) motivated by Corollary 2.1 and Theorem 2.3, and their use in other randomized algorithms [14, 39, 42]. We show (Theorem 2.4) that for rank-one matrices, Algorithm 1 with "optimal" probabilities produces the exact result with a single sample. Numerical experiments (Section 2.3.2) illustrate that sampling with "optimal" probabilities

results in smaller two-norm relative errors than sampling with leverage score probabilities, and that the two types of probabilities can differ significantly.

"Optimal" probabilities [36]

They are defined by

$$p_j^{opt} = \frac{\|A_j\|_2^2}{\|\mathbf{A}\|_F^2}, \qquad 1 \le j \le n \tag{2.1}$$

and are called "optimal" because they minimize $\mathbb{E}\left[\left\|\mathbf{X} - \mathbf{A}\mathbf{A}^T\right\|_F^2\right]$ [36, Lemma 4]. The "optimal" probabilities can be computed in $\mathcal{O}(mn)$ flops.

The analyses in [36, Section 4.4] apply to the more general "nearly optimal" probabilities p_j^β, which satisfy $\sum_{j=1}^n p_j^\beta = 1$ and are constrained by

$$p_j^\beta \ge \beta\, p_j^{opt}, \qquad 1 \le j \le n, \tag{2.2}$$

where $0 < \beta \le 1$ is a scalar. In the special case $\beta = 1$, they revert to the optimal probabilites, $p_j^\beta = p_j^{opt}$, $1 \le j \le n$. Hence β can be viewed as the deviation of the probabilities p_j^β from the "optimal" probabilities p_j^{opt}.

Leverage score probabilities [12, 14]

The exact representation in Theorem 2.3 suggests probabilities based on the leverage scores of $\mathbf{V}^T$,

$$p_j^{lev} = \frac{\left\|\mathbf{V}^T e_j\right\|_2^2}{\|\mathbf{V}\|_F^2} = \frac{\|\mathbf{V}^T e_j\|_2^2}{k}, \qquad 1 \le j \le n, \tag{2.3}$$

where $k = \mathsf{rank}(\mathbf{A})$.

Since the leverage score probabilities are proportional to the squared column norms of $\mathbf{V}^T$, they are the "optimal" probabilities for approximating $\mathbf{V}^T\mathbf{V} = \mathbf{I}_k$. Exact computation of leverage score probabilities, via SVD or QR decomposition, requires $\mathcal{O}(m^2 n)$ flops; thus, it is more expensive than the computation of the "optimal" probabilities.

In the special case of rank-one matrices, the "optimal" and leverage score probabilities are identical; and Algorithm 1 with "optimal" probabilities computes the exact result with any number of samples, and in particular a single sample. This follows directly from Corollary 2.1.

Theorem 2.4. *If* $\mathsf{rank}(\mathbf{A}) = 1$, *then* $p_j^{lev} = p_j^{opt}$, $1 \le j \le n$.

If $\mathbf{X}$ *is computed by Algorithm 1 with any* $c \ge 1$ *and probabilities* p_j^{opt}, *then* $\mathbf{X} = \mathbf{A}\mathbf{A}^T$.

Table 2.4: Eight datasets from [3], and the dimensions, rank and stable rank of the associated matrices $\mathbf{A}$.

Dataset	$m \times n$	$\mathsf{rank}(\mathbf{A})$	$\mathsf{sr}(\mathbf{A})$
Solar Flare	10×1389	10	1.10
EEG Eye State	15×14980	15	1.31
QSAR biodegradation	41×1055	41	1.13
Abalone	8×4177	8	1.002
Wilt	5×4399	5	1.03
Wine Quality - Red	12×1599	12	1.03
Wine Quality - White	12×4898	12	1.01
Yeast	8×1484	8	1.05

Comparison of sampling probabilities

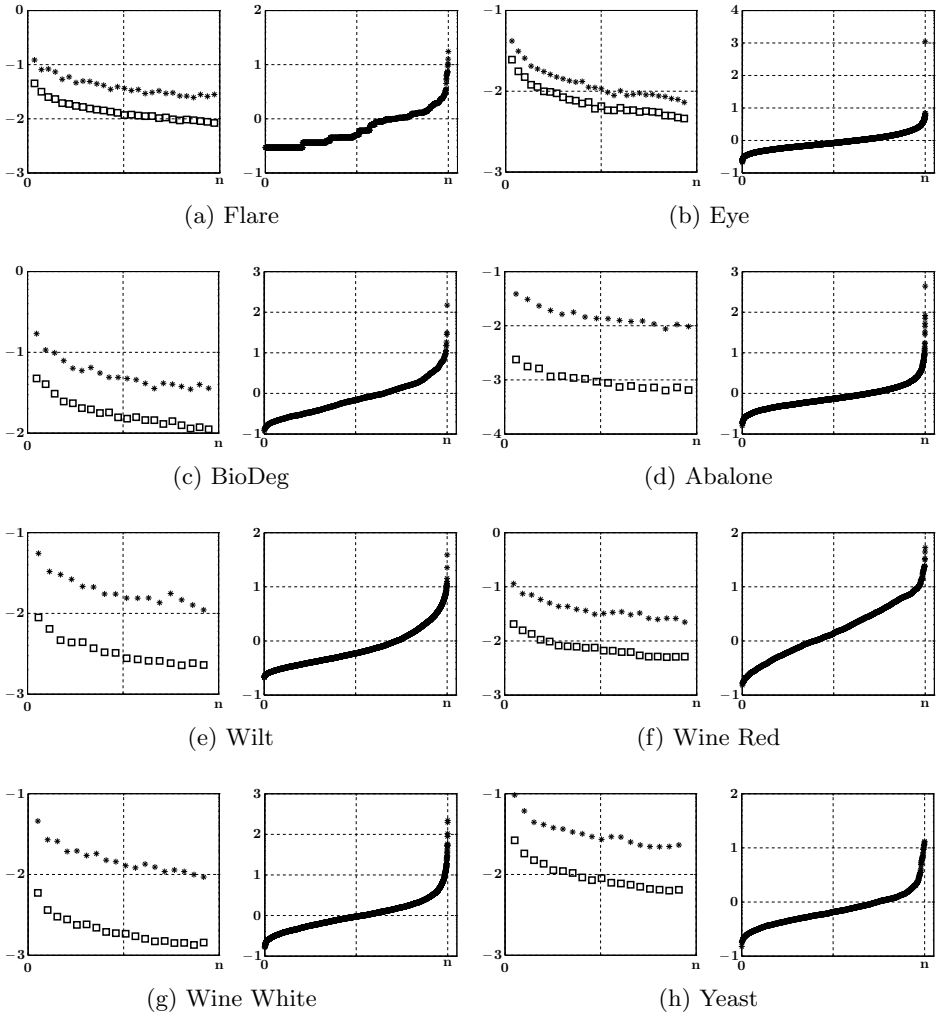
We compare the norm-wise relative errors due to randomization of Algorithm 1 when it samples with "optimal" probabilites and leverage score probabilities.

Experimental set up We present experiments with eight representative matrices, described in Table 2.4, from the UCI Machine Learning Repository [3].

For each matrix, we ran Algorithm 1 twice: once sampling with "optimal" probabilities p_j^{opt}, and once sampling with leverage score probabilities p_j^{lev}. The sampling amounts c range from 1 to n, with 100 runs for each value of c.

Figure 2.1 contains two plots for each matrix: The left plot shows the two-norm relative errors due to randomization, $\|\mathbf{X}-\mathbf{A}\mathbf{A}^T\|_2/\|\mathbf{A}\mathbf{A}^T\|_2$, averaged over 100 runs, versus the sampling amount c. The right plot shows the ratios of leverage score over "optimal" probabilities p_j^{lev}/p_j^{opt}, $1 \leq j \leq n$.

Conclusions Sampling with "optimal" probabilities produces average errors that are lower, by as much as a factor of 10, than those from sampling with leverage score probabilities, for all sampling amounts c. Furthermore, corresponding leverage score and "optimal" probabilities tend to differ by several orders of magnitude.



(a) Flare (b) Eye

(c) BioDeg (d) Abalone

(e) Wilt (f) Wine Red

(g) Wine White (h) Yeast

Figure 2.1: Relative errors due to randomization, and ratios of leverage score over "optimal" probabilities for the matrices in Table 2.4. Plots in columns 1 and 3: The average over 100 runs of $\left\|\mathbf{X}-\mathbf{A}\mathbf{A}^T\right\|_2 / \left\|\mathbf{A}\mathbf{A}^T\right\|_2$ when Algorithm 1 samples with "optimal probabilities" ($\square$) and with leverage score probabilities ($*$), versus the number c of sampled columns in $\mathbf{X}$. The vertical axes are logarithmic, and the labels correspond to powers of 10. Plots in columns 2 and 4: Ratios p_j^{lev}/p_j^{opt}, $1 \leq j \leq n$, sorted in increasing magnitude from left to right.

2.4 Error due to randomization, for sampling with "nearly optimal" probabilities

We present two new probabilistic bounds (Sections 2.4.1 and 2.4.2) for the two-norm relative error due to randomization, when Algorithm 1 samples with the "nearly optimal" probabili-

ties in (2.2). The bounds depend on the stable rank or the rank of $\mathbf{A}$, but not on the matrix dimensions. Neither bound is always better than the other (Section 2.4.3). The numerical experiments (Section 2.4.4) illustrate that the bounds are informative, even for stringent success probabilities and matrices of small dimension.

2.4.1 First bound

The first bound depends on the stable rank of $\mathbf{A}$ and also, weakly, on the rank.

Theorem 2.5. *Let $\mathbf{A} \neq \mathbf{0}$ be an $m \times n$ matrix, and let $\mathbf{X}$ be computed by Algorithm 1 with the "nearly optimal" probabilities p_j^{β} in (2.2).*

Given $0 < \delta < 1$ and $0 < \epsilon \leq 1$, if the number of columns sampled by Algorithm 1 is at least

$$c \geq c_0(\epsilon)\, \mathsf{sr}(\mathbf{A})\, \frac{\ln\left(\mathsf{rank}(\mathbf{A})/\delta\right)}{\beta\, \epsilon^2}, \qquad \textit{where} \quad c_0(\epsilon) \equiv 2 + \frac{2\epsilon}{3},$$

then with probability at least $1 - \delta$,

$$\frac{\left\|\mathbf{X} - \mathbf{A}\mathbf{A}^T\right\|_2}{\|\mathbf{A}\mathbf{A}^T\|_2} \leq \epsilon.$$

Proof. See Section 2.7.5. □

As the required error ϵ becomes smaller, so does the constant $c_0(\epsilon)$ in the lower bound for the number of samples, that is, $c_0(\epsilon) \to 2$ as $\epsilon \to 0$.

2.4.2 Second bound

This bound depends only on the stable rank of $\mathbf{A}$.

Theorem 2.6. *Let $\mathbf{A} \neq \mathbf{0}$ be an $m \times n$ matrix, and let $\mathbf{X}$ be computed by Algorithm 1 with the "nearly optimal" probabilities p_j^{β} in (2.2).*

Given $0 < \delta < 1$ and $0 < \epsilon \leq 1$, if the number of columns sampled by Algorithm 1 is at least

$$c \geq c_0(\epsilon)\, \mathsf{sr}(\mathbf{A})\, \frac{\ln\left(4\mathsf{sr}(\mathbf{A})/\delta\right)}{\beta\, \epsilon^2}, \qquad \textit{where} \quad c_0(\epsilon) \equiv 2 + \frac{2\epsilon}{3},$$

then with probability at least $1 - \delta$,

$$\frac{\left\|\mathbf{X} - \mathbf{A}\mathbf{A}^T\right\|_2}{\|\mathbf{A}\mathbf{A}^T\|_2} \leq \epsilon.$$

Proof. See Section 2.7.6. □

2.4.3 Comparison

The bounds in Theorems 2.5 and 2.6 differ only in the arguments of the logarithms.

On the one hand, Theorem 2.6 is tighter than Theorem 2.5 if $4\,\mathsf{sr}(\mathbf{A}) < \mathsf{rank}(\mathbf{A})$. On the other hand, Theorem 2.5 is tighter for matrices with large stable rank, and in particular for matrices $\mathbf{A}$ with orthonormal rows where $\mathsf{sr}(\mathbf{A}) = \mathsf{rank}(\mathbf{A})$.

In general, Theorem 2.6 is tighter than all the bounds in Table 2.2, that is, to our knowledge, all published bounds.

Table 2.5: Matrices from [33], their dimensions, rank and stable rank; and key quantities from (2.4) and (2.5).

Matrix	$m \times n$	$\mathsf{rank}(\mathbf{A})$	$\mathsf{sr}(\mathbf{A})$	$c\,\gamma_1$	$c\,\gamma_2$
us04	163×28016	115	5.27	16.43	13.44
bibd_16_8	163×28016	120	4.29	13.43	10.65

2.4.4 Numerical experiments

We compare the bounds in Theorems 2.5 and 2.6 to the errors of Algorithm 1 for sampling with "optimal" probabilities.

Experimental set up We present experiments with two matrices from the University of Florida Sparse Matrix Collection [33]. The matrices have the same dimension, and similar high ranks and low stable ranks, see Table 2.5. Note that only for low stable ranks can Algorithm 1 achieve any accuracy.

The sampling amounts c range from 1 to n, the number of columns, with 100 runs for each value of c. From the 100 errors $\|\mathbf{X} - \mathbf{A}\mathbf{A}^T\|_2 / \|\mathbf{A}\mathbf{A}^T\|_2$ for each c value, we plot the smallest, largest, and average.

In Theorems 2.5 and 2.6, the success probability is 99 percent, that is, a failure probability of $\delta = .01$. The error bounds are plotted as a function of c. That is, for Theorem 2.5 we plot (see Theorem 2.15)

$$\frac{\left\|\mathbf{X} - \mathbf{A}\mathbf{A}^T\right\|_2}{\|\mathbf{A}\mathbf{A}^T\|_2} \leq \gamma_1 + \sqrt{\gamma_1\,(6 + \gamma_1)}, \qquad \gamma_1 \equiv \mathsf{sr}(\mathbf{A})\,\frac{\ln\left(\mathsf{rank}(\mathbf{A})/.01\right)}{3\;c} \tag{2.4}$$

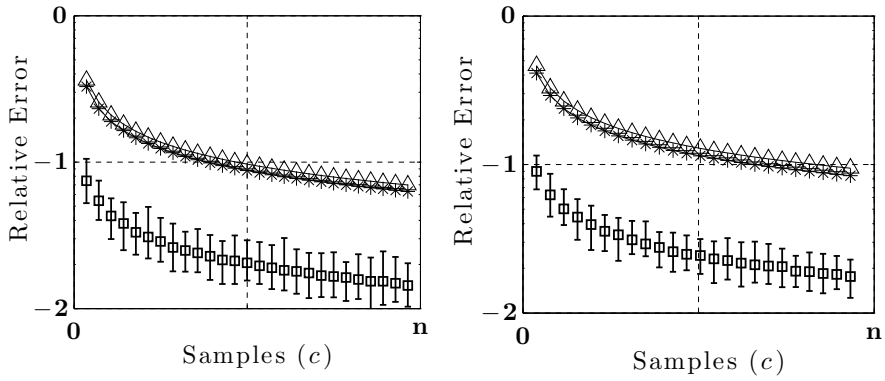


Figure 2.2: Relative errors due to randomization from Algorithm 1, and bounds (2.4) and (2.5) versus sampling amount c, for matrices us04 (left) and bidb_16_8 (right). Error bars represent the maximum and minimum of the errors $\left\|\mathbf{X}-\mathbf{A}\mathbf{A}^T\right\|_2 / \left\|\mathbf{A}\mathbf{A}^T\right\|_2$ from Algorithm 1 over 100 runs, while the squares represent the average. The triangles ($\triangle$) represent the bound (2.4), while the stars ($*$) represent (2.5). The vertical axes are logarithmic, and the labels correspond to powers of 10.

while for Theorem 2.6 we plot (see Theorem 2.17)

$$\frac{\left\|\mathbf{X}-\mathbf{A}\mathbf{A}^T\right\|_2}{\|\mathbf{A}\mathbf{A}^T\|_2} \leq \gamma_2 + \sqrt{\gamma_2\,(6+\gamma_2)}, \qquad \gamma_2 \equiv \mathsf{sr}(\mathbf{A})\,\frac{\ln\left(4\mathsf{sr}(\mathbf{A})/.01\right)}{3\,c} \tag{2.5}$$

The key quantities $c\,\gamma_1$ and $c\,\gamma_2$ are shown for both matrices in Table 2.5.

Figure 2.2 contains two plots, the left one for matrix us04, and the right one for matrix bibd_16_8. The plots show the relative errors $\|\mathbf{X}-\mathbf{A}\mathbf{A}^T\|_2/\|\mathbf{A}\mathbf{A}^T\|_2$ and the bounds (2.4) and (2.5) versus the sampling amount c.

Conclusions In both plots, the bounds corresponding to Theorems 2.5 and 2.6 are virtually indistinguishable, as was is already predicted by the key quantities $c\,\gamma_1$ and $c\,\gamma_2$ in Table 2.5. The bounds overestimate the worst case error from Algorithm 1 by a factor of at most 10. Hence they are informative, even for matrices of small dimension and a stringent success probability.

2.5 Error due to randomization, for sampling with leverage score probabilities

For completeness, we present a normwise relative bound for the error due to randomization, when Algorithm 1 samples with leverage score probabilities (2.3). The bound corroborates the numerical experiments in Section 2.3.2, and suggests that sampling with leverage score

probabilities produces a larger error due to randomization than sampling with "nearly optimal" probabilities.

Theorem 2.7. *Let $\mathbf{A} \neq \mathbf{0}$ be an $m \times n$ matrix, and let $\mathbf{X}$ be computed by Algorithm 1 with the leverage score probabilites p_j^{lev} in (2.3).*

Given $0 < \delta < 1$ and $0 < \epsilon \leq 1$, if the number of columns sampled by Algorithm 1 is at least

$$c \geq c_0(\epsilon)\, \mathsf{rank}(\mathbf{A})\, \frac{\ln(\mathsf{rank}(\mathbf{A})/\delta)}{\epsilon^2}, \qquad \textit{where} \quad c_0(\epsilon) = 2 + \frac{2\epsilon}{3},$$

then with probability at least $1 - \delta$,

$$\frac{\left\|\mathbf{X} - \mathbf{A}\mathbf{A}^T\right\|_2}{\left\|\mathbf{A}\mathbf{A}^T\right\|_2} \leq \epsilon.$$

Proof. See Section 2.7.7. □

In the special case when $\mathbf{A}$ has orthonormal columns, the leverage score probabilities p_j^{lev} are equal to the "optimal" probabilities p_j^{opt} in (2.1). Furthermore, $\mathsf{rank}(\mathbf{A}) = \mathsf{sr}(\mathbf{A})$, so that Theorem 2.7 is equal to Theorem 2.5. For general matrices $\mathbf{A}$, though, $\mathsf{rank}(\mathbf{A}) \geq \mathsf{sr}(\mathbf{A})$, and Theorem 2.7 is not as tight as Theorem 2.5.

2.6 Singular value and condition number bounds

As in [43], we apply the bounds for the Gram matrix approximation to a matrix with orthonormal rows, and derive bounds for the smallest singular value (Section 2.6.1) and condition number (Section 2.6.2) of a sampled matrix.

Specifically, let $\mathbf{Q}$ be a real $m \times n$ matrix with orthonormal rows, $\mathbf{Q}\mathbf{Q}^T = \mathbf{I}_m$. Then, as discussed in Section 2.3.2, the "optimal" probabilities (2.1) for $\mathbf{Q}$ are equal to the leverage score probabilities (2.3),

$$p_j^{opt} = \frac{\|Q_j\|_2^2}{\|\mathbf{Q}\|_F^2} = \frac{\|Q_j\|_2^2}{m} = p_j^{lev}, \qquad 1 \leq j \leq m.$$

The connection between Gram matrix approximations $(\mathbf{QS})\,(\mathbf{QS})^T$ and singular values of the sampled matrix $\mathbf{QS}$ comes from the well-conditioning of singular values [54, Corollary 2.4.4],

$$\begin{aligned} \left|1 - \sigma_j\left(\mathbf{QS}\right)^2\right| &= \left|\sigma_j\left(\mathbf{Q}\mathbf{Q}^T\right) - \sigma_j\left(\left(\mathbf{QS}\right)\left(\mathbf{QS}\right)^T\right)\right| \\ &\leq \left\|\mathbf{Q}\mathbf{Q}^T - \left(\mathbf{QS}\right)\left(\mathbf{QS}\right)^T\right\|_2, \qquad 1 \leq j \leq m. \end{aligned} \tag{2.6}$$

2.6.1 Singular value bounds

We present two bounds for the smallest singular value of a sampled matrix, for sampling with the "nearly optimal" probabilities (2.2), and for uniform sampling with and without replacement.

The first bound is based on the Gram matrix approximation in Theorem 2.5.

Theorem 2.8. *Let* $\mathbf{Q}$ *be an* $m \times n$ *matrix with orthonormal rows and coherence* μ*, and let* $\mathbf{QS}$ *be computed by Algorithm 1. Given* $0 < \epsilon < 1$ *and* $0 < \delta < 1$*, we have* $\sigma_m(\mathbf{QS}) \geq \sqrt{1-\epsilon}$ *with probability at least* $1 - \delta$*, if Algorithm 1*

- *either samples with the "nearly optimal" probabilities* p_j^β*, and*

$$c \geq c_0(\epsilon)\, m\, \frac{\ln(m/\delta)}{\beta \epsilon^2},$$

- *or samples with uniform probabilities* $1/n$*, and*

$$c \geq c_0(\epsilon)\, n\, \mu\, \frac{\ln(m/\delta)}{\epsilon^2}.$$

Here $c_0(\epsilon) \equiv 2 + \frac{2}{3}\,\epsilon$*.*

Proof. See Section 2.7.8. □

Since $c_0(\epsilon) \geq 2$, the above bound for uniform sampling is slightly less tight than the last bound in Table 2.3, i.e. [49, Lemma 1]. Although that bound technically holds only for uniform sampling *without* replacement, the same proof gives the same bound for uniform sampling *with* replacement.

This inspired us to derive a second bound, by modifying the argument in [49, Lemma 1], to obtain a slightly tighter constant. This is done with a direct application of a Chernoff bound (Theorem 2.18). The only difference between the next and the previous result is the smaller constant $c_1(\epsilon)$, and the added application to sampling without replacement.

Theorem 2.9. *Let* $\mathbf{Q}$ *be an* $m \times n$ *matrix with orthonormal rows and coherence* μ*, and let* $\mathbf{QS}$ *be computed by Algorithm 1. Given* $0 < \epsilon < 1$ *and* $0 < \delta < 1$*, we have* $\sigma_m(\mathbf{QS}) \geq \sqrt{1-\epsilon}$ *with probability at least* $1 - \delta$*, if Algorithm 1*

- *either samples with the "nearly optimal" probabilities* p_j^β*, and*

$$c \geq c_1(\epsilon)\, m\, \frac{\ln(m/\delta)}{\beta \epsilon^2},$$

- *or samples with uniform probabilities* $1/n$*, with or without replacement, and*

$$c \geq c_1(\epsilon)\, n\, \mu\, \frac{\ln(m/\delta)}{\epsilon^2}.$$

Here $c_1(\epsilon) \equiv \frac{\epsilon^2}{(1-\epsilon)\ln(1-\epsilon)+\epsilon}$*, and* $1 \leq c_1(\epsilon) \leq 2$*.*

Proof. See Section 2.7.9. □

The constant $c_1(\epsilon)$ is slightly smaller than the constant 2 in [49, Lemma 1], which is the last bound in Table 2.3.

2.6.2 Condition number bounds

We present two bounds for the condition number $\kappa(\mathbf{QS}) \equiv \sigma_1(\mathbf{QS})/\sigma_m(\mathbf{QS})$ of a sampled matrix $\mathbf{QS}$ with full row-rank.

The first condition number bound is based on a Gram matrix approximation, and is analogous to Theorem 2.8.

Theorem 2.10. *Let* $\mathbf{Q}$ *be an* $m \times n$ *matrix with orthonormal rows and coherence* μ*, and let* $\mathbf{QS}$ *be computed by Algorithm 1. Given* $0 < \epsilon < 1$ *and* $0 < \delta < 1$*, we have* $\kappa(\mathbf{QS}) \leq \frac{\sqrt{1+\epsilon}}{\sqrt{1-\epsilon}}$ *with probability at least* $1 - \delta$*, if Algorithm 1*

- *either samples with the "nearly optimal" probabilities* p_j^β*, and*

$$c \geq c_0(\epsilon)\, m\, \frac{\ln(m/\delta)}{\beta\epsilon^2},$$

- *or samples with uniform probabilities* $1/n$*, and*

$$c \geq c_0(\epsilon)\, n\, \mu\, \frac{\ln(m/\delta)}{\epsilon^2}.$$

Here $c_0(\epsilon) \equiv 2 + \frac{2}{3}\,\epsilon$*.*

Proof. See Section 2.7.10. □

The second condition number bound is based on a Chernoff inequality, and is analogous to Theorem 2.9, but with a different constant, and an additional factor of two in the logarithm.

Theorem 2.11. *Let* $\mathbf{Q}$ *be an* $m \times n$ *matrix with orthonormal rows and coherence* μ*, and let* $\mathbf{QS}$ *be computed by Algorithm 1. Given* $0 < \epsilon < 1$ *and* $0 < \delta < 1$*, we have* $\kappa(\mathbf{QS}) \leq \frac{\sqrt{1+\epsilon}}{\sqrt{1-\epsilon}}$ *with probability at least* $1 - \delta$*, if Algorithm 1*

- *either samples with the "nearly optimal" probabilities* p_j^β*, and*

$$c \geq c_2(\epsilon)\, m\, \frac{\ln(2m/\delta)}{\beta\epsilon^2},$$

- *or samples with uniform probabilities* $1/n$*, with or without replacement, and*

$$c \geq c_2(\epsilon)\, n\, \mu\, \frac{\ln(2m/\delta)}{\epsilon^2}.$$

Here $c_2(\epsilon) \equiv \frac{\epsilon^2}{(1+\epsilon)\ln(1+\epsilon)-\epsilon}$*, and* $2 \leq c_2(\epsilon) \leq 2.6$*.*

Proof. See Section 2.7.11. □

It is difficult to compare the two condition number bounds, and neither bound is always tighter than the other. On the one hand, Theorem 2.11 has a smaller constant than Theorem 2.10 since $c_2(\epsilon) \leq c_1(\epsilon)$. On the other hand, though, Theorem 2.10 has an additional factor of two in the logarithm. For very large m/δ, the additional factor of 2 in the logarithm does not matter much and Theorem 2.11 is tighter.

In general, Theorem 2.11 is not always tighter than Theorem 2.10. For example, if $m = 100$, $\delta = 0.01$, $\epsilon = 0.1$, $\beta = 1$, and Algorithm 1 samples with "nearly optimal" probabilities, then Theorem 2.11 requires $1.57 \cdot 10^5$ samples, while Theorem 2.10 requires only $1.43 \cdot 10^5$; hence, it is tighter.

Acknowledgements

We thank Petros Drineas and Michael Mahoney for useful discussions, and the four anonymous reviewers whose suggestions helped us to improve the quality of the paper.

2.7 Proofs

We present proofs for the results in Sections 2.2 – 2.6.

2.7.1 Proof of Theorem 2.1

We will use the two lemmas below. The first one is a special case of [45, Theorem 2.1] where the rank of the approximation is not restricted.

Lemma 2.1. *Let* $\mathbf{H}$ *be* $m \times n$, $\mathbf{B}$ *be* $m \times p$, *and* $\mathbf{C}$ *be* $q \times n$ *matrices, and let* $\mathbf{P_B}$ *be the orthogonal projector onto* $\mathsf{range}(\mathbf{B})$, *and* $\mathbf{P_{C^T}}$ *the orthogonal projector onto* $\mathsf{range}(\mathbf{C}^T)$. *Then the solution of*

$$\min_{\mathbf{W}} \|\mathbf{H} - \mathbf{B}\,\mathbf{W}\,\mathbf{C}\|_F$$

with minimal Frobenius norm is

$$\mathbf{W} = \mathbf{B}^\dagger\,\mathbf{P_B}\,\mathbf{H}\,\mathbf{P_{C^T}}\,\mathbf{C}^\dagger.$$

Lemma 2.2. *If* $\mathbf{B}$ *is* $m \times p$ *and* $\mathbf{C}$ *is* $p \times n$, *with* $\mathsf{rank}(\mathbf{B}) = p = \mathsf{rank}(\mathbf{C})$, *then* $(\mathbf{BC})^\dagger = \mathbf{C}^\dagger\mathbf{B}^\dagger$.

Proof. Set $\mathbf{Y} \equiv \mathbf{BC}$, and use $\mathbf{B}^\dagger\mathbf{B} = \mathbf{I}_p = \mathbf{C}\mathbf{C}^\dagger$ to verify that $\mathbf{Z} \equiv \mathbf{C}^\dagger\mathbf{B}^\dagger$ satisfies the four conditions defining the Moore-Penrose inverse

$$\mathbf{YZY} = \mathbf{Y}, \quad \mathbf{ZYZ} = \mathbf{Z}, \quad (\mathbf{YZ})^T = \mathbf{YZ}, \quad (\mathbf{ZY})^T = \mathbf{ZY}. \tag{2.7}$$

□

Proof of Theorem 2.1

Abbreviate $\mathbf{A}_1 \equiv \mathbf{AS}$ and $\mathbf{V}_1^T \equiv \mathbf{V}^T\mathbf{S}$.

In Lemma 2.1, set $\mathbf{H} = \mathbf{A}\mathbf{A}^T$, $\mathbf{B} = \mathbf{A}_1$, and $\mathbf{C} = \mathbf{A}_1^T$. Then $\mathbf{P_B} = \mathbf{A}_1\mathbf{A}_1^\dagger = \mathbf{P_{C^T}}$, and

$$\mathbf{W}_{opt} = \mathbf{A}_1^\dagger\,\mathbf{A}_1\mathbf{A}_1^\dagger\,\mathbf{A}\mathbf{A}^T\,\mathbf{A}_1\mathbf{A}_1^\dagger\,(\mathbf{A}_1^\dagger)^T.$$

The conditions for the Moore-Penrose inverse (2.7) imply $\mathbf{A}_1^\dagger\mathbf{A}_1\mathbf{A}_1^\dagger = \mathbf{A}_1^\dagger$, and

$$\mathbf{A}_1\mathbf{A}_1^\dagger\,(\mathbf{A}_1^\dagger)^T = \left(\mathbf{A}_1\mathbf{A}_1^\dagger\right)^T\,(\mathbf{A}_1^\dagger)^T = (\mathbf{A}_1^\dagger)^T\,\mathbf{A}_1^T\,(\mathbf{A}_1^\dagger)^T = (\mathbf{A}_1^\dagger)^T.$$

Hence $\mathbf{W}_{opt} = \mathbf{A}_1^\dagger\,\mathbf{A}\mathbf{A}^T\,(\mathbf{A}_1^\dagger)^T$.

Special case $\mathsf{rank}(\mathbf{A}_1) = \mathsf{rank}(\mathbf{A})$ This means the number of columns c in $\mathbf{A}_1 = \mathbf{U}\mathbf{\Sigma}\mathbf{V}_1^T$ is at least as large as $k \equiv \mathsf{rank}(\mathbf{A})$. Hence $\mathbf{V}_1^T$ is $k \times c$ with $c \geq k$, and $\mathsf{rank}(\mathbf{V}_1^T) = k = \mathsf{rank}(\mathbf{U}\mathbf{\Sigma})$. From Lemma 2.2 follows $\mathbf{A}_1^\dagger = (\mathbf{V}_1^\dagger)^T\,\mathbf{\Sigma}^{-1}\mathbf{U}^T$. Hence

$$\mathbf{W}_{opt} = (\mathbf{V}_1^\dagger)^T\,\mathbf{V}^T\mathbf{V}\,\mathbf{V}_1^\dagger = (\mathbf{V}_1^\dagger)^T\,\mathbf{V}_1^\dagger.$$

Furthermore $\mathsf{rank}(\mathbf{A}_1) = \mathsf{rank}(\mathbf{A})$ implies that $\mathbf{A}_1$ has the same column space as $\mathbf{A}$. Hence the residual in Theorem 2.1 is zero, and $\mathbf{A}_1\mathbf{W}_{opt}\mathbf{A}_1^T = \mathbf{A}\mathbf{A}^T$.

Special case $c = \mathsf{rank}(\mathbf{A}_1) = \mathsf{rank}(\mathbf{A})$ This means $c = k$, so that $\mathbf{V}_1$ is a $k \times k$ matrix. From $\mathsf{rank}(\mathbf{A}) = k$ follows $\mathsf{rank}(\mathbf{V}_1) = k$, so that $\mathbf{V}_1$ is nonsingular and $\mathbf{V}_1^\dagger = \mathbf{V}_1^{-1}$.

2.7.2 Proof of Theorem 2.2

Abbreviate

$$\mathbf{A}_1 \equiv \begin{pmatrix} A_{t_1} & \cdots & A_{t_c} \end{pmatrix}, \qquad \mathbf{V}_1^T \equiv \mathbf{V}^T \begin{pmatrix} e_{t_1} & \cdots & e_{t_c} \end{pmatrix},$$

so that the sum of outer products can be written as $\sum_{j=1}^c w_j \, A_{t_j} A_{t_j}^T = \mathbf{A}_1 \mathbf{W} \mathbf{A}_1^T$, where $\mathbf{W} \equiv \mathsf{diag} \begin{pmatrix} w_1 & \cdots & w_c \end{pmatrix}$.

1. Show: If $\mathbf{A}_1 \mathbf{W} \mathbf{A}_1^T = \mathbf{A}\mathbf{A}^T$ for a diagonal $\mathbf{W}$ with non-negative diagonal, then $\mathbf{V}_1^T \mathbf{W}^{1/2}$ has orthonormal rows From $\mathbf{A}\mathbf{A}^T = \mathbf{A}_1 \mathbf{W} \mathbf{A}_1^T$ follows

$$\mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^T = \mathbf{A}\mathbf{A}^T = \mathbf{A}_1 \mathbf{W} \mathbf{A}_1^T = \mathbf{U}\mathbf{\Sigma}\, \mathbf{V}_1^T \mathbf{W} \mathbf{V}_1\, \mathbf{\Sigma}\mathbf{U}^T. \tag{2.8}$$

Multiplying by $\mathbf{\Sigma}^{-1}\mathbf{U}^T$ on the left and by $\mathbf{U}\mathbf{\Sigma}^{-1}$ on the right gives $\mathbf{I}_k = \mathbf{V}_1^T \mathbf{W} \mathbf{V}_1$. Since $\mathbf{W}$ is positive semi-definite, it has a symmetric positive semi-definite square root $\mathbf{W}^{1/2}$. Hence $\mathbf{I}_k = \mathbf{V}_1^T \mathbf{W} \mathbf{V}_1 = (\mathbf{V}_1^T \mathbf{W}^{1/2})\,(\mathbf{V}_1^T \mathbf{W}^{1/2})^T$, and $\mathbf{V}_1^T \mathbf{W}^{1/2}$ has orthonormal rows.

2. Show: If $\mathbf{V}_1^T \mathbf{W}^{1/2}$ has orthonormal rows, then $\mathbf{A}_1 \mathbf{W} \mathbf{A}_1^T = \mathbf{A}\mathbf{A}^T$ Inserting $\mathbf{I}_k = (\mathbf{V}_1^T \mathbf{W}^{1/2})\,(\mathbf{V}_1^T \mathbf{W}^{1/2})^T = \mathbf{V}_1^T \mathbf{W} \mathbf{V}_1$ into $\mathbf{A}_1 \mathbf{W} \mathbf{A}_1^T$ gives

$$\mathbf{A}_1 \mathbf{W} \mathbf{A}_1^T = \mathbf{U}\mathbf{\Sigma}\, \left(\mathbf{V}_1^T \mathbf{W} \mathbf{V}_1\right)\, \mathbf{\Sigma}\mathbf{U}^T = \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^T = \mathbf{A}\mathbf{A}^T.$$

2.7.3 Proof of Corollary 2.1

Since $\mathsf{rank}(\mathbf{A}) = 1$, the right singular vector matrix $\mathbf{V} = \begin{pmatrix} v_1 & \cdots & v_n \end{pmatrix}^T$ is a $n \times 1$ vector. Since $\mathbf{A}$ has only a single non-zero singular value, $\|A_j\|_2 = \|\mathbf{U}\mathbf{\Sigma}\, v_j\|_2 = \|\mathbf{A}\|_F v_j$. Clearly $A_j \neq 0$ if and only $v_j \neq 0$, and $\|\mathbf{V}^T e_j\|_2^2 = v_j^2 = \|A_j\|_2^2 / \|\mathbf{A}\|_F^2$. Let A_{t_j} be any c non-zero columns of $\mathbf{A}$. Then

$$\sum_{j=1}^c w_j A_{t_j} A_{t_j}^T = \mathbf{U}\mathbf{\Sigma}\, \left(\sum_{j=1}^c w_j v_{t_j}^2\right)\, \mathbf{\Sigma}\mathbf{U}^T = \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^T = \mathbf{A}\mathbf{A}^T$$

if and only if $\sum_{j=1}^c w_j v_{t_j}^2 = 1$. This is true if $w_j = 1/(c v_{t_j}^2)$, $1 \leq j \leq c$.

2.7.4 Proof of Theorem 2.3

Since Theorem 2.3 is a special case of Theorem 2.2, we only need to derive the expression for the weights. From $c = k$ follows that $\mathbf{V}_1^T \mathbf{W}^{1/2}$ is $k \times k$ with orthonormal rows. Hence $\mathbf{V}_1^T \mathbf{W}^{1/2}$ is an orthogonal matrix, and must have orthonormal columns as well, $(\mathbf{W}^{1/2}\mathbf{V}_1)\,(\mathbf{W}^{1/2}\mathbf{V}_1)^T = \mathbf{I}_k$. Thus

$$\mathbf{V}_1 \mathbf{V}_1^T = \mathsf{diag}\left(\left\|\mathbf{V}^T e_{t_1}\right\|_2^2 \quad \dots \quad \left\|\mathbf{V}^T e_{t_c}\right\|_2^2 \right) = \mathbf{W}^{-1}.$$

This and $\mathbf{W}^{1/2}$ being diagonal implies $w_j = 1/\left\|\mathbf{V}^T e_{t_j}\right\|_2^2$.

2.7.5 Proof of Theorem 2.5

We present two auxiliary results, a matrix Bernstein concentration inequality (Theorem 2.12) and a bound for the singular values of a difference of positive semi-definite matrices (Theorem 2.13), before deriving a probabilistic bound (Theorem 2.14). The subsequent combination of Theorem 2.14 and the invariance of the two-norm under unitary transformations yields Theorem 2.15 which, at last, leads to a proof for the desired Theorem 2.5.

Theorem 2.12 (Theorem 1.4 in [99])**.** *Let $\mathbf{X}_j$ be c independent real symmetric random $m \times m$ matrices. Assume that, with probability one, $\mathbb{E}[\mathbf{X}_j] = \mathbf{0}$, $1 \le j \le c$ and $\max_{1 \le j \le c} \|\mathbf{X}_j\|_2 \le \rho_1$. Let $\left\|\sum_{j=1}^c \mathbb{E}[\mathbf{X}_j^2]\right\|_2 \le \rho_2$.*

Then for any $\epsilon \ge 0$

$$\mathbb{P}\left[\left\| \sum_{j=1}^{c} \mathbf{X}_j \right\|_2 \ge \epsilon \right] \le m \, \exp\left(-\frac{\epsilon^2/2}{\rho_2 + \rho_1 \epsilon/3} \right).$$

Theorem 2.13 (Theorem 2.1 in [102])**.** *If $\mathbf{B}$ and $\mathbf{C}$ are $m \times m$ real symmetric positive semi-definite matrices, with singular values $\sigma_1(\mathbf{B}) \ge \ldots \ge \sigma_m(\mathbf{B})$ and $\sigma_1(\mathbf{C}) \ge \ldots \ge \sigma_m(\mathbf{C})$, then the singular values of the difference are bounded by*

$$\sigma_j(\mathbf{B} - \mathbf{C}) \le \sigma_j \begin{pmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{0} & \mathbf{C} \end{pmatrix}, \qquad 1 \le j \le m.$$

In particular, $\|\mathbf{B} - \mathbf{C}\|_2 \le \max\{\|\mathbf{B}\|_2, \|\mathbf{C}\|_2\}$.

Theorem 2.14. *Let $\mathbf{A} \ne \mathbf{0}$ be an $m \times n$ matrix, and let $\mathbf{X}$ be computed by Algorithm 1 with the "nearly optimal" probabilites p_j^β in (2.2).*

For any $\delta > 0$, with probability at least $1 - \delta$,

$$\frac{\left\|\mathbf{X} - \mathbf{A}\mathbf{A}^T\right\|_2}{\|\mathbf{A}\mathbf{A}^T\|_2} \leq \gamma_0 + \sqrt{\gamma_0\,(6 + \gamma_0)}, \qquad \textit{where} \quad \gamma_0 \equiv \mathsf{sr}(\mathbf{A})\,\frac{\ln(m/\delta)}{3\,\beta\,c}.$$

Proof. In order to apply Theorem 2.12, we need to change variables, and check that the assumptions are satisfied.

1. Change of variables Define the $m \times m$ real symmetric matrix random variables $\mathbf{Y}_j \equiv \frac{1}{c\,p_{t_j}}\,A_{t_j}A_{t_j}^T$, and write the output of Algorithm 1 as

$$\mathbf{X} = (\mathbf{AS})\,(\mathbf{AS})^T = \mathbf{Y}_1 + \cdots + \mathbf{Y}_c.$$

Since $\mathbb{E}[\mathbf{Y}_j] = \mathbf{A}\mathbf{A}^T/c$, but Theorem 2.12 requires random variables with zero mean, set $\mathbf{X}_j \equiv \mathbf{Y}_j - \frac{1}{c}\mathbf{A}\mathbf{A}^T$. Then

$$\mathbf{X} - \mathbf{A}\mathbf{A}^T = (\mathbf{AS})\,(\mathbf{AS})^T - \mathbf{A}\mathbf{A}^T = \sum_{j=1}^{c}\left(\mathbf{Y}_j - \frac{1}{c}\,\mathbf{A}\mathbf{A}^T\right) = \sum_{j=1}^{c}\mathbf{X}_j.$$

Hence, we show $\left\|\mathbf{X} - \mathbf{A}\mathbf{A}^T\right\|_2 \leq \epsilon$ by showing $\left\|\sum_{j=1}^{c}\mathbf{X}_j\right\|_2 \leq \epsilon$.

Next we have to check that the assumptions of Theorem 2.12 are satisfied. In order to derive bounds for $\max_{1\leq j\leq c}\|\mathbf{X}_j\|_2$ and $\left\|\sum_{j=1}^{c}\mathbb{E}[\mathbf{X}_j^2]\right\|_2$, we assume general non-zero probabilities p_j for the moment, that is, $p_j > 0$, $1 \leq j \leq n$.

2. Bound for $\max_{1\leq j\leq c}\|\mathbf{X}_j\|_2$ Since $\mathbf{X}_j$ is a difference of positive semidefinite matrices, apply Theorem 2.13 to obtain

$$\|\mathbf{X}_j\|_2 \leq \max\left\{\|\mathbf{Y}_j\|_2,\, \tfrac{1}{c}\left\|\mathbf{A}\mathbf{A}^T\right\|_2\right\} \leq \frac{\hat{\rho}_1}{c}, \qquad \hat{\rho}_1 \equiv \max_{1\leq i\leq n}\left\{\frac{\|A_i\|_2^2}{p_i},\, \|\mathbf{A}\|_2^2\right\}.$$

3. Bound for $\left\|\sum_{j=1}^{c}\mathbb{E}[\mathbf{X}_j^2]\right\|_2$ To determine the expected value of

$$\mathbf{X}_j^2 = \mathbf{Y}_j^2 - \tfrac{1}{c}\,\mathbf{A}\mathbf{A}^T\,\mathbf{Y}_j - \tfrac{1}{c}\mathbf{Y}_j\,\mathbf{A}\mathbf{A}^T + \tfrac{1}{c^2}(\mathbf{A}\mathbf{A}^T)^2$$

use the linearity of the expected value and $\mathbb{E}[\mathbf{Y}_j] = \mathbf{A}\mathbf{A}^T/c$ to obtain

$$\mathbb{E}[\mathbf{X}_j^2] = \mathbb{E}[\mathbf{Y}_j^2] - \frac{1}{c^2}\,(\mathbf{A}\mathbf{A}^T)^2.$$

Applying the definition of expected value again yields

$$\mathbb{E}[\mathbf{Y}_j^2] = \frac{1}{c^2} \sum_{i=1}^{n} p_i \frac{(A_i A_i^T)^2}{p_i^2} = \frac{1}{c^2} \sum_{i=1}^{n} \frac{(A_i A_i^T)^2}{p_i}.$$

Hence

$$\begin{aligned} \sum_{j=1}^{c} \mathbb{E}[\mathbf{X}_j^2] &= \frac{1}{c} \left(\sum_{i=1}^{n} \frac{(A_i A_i^T)^2}{p_i} - (\mathbf{A}\mathbf{A}^T)^2 \right) = \frac{1}{c} \mathbf{A} \left(\sum_{i=1}^{n} e_i \frac{\|A_i\|_2^2}{p_i} e_i^T - \mathbf{A}^T \mathbf{A} \right) \mathbf{A}^T \\ &= \frac{1}{c} \mathbf{A} \left(\mathbf{L} - \mathbf{A}^T \mathbf{A} \right) \mathbf{A}^T, \end{aligned}$$

where $\mathbf{L} \equiv \mathsf{diag}\left(\|A_1\|_2^2/p_1 \quad \dots \quad \|A_n\|_2^2/p_n \right)$. Taking norms and applying Theorem 2.13 to $\|\mathbf{L} - \mathbf{A}^T\mathbf{A}\|_2$ gives

$$\left\| \sum_{j=1}^{c} \mathbb{E}[\mathbf{X}_j^2] \right\|_2 \leq \frac{\|\mathbf{A}\|_2^2}{c} \max\left\{ \|\mathbf{L}\|_2, \|\mathbf{A}\|_2^2 \right\} = \frac{\|\mathbf{A}\|_2^2}{c} \hat{\rho}_1.$$

4. Application of Theorem 2.12 The required upper bounds for Theorem 2.12 are

$$\|\mathbf{X}_j\|_2 \leq \rho_1 \equiv \frac{\hat{\rho}_1}{c} \qquad \textit{and} \qquad \left\| \sum_{j=1}^{c} \mathbb{E}[\mathbf{X}_j^2] \right\|_2 \leq \rho_2 \equiv \frac{\|\mathbf{A}\|_2^2}{c} \hat{\rho}_1.$$

Inserting these bounds into Theorem 2.12 gives

$$\mathbb{P}\left[\left\| \sum_{j=1}^{c} \mathbf{X}_j \right\|_2 > \epsilon \right] \leq m \exp\left(\frac{-c\epsilon^2}{2\hat{\rho}_1 \left(\|\mathbf{A}\|_2^2 + \epsilon/3 \right)} \right).$$

Hence $\left\| \sum_{j=1}^{c} \mathbf{X}_j \right\|_2 \leq \epsilon$ with probability at least $1 - \delta$, where

$$\delta \equiv m \exp\left(\frac{-c\epsilon^2}{2\hat{\rho}_1 \left(\|\mathbf{A}\|_2^2 + \epsilon/3 \right)} \right).$$

Solving for ϵ gives

$$\epsilon = \tau_1 \hat{\rho}_1 + \sqrt{\tau_1 \hat{\rho}_1 \left(6\|\mathbf{A}\|_2^2 + \tau_1 \hat{\rho}_1 \right)}, \qquad \tau_1 \equiv \frac{\ln(m/\delta)}{3c}.$$

5. Specialization to "nearly optimal" probabilities We remove zero columns from the matrix. This does not change the norm or the stable rank. The "nearly optimal" probabilities

for the resulting submatrix are $p_j^\beta = \beta \|A_j\|_2^2 / \|\mathbf{A}\|_F^2$, with $p_j > 0$ for all j. Now replace p_j^β by their lower bounds (2.2). This gives $\hat{\rho}_1 \leq \|\mathbf{A}\|_2^2 \, \tau_2$ where $\tau_2 \equiv \mathsf{sr}(\mathbf{A})/\beta \geq 1$, and

$$\epsilon \leq \|\mathbf{A}\|_2^2 \left(\tau_1 \tau_2 + \sqrt{\tau_1 \tau_2 \ (6 + \tau_1 \tau_2)} \right).$$

Finally observe that $\gamma_0 = \tau_1 \tau_2$, and divide by $\|\mathbf{A}\|_2^2 = \|\mathbf{A}\mathbf{A}^T\|_2$. □

We make Theorem 2.14 tighter and replace the dimension m by $\mathsf{rank}(\mathbf{A})$. The idea is to apply Theorem 2.14 to the $k \times k$ matrix $(\mathbf{\Sigma}\mathbf{V}^T)\,(\mathbf{\Sigma}\mathbf{V}^T)^T$ instead of the $m \times m$ matrix $\mathbf{A}\mathbf{A}^T$.

Theorem 2.15. *Let $\mathbf{A} \neq \mathbf{0}$ be an $m \times n$ matrix, and let $\mathbf{X}$ be computed by Algorithm 1 with the "nearly optimal" probabilites p_j^β in (2.2).*

For any $\delta > 0$, with probability at least $1 - \delta$,

$$\frac{\left\| \mathbf{X}\mathbf{X}^T - \mathbf{A}\mathbf{A}^T \right\|_2}{\|\mathbf{A}\mathbf{A}^T\|_2} \leq \gamma_1 + \sqrt{\gamma_1 \, (6 + \gamma_1)}, \qquad \textit{where} \quad \gamma_1 \equiv \mathsf{sr}(\mathbf{A}) \, \frac{\ln \left(\mathsf{rank}(\mathbf{A})/\delta \right)}{3 \, \beta \, c}.$$

Proof. The invariance of the two-norm under unitary transformations implies

$$\|\mathbf{X}\mathbf{X}^T - \mathbf{A}\mathbf{A}^T\|_2 = \left\| (\mathbf{\Sigma}\mathbf{V}^T\mathbf{S})\,(\mathbf{\Sigma}\mathbf{V}^T\mathbf{S})^T - (\mathbf{\Sigma}\mathbf{V}^T)\,(\mathbf{\Sigma}\mathbf{V}^T)^T \right\|_2.$$

Apply Theorem 2.14 to the $k \times n$ matrix $B \equiv \mathbf{\Sigma}\mathbf{V}^T$ with probabilities

$$p_j^\beta \geq \beta \, \frac{\|A_j\|_2^2}{\|\mathbf{A}\|_F^2} = \beta \, \frac{\|B_j\|_2^2}{\|\mathbf{B}\|_F^2}.$$

□

Note that Algorithm 1 is still applied to the original matrix $\mathbf{A}$, with probabilities (2.2) computed from $\mathbf{A}$. It is only the bound that has changed.

Proof of Theorem 2.5

At last, we set $\gamma_1 + \sqrt{\gamma_1 \, (6 + \gamma_1)} \leq \epsilon$ and solve for c as follows. In $\gamma_1 + \sqrt{\gamma_1 \, (6 + \gamma_1)}$, write

$$\gamma_1 = \tfrac{\ln \left(\mathsf{rank}(\mathbf{A})/\delta \right)}{3 \, \beta \, c} \, \mathsf{sr}(\mathbf{A}) = \tfrac{t}{3c}, \qquad \text{where} \quad t \equiv \frac{\ln \left(\mathsf{rank}(\mathbf{A})/\delta \right) \mathsf{sr}(\mathbf{A})}{\beta}.$$

We want to determine $\alpha > 0$ so that $c = \alpha t / \epsilon^2$ satisfies

$$\gamma_1 + \sqrt{\gamma_1 \, (6 + \gamma_1)} = \frac{t}{3c} + \sqrt{\frac{t}{3c} \left(6 + \frac{t}{3c} \right)} \leq \epsilon.$$

Solving for α gives $\alpha \geq 2 + 2\epsilon/3 = c_0(\epsilon)$.

2.7.6 Proof of Theorem 2.6

To start with, we need a matrix Bernstein concentration inequality, along with the the Löwner partial ordering [64, Section 7.7]. and the instrinsic dimension [97, Section 7].

If $\mathbf{A}_1$ and $\mathbf{A}_2$ are $m \times m$ real symmetric matrices, then $\mathbf{A}_1 \preceq \mathbf{A}_2$ means that $\mathbf{A}_2 - \mathbf{A}_1$ is positive semi-definite [64, Definition 7.7.1]. The *intrinsic dimension* of a $m \times m$ symmetric positive semi-definite matrix $\mathbf{A}$ is [97, Definition 7.1.1]:

$$\mathsf{intdim}(\mathbf{A}) \equiv \mathsf{trace}(\mathbf{A}) / \left\|\mathbf{A}\right\|_2,$$

where $1 \leq \mathsf{intdim}(\mathbf{A}) \leq \mathsf{rank}(\mathbf{A}) \leq m$.

Theorem 2.16 (Theorem 7.3.1 and (7.3.2) in [97])**.** *Let $\mathbf{X}_j$ be c independent real symmetric random matrices, with $\mathbb{E}[\mathbf{X}_j] = \mathbf{0}$, $1 \leq j \leq c$. Let $\max_{1 \leq j \leq c} \|\mathbf{X}_j\|_2 \leq \rho_1$, and let $\mathbf{P}$ be a symmetric positive semi-definite matrix so that $\sum_{j=1}^{c} \mathbb{E}[\mathbf{X}_j^2] \preceq \mathbf{P}$. Then for any $\epsilon \geq \|\mathbf{P}\|_2^{1/2} + \rho_1/3$*

$$\mathbb{P}\left[\left\|\sum_{j=1}^{c} \mathbf{X}_j\right\|_2 \geq \epsilon\right] \leq 4\,\mathsf{intdim}(\mathbf{P})\,\exp\left(\frac{-\epsilon^2/2}{\|\mathbf{P}\|_2 + \rho_1\epsilon/3}\right).$$

Now we apply the above theorem to sampling with "nearly optimal" probabilities.

Theorem 2.17. *Let $\mathbf{A} \neq \mathbf{0}$ be an $m \times n$ matrix, and let $\mathbf{X}$ be computed by Algorithm 1 with the "nearly optimal" probabilities p_j^{β} in (2.2).*

For any $0 < \delta < 1$, with probability at least $1 - \delta$,

$$\frac{\left\|\mathbf{X} - \mathbf{A}\mathbf{A}^T\right\|_2}{\|\mathbf{A}\mathbf{A}^T\|_2} \leq \gamma_2 + \sqrt{\gamma_2\,(6 + \gamma_2)}, \qquad \textit{where} \quad \gamma_2 \equiv \mathsf{sr}(\mathbf{A})\,\frac{\ln\left(4\mathsf{sr}(\mathbf{A})/\delta\right)}{3\,\beta\,c}.$$

Proof. In order to apply Theorem 2.16, we need to change variables, and check that the assumptions are satisfied.

1. Change of variables As in item 1 of the proof of Theorem 2.14, we define the real symmetric matrix random variables $\mathbf{Y}_j \equiv \frac{1}{c\,p_{t_j}}\,A_{t_j}A_{t_j}^T$, and write the output of Algorithm 1 as

$$\mathbf{X} = (\mathbf{A}\mathbf{S})\,(\mathbf{A}\mathbf{S})^T = \mathbf{Y}_1 + \cdots + \mathbf{Y}_c.$$

The zero mean versions are $\mathbf{X}_j \equiv \mathbf{Y}_j - \frac{1}{c}\mathbf{A}\mathbf{A}^T$, so that $\mathbf{X} - \mathbf{A}\mathbf{A}^T = \sum_{j=1}^{c} \mathbf{X}_j$.

Next we have to check that the assumptions of Theorem 2.16 are satisfied, for the "nearly optimal" probabilities $p_j^\beta = \beta\|A_j\|_2^2/\|\mathbf{A}\|_F^2$. Since Theorem 2.16 does not depend on the matrix dimensions, we can assume that all zero columns of $\mathbf{A}$ have been removed, so that all $p_j^\beta > 0$.

2. Bound for $\max_{1\le j\le c}\|\mathbf{X}_j\|_2$ From item 2 in the proof of Theorem 2.14 follows $\|\mathbf{X}_j\|_2 \le \rho_1$, where

$$\rho_1 = \frac{1}{c}\max_{1\le j\le n}\left\{\frac{\|A_j\|_2^2}{p_j^\beta},\ \|\mathbf{A}\|_2^2\right\} \le \frac{\|\mathbf{A}\|_F^2}{\beta c}.$$

3. The matrix P From item 3 in the proof of Theorem 2.14 follows

$$\sum_{j=1}^{c}\mathbb{E}[\mathbf{X}_j^2] = \tfrac{1}{c}\mathbf{A}\mathbf{L}\mathbf{A}^T - \tfrac{1}{c}\mathbf{A}\mathbf{A}^T\mathbf{A}\mathbf{A}^T,$$

where $\mathbf{L} \equiv \mathsf{diag}\begin{pmatrix}\|A_1\|_2^2/p_1^\beta & \cdots & \|A_n\|_2^2/p_n^\beta\end{pmatrix} \preceq (\|\mathbf{A}\|_F^2/\beta)\,\mathbf{I}_n$. Since $\mathbf{A}\mathbf{A}^T\mathbf{A}\mathbf{A}^T$ is positive semi-definite, so is

$$\tfrac{1}{c}\mathbf{A}\mathbf{A}^T\mathbf{A}\mathbf{A}^T = \tfrac{1}{c}\mathbf{A}\mathbf{L}\mathbf{A}^T - \tfrac{1}{c}\left(\mathbf{A}\mathbf{L}\mathbf{A}^T - \mathbf{A}\mathbf{A}^T\mathbf{A}\mathbf{A}^T\right) = \tfrac{1}{c}\mathbf{A}\mathbf{L}\mathbf{A}^T - \sum_{j=1}^{c}\mathbb{E}[\mathbf{X}_j^2].$$

Thus, $\sum_{j=1}^{c}\mathbb{E}[\mathbf{X}_j^2] \preceq \frac{1}{c}\mathbf{A}\mathbf{L}\mathbf{A}^T \preceq \frac{\|\mathbf{A}\|_F^2}{\beta c}\mathbf{A}\mathbf{A}^T$, where the the second inequality follows from [64, Theorem 7.7.2(a)]. Set $\mathbf{P} \equiv \frac{\|\mathbf{A}\|_F^2}{\beta c}\mathbf{A}\mathbf{A}^T$. Then

$$\|\mathbf{P}\|_2 = \frac{\|\mathbf{A}\|_2^2\|\mathbf{A}\|_F^2}{\beta c} \qquad \textit{and} \qquad \mathsf{intdim}(\mathbf{P}) = \frac{\|\mathbf{A}\|_F^4}{\|\mathbf{A}\|_F^2\,\|\mathbf{A}\|_2^2} = \mathsf{sr}(\mathbf{A}).$$

4. Application of Theorem 2.16 Substituting the above expressions for $\|\mathbf{P}\|_2$, $\mathsf{intdim}(\mathbf{P})$ and $\rho_1 = \frac{\|\mathbf{A}\|_F^2}{\beta\, c}$ into Theorem 2.16 gives

$$\mathbb{P}\left[\left\|\sum_{j=1}^{c}\mathbf{X}_j\right\|_2 \ge \epsilon\right] \le 4\,\mathsf{sr}(\mathbf{A})\,\exp\left(\frac{-\epsilon^2\beta c}{2\,\|\mathbf{A}\|_F^2\left(\|\mathbf{A}\|_2^2 + \epsilon/3\right)}\right).$$

Hence $\left\|\sum_{j=1}^{c}\mathbf{X}_j\right\|_2 \le \epsilon$ with probability at least $1-\delta$, where

$$\delta \equiv 4\,\mathsf{sr}(\mathbf{A})\,\exp\left(\frac{-\epsilon^2\beta c}{2\,\|\mathbf{A}\|_F^2\left(\|\mathbf{A}\|_2^2 + \epsilon/3\right)}\right).$$

Solving for ϵ gives

$$\epsilon = \hat{\gamma}_2 + \sqrt{\hat{\gamma}_2 \left(6 \left\|\mathbf{A}\right\|_2^2 + \hat{\gamma}_2\right)}, \qquad \text{where} \quad \hat{\gamma}_2 \equiv \left\|\mathbf{A}\right\|_F^2 \frac{\ln(4\,\mathsf{sr}(\mathbf{A})/\delta)}{3\beta c} = \left\|\mathbf{A}\right\|_2^2 \gamma_2.$$

It remains to show the last requirement of Theorem 2.16, that is, $\epsilon \geq \left\|\mathbf{P}\right\|_2^{1/2} + \rho_1/3$. Replacing ϵ by its above expression in terms of $\hat{\gamma}_2$ shows that the requirement is true if $\hat{\gamma}_2 \geq \rho_1/3$ and $\sqrt{6\|\mathbf{A}\|_2^2 \hat{\gamma}_2} \geq \left\|\mathbf{P}\right\|_2^{1/2}$. This is the case if $\ln(4\,\mathsf{sr}(\mathbf{A})/\delta) > 1$. Since $\mathsf{sr}(\mathbf{A}) \geq 1$, this is definitely true if $\delta < 4/e$. Since we assumed $\delta < 1$ from the start, the requirement is fulfilled automatically.

At last, divide both sides of $\left\|\mathbf{X} - \mathbf{A}\mathbf{A}^T\right\|_2 \leq \hat{\gamma}_2 + \sqrt{\hat{\gamma}_2 \left(6 \left\|\mathbf{A}\right\|_2^2 + \hat{\gamma}_2\right)}$ by $\left\|\mathbf{A}\mathbf{A}^T\right\|_2 = \left\|\mathbf{A}\right\|_2^2$. □

Proof of Theorem 2.6

As in the proof of Theorem 2.5, solve for c in $\gamma_2 + \sqrt{\gamma_2 \left(6 + \gamma_2\right)} \leq \epsilon$.

2.7.7 Proof of Theorem 2.7

To get a relative error bound, substitute the thin SVD $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ into

$$\begin{aligned}
\|\mathbf{X} - \mathbf{A}\mathbf{A}^T\|_2 &= \|(\mathbf{A}\mathbf{S})\,(\mathbf{A}\mathbf{S})^T - \mathbf{A}\mathbf{A}^T\|_2 = \|(\mathbf{\Sigma}\mathbf{V}^T\mathbf{S})\,(\mathbf{\Sigma}\mathbf{V}^T\mathbf{S})^T - \mathbf{\Sigma}\mathbf{V}^T\mathbf{V}\mathbf{\Sigma}\|_2 \\
&\leq \|\mathbf{\Sigma}\|_2^2\,\|(\mathbf{V}^T\mathbf{S})\,(\mathbf{V}^T\mathbf{S})^T - \mathbf{V}^T\mathbf{V}\|_2 \\
&= \|\mathbf{A}\mathbf{A}^T\|_2\,\|(\mathbf{V}^T\mathbf{S})\,(\mathbf{V}^T\mathbf{S})^T - \mathbf{V}^T\mathbf{V}\|_2.
\end{aligned}$$

The last term can be viewed as sampling columns from $\mathbf{V}^T$ to approximate the product $\mathbf{V}^T\mathbf{V} = \mathbf{I}_n$. Now apply Theorem 2.5, where $\|\mathbf{V}\|_F^2 = k = \mathsf{rank}(\mathbf{A})$ and $\|\mathbf{V}\|_2^2 = 1$, so that $\mathsf{sr}(\mathbf{V}) = k = \mathsf{rank}(\mathbf{A})$.

2.7.8 Proof of Theorem 2.8

We present separate proofs for the two types of sampling probabilities.

Sampling with "nearly optimal" probabilities Applying Theorem 2.5 shows that

$$\left\|\mathbf{Q}\mathbf{Q}^T - (\mathbf{Q}\mathbf{S})\,(\mathbf{Q}\mathbf{S})^T\right\|_2 \leq \epsilon$$

with probability at least $1 - \delta$, if $c \geq c_0(\epsilon)\,\frac{m}{\beta\epsilon^2}\,\ln(m/\delta)$.

Sampling with uniform probabilities Use the β factor to express the uniform probabilities as "nearly optimal" probabilities,

$$\frac{1}{n} = \frac{m}{n\,\mu}\,\frac{\mu}{m} \geq \frac{m}{n\,\mu}\,\frac{\|Q_j\|_2^2}{\|\mathbf{Q}\|_F^2} = \beta\,\frac{\|Q_j\|_2^2}{\|\mathbf{Q}\|_F^2} = \beta\,p_j^{opt} \qquad 1 \leq j \leq n.$$

Now apply Theorem 2.5 with $\beta = m/(n\mu)$.

For both sampling methods, the connection (2.6) implies that $\sigma_m(\mathbf{QS}) \geq \sqrt{1-\epsilon}$ with probability at least $1-\delta$.

2.7.9 Proof of Theorem 2.9

First we present the concentration inequality on which the proof is based. Below $\lambda_{min}(\mathbf{X})$ and $\lambda_{max}(\mathbf{X})$ denote the smallest and largest eigenvalues, respectively, of the symmetric positive semi-definite matrix $\mathbf{X}$.

Theorem 2.18 (Theorem 5.1.1 in [97]). *Let $\mathbf{X}_j$ be c independent $m \times m$ real symmetric positive semi-definite random matrices, with $\max_{1\leq j\leq c} \|\mathbf{X}_j\|_2 \leq \rho$. Define*

$$\rho_{max} \equiv \lambda_{max}\left(\mathbb{E}\left[\sum_{j=1}^{c}\mathbf{X}_j\right]\right), \qquad \rho_{min} \equiv \lambda_{min}\left(\mathbb{E}\left[\sum_{j=1}^{c}\mathbf{X}_j\right]\right),$$

and $f(x) \equiv e^x/(1+x)^{1+x}$. Then, for any $0 < \epsilon < 1$

$$\mathbb{P}\left[\lambda_{min}\left(\sum_{j=1}^{c}\mathbf{X}_j\right) \leq (1-\epsilon)\rho_{min}\right] \leq m\,f(-\epsilon)^{\rho_{min}/\rho},$$

and

$$\mathbb{P}\left[\lambda_{max}\left(\sum_{j=1}^{c}\mathbf{X}_j\right) \geq (1+\epsilon)\rho_{max}\right] \leq m\,f(\epsilon)^{\rho_{max}/\rho}.$$

Proof of Theorem 2.9

Write $(\mathbf{QS})(\mathbf{QS})^T = \sum_{j=1}^{c}\mathbf{X}_j$, where $\mathbf{X}_j \equiv \frac{Q_{t_j}Q_{t_j}^T}{c\,p_{t_j}}$. To apply Theorem 2.18 we need to compute ρ, ρ_{min}, and ρ_{max}.

Sampling with "nearly optimal" probabilities The definition of "nearly optimal" probabilities (2.2) and the fact that $\|\mathbf{Q}\|_F^2 = m$ imply $\|\mathbf{X}_j\|_2 = \frac{\|Q_{t_j}\|_2^2}{cp_{t_j}^{\beta}} \leq \frac{m}{c\,\beta}$. Hence we can set

$\rho \equiv \frac{m}{c\,\beta}$. The definition of $\mathbf{X}_j$ implies

$$\mathbb{E}\left[\sum_{j=1}^{c} X_{t_j}\right] = \frac{1}{c}\sum_{j=1}^{c}\sum_{i=1}^{n} Q_i Q_i^T = \mathbf{Q}\mathbf{Q}^T = \mathbf{I}_m,$$

so that $\rho_{min} = 1$. Now apply Theorem 2.18 to conclude

$$\mathbb{P}\left[\lambda_{min}\left(\sum_{j=1}^{c} X_j\right) \leq (1-\epsilon)\right] \leq m f(-\epsilon)^{c\beta/m}.$$

Setting the right hand side equal to δ and solving for c gives

$$c = \frac{m}{\beta}\,\frac{\ln(\delta/m)}{\ln f(-\epsilon)} = c_1(\epsilon)\, m\, \frac{\ln(m/\delta)}{\beta\epsilon^2},$$

where the second equality follows from $\ln f(x) = x - (1+x)\ln(1+x)$. The function $c_1(x)$ is decreasing in $[0,1]$, and L'Hôpital's rule implies that $c_1(\epsilon) \to 2$ as $\epsilon \to 0$ and $c_1(\epsilon) \to 1$ as $\epsilon \to 1$.

Sampling with uniform probabilities An analogous proof with $p_j = 1/n$ shows that $\|\mathbf{X}_j\|_2 \leq \rho \equiv n\mu/c$.

Uniform sampling without replacement Theorem 2.18 also holds when the matrices $\mathbf{X}_j$ are sampled uniformly without replacement [98, Theorem 2.2].

For all three sampling methods, the connection (2.6) implies that $\sigma_m(\mathbf{QS}) \geq \sqrt{1-\epsilon}$ with probability at least $1-\delta$.

2.7.10 Proof of Theorem 2.10

The proof follows from Theorem 2.8, and the connection (2.6), since $|1-\sigma_j^2(\mathbf{QS})| \leq \epsilon, 1 \leq j \leq m$, implies that both, $\sigma_m(\mathbf{QS}) \geq \sqrt{1-\epsilon}$ and $\sigma_1(\mathbf{QS}) \leq \sqrt{1+\epsilon}$.

2.7.11 Proof of Theorem 2.11

We derive separate bounds for the smallest and largest singular values of $\mathbf{QS}$.

Sampling with "nearly optimal" probabilities The proof Theorem 2.9 implies that

$$\mathbb{P}\left[\lambda_{min}\left(\sum_{j=1}^{c} X_j\right) \leq (1-\epsilon)\right] \leq m f(-\epsilon)^{c\beta/m}.$$

Similarly, we can apply Theorem 2.18 with $\rho_{max} = 1$ to conclude

$$\mathbb{P}\left[\lambda_{max}\left(\sum_{j=1}^{c} X_j\right) \geq (1+\epsilon)\right] \leq m f(\epsilon)^{c\beta/m}.$$

Since $f(-\epsilon) \leq f(\epsilon)$, Boole's inequality implies

$$\mathbb{P}\left[\lambda_{min}\left(\sum_{j=1}^{c} X_j\right) \leq (1-\epsilon) \text{ and } \lambda_{max}\left(\sum_{j=1}^{c} X_j\right) \geq (1+\epsilon)\right] \leq 2m f(\epsilon)^{c\beta/m}.$$

Hence, $\sigma_m(\mathbf{QS}) \geq \sqrt{1-\epsilon}$ and $\sigma_1(\mathbf{QS}) \leq \sqrt{1+\epsilon}$ hold simultaneously with probability at least $1-\delta$, if

$$c \geq c_2(\epsilon)\, m\, \frac{\ln(2m/\delta)}{\beta\epsilon^2}.$$

This bound for c also ensures that $\kappa(\mathbf{QS}) \leq \frac{\sqrt{1+\epsilon}}{\sqrt{1-\epsilon}}$ with probability at least $1-\delta$. The function $c_2(x)$ is increasing in $[0,1]$, and L'Hôpital's rule implies that $c_2(\epsilon) \to 2$ as $\epsilon \to 0$ and $c_2(\epsilon) \to 1/(2\ln(2)-1) \leq 2.6$ as $\epsilon \to 1$.

Uniform sampling, with or without replacement The proof is analogous to the corresponding part of the proof Theorem 2.9.

Chapter 3

Perturbation of Leverage Scores

3.1 Introduction

Leverage scores are scalar quantities associated with the column space of a matrix, and can be computed from the rows of *any* orthonormal basis for this space.

Leverage scores

We restrict our discussion here to leverage scores of full column rank matrices.

Definition 3.1. *Let* $\mathbf{A}$ *be a real* $m \times n$ *matrix with* $rank(\mathbf{A}) = n$*. If* $\mathbf{Q}$ *is any* $m \times n$ *matrix whose columns form an orthonormal basis for* range $(\mathbf{A})$*, then the leverage scores of* $\mathbf{A}$ *are*

$$\ell_j \equiv \left\| e_j^T \mathbf{Q} \right\|_2^2, \qquad 1 \le j \le m.$$

Here e_j *denotes the* j*th column of the* $m \times m$ *identity matrix, and* $e_j^T \mathbf{Q}$ *denotes the* j*th row of* $\mathbf{Q}$*.*

Note that leverage scores are independent of the orthonormal basis, since

$$\left\| e_j^T \mathbf{Q} \right\|_2^2 = e_j^T \mathbf{Q}\mathbf{Q}^T e_j = (\mathbf{Q}\mathbf{Q}^T)_{jj}, \quad 1 \le j \le m$$

and $\mathbf{Q}\mathbf{Q}^T$ is the unique orthogonal projector onto range $(\mathbf{A})$.

The basic properties of leverage scores are

$$0 \le \ell_j \le 1, \qquad 1 \le j \le m, \qquad \text{and} \qquad \sum_{j=1}^{m} \ell_j = n.$$

Hoaglin and Welsch introduced statistical leverage scores in 1978 to detect outliers in regression problems [60, Section 2], [66, Section 5.1], [100, Section 2.2]. About thirty years later, Mahoney,

Drineas and their coauthors started to advocate the use of leverage scores in randomized matrix algorithms [38, 40, 79]. More specifically, leverage scores are the basis for importance sampling strategies, in the context of low rank approximations [40], CUR decompositions [41], subset selection [15], Nyström approximations [96], least squares problems [38], and matrix completion [16], to name just a few. Leverage scores also play a crucial role in the analysis of randomized algorithms [66], and fast algorithms have been developed for their approximation [37, 72, 75].

Motivation

Since leverage scores depend only on the column space, and are not tied to any particular orthonormal basis, the question is how to compute them. Many existing papers, among them the survey monograph [79, Definition 1], define leverage scores as row norms of a thin left singular vector matrix. However, the sensitivity of singular vectors is determined by the corresponding singular value gaps.

This, and the fact that QR decompositions, when implemented via Householder transformations or Givens rotations, are numerically stable [59, Sections 19.1–19.7], motivated us to investigate QR decompositions for the computation of leverage scores. In this chapter, we derive bounds on the difference between the leverage scores of a matrix $\mathbf{A}$ and a perturbation $\mathbf{A} + \mathbf{\Delta A}$, when the leverage scores are computed from a QR decomposition. Note that we do not assume a particular implementation of the QR decomposition and assume that quantities are computed in exact arithmetic. We consider our results to be a first step towards determining whether computing leverage scores with a QR decomposition is numerically stable. Since most of our bounds do not exploit the zero structure of the upper triangular factor, they can be readily extended to polar decompositions.

3.1.1 Overview

We present a short overview of the contents of the chapter and the main results. For brevity, we display only the first order terms in the bounds, and omit the technical assumptions.

Notation

Matrices $\mathbf{A}$ always appear in boldface. The $m \times m$ identity matrix is $\mathbf{I}_m$, with columns e_j and rows e_j^T, $1 \leq j \leq m$.

For a real $m \times n$ matrix $\mathbf{A}$ with $\mathsf{rank}(\mathbf{A}) = n$, the two-norm condition number with respect to left inversion is $\kappa_2(\mathbf{A}) \equiv \|\mathbf{A}\|_2 \|\mathbf{A}^\dagger\|_2$, where $\mathbf{A}^\dagger$ is the Moore-Penrose inverse. The stable rank is $\mathrm{sr}(\mathbf{A}) \equiv \|\mathbf{A}\|_F^2 / \|\mathbf{A}\|_2^2$, where $\mathrm{sr}(\mathbf{A}) \leq \mathsf{rank}(\mathbf{A})$.

We denote the leverage scores of a perturbed matrix $\mathbf{A}+\mathbf{\Delta A}$ by $\widetilde{\ell}_j$ and refer to the quantities $|\widetilde{\ell}_j - \ell_j|$ and $|\widetilde{\ell}_j - \ell_j|/\ell_j$ as the the absolute leverage score difference and relative leverage

score difference, respectively. We assume, tacitly, that relative leverage score difference bounds $|\widetilde{\ell}_j - \ell_j|/\ell_j$ apply only for $\ell_j > 0$.

Leverage scores computed with a QR decomposition (Section 3.2)

We present perturbation bounds that represent the first step in assessing the numerical stability of the QR decomposition for computing leverage scores.

Section 3.2.1 Our first result is a bound derived from existing QR perturbation results that make no reference to a particular implementation. If $\epsilon_F = \|\mathbf{\Delta A}\|_F/\|\mathbf{A}\|_F$ is the total mass of the perturbation, then the leverage scores $\widetilde{\ell}_j$ computed from a QR decomposition of $\mathbf{A} + \mathbf{\Delta A}$ satisfy

$$\frac{|\widetilde{\ell}_j - \ell_j|}{\ell_j} \leq 12 \sqrt{\frac{1-\ell_j}{\ell_j}}\, \mathrm{sr}\,(\mathbf{A})^{1/2}\ \kappa_2(\mathbf{A})\, \epsilon_F + \mathcal{O}(\epsilon_F^2), \qquad 1 \leq j \leq m.$$

Therefore, if $\mathbf{\Delta A}$ is a general matrix perturbation, then leverage scores, computed from a QR decomposition of $\mathbf{A}+\mathbf{\Delta A}$ are well-conditioned in the norm-wise sense, provided they have large magnitude and $\mathbf{A}$ is well-conditioned.

Section 3.2.2 The next bound is derived from scratch and does not rely on existing QR perturbation results. Again, it makes no assumptions on the matrix perturbation $\mathbf{\Delta A}$, but is able to recognize norm-wise row-scaling in $\mathbf{\Delta A}$. If $\epsilon_j = \|e_j^T\mathbf{\Delta A}\|_2/\|e_j^T\mathbf{A}\|_2$, $1 \leq j \leq m$, are norm-wise perturbations of the rows of $\mathbf{A}$, then the leverage scores $\widetilde{\ell}_j$ computed from a QR decomposition of $\mathbf{A} + \mathbf{\Delta A}$ satisfy

$$\frac{\left|\widetilde{\ell}_j - \ell_j\right|}{\ell_j} \leq 2\ \left(\epsilon_j + \sqrt{2}\,\mathrm{sr}\,(\mathbf{A})^{1/2}\ \epsilon_F\right)\ \kappa_2(\mathbf{A}) + \mathcal{O}(\epsilon_F^2), \qquad 1 \leq j \leq m.$$

The perturbation ϵ_j represents the *local* effect of $\mathbf{\Delta A}$, because it indicates how the jth relative leverage score difference depends on the perturbation in row j of $\mathbf{A}$. In contrast, ϵ_F, containing the total mass of the perturbation, represents the *global* effect on all leverage scores.

A similar bound holds for projected perturbations $\epsilon_F^{\perp} = \|(\mathbf{I}_m - \mathbf{A}\mathbf{A}^{\dagger})\,\mathbf{A}\|_F/\|\mathbf{A}\|_F$ and $\epsilon_j^{\perp} = \|e_j^T(\mathbf{I}_m - \mathbf{A}\mathbf{A}^{\dagger})\,\mathbf{\Delta A}\|_2/\|e_j^T\mathbf{A}\|_2$, $1 \leq j \leq m$.

Section 3.2.3 The natural follow up question is: What if $\mathbf{\Delta A}$ does indeed represent a row-scaling of $\mathbf{A}$? Can we get tighter bounds? The answer is yes. If $|e_j^T\mathbf{\Delta A}| \leq \eta_j\,|e_j^T\mathbf{A}|$, $1 \leq j \leq m$, with $\eta = \max_{1\leq j\leq m}\eta_j$, are component-wise row-scaled perturbations, then the leverage scores

$\widetilde{\ell}_j$ computed from a QR decomposition of $\mathbf{A} + \mathbf{\Delta A}$ satisfy

$$\frac{\left|\widetilde{\ell}_j - \ell_j\right|}{\ell_j} \leq 2\,\left(\eta_j + \sqrt{2}\,n\,\eta\right) + \mathcal{O}(\eta^2), \qquad 1 \leq j \leq m.$$

Thus, under component-wise row-scaled perturbations, leverage scores computed with a QR decomposition have relative leverage score differences that depend, to first order, neither on the condition number nor on the magnitudes of the leverage scores.

Numerical experiments (Section 3.2)

After each of the bounds presented in Section 3.2, we perform numerical experiments that illustrate that the bounds correctly capture the relative leverage score differences under different types of perturbations.

Summary (Section 3.3)

We summarize the results in this chapter and describe a few directions for future research.

Appendix (Section 3.4)

We present the proofs for all results in Section 3.2.

3.2 Leverage scores computed with a QR decomposition

We derive bounds for relative leverage score differences for leverage scores that are computed with a QR decomposition. The bounds assume exact arithmetic and are based on perturbation results for QR decompositions; they make no reference to particular QR implementations.

Specifically, our bounds include: Norm-wise bounds for general matrix perturbations (Section 3.2.1), bounds for general perturbations that recognize row-scaling in the perturbations (Section 3.2.2), and bounds for component-wise row-scaled perturbations (Section 3.2.3). Since the bounds do not exploit the zero structure of the triangular factor in the QR decomposition, they can be readily extended to the polar decomposition as well.

3.2.1 General normwise perturbations

The first bound is derived from a normwise perturbation result for QR decompositions [93, Theorem 1.6]. Among the existing and sometimes tighter QR perturbation bounds [10, 19, 20, 21, 22, 90, 91, 94, 95, 101], we chose [93, Theorem 1.6] because it is simple and has the required key ingredients.

Theorem 3.1. *Let* $\mathbf{A}$ *and* $\mathbf{A}+\mathbf{\Delta A}$ *be real* $m\times n$ *matrices with* $\mathsf{rank}(\mathbf{A}) = n$ *and* $\|\mathbf{\Delta A}\|_2 \left\|\mathbf{A}^\dagger\right\| \leq 1/2$. *The leverage scores* $\widetilde{\ell}_j$ *computed from a QR decomposition of* $\mathbf{A}+\mathbf{\Delta A}$ *satisfy*

$$\frac{|\widetilde{\ell}_j - \ell_j|}{\ell_j} \leq 12 \left(\sqrt{\frac{1-\ell_j}{\ell_j}} + 3\,\frac{\kappa_2(\mathbf{A})\,\mathrm{sr}\,(\mathbf{A})^{1/2}}{\ell_j}\,\epsilon_F \right) \kappa_2(\mathbf{A})\,\mathrm{sr}\,(\mathbf{A})^{1/2}\;\epsilon_F, \quad 1 \leq j \leq m.$$

Proof. See Section 3.4.1. □

The perturbation bound in Theorem 3.1 sends the message that: If $\mathbf{\Delta A}$ is a general perturbation, then leverage scores computed from a QR decomposition of $\mathbf{A}+\mathbf{\Delta A}$, are well-conditioned in the norm-wise relative sense, if they have large magnitude and if $\mathbf{A}$ is well-conditioned. We demonstrate that this conclusion is valid in the following experiment.

Numerical experiments: Figure 3.1

The matrix $\mathbf{A}$ has dimension 1000×25, $\kappa_2(\mathbf{A}) = 1$, and leverage scores that increase in four steps, from 10^{-10} to about 10^{-1}. It is generated with the Matlab commands

$$\begin{aligned} \mathbf{A1} &= \mathsf{diag}\left(\mathbf{I}_{250} \quad 10^2\,\mathbf{I}_{250} \quad 10^3\,\mathbf{I}_{250} \quad 10^4\,\mathbf{I}_{250}\right) \mathsf{randn}(1000, 25) \\ [\mathbf{A}, \sim] &= \mathsf{qr}(\mathbf{A1}, 0). \end{aligned} \tag{3.1}$$

The leverage scores of the perturbed matrix $\mathbf{A}+\mathbf{\Delta A}$ are computed with the MATLAB QR decomposition $\mathsf{qr}(\mathbf{A}+\mathbf{\Delta A}, 0)$.

For matrices $\mathbf{A}$ in (3.1), Figure 3.1 shows the relative leverage score differences $|\widetilde{\ell}_j - \ell_j|/\ell_j$ from norm-wise perturbations $\epsilon_F = \|\mathbf{\Delta A}\|_F/\|\mathbf{A}\|_F$ and the bound from Theorem 3.1, for two different perturbations: $\epsilon_F = 10^{-8}$ and $\epsilon_F = 10^{-5}$.

Figure 3.1 illustrates that the relative leverage score differences decrease with the same step size with which the leverage score magnitude increases. In particular, for $\epsilon_F = 10^{-8}$ in (a), the relative leverage score differences decrease from 10^{-5} for the smallest leverage scores to about 10^{-9} for the largest leverage scores. The differences for $\epsilon_F = 10^{-5}$ in (b) are larger by a factor of 1000; the 250 smallest leverage scores have lost all accuracy because they are smaller than the perturbation ϵ_F.

The bound in Theorem 3.1 differs from the actual differences by several orders of magnitude, but reflects the qualitative behavior of the relative leverage score differences.

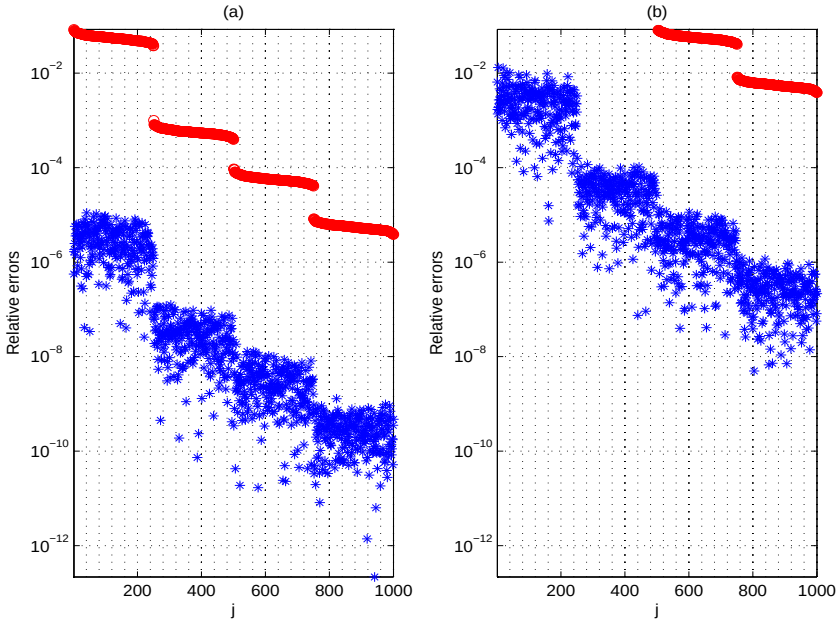


Figure 3.1: Relative leverage score differences $|\widetilde{\ell}_j - \ell_j|/\ell_j$ (blue stars) and the bound from Theorem 3.1 (red line above the stars) vs index j for $\epsilon_F = 10^{-8}$ (a) and $\epsilon_F = 10^{-5}$ (b).

3.2.2 General normwise perturbation bounds that detect row scaling in the perturbations

The two first-order bounds presented here are based on a perturbation of the QR decomposition. Although the bounds make no assumptions on the perturbations $\mathbf{\Delta A}$, they are able to recognize row-scaling in $\mathbf{\Delta A}$ of the form

$$\epsilon_j \equiv \frac{\|e_j^T \mathbf{\Delta A}\|_2}{\|e_j^T \mathbf{A}\|_2}, \qquad 1 \le j \le m.$$

Theorem 3.2. *Let* $\mathbf{A}$ *and* $\mathbf{A} + \mathbf{\Delta A}$ *be real* $m \times n$ *matrices such that* $\mathsf{rank}(\mathbf{A}) = n$ *and* $\|\mathbf{\Delta A}\|_2 \left\|\mathbf{A}^\dagger\right\|_2 < 1$. *The leverage scores* $\widetilde{\ell}_j$ *computed from a QR decomposition of* $\mathbf{A} + \mathbf{\Delta A}$ *satisfy*

$$\frac{\left|\widetilde{\ell}_j - \ell_j\right|}{\ell_j} \le 2\left(\epsilon_j + \sqrt{2}\,\mathrm{sr}\,(\mathbf{A})^{1/2}\,\epsilon_F\right)\kappa_2(\mathbf{A}) + \mathcal{O}(\epsilon_F^2), \qquad 1 \le j \le m.$$

Proof. See Section 3.4.2. □

The relative leverage score difference bound for the jth leverage score in Theorem 3.2

contains three main ingredients:

1. The two-norm condition number of A with respect to left inversion, $\kappa_2(\mathbf{A})$.
   It indicates leverage scores computed from matrices with smaller condition numbers have smaller relative leverage score differences.

2. The relative normwise perturbation in the jth row of $\mathbf{A}$, ϵ_j.
   This perturbation represents the *local* effect of $\mathbf{\Delta A}$, because it shows how the jth relative leverage score difference depends on the perturbation in row j of $\mathbf{A}$.

3. The total normwise perturbation ϵ_F.
   This is the total relative mass of the perturbation, since

$$\epsilon_F^2 = \sum_{i=1}^{m} \left\| e_i^T \mathbf{\Delta A} \right\|_2^2 / \left\| \mathbf{A} \right\|_F^2$$

   represents the *global* effect of $\mathbf{\Delta A}$.

Numerical experiments: Figure 3.2

We illustrate that the local effect described above is real by examining the effect of row scaled perturbations on the relative accuracy of leverage scores computed with a QR decomposition.

Figure 3.2 shows the relative leverage score difference $|\widetilde{\ell}_j - \ell_j|/\ell_j$ from norm wise perturbations $\epsilon_F = \|\mathbf{\Delta A}\|_F/\|\mathbf{A}\|_F = 10^{-8}$ and the bound from Theorem 3.2. In panel (a), only rows 501–750 of $\mathbf{A}$ are perturbed, while in panel (b) the perturbation has the same row scaling as $\mathbf{A}$, that is, $\mathbf{\Delta A} = 10^{-8}\mathbf{A1}/\|\mathbf{A1}\|_F$, where $\mathbf{A1}$ is of the form (3.1).

In panel (a), the leverage scores corresponding to rows 1–500 and 751-1000 have relative leverage score differences between 10^{-12} and 10^{-10}, which illustrates that the local perturbation in rows 501–750 has a global effect on all leverage scores. However, the leverage scores corresponding to the perturbed rows 501–750 have larger relative differences of 10^{-8} or more, which illustrates the strong effect of local perturbations. The bound from Theorem 3.2 hovers around 10^{-7}, but is slightly larger for the leverage scores corresponding to the perturbed rows. Thus, Theorem 3.2 is able to detect strongly local row scaling in norm wise perturbations.

In panel (b), almost all leverage scores have relative differences between 10^{-10} and 10^{-9}, and the bound from Theorem 3.2 is flat at 10^{-7}. Thus, the relative leverage scores differences tend to be more uniform when the norm wise perturbations have the same row scaling as the matrix. This effect is recognized by Theorem 3.2.

Therefore, although Theorem 3.2 makes no assumptions about the perturbations ΔA, it is able to detect row scaling in norm wise perturbations, and correctly predicts the qualitative behavior of relative leverage score differences.

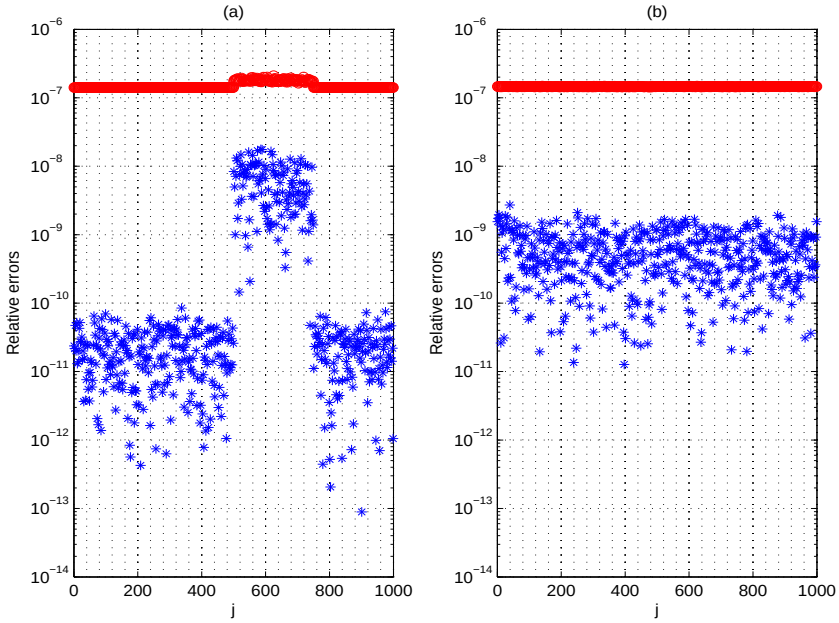


Figure 3.2: Relative leverage score difference $|\widetilde{\ell}_j - \ell_j|/\ell_j$ (blue stars) and bound from Theorem 3.2 (red line above the stars) vs index j for row-wise scaled perturbations with $\epsilon_F = 10^{-8}$. In (a) only rows 501–750 of $\mathbf{A}$ are perturbed, while in (b) the perturbation has the same row scaling as $\mathbf{A}$.

Projected perturbations

The following bound is a refinement of Theorem 3.2 that projects out the part of the perturbation that lies in range $(\mathbf{A})$ and does not contribute to a change in leverage scores,

$$\epsilon_F^{\perp} \equiv \frac{\|(\mathbf{I}_m - \mathbf{A}\mathbf{A}^{\dagger})\,\mathbf{\Delta A}\|_F}{\|\mathbf{A}\|_F}, \qquad \epsilon_j^{\perp} \equiv \frac{\|e_j^T(\mathbf{I}_m - \mathbf{A}\mathbf{A}^{\dagger})\,\mathbf{\Delta A}\|_2}{\|e_j^T\mathbf{A}\|_2}, \qquad 1 \le j \le m.$$

Theorem 3.3 (Projected perturbations)**.** *Let* $\mathbf{A}$ *and* $\mathbf{A} + \mathbf{\Delta A}$ *be real* $m \times n$ *matrices with* $\mathsf{rank}(\mathbf{A}) = n$ *and* $\|\mathbf{\Delta A}\|_2 \left\|\mathbf{A}^{\dagger}\right\|_2 \le 1/2$*. The leverage scores* $\widetilde{\ell}_j$ *computed from a QR decomposition of* $\mathbf{A} + \mathbf{\Delta A}$ *satisfy*

$$\frac{\left|\widetilde{\ell}_j - \ell_j\right|}{\ell_j} \le 4\,\left(\epsilon_j^{\perp} + \sqrt{2}\,\mathrm{sr}\,(\mathbf{A})^{1/2}\,\epsilon_F^{\perp}\right)\kappa_2(\mathbf{A}) + \mathcal{O}\left((\epsilon_F^{\perp})^2\right), \qquad 1 \le j \le m.$$

Proof. See Section 3.4.3. □

It is not clear that Theorem 3.3 is tighter than Theorem 3.2. First, Theorem 3.3 contains an additional factor of 2 in the bound. Second, although the total projected perturbation is smaller, i.e. $\epsilon_F^{\perp} \leq \epsilon_F$, this is not necessarily true for $\epsilon_j^{\perp}$ and ϵ_j. For instance, if

$$\mathbf{A} = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ 1 & 1 \\ 1 & -1 \end{pmatrix}, \qquad \mathbf{\Delta A} = \begin{pmatrix} 1 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix},$$

then

$$(\mathbf{I} - \mathbf{A}\mathbf{A}^{\dagger})\,\mathbf{\Delta A} = (\mathbf{I} - \mathbf{A}\mathbf{A}^{T})\,\mathbf{\Delta A} = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 0 & 0 \\ 1 & 1 \\ 0 & 0 \end{pmatrix}.$$

Here we have $\epsilon_3 = \left\|e_3^T \mathbf{\Delta A}\right\|_2 / \left\|e_3^T \mathbf{A}\right\|_2 = 0$ and $\epsilon_3^{\perp} = \left\|e_3^T (\mathbf{I} - \mathbf{A}\mathbf{A}^{\dagger})\,\mathbf{\Delta A}\right\|_2 / \left\|e_3^T \mathbf{A}\right\|_2 = 1$, so that $\epsilon_3^{\perp} > \epsilon_3$.

3.2.3 Componentwise row-scaled perturbations

Motivated by Section 3.2.2, where bounds for general perturbations $\mathbf{\Delta A}$ can recognize row scaling in $\mathbf{\Delta A}$, we ask the natural follow-up question: What if $\mathbf{\Delta A}$ does indeed represent a row scaling of $\mathbf{A}$? Can we get tighter bounds? To this end, we consider componentwise row perturbations of the form $|e_j^T \mathbf{\Delta A}| \leq \eta_j\, |e_j^T \mathbf{A}|$, where $\eta_j \geq 0$, $1 \leq j \leq m$, and model them as

$$e_j^T \mathbf{\Delta A} = \zeta_j\, \eta_j\, e_j^T \mathbf{A}, \qquad 1 \leq j \leq m, \qquad \eta \equiv \max_{1 \leq j \leq m} \eta_j, \tag{3.2}$$

where ζ_j are uniform random variables in $[-1, 1]$, $1 \leq j \leq m$. We show that, under component wise row-scaled perturbations (3.2), leverage scores computed with a QR decomposition have relative differences that do not depend, to first order, on the condition number or the magnitudes of the leverage scores.

Theorem 3.4. *Let $\mathbf{A}$ be a real $m \times n$ matrix with $\mathsf{rank}(\mathbf{A}) = n$, and let the perturbations $\mathbf{\Delta A}$ be of the form (3.2) with $\eta\, \kappa_2(\mathbf{A}) < 1$. The leverage scores $\widetilde{\ell}_j$ computed from a QR decomposition*

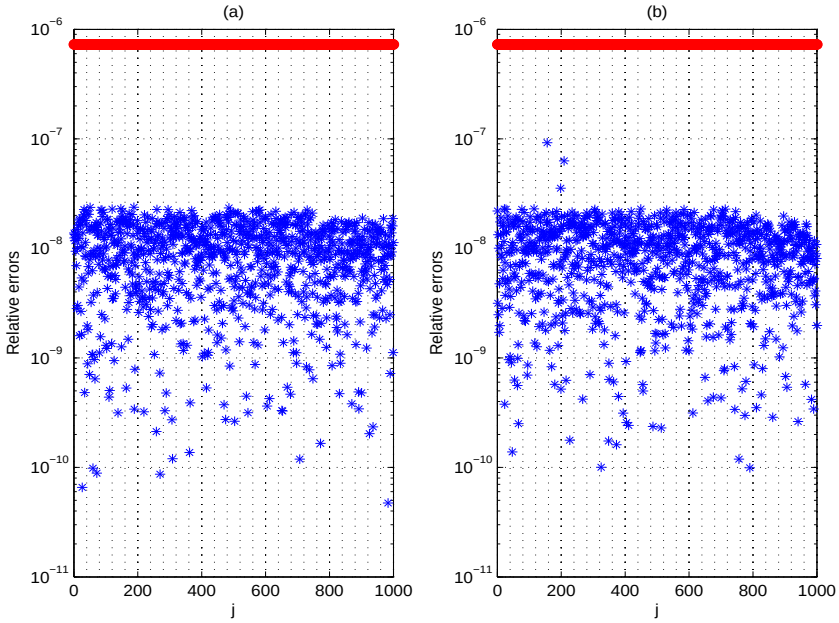


Figure 3.3: Relative leverage score differences $|\widetilde{\ell}_j - \ell_j|/\ell_j$ (blue stars) and the bound from Theorem 3.3 (red line above the stars) vs index j for component wise row-wise scaled perturbations with $\eta_j = 10^{-8}$, $1 \le j \le m$.

of $\mathbf{A} + \mathbf{\Delta A}$ *satisfy*

$$\frac{\left|\widetilde{\ell}_j - \ell_j\right|}{\ell_j} \le 2\,\left(\eta_j + \sqrt{2}\,n\,\eta\right) + \mathcal{O}(\eta^2), \qquad 1 \le j \le m.$$

Proof. See Section 3.4.4. □

The quantities η_j represent the local effects of the individual row-wise perturbations, while the factor $n\,\eta$ represents the global effect of all perturbations. In contrast to our previous results, the bound does not depend on either the condition number or the leverage score magnitude.

Numerical experiments: Figure 3.3

We illustrate the effect of component-wise row-scaled perturbations on the relative accuracy of leverage scores that are computed with a QR decomposition.

Figure 3.3 shows the relative leverage score differences from a well-conditioned matrix $\mathbf{A}$ with $\kappa_2(\mathbf{A}) = 1$ in (a), and from a worse conditioned matrix $\mathbf{B}$ in (b). The condition number

of $\mathbf{B}$ is $\kappa_2(\mathbf{B}) \approx 10^5$ and $\mathbf{B}$ has leverage scores like those of A. Specifically,

$$B = \mathsf{diag}\left(I_{250} \quad 10^2\, I_{250} \quad 10^3\, I_{250} \quad 10^4\, I_{250} \right) \mathsf{gallery}('\mathsf{randsvd}', [\mathsf{m}, \mathsf{n}], \mathsf{10^6}, 3). \tag{3.3}$$

The component-wise row-scaled perturbations from (3.2) are $\eta = \eta_j = 10^{-8}$ for $1 \leq j \leq m$.

Figure 3.3 shows that the relative leverage score differences for both matrices look almost the same, hovering around 10^{-8}, except for a few outliers. Thus, the relative accuracy of most leverage scores does not depend on the condition number, but a few small leverage scores do show a slight effect. Note that Theorem 3.3 is based only on a perturbation analysis, not a round off error analysis of the QR decomposition, and that we did not take into errors arising in the computation of the two norm.

Furthermore, Figure 3.3 shows that the relative leverage score differences do not depend on the leverage score magnitude. Hence Theorem 3.3 captures the relative leverage score accuracy under component-wise row-scaled perturbations.

3.3 Summary

We took the first steps in assessing the numerical stability of QR decompositions for computing leverage scores (Section 3.2). To this end, we derived several bounds for the relative accuracy of individual leverage scores. The bounds are expressed for three classes of matrix perturbations: General norm-wise, norm-wise row-scaled, and component-wise row-scaled.

Since most of the bounds in Section 3.2 do not exploit the zero structure of the upper triangular factor, they are readily extended to polar decompositions as well.

Future research

The next step is to extend the results in Section 3.2.3 to component-wise perturbations

$$|\mathbf{\Delta A}_{jk}| \leq \eta_{jk} |\mathbf{A}_{jk}|, \ 1 \leq j \leq m, \ 1 \leq k \leq n.$$

Numerical experiments strongly suggest that leverage scores computed from QR decompositions of such perturbed matrices have relative leverage score differences that do not depend on the magnitude of the leverage scores.

The most popular method for computing leverage scores is the singular value decomposition. The numerical stability of the SVD in this context needs to be investigated, and whether the sensitivity of the singular vectors to singular value gaps matters for leverage score computations.

Another issue is the numerically stable computation of "k-leverage scores". These are leverage scores of the best rank k approximation to A in the two-norm. Determining leverage scores

from a truncated SVD is necessary when A is (numerically) rank deficient, or when noisy data are well represented, as in the case of PCA, by only a few dominant singular vectors.

3.4 Proofs

We present proofs for the results in Section 3.2.

3.4.1 Proof of Theorem 3.1

We start with a special case of [62, Theorem 2.4] applied to $m \times n$ matrices $\mathbf{Q}$ and $\mathbf{Q} + \mathbf{\Delta Q}$ with orthonormal columns and leverage scores $\ell_j = \|e_j^T \mathbf{Q}\|_2^2$ and $\widetilde{\ell}_j = \|e_j^T (\mathbf{Q} + \mathbf{\Delta Q})\|_2^2$. Since $\|\mathbf{Q}\|_2 = \kappa_2(\mathbf{Q}) = 1$, we obtain

$$\frac{|\widetilde{\ell}_j - \ell_j|}{\ell_j} \leq \left(2\sqrt{\frac{1-\ell_j}{\ell_j}} + \frac{\|\mathbf{\Delta Q}\|_2}{\ell_j} \right) \|\mathbf{\Delta Q}\|_2, \qquad 1 \leq j \leq m. \tag{3.4}$$

The bound for $\|\mathbf{\Delta Q}\|_2 \leq \|\mathbf{\Delta Q}\|_F$ is obtained from a simpler version of the lemma below.

Lemma 3.1 (Theorem 1.6 in [93])**.** *Let* $\mathbf{A}$ *and* $\mathbf{\Delta A}$ *be real* $m \times n$ *matrices with* $\mathsf{rank}(\mathbf{A}) = n$*, and* $\left\|\mathbf{A}^\dagger\right\|_2 \|\mathbf{\Delta A}\|_2 < 1$*. If* $\mathbf{A} + \mathbf{\Delta A} = (\mathbf{Q} + \mathbf{\Delta Q})\,\widetilde{\mathbf{R}}$ *is the thin QR decomposition, then*

$$\|\mathbf{\Delta Q}\|_F \leq \frac{1+\sqrt{2}}{1 - \|\mathbf{A}^\dagger\|_2 \|\mathbf{\Delta A}\|_2} \|\mathbf{A}^\dagger\|_2 \|\mathbf{\Delta A}\|_F .$$

Below is a simpler but not much more restrictive version of Lemma 3.1. If $\|\mathbf{\Delta A}\|_2 \|\mathbf{A}^\dagger\|_2 \leq 1/2$, then

$$\|\mathbf{\Delta Q}\|_F \leq 6 \|\mathbf{A}^\dagger\|_2 \|\mathbf{\Delta A}\|_F = 6\,\mathrm{sr}\,(\mathbf{A})^{1/2}\ \kappa_2(\mathbf{A})\ \epsilon_F.$$

Substituting this into (3.4) gives Theorem 3.1.

3.4.2 Proof of Theorem 3.2

We start with a simplified version of [62, Theorem 2.4]. Let $\mathbf{Q}$ and $\mathbf{Q} + \mathbf{\Delta Q}$ be $m \times n$ matrices with orthonormal columns, and $\ell_j = \|e_j^T \mathbf{Q}\|_2^2$ and $\widetilde{\ell}_j = \|e_j^T (\mathbf{Q} + \mathbf{\Delta Q})\|_2^2$, $1 \leq j \leq m$, their leverage scores. Multiplying out the inner product in $\widetilde{\ell}_j$ and using triangle and submultiplicative inequalities gives

$$\frac{|\widetilde{\ell}_j - \ell_j|}{\ell_j} \leq 2\,\frac{\|e_j^T \mathbf{\Delta Q}\|_2}{\sqrt{\ell_j}} + \frac{\|e_j^T \mathbf{\Delta Q}\|_2^2}{\ell_j}, \qquad 1 \leq j \leq m. \tag{3.5}$$

Next we derive bounds for $\|e_j^T \mathbf{\Delta Q}\|_2$ in terms of $\mathbf{\Delta A}$. To this end we represent the perturbed matrix by a function $A(t)$, with a smooth decomposition $A(t) = Q(t)R(t)$.

This is a very common approach, see for instance [19, Section 3], [20, Section 4], [21, Section 3], [22, Section 5], [34, Section 2.1] [59, Section 2.4], [90, Section 3], [93, Section 2], [94, Section 4], [95, Section 5], and [101, Section].

Define the function

$$\mathbf{A}(t) \equiv \mathbf{A} + \frac{t}{\epsilon_F}\,\mathbf{\Delta A}, \qquad 0 \le t \le \epsilon_F \equiv \frac{\|\mathbf{\Delta A}\|_F}{\|\mathbf{A}\|_F}.$$

Let $\mathbf{A}(t) = \mathbf{Q}(t)\mathbf{R}(t)$ be a thin QR decomposition, where we set $\mathbf{Q} \equiv \mathbf{Q}(0)$, $\mathbf{R} \equiv \mathbf{R}(0)$, $\mathbf{Q} + \mathbf{\Delta Q} \equiv \mathbf{Q}(\epsilon_F)$ and $\mathbf{R} + \mathbf{\Delta R} \equiv \mathbf{R}(\epsilon_F)$. The derivative of $\mathbf{R}$ with regard to t is $\dot{\mathbf{R}}$.

Theorem 3.5. *Let* $\mathbf{A}$ *and* $\mathbf{A} + \mathbf{\Delta A}$ *be real* $m \times n$ *matrices such that* $\mathsf{rank}(\mathbf{A}) = n$ *and* $\|\mathbf{\Delta A}\|_2 \|\mathbf{A}^\dagger\|_2 < 1$. *Then*

$$\mathbf{\Delta Q} = \mathbf{\Delta A}\,\mathbf{R}^{-1} - \epsilon_F\,\mathbf{Q}\,\dot{\mathbf{R}}\,\mathbf{R}^{-1} + \mathcal{O}(\epsilon_F^2),$$

where $\|\dot{\mathbf{R}}\,\mathbf{R}^{-1}\|_F \le \sqrt{2}\,\mathrm{sr}\,(\mathbf{A})^{1/2}\;\kappa_2(\mathbf{A})$.

Proof. The proof is inspired in particular by [20, Section 4] and [58, Section 2.4].

Smooth decomposition From $\mathsf{rank}(\mathbf{A}) = n$, $\|\frac{t}{\epsilon_F}\,\mathbf{\Delta A}\|_2 \le \|\mathbf{\Delta A}\|_2$ for $0 \le t \le \epsilon_F$, and $\|\mathbf{\Delta A}\|_2 \|\mathbf{A}^\dagger\|_2 < 1$ follows $\mathsf{rank}(A(t)) = n$. Furthermore, since $\mathbf{A}(t)$ has at least two continuous derivatives, so do $\mathbf{Q}(t)$ and $\mathbf{R}(t)$ [34, Proposition 2.3].

Expression for $\mathbf{\Delta Q}$ The existence of two derivatives allows us to take a Taylor expansion of $\mathbf{Q}(t)$ around $t = 0$, and get $\mathbf{Q}(t) - \mathbf{Q}(0) = t\,\dot{\mathbf{Q}}(0) + \mathcal{O}(t^2)$. Evaluating at $t = \epsilon_F$ gives

$$\mathbf{\Delta Q} = (\mathbf{Q} + \mathbf{\Delta Q}) - \mathbf{Q} = \mathbf{Q}(\epsilon_F) - \mathbf{Q}(0) = \epsilon_F\,\dot{\mathbf{Q}} + \mathcal{O}(\epsilon_F^2). \tag{3.6}$$

To get an expression for $\dot{\mathbf{Q}}$, differentiate $\mathbf{A}(t) = \mathbf{Q}(t)\mathbf{R}(t)$,

$$\frac{\mathbf{\Delta A}}{\epsilon_F} = \dot{\mathbf{Q}}(t)\,\mathbf{R}(t) + \mathbf{Q}(t)\,\dot{\mathbf{R}}(t),$$

and evaluate at $t = 0$,

$$\dot{\mathbf{Q}} = \frac{\mathbf{\Delta A}}{\epsilon_F}\,\mathbf{R}^{-1} - \mathbf{Q}\,\dot{\mathbf{R}}\,\mathbf{R}^{-1}.$$

Inserting the above into (3.6) gives the expression for $\mathbf{\Delta Q}$ in Theorem 3.2.

Bound for $\|\dot{\mathbf{R}}\mathbf{R}^{-1}\|_F$ Differentiating $\mathbf{A}(t)^T\mathbf{A}(t) = \mathbf{R}(t)^T\mathbf{R}(t)$ gives

$$\frac{1}{\epsilon_F}\left((\boldsymbol{\Delta}\mathbf{A})^T\mathbf{A} + \mathbf{A}^T\boldsymbol{\Delta}\mathbf{A} + \frac{2t}{\epsilon}(\boldsymbol{\Delta}\mathbf{A})^T\boldsymbol{\Delta}\mathbf{A}\right) = \dot{\mathbf{R}}(t)^T\mathbf{R}(t) + \mathbf{R}(t)^T\dot{\mathbf{R}}(t),$$

and evaluating at $t = 0$ yields

$$\frac{1}{\epsilon_F}\left((\boldsymbol{\Delta}\mathbf{A})^T\mathbf{A} + \mathbf{A}^T\boldsymbol{\Delta}\mathbf{A}\right) = \dot{\mathbf{R}}^T\mathbf{R} + \mathbf{R}^T\dot{\mathbf{R}}.$$

Multiplying by $\mathbf{R}^{-T}$ on the left and by $\mathbf{R}^{-1}$ on the right gives

$$\dot{\mathbf{R}}\mathbf{R}^{-1} + \left(\dot{\mathbf{R}}\mathbf{R}^{-1}\right)^T = \frac{1}{\epsilon_F}\left(\mathbf{Q}^T\boldsymbol{\Delta}\mathbf{A}\mathbf{R}^{-1} + \left(\mathbf{Q}^T\boldsymbol{\Delta}\mathbf{A}\mathbf{R}^{-1}\right)^T\right). \tag{3.7}$$

Now we take advantage of the fact that $\dot{\mathbf{R}}\mathbf{R}^{-1}$ is upper triangular, and define a function that extracts the upper triangular part of a square matrix $\mathbf{Z}$ via

$$\mathrm{up}(\mathbf{Z}) \equiv \frac{1}{2}\mathrm{diagonal}(\mathbf{Z}) + \text{strictly upper triangular part}(\mathbf{Z}).$$

Applying the function to (3.7) gives

$$\dot{\mathbf{R}}\mathbf{R}^{-1} = \frac{1}{\epsilon_F}\,\mathrm{up}\left(\mathbf{Q}^T\boldsymbol{\Delta}\mathbf{A}\mathbf{R}^{-1} + \left(\mathbf{Q}^T\boldsymbol{\Delta}\mathbf{A}\mathbf{R}^{-1}\right)^T\right).$$

Taking norms yields [20, Equation (3.5)]

$$\begin{aligned}\left\|\dot{\mathbf{R}}\mathbf{R}^{-1}\right\|_F &\leq \frac{\sqrt{2}}{\epsilon_F}\left\|\mathbf{Q}^T\boldsymbol{\Delta}\mathbf{A}\mathbf{R}^{-1}\right\|_F \\ &\leq \frac{\sqrt{2}}{\epsilon_F}\left\|\boldsymbol{\Delta}\mathbf{A}\right\|_F\|\mathbf{R}^{-1}\|_2 = \sqrt{2}\,\mathrm{sr}\,(\mathbf{A})^{1/2}\,\kappa_2(\mathbf{A}).\end{aligned} \tag{3.8}$$

□

Now we are ready to derive a bound for the row norms of $\boldsymbol{\Delta}\mathbf{Q}$. Combining the two bounds from Theorem 3.5, that is, inserting $\|\dot{\mathbf{R}}\,\mathbf{R}^{-1}\|_F \leq \sqrt{2}\,\mathrm{sr}\,(\mathbf{A})^{1/2}\,\kappa_2(\mathbf{A})$ into

$$\left\|e_j^T\boldsymbol{\Delta}\mathbf{Q}\right\|_2 \leq \left\|e_j^T\boldsymbol{\Delta}\mathbf{A}\right\|_2\left\|\mathbf{A}^\dagger\right\|_2 + \epsilon_F\sqrt{\ell_j}\left\|\dot{\mathbf{R}}\mathbf{R}^{-1}\right\|_2 + \mathcal{O}(\epsilon^2), \qquad 1 \leq j \leq m,$$

gives

$$\left\|e_j^T\boldsymbol{\Delta}\mathbf{Q}\right\|_2 \leq \left\|e_j^T\boldsymbol{\Delta}\mathbf{A}\right\|_2\|\mathbf{A}^\dagger\|_2 + \sqrt{2\,\ell_j}\,\sqrt{\mathrm{sr}\,(\mathbf{A})}\,\epsilon_F\,\kappa_2(\mathbf{A}) + \mathcal{O}(\epsilon_F^2).$$

Into the first summand substitute

$$\|e_j^T \boldsymbol{\Delta}\mathbf{A}\|_2 = \epsilon_j \, \|e_j^T \mathbf{A}\|_2 \leq \epsilon_j \, \|e_j^T \mathbf{Q}\|_2 \, \|\mathbf{R}\|_2 = \epsilon_j \, \sqrt{\ell_j} \, \|\mathbf{A}\|_2,$$

and obtain

$$\|e_j^T \Delta\mathbf{Q}\|_2 \leq \sqrt{\ell_j} \, \left(\epsilon_j + \sqrt{2}\,\sqrt{\mathrm{sr}\,(\mathbf{A})}\,\epsilon_F\right) \, \kappa_2(\mathbf{A}) + \mathcal{O}(\epsilon_F^2), \qquad 1 \leq j \leq m.$$

Inserting the above into (3.5) and focussing on the first order terms in ϵ_F gives Theorem 3.2.

3.4.3 Proof of Theorem 3.3

To remove the contribution of $\boldsymbol{\Delta}\mathbf{A}$ in range $(\mathbf{A})$, let $\mathbf{P} \equiv \mathbf{A}\mathbf{A}^\dagger$ be the orthogonal projector onto range $(\mathbf{A})$, and $\mathbf{P}^\perp \equiv \mathbf{I}_m - \mathbf{P}$ the orthogonal projector onto range $(\mathbf{A})^\perp$. Extracting the contribution in range $(\mathbf{A})$ gives

$$\mathbf{A} + \boldsymbol{\Delta}\mathbf{A} = \mathbf{A} + \mathbf{P}\,\boldsymbol{\Delta}\mathbf{A} + \mathbf{P}^\perp\,\boldsymbol{\Delta}\mathbf{A} = (\mathbf{A} + \mathbf{P}\,\boldsymbol{\Delta}\mathbf{A}) + \mathbf{P}^\perp\,\boldsymbol{\Delta}\mathbf{A} = \mathbf{M} + \boldsymbol{\Delta}\mathbf{M},$$

where $\mathbf{M} \equiv \mathbf{A} + \mathbf{P}\,\boldsymbol{\Delta}\mathbf{A}$ and $\boldsymbol{\Delta}\mathbf{M} \equiv \mathbf{P}^\perp\,\boldsymbol{\Delta}\mathbf{A}$.

Leverage scores Here $\mathsf{rank}(\mathbf{M}) = n$, because $\mathbf{P}$ is an orthogonal projector, and we have that $\|\mathbf{P}\,\boldsymbol{\Delta}\mathbf{A}\|_2 \|\mathbf{A}^\dagger\|_2 \leq \|\boldsymbol{\Delta}\mathbf{A}\|_2 \|\mathbf{A}^\dagger\|_2 < 1$. With $\mathbf{M} = \mathbf{P}\,(\mathbf{A} + \boldsymbol{\Delta}\mathbf{A})$ this implies range $(\mathbf{M}) =$ range $(\mathbf{A})$. Furthermore $\mathsf{rank}(\mathbf{M} + \boldsymbol{\Delta}\mathbf{M}) = \mathsf{rank}(\mathbf{A} + \boldsymbol{\Delta}\mathbf{A}) = n$. Thus $\mathbf{M}$ and $\mathbf{M} + \boldsymbol{\Delta}\mathbf{M}$ have thin QR decompositions $\mathbf{M} = \mathbf{Q}\mathbf{X}$ and $\mathbf{M} + \boldsymbol{\Delta}\mathbf{M} = (\mathbf{Q} + \boldsymbol{\Delta}\mathbf{Q})\widetilde{\mathbf{X}}$, and have the same leverage scores ℓ_j and $\widetilde{\ell}_j$, respectively, as $\mathbf{A}$ and $\mathbf{A} + \boldsymbol{\Delta}\mathbf{A}$.

Ultimately, we want to apply Theorem 3.2 to $\mathbf{M}$ and $\mathbf{M} + \boldsymbol{\Delta}\mathbf{M}$, but the perturbation $\boldsymbol{\Delta}\mathbf{M} = \mathbf{P}^\perp\,\boldsymbol{\Delta}\mathbf{A}$ is to be related to $\mathbf{A}$ rather than to $\mathbf{M}$, and the bound is to be expressed in terms of $\kappa_2(\mathbf{A})$ rather than $\kappa_2(\mathbf{M})$.

Applying Theorem 3.5 With

$$\mathbf{M}(t) \equiv \mathbf{M} + \tfrac{t}{\mu}\boldsymbol{\Delta}\mathbf{M}, \qquad 0 \leq t \leq \mu \equiv \frac{\|\boldsymbol{\Delta}\mathbf{M}\|_F}{\|\mathbf{A}\|_F} = \epsilon_F^\perp,$$

Theorem 3.5 implies

$$\boldsymbol{\Delta}\mathbf{Q} = \boldsymbol{\Delta}\mathbf{M}\,\mathbf{X}^{-1} - \mu\,\mathbf{Q}\,\dot{\mathbf{X}}\,\mathbf{X}^{-1} + \mathcal{O}(\mu^2). \tag{3.9}$$

To bound $\|\dot{\mathbf{X}}\mathbf{X}^{-1}\|_F$, we apply (3.8) and obtain

$$\|\dot{\mathbf{X}}\,\mathbf{X}^{-1}\|_F \leq \frac{\sqrt{2}}{\mu}\,\|\mathbf{\Delta M}\|_F\,\|\mathbf{X}^{-1}\|_2. \tag{3.10}$$

Bounding $\|e_j^T\mathbf{\Delta Q}\|_2$ Combining (3.9) and (3.10) gives

$$\begin{aligned}\|e_j^T\mathbf{\Delta Q}\|_2 &\leq \left(\|e_j^T\mathbf{\Delta M}\|_2 + \sqrt{2}\,\|e_j^T\mathbf{Q}\|_2\,\|\mathbf{\Delta M}\|_F\right)\,\|\mathbf{X}^{-1}\|_2 + \mathcal{O}(\mu^2), \qquad 1\leq j\leq m\\ &= \left(\epsilon_j^{\perp}\,\frac{\|e_j^T\mathbf{A}\|_2}{\|\mathbf{A}\|_2} + \sqrt{2}\,\sqrt{\ell_j}\,\mu\,\mathrm{sr}\,(\mathbf{A})^{1/2}\right)\,\|\mathbf{A}\|_2\|\mathbf{M}^{\dagger}\|_2 + \mathcal{O}(\mu^2).\end{aligned}$$

From $\|e_j^T\mathbf{A}\|_2 \leq \|e_j^T\mathbf{Q}\|_2\|\mathbf{A}\|_2 = \sqrt{\ell_j}\,\|\mathbf{A}\|_2$ follows

$$\|e_j^T\mathbf{\Delta Q}\|_2 \leq \sqrt{\ell_j}\,\left(\epsilon_j^{\perp} + \sqrt{2}\,\mu\,\mathrm{sr}\,(\mathbf{A})^{1/2}\right)\,\|\mathbf{A}\|_2\|\mathbf{M}^{\dagger}\|_2 + \mathcal{O}(\mu^2). \tag{3.11}$$

It remains to express $\|\mathbf{M}^{\dagger}\|_2$ in terms of $\|\mathbf{A}^{\dagger}\|_2$. The well-conditioning of singular values [54, Corollary 8.6.2] applied to $\mathbf{M} = \mathbf{A} + \mathbf{Z}$, where $\mathbf{Z} \equiv \mathbf{P}\,\mathbf{\Delta A}$, implies

$$\|\mathbf{M}^{\dagger}\|_2 = \|(\mathbf{A}+\mathbf{Z})^{\dagger}\|_2 \leq \frac{\|\mathbf{A}^{\dagger}\|_2}{1-\|\mathbf{Z}\|_2\|\mathbf{A}^{\dagger}\|_2} \leq 2\,\|\mathbf{A}^{\dagger}\|_2,$$

where the last inequality is due to the assumption $\|\mathbf{Z}\|_2\|\mathbf{A}^{\dagger}\|_2 \leq \|\mathbf{\Delta A}\|_2\|\mathbf{A}^{\dagger}\|_2 \leq 1/2$. Inserting this bound for $\|\mathbf{M}^{\dagger}\|_2$ into (3.11) yields

$$\|e_j^T\mathbf{\Delta Q}\|_2 \leq 2\,\sqrt{\ell_j}\,\left(\epsilon_j^{\perp} + \sqrt{2}\,\mu\,\mathrm{sr}\,(\mathbf{A})^{1/2}\right)\,\kappa_2(\mathbf{A}) + \mathcal{O}(\mu^2), \qquad 1\leq j\leq m.$$

At last, substituting the above into (3.5) and focussing on the first order terms in $\mu = \epsilon_F^{\perp}$ gives Theorem 3.3.

3.4.4 Proof of Theorem 3.4

Write the perturbations (3.2) as $\mathbf{\Delta A} = \mathbf{DA}$, where $\mathbf{D}$ is a diagonal matrix with diagonal elements $\mathbf{D}_{jj} = \zeta_j\eta_j$, $1 \leq j \leq m$. By assumption $\|\mathbf{\Delta A}\|_2\|\mathbf{A}^{\dagger}\|_2 \leq \eta\,\kappa_2(\mathbf{A}) < 1$, so that $\mathsf{rank}(\mathbf{A}+\mathbf{\Delta A}) = n$.

As in the proof of Theorem 3.2, we start with (3.5). To derive bounds for $\|e_j^T\mathbf{\Delta Q}\|_2$ in terms of η_j and η, represent the perturbed matrix by

$$\mathbf{A}(t) \equiv \mathbf{A} + \frac{t}{\eta}\,\mathbf{\Delta A}, \qquad 0\leq t\leq\eta.$$

Let $\mathbf{A}(t) = \mathbf{Q}(t)\mathbf{R}(t)$ be a thin QR decomposition, where $\mathbf{Q}\equiv\mathbf{Q}(0)$, $\mathbf{R}\equiv\mathbf{R}(0)$, $\mathbf{Q}+\mathbf{\Delta Q}\equiv\mathbf{Q}(\eta)$

and $\mathbf{R} + \mathbf{\Delta R} \equiv \mathbf{R}(\eta)$. The derivative of $\mathbf{R}$ with respect to t is $\dot{\mathbf{R}}$.

Theorem 3.5 implies

$$\mathbf{\Delta Q} = \mathbf{\Delta A}\,\mathbf{R}^{-1} - \epsilon\,\mathbf{Q}\,\dot{\mathbf{R}}\,\mathbf{R}^{-1} + \mathcal{O}(\eta^2) = \mathbf{DQ} - \mathbf{Q}\,\dot{\mathbf{R}}\,\mathbf{R}^{-1} + \mathcal{O}(\eta^2).$$

With $\mathbf{\Delta A} = \mathbf{DA}$ this gives

$$e_j^T\,\mathbf{\Delta Q} = \eta_j\,e_j^T\mathbf{Q} + \eta\,e_j^T\mathbf{Q}\,\dot{\mathbf{R}}\,\mathbf{R}^{-1} + \mathcal{O}(\eta^2), \qquad 1 \le j \le m.$$

Taking norms gives

$$\|e_j^T\,\mathbf{\Delta Q}\|_2 \le \sqrt{\ell_j}\,\left(\eta_j + \eta\,\|\dot{\mathbf{R}}\,\mathbf{R}^{-1}\|_2\right) + \mathcal{O}(\eta^2), \qquad 1 \le j \le m. \tag{3.12}$$

From (3.8) follows

$$\begin{aligned}\|\dot{\mathbf{R}}\,\mathbf{R}^{-1}\|_2 &\le \left\|\dot{\mathbf{R}}\mathbf{R}^{-1}\right\|_F \le \frac{\sqrt{2}}{\eta}\,\left\|\mathbf{Q}^T\mathbf{\Delta A}\mathbf{R}^{-1}\right\|_F = \frac{\sqrt{2}}{\eta}\,\left\|\mathbf{Q}^T\mathbf{DQ}\right\|_F \\ &\le \frac{\sqrt{2}}{\eta}\,\left\|\mathbf{Q}^T\right\|_F\,\|\mathbf{DQ}\|_2 \le \sqrt{2}n.\end{aligned}$$

Combining this with (3.12) yields

$$\|e_j^T\,\mathbf{\Delta Q}\|_2 \le \sqrt{\ell_j}\,\left(\eta_j + \sqrt{2}\,n\,\eta\right) + \mathcal{O}(\eta^2), \qquad 1 \le j \le m.$$

Inserting the above into (3.5) and focussing on the first order terms in η gives Theorem 3.4.

Chapter 4

Approximating functions over active subspaces

4.1 Introduction

Consider a differentiable function $f : \mathbb{R}^m \to \mathbb{R}$ that is expensive to evaluate. Significant work has gone into determining functions $\widehat{f}$ that are "close" to f in some sense and are much cheaper to evaluate. Such a function $\widehat{f}$ is known as a *response surface*. The basic idea of constructing a response surface is to evaluate f at a number of training points and then fit a surface $\widehat{f}$ to the training points. If m is large, we may need to evaluate many training points (i.e. the "curse of dimensionality") in order to construct a good approximation to f.

To attempt to reduce the difficulty of constructing a response surface, we can apply a dimension reduction technique that determines an *active subspace*[1] [28, 85] of the parameter space. The idea is to determine linear combinations of parameters to which f is most sensitive. To put it another way, we want to determine orthogonal directions in the m-dimensional parameter space along which f changes significantly. Computationally, these directions are just the dominant eigenvectors of a Monte Carlo approximation to the $m \times m$ matrix

$$E = \int_{\mathbb{R}^m} \nabla f(\mathbf{x}) \left(\nabla f(\mathbf{x})\right)^T \rho(\mathbf{x}) d\mathbf{x}.$$

If there are only a few (k) dominant eigenvalues, then we can construct a response surface $\widehat{f}$ over the related k-dimensional subspace. We call this k-dimensional subspace an active subspace. Since $k << m$, the number of training points we need to evaluate to construct the response surface is smaller than if we trained $\widehat{f}$ over the full m-dimensional space.

[1] Active subspaces are not unique, so we will usually say "an" active subspace, rather than "the" active subspace, unless we are talking about a particular active subspace.

4.1.1 Related Literature

The idea of finding an active subspace originates in work by Russi [85, Chapter 6] and was formalized by Constantine et al. [28]. Active subspaces are also investigated in [92, Algorithm 1] and [4, Section 4.2.2]. The computation of active subspaces is based on the eigendecomposition of a covariance matrix and is highly related to principal component analysis (PCA) [67].

Active subspaces have been identified and exploited in a number of engineering problems. In [24], two functions related to the manufacturing error of airfoils, both of which depend on twenty variables, are approximated by a response surface over a one-dimensional active subspace. A function related to the wall pressure of combustors, that depends on six variables, is approximated over an active subspace of three variables in [27]. Functions defined in terms of solutions to PDEs containing a coefficient that depends on a random field are approximated with a response surface over an active subspace of a single variable in [28] and [31]. A model of an annular combustor in 38 variables is approximated using only three variables in [8]. An active subspace of one dimension is identified for the power of a photovoltaic cell, which depends on five variables, in [32]. Active subspaces are used in combination with kriging to form response surfaces for test problems and an airfoil design problem in [80].

4.1.2 Our contributions

We have three major contributions:

1. In Theorem 4.5, we show a tighter bound on the number of Monte Carlo samples necessary to approximate an active subspace with error less than ϵ. The bound has the advantage of not depending on m, the total number of parameters.

2. In Section 4.3, we describe three different ways to measure the error of response surfaces over active subspaces. We are careful to separate the error caused by the response surface construction method and the error caused by approximating the function over a subspace of the full parameter space. We emphasize that if care is not taken to construct a good response surface over an active subspace, then the response surface does a poor job of approximating f.

3. We extend a simple test problem, which defines a function with a one-dimensional active subspace, to obtain a test problem that defines a function with a k-dimensional active subspace. For the ten-dimensional version of the test problem, we approximate a function of 3556 variables over a ten-dimensional active subspace with relative accuracy.

4.1.3 Outline

In Section 4.2, we formally define active subspaces, discuss the ideal response surface, give algorithms to approximate both active subspaces and ideal response surface, and discuss the error incurred by the approximations. In Section 4.3, we give our perspective on how to, in practice, evaluate the error between f and the response surface. Section 4.4 describes a test problem with a one-dimensional active subspace and an extension with a higher-dimensional active subspace. Finally, in Section 4.5, we compute an active subspace and associated response surface for the problem described in Section 4.4 and use the criteria discussed in Section 4.3 to evaluate the error.

Sections 4.6 - 4.9 contain proofs and supplementary material.

4.2 Active subspace identification and response surface construction

In this section, we present the material from [28, Sections 2-4] which is relevant to our discussion of active subspaces and response surfaces. All of the results in this section are due to Constantine, Dow, and Wang [28], except for Theorem 4.5, which is a new result.

We assume that $f(\mathbf{x}) : \mathbb{R}^m \to \mathbb{R}$ is a continuously differentiable function of m random variables $\mathbf{x} = (x_1, \ldots, x_m)^T$, where the random variables $\mathbf{x}$ have probability density function $\rho(\mathbf{x})$. We also assume that $\rho(\mathbf{x}) > 0$ for all $\mathbf{x} \in \mathbb{R}^m$ and that $\rho(\mathbf{x})$ is bounded for all $\mathbf{x} \in \mathbb{R}^m$. Assuming that $\rho(\mathbf{x}) > 0$ ensures that the conditional probability density functions used in Sections 4.2.1 and 4.2.2 are well-defined, while assuming $\rho(\mathbf{x})$ is bounded ensures that we can integrate with respect to $\rho(\mathbf{x})$. Since f is continuously differentiable and therefore Lipschitz continuous, there exists an $L > 0$ such that $\|\nabla f(\mathbf{x})\|_2 \leq L$, for all $\mathbf{x} \in \mathbb{R}^m$.

In this section, we describe how to identify an active subspace of f and construct a response surface. In Section 4.2.1, we define active subspaces, show how they can be computed in theory, and discuss the ideal response surface. We bound the root mean square error between f and the ideal response surface in Theorem 4.2. In Section 4.2.2, we discuss the computational difficulties associated with computing active subspaces and the ideal response surface. We describe how to approximate active subspaces and the ideal response surface with Monte Carlo (see Algorithms 2 and 3). Assuming that we have approximated an active subspace with error less than ϵ, we bound the root mean square error between f and our approximation to the ideal response surface in Theorem 4.3. In Section 4.2.3, we show how many Monte Carlo samples in Algorithm 2 are needed to approximate an active subspace with error less than ϵ.

4.2.1 Constructing the ideal response surface

As in [28, Lemma 2.1], we construct an active subspace using orthogonal directions $\mathbf{v}$ along which f changes significantly. We use the expected value of the squared directional derivative of f along $\mathbf{v}$ to measure the change in f along $\mathbf{v}$. To be specific, the expected value of the squared directional derivative along $\mathbf{v}$ ($\|\mathbf{v}\|_2 = 1$) is

$$\mathbf{E}\left[\left(\mathbf{v}^T \nabla f(\mathbf{x})\right)^2\right] = \int_{\mathbb{R}^m} \left(\mathbf{v}^T \nabla f(\mathbf{x})\right)^2 \rho(\mathbf{x}) d\mathbf{x}.$$

Expected values of squared derivatives have been applied previously to measure change along coordinate directions, for an example see [88]. The advantage of using the directional derivative is that we can measure the change along any direction, not just the coordinate directions.

In the following theorem, we show that the expected value of the squared directional derivative along the eigenvectors of

$$E \equiv \int_{\mathbb{R}^m} \nabla f(\mathbf{x}) \left(\nabla f(\mathbf{x})\right)^T \rho(\mathbf{x}) d\mathbf{x}, \tag{4.1}$$

are the eigenvalues of E. We will use these eigenvectors to define an active subspace.

Theorem 4.1 (Lemma 2.1 in [28])**.** *Assume that $f(\mathbf{x}) : \mathbb{R}^m \to \mathbb{R}$ is continuously differentiable and that $\rho(\mathbf{x})$ is a probability density function such that $\rho(\mathbf{x})$ is bounded and strictly positive.*

If $(\lambda_i, \mathbf{v}_i),\ \|\mathbf{v}_i\|_2 = 1,\ 1 \le i \le m$ is an eigenvalue-eigenvector pair of the $m \times m$ matrix

$$E = \int_{\mathbb{R}^m} \nabla f(\mathbf{x}) \left(\nabla f(\mathbf{x})\right)^T \rho(\mathbf{x}) d\mathbf{x},$$

then

$$\mathbf{E}\left[(\mathbf{v}_i^T \nabla f(\mathbf{x}))^2\right] = \lambda_i.$$

Proof. Observe that

$$\mathbf{E}\left[(\mathbf{v}_i^T \nabla f(\mathbf{x}))^2\right] = \mathbf{v}_i^T E \mathbf{v}_i = \mathbf{v}_i^T (\lambda_i \mathbf{v}_i) = \lambda_i.$$

□

The matrix E is symmetric positive semi-definite and so the eigenvalues are all non-negative. Thus, if we can compute the eigenvalues and eigenvectors of E, we obtain a set of m orthogonal directions that are ordered (by the eigenvalues) according to the change in f along those directions. If there are only a few large eigenvalues (say k), then the function $f(\mathbf{x})$ changes primarily over the k directions defined by the k dominant eigenvectors of E. We write the

eigendecomposition of E as

$$E = V\Lambda V^T, \text{ where } V = [\underbrace{V_1}_{k} \underbrace{V_2}_{m-k}], \text{ and } \Lambda = \mathsf{diag}(\lambda_1, \ldots, \lambda_m), \tag{4.2}$$

where V is an $m \times m$ real orthogonal matrix, $\lambda_1 \geq \ldots \geq \lambda_k > \lambda_{k+1} \geq \ldots \geq \lambda_m$ and define an active subspace as $\mathsf{range}\,(V_1)$.

We now want to approximate f with a response surface over an active subspace. In the discussion below, we will make use of the following "decomposition" of a random vector $\mathbf{x} \in \mathbb{R}^m$ into the sum of a random vector in the active subspace and a random vector orthogonal to the active subspace. In other words, decompose $\mathbf{x}$ into

$$\mathbf{x} = [V_1\ V_2][V_1\ V_2]^T\mathbf{x} = V_1V_1^T\mathbf{x} + V_2V_2^T\mathbf{x} = V_1\mathbf{y} + V_2\mathbf{z},$$

where $\mathbf{y} = V_1^T\mathbf{x}$ and $\mathbf{z} = V_2^T\mathbf{x}$ are random vectors. The random vector $\mathbf{y} \in \mathbb{R}^k$ represents the coordinates of the projection of $\mathbf{x}$ onto the active subspace, while the random vector $\mathbf{z} \in \mathbb{R}^{m-k}$ represents the coordinates of the projection of $\mathbf{x}$ onto the orthogonal complement of the active subspace.

We want to approximate $f(\mathbf{x})$ with a function that depends only on $\mathbf{y}$. A simple approximation to $f(\mathbf{x})$ would be $f(V_1\mathbf{y}) = f(V_1V_1^T\mathbf{x})$.[2] In fact, if f does not change at all over the $m-k$ directions in V_2 (i.e. $\lambda_{k+1} = \ldots = \lambda_m = 0$), then $f(V_1\mathbf{y}) = f(\mathbf{x})$. In the case where $\lambda_{k+1}, \ldots, \lambda_m$ are not all zero, we can improve our approximation by averaging over all $\mathbf{z}$. Specifically, we will approximate $f(\mathbf{x})$ with the conditional expectation of $f(\mathbf{x})$ given $\mathbf{y} = \mathbf{y}^*$[3]

$$f(\mathbf{x}) = f(V_1\mathbf{y} + V_2\mathbf{z}) \approx \mathbf{E}\left[f(V_1\mathbf{y} + V_2\mathbf{z}) \mid \mathbf{y} = \mathbf{y}^*\right] = \int_{\mathbb{R}^{m-k}} f(V_1\mathbf{y}^* + V_2\mathbf{z})\rho_{\mathbf{z}|\mathbf{y}}(\mathbf{z} \mid \mathbf{y}^*)\, d\mathbf{z}.$$

In the above, $\rho_{\mathbf{z}|\mathbf{y}}$ is the conditional probability density function of $\mathbf{z}$ given $\mathbf{y} = \mathbf{y}^*$. See Section 4.7 for a brief review of the conditional density function.

In the sense of the mean squared error, the conditional expectation of $f(\mathbf{x}) = f(V_1\mathbf{y} + V_2\mathbf{z})$ given $\mathbf{y} = \mathbf{y}^*$ is the best approximation to $f(\mathbf{x}) = f(V_1\mathbf{y} + V_2\mathbf{z})$ given $\mathbf{y} = \mathbf{y}^*$ [55, Section 7.9: Theorem 17]. Specifically,

$$\mathbf{E}\left[\left(\left(f(\mathbf{x}) - \mathbf{E}\left[f(\mathbf{x}) \mid \mathbf{y} = \mathbf{y}^*\right]\right)^2 \mid \mathbf{y} = \mathbf{y}^*\right] \leq \mathbf{E}\left[\left(f(\mathbf{x}) - g(\mathbf{z})\right)^2 \mid \mathbf{y} = \mathbf{y}^*\right]$$

[2] Since we have assumed that f is defined over all of $\mathbb{R}^m$, $V_1V_1^T\mathbf{x}$ is in the domain of f. If we let the domain of f be a subset of $\mathbb{R}^m$, this may not be true. See [92, Section 1.1] and [28, Section 4.1.2].

[3] We use $\mathbf{y}^*$ to denote a specific instance of the random variable $\mathbf{y}$.

for all functions $g(\mathbf{z})$ such that $\mathbf{E}\left[g(\mathbf{z})^2\right] < \infty$.

The following theorem bounds the root mean squared error incurred by approximating $f(\mathbf{x})$ given $\mathbf{y} = \mathbf{y}^*$ with the conditional expectation $\mathbf{E}\left[f(\mathbf{x}) \mid \mathbf{y} = \mathbf{y}^*\right]$.

Theorem 4.2 (Theorem 3.1 in [28]). *Assume that $f(\mathbf{x}) : \mathbb{R}^m \to \mathbb{R}$ is continuously differentiable and that $\rho(\mathbf{x})$ is a probability density function such that $\rho(\mathbf{x})$ is bounded and strictly positive.*

Let E be defined as in (4.1) and let E have the eigendecomposition described in (4.2). Then,

$$\sqrt{\int_{\mathbb{R}^m} \left(f(\mathbf{x}) - \mathbf{E}\left[f(\mathbf{x}) \mid \mathbf{y} = \mathbf{y}^*\right]\right)^2 \rho(\mathbf{x}) d\mathbf{x}} \leq c_1 \sqrt{\lambda_{k+1} + \ldots + \lambda_m},$$

where c_1 is a constant that depends on $\rho(\mathbf{x})$.

Theorem 4.2 indicates that if the eigenvalues corresponding to the orthogonal complement of the active subspace are small, then the error incurred by approximating f with the conditional expectation is small.

4.2.2 Approximation to ideal response surface

There are a few computational problems with the approach outlined in the previous section.

The first computational difficulty is that computing E (and thus the eigenvalues and eigenvectors) is difficult because all of the entries of E are integrals over an m-dimensional space. We estimate E with a Monte Carlo approximation as in [28, (2.16)]. In other words, approximate E with

$$\widehat{E} = \frac{1}{n_1} \sum_{i=1}^{n_1} \nabla f(\mathbf{x}_i) \left(\nabla f(\mathbf{x}_i)\right)^T$$

where $\mathbf{x}_i \in \mathbb{R}^m$ are chosen randomly according to $\rho(\mathbf{x})$ and we choose $n_1 \leq m$. Then, compute the eigenvalues and eigenvectors of $\widehat{E} = \widehat{V}\widehat{\Lambda}\widehat{V}^T$. Note that, by construction, $\widehat{E}$ is also symmetric positive semidefinite. The number of samples n_1 necessary to approximate the eigenvectors of E is discussed in Section 4.2.3.

We can avoid computing the $m \times m$ matrix $\widehat{E}$ if we define

$$G = \begin{bmatrix} \nabla f(\mathbf{x}_1) & \ldots & \nabla f(\mathbf{x}_{n_1}) \end{bmatrix}$$

and compute the singular value decomposition of G. The left singular vectors of G are the eigenvectors of $\widehat{E}$. We also have that $\frac{1}{n_1}\sigma_i^2(G) = \lambda_i(\widehat{E})$, $1 \leq i \leq n_1$, where $\sigma_i(G)$ are the singular values of G.

At this point we must determine, based on the magnitudes of the eigenvalues, the dimension (k) of an active subspace. Constantine et al. [30, Section 4.1] recommend looking for large gaps

Algorithm 2 Approximate active subspace of f

Input:

$f(\mathbf{x}) : \mathbb{R}^m \to \mathbb{R}$

- f continuously differentiable

$\rho(\mathbf{x})$: probability density function

- $\rho(\mathbf{x}) > 0$ for all $\mathbf{x} \in \mathbb{R}^m$
- $\rho(\mathbf{x})$ is bounded

$0 < n_1 \leq m,\ n_1 \in \mathbb{Z}$

Output:

$m \times k$ matrix $\widehat{V}_1$

$G = \mathbf{0}_{m \times n_1}$
for $i = 1 : n_1$ **do**
  Sample $\mathbf{x}_i \in \mathbb{R}^m$ according to $\rho(\mathbf{x})$
  Set $G(:, i) = \nabla f(\mathbf{x}_i)$
end for
Compute SVD of $G = \widehat{V} \Sigma W^T$, $\Sigma = \mathsf{diag}(\sigma_1, \ldots, \sigma_{\widetilde{n_1}})$
Choose an integer k such that $\sigma_{k+1}^2 / \sigma_k^2$ is large
Set $\widehat{V}_1 = \widehat{V}(:, 1 : k)$, the first k columns of $\widetilde{V}$

between eigenvalues. In other words, choose k so that λ_{k+1}/λ_k is large. Having chosen a k, partition $\widehat{V} = [\widehat{V}_1\ \widehat{V}_2]$, where $\widehat{V}_1$ has k columns and $\widehat{V}_2$ has $m - k$ columns. We summarize our approximation to an active subspace of E in Algorithm 2.

The second computational difficulty, having identified approximations ($\widehat{V}_1$) to the dominant eigenvectors of E, is that we need to compute the conditional expectation of $f(\mathbf{x}) = f(\widehat{V}_1\mathbf{y} + \widehat{V}_2\mathbf{x})$ given $\widehat{\mathbf{y}} = \widehat{\mathbf{y}}^*$, where $\widehat{\mathbf{y}} = \widehat{V}_1^T\mathbf{x}$ and $\widehat{\mathbf{z}} = \widehat{V}_2^T\mathbf{x}$. More precisely, we need to compute the conditional expectation for a set of training points $\{\widehat{\mathbf{y}}_i\}$, which we will use to build the response surface.

The conditional expectation of $f(\mathbf{x})$, given $\widehat{\mathbf{y}} = \widehat{\mathbf{y}}_i$, is again an integral over a high-dimensional space and thus difficult to compute. We will approximate the conditional expectation at T training points $\widehat{\mathbf{y}}_i$ using Monte Carlo. Specifically, the approximation at the training points is

$$\mathbf{E}\left[f\left(\widehat{V}_1\widehat{\mathbf{y}} + \widehat{V}_2\widehat{\mathbf{z}}\right)\ \middle|\ \widehat{\mathbf{y}} = \widehat{\mathbf{y}}_i\right] = \int_{\mathbb{R}^{m-k}} f\left(\widehat{V}_1\widehat{\mathbf{y}}_i + \widehat{V}_2\widehat{\mathbf{z}}\right) \rho_{\widehat{\mathbf{z}}|\widehat{\mathbf{y}}}\left(\widehat{\mathbf{z}} \mid \widehat{\mathbf{y}}_i\right) d\widehat{\mathbf{z}}$$
$$\approx \frac{1}{n_2} \sum_{j=1}^{n_2} f\left(\widehat{V}_1\widehat{\mathbf{y}}_i + \widehat{V}_2\widehat{\mathbf{z}}_j\right) \equiv \widehat{\mathbf{t}}_i,$$

where the $\widehat{\mathbf{z}}_j$ are chosen at random according to the conditional density function $\rho_{\widehat{\mathbf{z}}|\widehat{\mathbf{y}}}$. In gen-

Algorithm 3 Compute training pairs

Input:

$f(\mathbf{x}) : \mathbb{R}^m \to \mathbb{R}$

- f continuously differentiable

$\rho(\mathbf{x})$: probability density function

- $\rho(\mathbf{x}) > 0$ for all $\mathbf{x} \in \mathbb{R}^m$
- $\rho(\mathbf{x})$ is bounded

$m \times k$ matrix $\widehat{V}_1$, with orthonormal columns

$n_2 > 0,\ n_2 \in \mathbb{Z}$

T training points $\widehat{\mathbf{y}}_i$ (coordinates in the active subspace), $1 \leq i \leq T$

Output:

T training points $(\widehat{\mathbf{y}}_i, \mathbf{t}_i)$, $1 \leq i \leq T$

for $i = 1 : T$ **do**
  $\mathbf{t}_i = 0$
  for $j = 1 : n_2$ **do**
    Sample $\mathbf{z}_j \in \mathsf{range}\left(\widehat{V}_2\right)$ according to $\rho_{\widehat{\mathbf{z}}|\widehat{\mathbf{y}}}$
    $\mathbf{t}_i = \mathbf{t}_i + f(\widehat{V}_1\widehat{\mathbf{y}}_i + \widehat{V}_2\widehat{\mathbf{z}}_j)$
  end for
end for

eral, the conditional density function may be very complicated. However, if $\rho(\mathbf{x})$ is a standard Gaussian density, that is, $\mathbf{x}$ is a standard Gaussian random vector, then $\mathbf{y}$ and $\mathbf{z}$ are standard Gaussian random vectors and $\rho_{\widehat{\mathbf{z}}|\widehat{\mathbf{y}}}$ is also a standard Gaussian density [82, p. 200]. Algorithm 3 summarizes our computation of the training points.

We now have training pairs $(\widehat{\mathbf{y}}_i, \mathbf{t}_i)_{i=1}^T$, $1 \leq i \leq T$, and can fit a response surface to them (i.e. using some form of regression or interpolation). A natural question to ask at this point is: How much error is incurred by the Monte Carlo approximation to E, the Monte Carlo approximation to the conditional expectation at the training points, and by fitting a response surface to the training pairs?

We answer the question in Theorem 4.3, which is a restatement of [28, Theorem 3.7]. Since we have not specified how to compute the response surface, we will merely assume that the root mean squared error between the response surface, which we call $\widehat{f}$, and the conditional expectation of f, is bounded above by some constant. The actual error is obviously affected by the number of training points, the location of the training points, and the response surface construction method.

Theorem 4.3 (Theorem 3.7 in [28], [29]). *Make the following assumptions:*

1. *$f(\mathbf{x}) : \mathbb{R}^m \to \mathbb{R}$ is continuously differentiable and $\rho(\mathbf{x})$ is a probability density function such that $\rho(\mathbf{x})$ is bounded and strictly positive*
2. *Algorithm 2 produces $\widehat{V}_1$ such that $\left\| V_1 V_1^T - \widehat{V}_1 \widehat{V}_1^T \right\|_2 \leq \epsilon$*
3. *Algorithm 3 approximates the conditional expectation of f given $\widehat{\mathbf{y}} = \widehat{\mathbf{y}}_i$ with n_2 samples at training points $\widehat{\mathbf{y}}_i$ to produce training pairs $(\widehat{\mathbf{y}}_i, \widehat{\mathbf{t}}_i)$*
4. *We construct $\widehat{f}$ using the training pairs such that the root mean squared error between $\widehat{f}$ and the conditional expectation of f given $\widehat{\mathbf{y}} = \widehat{\mathbf{y}}^*$ is bounded by c_2.*

Then,

$$\sqrt{\int_{\mathbb{R}^m} \left(f(\mathbf{x}) - \widehat{f}(\widehat{\mathbf{y}}) \right)^2 \rho(\mathbf{x}) d\mathbf{x}} \leq c_1 \left(1 + \frac{1}{\sqrt{n_2}} \right) \left(\epsilon \sqrt{\lambda_1 + \ldots + \lambda_k} + \sqrt{\lambda_{k+1} + \ldots + \lambda_m} \right) + c_2,$$

where c_1 is a constant that depends on the probability density function $\rho(\mathbf{x})$.

Remark 4.1. *We make the following observations about Theorem 4.3:*

1. *The quality of the response surface approximation depends on the eigenvalues of E, the accuracy (ϵ) of the approximation to V_1, the number of Monte Carlo samples (n_2) used to estimate the conditional expectation, a constant c_1, and the method used to construct the response surface (through c_2).*
2. *The method we choose to construct the response surface is of critical importance. Since c_2 appears as an additive term, if a large error is incurred while constructing the response surface, the overall error will be large, regardless of how well we have approximated an active subspace.*
3. *The influence of n_2, the number of Monte Carlo samples used to approximate the conditional expectation, on the root mean square error is weak, since $(1 + \frac{1}{\sqrt{n_2}})$ is between one and two.*

In Theorem 4.3, we assumed that we computed $\widehat{V}_1$ so that $\left\| V_1 V_1^T - \widehat{V}_1 \widehat{V}_1^T \right\|_2 \leq \epsilon$. In the next section, we show how many samples n_1 are needed to obtain an absolute error of at most ϵ.

4.2.3 Bounds on estimating eigenvectors

In this section, we answer the question: How large does n_1 need to be to approximate the eigenvalues and eigenvectors of E? We present a bound by Constantine et al. [30, Corollary 3.6] on the number of samples n_1 to ensure, with high probability, that the distance between the subspaces defined by V_1 and $\widehat{V}_1$, $\left\|V_1 V_1^T - \widehat{V}_1 \widehat{V}_1^T\right\|_2$ (see [54, Section 2.5.3]), is not too large.

Theorem 4.4 (Corollary 3.6 in [30])**.** *Assume that $f(\mathbf{x}) : \mathbb{R}^m \to \mathbb{R}$ is continuously differentiable and that $\rho(\mathbf{x})$ is a probability density function such that $\rho(\mathbf{x})$ is bounded and strictly positive.*

Let E be defined as in (4.1) and let E have the eigendecomposition described in (4.2). Assume $0 < \epsilon < \frac{\lambda_k - \lambda_{k+1}}{5\lambda_1}$ and $0 < \delta < 1$. If

$$n_1 \geq \frac{3}{\epsilon^2} \frac{L^2}{\lambda_1} \ln\left(\frac{2m}{\delta}\right),$$

then, with probability at least $1 - \delta$,

$$\left\|V_1 V_1^T - \widehat{V}_1 \widehat{V}_1^T\right\|_2 \leq \frac{4\epsilon}{\lambda_k - \lambda_{k+1}}.$$

We derive a bound on the number of samples necessary to approximate

$$\left\|V_1 V_1^T - \widehat{V}_1 \widehat{V}_1^T\right\|_2$$

that does not depend on the total number of parameters m. The structure of our bound is slightly different, since we want it to hold for any $0 < \epsilon < 1$, and do not want terms other than ϵ on the right hand side.

Theorem 4.5. *Assume that $f(\mathbf{x}) : \mathbb{R}^m \to \mathbb{R}$ is continuously differentiable and that $\rho(\mathbf{x})$ is a probability density function such that $\rho(\mathbf{x})$ is bounded and strictly positive.*

Let E be defined as in (4.1) and let E have the eigendecomposition described in (4.2). Assume $0 < \epsilon < 1$ and $0 < \delta < 1$. If

$$n_1 \geq \left(2 + \frac{2\epsilon}{3}\right) \frac{25L^2}{\epsilon^2 \min\{1, 1/\lambda_1\}^2 (\lambda_k - \lambda_{k+1})^2} \ln\left(\frac{4(\lambda_1 + \ldots + \lambda_m)}{\lambda_1 \delta}\right),$$

then, with probability at least $1 - \delta$,

$$\left\|V_1 V_1^T - \widehat{V}_1 \widehat{V}_1^T\right\|_2 \leq \epsilon.$$

Proof. See Section 4.6. □

Remark 4.2. *We have the following comments about Theorem 4.4 and 4.5.*

1. *The proof of Theorem 4.5, see Section 4.6, relies on a deterministic bound on the distance between subspaces and a probabilistic bound on the sum of random matrices. The proof is similar to the earlier work by Constantine et al. but relies on a different probabilistic bound.*

2. *Both Theorems 4.4 and 4.5 are somewhat conceptual. In order to determine the number of samples necessary for ϵ accuracy, we need to know the true eigenvalues as well as an upper bound L on $\|\nabla f(\mathbf{x})\|_2$, for all $\mathbf{x} \in \mathbb{R}^m$.*

   While we are unlikely to know these quantities, the bounds tell us that approximating an active subspace is more difficult when the absolute eigenvalue gap between an active subspace and its orthogonal complement is small, f is not smooth, or the eigenvalues decay slowly.

3. *The logarithmic term in Theorem 4.5 is usually an improvement over the logarithmic term in Theorem 4.4, because it does not directly depend on the total number of parameters, m. For example, if the eigenvalues are such that $\frac{\lambda_k}{\lambda_1} \geq \gamma \frac{\lambda_{k+1}}{\lambda_1}$, for some $\gamma > 1$, then*

$$\sum_{i=1}^{k} \frac{\lambda_i}{\lambda_1} \leq k \quad \textit{and} \quad \sum_{i=k+1}^{m} \frac{\lambda_i}{\lambda_1} \leq \frac{1}{\gamma}(m-k).$$

   It follows that $\frac{4(\lambda_1+\ldots+\lambda_m)}{\lambda_1} \leq \frac{4}{\gamma}m$. If $\gamma \geq 2$, then

$$\frac{4(\lambda_1 + \ldots + \lambda_m)}{\lambda_1} \leq 2m.$$

4. *Theorem 4.5 emphasizes that the number of samples does not depend explicitly on m, which indicates that Algorithm 2 should scale well as the total number of parameters m becomes very large.*

4.3 Evaluating response surface error

In Section 4.2, we described how to identify an active subspace of a function f and approximate f with a response surface $\widehat{f}$ over the active subspace. In this section, we discuss how to evaluate the error incurred by approximating f with $\widehat{f}$.

One way to measure the error between f and $\widehat{f}$ is to compute the root mean square error. In Theorems 4.2 and 4.3, we presented a bound on the root mean square error between f and $\widehat{f}$.

The mean is taken over all $\mathbf{x} \in \mathbb{R}^m$, with respect to $\rho(\mathbf{x})$. The root mean square error provides a very general idea of how well $\widehat{f}$ approximates f. We say that it is very general because one would expect that the error for $\mathbf{x}$ in an active subspace would be considerably smaller than the error for $\mathbf{x}$ outside an active subspace. The root mean square error blends all these points together. It is also an absolute, rather than relative, measure of the error.

A more precise way to measure the error is to compute the relative error between $\widehat{f}$ and f at different testing locations. Specifically, we want to get a sense for the magnitude of

$$\frac{|f(\mathbf{x}) - \widehat{f}(\mathbf{y})|}{|f(\mathbf{x})|}, \tag{4.3}$$

where $\mathbf{y} = V_1^T \mathbf{x}$.

Consider the following decomposition of the error, which we obtain by adding and subtracting $f(V_1\mathbf{y})$, using the triangle inequality, and reordering the terms

$$\frac{|f(\mathbf{x}) - \widehat{f}(\mathbf{y})|}{|f(\mathbf{x})|} \leq \frac{|f(V_1\mathbf{y}) - \widehat{f}(\mathbf{y})|}{|f(\mathbf{x})|} + \frac{|f(\mathbf{x}) - f(V_1\mathbf{y})|}{|f(\mathbf{x})|}.$$

The magnitude of the first term depends primarily on the method used to construct the response surface. In particular, it should be small at the training points. The second term is the information lost by computing f using the projected version of $\mathbf{x}$. While we do not have a bound on this term, we expect that it depends primarily on the "size" of the information outside the active subspace $(\lambda_{k+1}, \ldots, \lambda_m)$.

With the intuition from the bound above, we want to see whether the relative error is small for two types of points $\mathbf{x}$. First, we should verify that $\widehat{f}$ is a good approximation to f on the active subspace, $\mathsf{range}\,(V_1)$, over which we trained the response surface. To do this, we choose testing points $\mathbf{x} \in \mathsf{range}\,(V_1)$. If these relative errors are small, then our response surface construction method has done an acceptable job of approximating f over the active subspace.

Second, we compute the relative error for points $\mathbf{x}$ outside the active subspace. If these relative errors are small, then our response surface is a good approximation to f for points outside the active subspace.

Note that if we fail to construct an accurate enough response surface over the active subspace (i.e. the errors are large for testing points $\mathbf{x}$ outside the active subspace.) it is unlikely that the response surface will well approximate points outside the active subspace. Thus, care should be taken to construct a response surface in such a way as to ensure some degree of accuracy.

Also note that even if we construct a "good" response surface over the active subspace, we may still have large errors for testing points outside the active subspace, if, for example, $\lambda_{k+1}, \ldots, \lambda_m$ are large or we do not accurately approximate the important directions.

4.4 Description of specific problem

We want to demonstrate our approach to evaluating the response surface error for a problem with an active subspace of more than a few dimensions. To do this, we extend a problem considered by Constantine et al. [28, Section 5]. We describe the original problem in Section 4.4.1 and our extension in Section 4.4.2.

4.4.1 Original problem

Our quantity of interest f is defined in terms of the solution of a partial differential equation. We begin by describing the PDE and how we obtain a numerical solution, and then define the quantity of interest.

Consider

$$-\nabla_{\mathbf{s}} \cdot (a(\mathbf{s}) \nabla_{\mathbf{s}} u(\mathbf{s}, a(\mathbf{s}))) = 1, \quad \mathbf{s} \in [0,1] \times [0,1],$$

with boundary conditions $u = 0$ on the top, left, and bottom boundaries and $\frac{\partial u}{\partial s_1} = 0$ on the right boundary. The coefficient $a(\mathbf{s})$ is a log-Gaussian second order random field with zero mean and covariance function $\mathcal{C}(\mathbf{s}, \mathbf{s}')$. The random field can be expressed in terms of the eigenvalues (μ_i) and orthonormal eigenfunctions $(\phi_i(\mathbf{s}))$ of $\mathcal{C}$ using a Karhunen-Loéve expansion [87, (5.5)]

$$\ln(a(\mathbf{s})) = \sum_{i=1}^{\infty} \sqrt{\mu_i} \phi_i(\mathbf{s}) x_i.$$

In the expansion above, x_i are independent standard Gaussian random variables. For more details about random fields and the Karhunen-Loéve expansion, see Section 4.8.

We discretize and solve the PDE with finite elements in MATLAB's PDE Toolbox. Let $\{\mathbf{n}_i\}_{i=1}^N$ be the nodes of the finite element discretization. The eigenvalues and eigenfunctions of $\mathcal{C}$ are approximated by the eigenvalues and eigenvectors of the $N \times N$ covariance matrix C with elements

$$C_{ij} = \mathcal{C}(\mathbf{n}_i, \mathbf{n}_j), \quad 1 \leq i \leq N, 1 \leq j \leq N.$$

We choose to discard eigenvalues that are smaller than 10^{-12}. Let $m < N$ be the number of eigenvalues larger than 10^{-12}. Then, we approximate

$$\ln(a(\mathbf{s})) \approx \sum_{i=1}^{m} \sqrt{\widehat{\mu}_i} \widehat{\phi}_i x_i \equiv \widehat{a}(\mathbf{s}, \mathbf{x}),$$

where $\widehat{\mu}_i$ and $\widehat{\phi}_i$ are the eigenvalues and eigenvectors of the covariance matrix C and $\mathbf{x}$ is a vector of m independent standard Gaussian random variables. Notice that $\widehat{a}(\mathbf{s}, \mathbf{x})$ approximates $\ln(a(\mathbf{s}))$ at the nodes $\{\mathbf{n}_i\}_{i=1}^N$.

Let $\mathbf{v}(\mathbf{s}, \widehat{a}(\mathbf{s}, \mathbf{x}))$ be the $N \times 1$ vector containing the solution to the discretized PDE at the nodes. We define our quantity of interest to be

$$f(\mathbf{x}) = \mathbf{r}^T M \mathbf{v} \approx \int_0^1 u\left(\begin{bmatrix} 1 \\ \\ s_2 \end{bmatrix}, a\left(\begin{bmatrix} 1 \\ \\ s_2 \end{bmatrix}\right)\right) ds_2$$

where M is the mass matrix of the discretization and $\mathbf{r}$ is a vector of zeros and ones, with ones for the nodes located on the right boundary.

The function $f(\mathbf{x})$ depends on m random variables. Constantine et al. ([28, Section 5.2] and [30, Section 5.2]) determined that, for two specific choices of covariance functions, $f(\mathbf{x})$ changes primarily along a single direction in the m-dimensional parameter space. In Section 4.4.2, we extend the problem to obtain a function f that changes along multiple directions.

4.4.2 Modified problem

Since we are interested in evaluating the effectiveness of approximating a function over an active subspace of several dimensions, we modify the function described in Section 4.4.1 to incorporate more than one direction of change. Consider a family of PDEs

$$-\nabla_{\mathbf{s}} \cdot (a(\mathbf{s}, w)\, \nabla_{\mathbf{s}} u(\mathbf{s}, a(\mathbf{s}, w)) = 1, \quad \mathbf{s} \in [0,1] \times [0,1], \quad 1 \leq w \leq W.$$

The boundary conditions for each separate PDE are identical to that of the original problem. The random fields are log-normal with mean zero and covariance function $\mathcal{C}_w$. Using the Karhunen-Loéve expansion, we express each random field in terms of m_w eigenvalues and eigenvectors of the related covariance matrix and m_w standard Gaussian random variables.

Let $\mathbf{v}_w(\mathbf{s}, \widehat{a}(\mathbf{s}, \mathbf{x}_w))$ be the solution to the wth discretized PDE. Our quantity of interest is

$$f(\mathbf{x}_1, \ldots, \mathbf{x}_W) = \sum_{w=1}^{W} \mathbf{r}^T M \mathbf{v}_w \approx \sum_{w=1}^{W} \int_0^1 u\left(\begin{bmatrix} 1 \\ \\ s_2 \end{bmatrix}, a\left(\begin{bmatrix} 1 \\ \\ s_2 \end{bmatrix}, w\right)\right) ds_2.$$

As before $\mathbf{r}$ is a vector of zeros and ones, with ones for the nodes located on the right boundary, and M is the mass matrix associated with the discretization. We hypothesize that the f defined above, which depends on $\sum_{w=1}^{W} m_w$ total parameters, should change primarily along W directions, provided that the covariance functions $\mathcal{C}_w$ are sufficiently different.

To use Algorithm 2 to identify an active subspace, we will need to compute $\nabla f(\mathbf{x}_1, \ldots, \mathbf{x}_W)$. Constantine et al. in [28, Section 5] outline a procedure to compute ∇f when $W = 1$. It is easy to extend the procedure to compute the gradient when $W > 1$.

4.5 Numerical example

To see if we can construct a response surface $\widehat{f}$ that is a good approximation to f, in the sense of the error measures discussed in Section 4.3, for a problem with an active subspace of more than a few dimensions, we consider the modified problem described in Section 4.4.2 with $W = 10$.

We choose the covariance functions from two families, the exponential and the rational quadratic (see [82, p. 86] and [1, Section 4.2.3 and 4.2.4]). Specifically, we choose

$$\mathcal{C} = \exp(-\left\|\mathbf{s} - \mathbf{s}'\right\|_2^{\alpha}) \quad \text{and} \quad \mathcal{C} = \left(1 + \frac{\|\mathbf{s} - \mathbf{s}'\|_2^2}{2\alpha}\right)^{\alpha}$$

for $\alpha = [2/5 \;\; 4/5 \;\; 6/5 \;\; 8/5 \;\; 10/5]$.

To compute values of f and ∇f, we must approximate the solution to each of the 10 PDEs. We use a finite elements on a mesh with $N = 727$ nodes and solve the PDEs with MATLAB's PDE Toolbox. For the covariance matrices that we chose, $\sum_{w=1}^{W} m_w = 3556$.

In the remainder of the section, we will use Algorithm 2 to identify an active subspace (Section 4.5.1), use Algorithm 3 to compute training points (Section 4.5.2), fit a response surface to the training points (Section 4.5.2), and evaluate the error between f and the response surface (Section 4.5.3) using the criteria discussed in Section 4.3.

4.5.1 Identify active subspace

We hypothesize, because of its construction, that there is a ten-dimensional active subspace. To verify this (and to find a basis $\widehat{V}_1$), we will run Algorithm 2 using $n_1 = 100$ and $n_1 = 1000$ Monte Carlo samples.

We show the normalized squared singular values computed by Algorithm 2 for $n_1 = 100$ and $n_1 = 1000$ in Figure 4.1. In both cases, there is clearly one largest gap, between the tenth and eleventh singular values. The gap is roughly 10^2, and indicates, as we anticipated, that it is be appropriate to approximate $f(\mathbf{x}) = f(\mathbf{x}_1, \ldots, \mathbf{x}_{10})$ over an active subspace of ten dimensions.

4.5.2 Compute training points and construct response surface

To construct a response surface $\widehat{f}$, we build a piecewise multilinear interpolation approximation to $f(\mathbf{x})$ using the Sparse Grid Interpolation Toolbox [69, 70] in MATLAB. We choose this method because it is designed to be practical in high-dimensions, is implemented in existing software, and is fairly simple. Of course, many other options for constructing the response surface exist. In particular, one could fit a Gaussian process, as was done in [28, Section 5]. Gaussian processes have the advantage of coming with "confidence intervals" around the constructed surface.

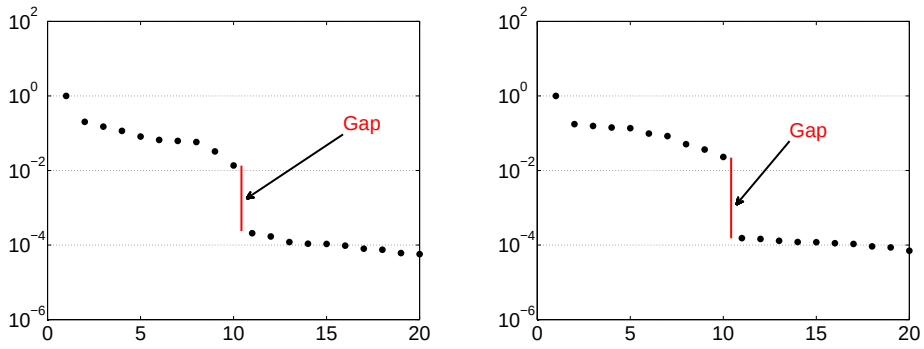


Figure 4.1: Normalized squared singular values of G. Left plot: G constructed with 100 gradient samples. Right plot: G constructed with 1000 gradient samples.

Though we are primarily interested in constructing a response surface in a 10-dimensional subspace, we construct surfaces in $k = 1, 2, \ldots, 14$ dimensions to get a sense for how well the interpolation method performs and to confirm that 10 is the "right" dimension. Since we cannot fit a surface over all of $\mathbb{R}^k$, we choose to fit the surface over $[-3, 3]^k = [-3, 3] \times \ldots \times [-3, 3]$. This comprises three standard deviations of the standard Gaussian random variables. The toolbox constructs a sparse grid over $[-3, 3]^k$ (here the coordinates are with respect to $\mathsf{range}\,(V_1)$). We set the relative tolerance of the toolbox to 10^{-1}.

Thus, our training points $\widehat{\mathbf{y}}_i$ are the sparse gridpoints chosen by the toolbox. We display the number of training points for each k in Table 4.1. With these in hand, we compute the training pairs using Algorithm 3. Because of the small effect on the root mean square error caused by increasing n_2 (see Theorem 4.3), we simply set $n_2 = 1$, as was done in [28, Section 4.2]. Since we have decided to estimate the conditional expectation using only one "sample," we set $\widehat{\mathbf{z}}_1 = 0$ in Algorithm 3, rather than choosing it randomly from the conditional distribution.

Table 4.1: Number of points in sparse grid for active subspaces of dimension k.

k	1	2	3	4	5	6	7
Training points	5	29	177	1 105	6 993	15 121	30 241
k	8	9	10	11	12	13	14
Training points	56 737	100 897	171 425	280 017	442 001	677 041	1 009 905

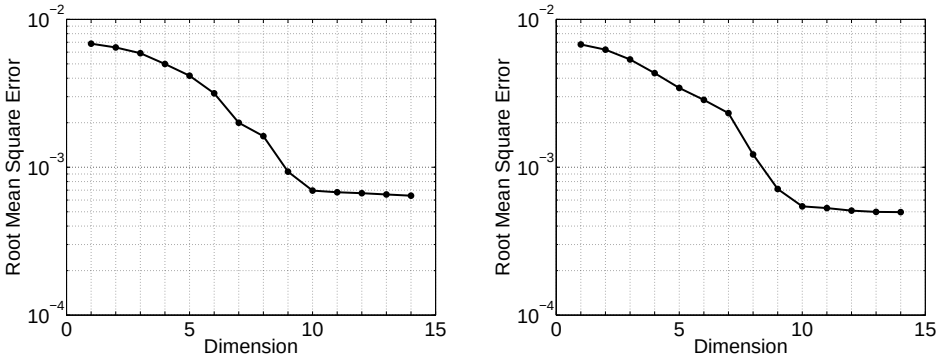


Figure 4.2: Root mean square error using 100 testing points between f and response surfaces $\widehat{f}$ constructed over $k = 1, \ldots, 14$ dimensions. Left plot: G constructed with 100 gradient samples. Right plot: G constructed with 1000 gradient samples.

4.5.3 Evaluate error

We are interested in several types of error (see Section 4.3) incurred by approximating f over the active subspace. To begin with, there is the root mean squared error that we bounded in Theorem 4.3, which averages together the squared error between $\widehat{f}$ and f over all of $\mathbb{R}^m$.

To approximate the root mean squared error, we choose 100 points $\mathbf{x}_i \in \mathbb{R}^m$ so that each entry of $\mathbf{x}_i$ is a standard Gaussian random variable. Figure 4.2 shows the root mean square error for the active subspaces of dimension $k = 1, \ldots, 14$

$$\sqrt{\frac{1}{100}\sum_{i=1}^{100}(f(\mathbf{x}_i) - \widehat{f}(\widehat{V}_1^T\mathbf{x}_i)^2}.$$

The root mean square error decreases as the dimension k increases. Notice that there is almost no gain moving past $k = 10$, which indicates, as expected, that f changes primarily over a $k = 10$ dimensional subspace. Additionally, we see that there is not a significant difference between the root mean square error when $n_1 = 100$ and $n_1 = 1000$.

To evaluate the relative error (4.3) for points in the active subspace, we compute the relative error between the response surface $\widehat{f}$ and the original function f at testing points $\mathbf{a}_i$ chosen at random from $[-3, 3]^k$. In the left plot of Figure 4.3, we display the maximum, mean, and minimum relative error computed at $10k$ points for surfaces computed over $k = 1, \ldots, 14$ dimensions. For each surface constructed, the maximum observed relative error is less than 10^{-1}. When G is constructed with $n_1 = 1000$ gradient samples, the maximum and mean relative error for $k = 9, \ldots, 14$ is slightly smaller than for $n_1 = 100$.

To get a better sense for the distribution of errors when $k = 10$, we plot all 100 relative

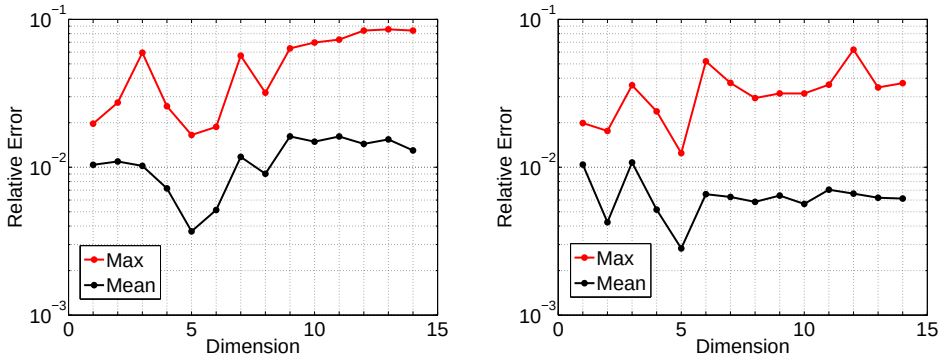


Figure 4.3: Maximum, mean, and minimum relative error $|\widehat{f} - f|/|f|$ at $10k$ testing points for response surfaces over active subspaces of dimension $k = 1, 2, \ldots, 14$. Testing points chosen from active subspace. Left plot: G constructed with 100 gradient samples. Right plot: G constructed with 1000 gradient samples.

errors in Figure 4.4. When $n_1 = 100$, about half of the relative errors are smaller than 10^{-2}. For $n_1 = 1000$, about 80% of the relative errors are smaller than 10^{-2}.

To evaluate the relative error for points outside the active subspace, we choose $10k$ random points $\mathbf{a}_i$ from $[-3, 3]^k$. For each $\mathbf{a}_i$, we choose 10 random points $\mathbf{o}_{ij}$ from the affine orthogonal subspace. Thus, we evaluate $100k$ testing points $\mathbf{a}_i + \mathbf{o}_{ij}$ that live (almost certainly) outside the active subspace. The relative error between the true value of f and the approximate value $\widehat{f}$ is computed.

In Figure 4.5, we display the maximum, mean, and minimum relative error over the $100k$ testing points for the surface computed over $k = 1, 2, \ldots, 14$ dimensions. Notice that as the dimension increases, the mean error decreases, until $k = 10$, where it levels off. The maximum for $n_1 = 100$ is just above 10^{-1}, while for $n_1 = 1000$, it is almost exactly 10^{-1}.

In Figure 4.6, we show all relative errors for the surfaces constructed over dimension $k = 10$. While the maximum error for $k = 10$ is above 10^{-1} for both choices of n_1, notice that nearly all relative errors are below 10^{-1}. The errors are somewhat smaller, on the whole, for $n_1 = 1000$, than for $n_1 = 100$.

Based on our examination of the errors in Figures 4.2 - 4.6, we conclude that the $k = 10$ dimensional response surface is able to approximate points inside and outside the active subspace with relative errors generally smaller than 10^{-1}.

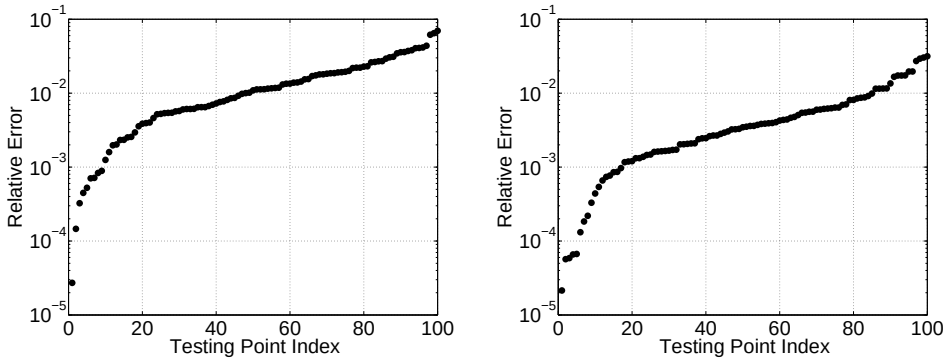


Figure 4.4: Relative errors $|\widehat{f} - f|/|f|$ at 100 testing points for response surface over active subspace of dimension 10. Testing points chosen from active subspace. Left plot: G constructed with 100 gradient evaluations. Right plot: G constructed with 1000 gradient evaluations.

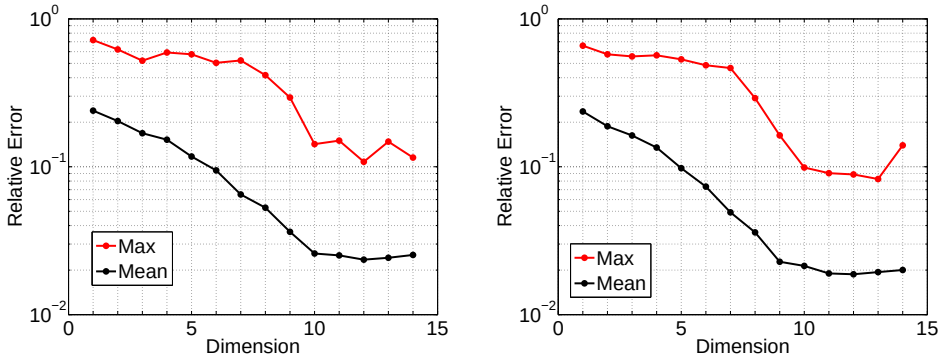


Figure 4.5: Maximum, mean, and minimum relative error $|\widehat{f} - f|/|f|$ at $10k$ testing points for response surfaces over active subspaces of dimension $k = 1, 2, \ldots, 11$. Testing points chosen from outside active subspace. Left plot: G constructed with 100 gradient samples. Right plot: G constructed with 1000 gradient samples.

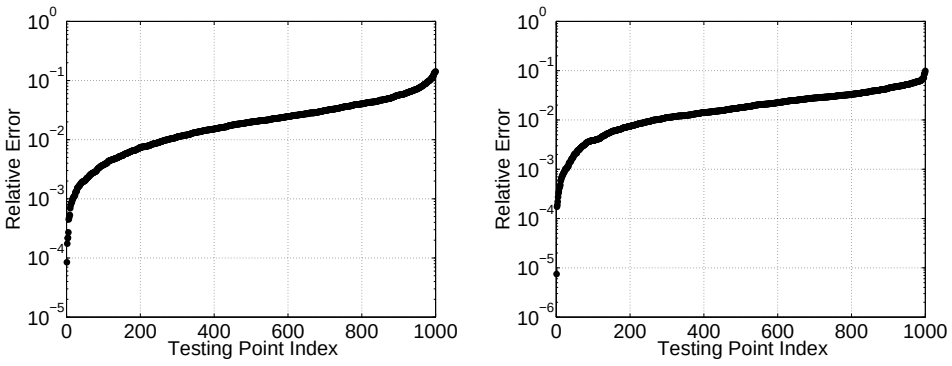


Figure 4.6: Relative errors $|\widehat{f} - f|/|f|$ at 1000 testing points for response surfaces over active subspace of dimension 10. Testing points chosen from outside active subspace. Left plot: G constructed with 100 gradient evaluations. Right plot: G constructed with 1000 gradient evaluations.

4.6 Proof of Theorem 4.5

To prove the bound, we will need a concentration inequality that relies on the *intrinsic dimension* of a matrix P. If P is an $m \times m$ symmetric positive semi-definite matrix then the intrinsic dimension is [97, Definition 7.1.1]:

$$\mathsf{intdim}\,(P) \equiv \mathsf{trace}\,(P) \,/\, \|P\|_2 \,.$$

It is easy to see that $1 \leq \mathsf{intdim}\,(P) \leq \mathsf{rank}\,(P) \leq m$.

The following matrix concentration inequality bounds a sum of random matrices in terms of the intrinsic dimension of a matrix (P) that succeeds (in the sense of the Löwner partial ordering [64, Section 7.7]) a sum of second moments.

Theorem 4.6 (Theorem 7.3.1 and (7.3.2) in [97])**.** *Let X_j be n_1 independent real symmetric random matrices, with $\mathbf{E}[X_j] = 0$, $1 \leq j \leq c$. Let $\max_{1\leq j\leq n_1} \|X_j\|_2 \leq p_1$, and let P be a symmetric positive semi-definite matrix so that $\sum_{j=1}^{n_1} \mathbf{E}\left[X_j^2\right] \preceq P$. Then for any $\epsilon \geq \|P\|_2^{1/2} + p_1/3$*

$$Pr\left[\left\|\sum_{j=1}^{n_1} X_j\right\|_2 \geq \epsilon\right] \leq 4\,\mathsf{intdim}\,(P)\,\exp\left(\frac{-\epsilon^2/2}{\|P\|_2 + p_1\epsilon/3}\right).$$

We are now ready to apply the theorem to our problem. Note that the proof of the following theorem is very similar to [61, Theorem 7.8].

Theorem 4.7. *Let E be defined as in (4.1) and let E have the eigendecomposition described*

in (4.2). For any $\delta > 0$, with probability at least $1 - \delta$,

$$\left\| \widehat{E} - E \right\|_2 \leq \gamma + \sqrt{\gamma(\gamma + 6)}, \text{ where } \gamma = \frac{1}{3n_1} L^2 \ln \left(\frac{4\mathsf{trace}\,(E)}{\|E\|_2\, \delta} \right).$$

Proof. To use Theorem 4.6, we define

$$X_j = \frac{1}{n_1} \nabla f(\mathbf{x}_j)(\nabla f(\mathbf{x}_j))^T - \frac{1}{n_1} E$$

and perform the following computations:

1. *Zero mean.*

$$\begin{aligned} \mathbf{E}\left[X_j\right] &= \frac{1}{n_1} \int (\nabla f(\mathbf{x})(\nabla f(\mathbf{x}))^T - E)\rho(\mathbf{x})\, d\mathbf{x} \\ &= \frac{1}{n_1} \int \nabla f(\mathbf{x})(\nabla f(\mathbf{x}))^T \rho(\mathbf{x})\, d\mathbf{x} - \frac{1}{n_1} \int E\rho(\mathbf{x})\, d\mathbf{x} = 0. \end{aligned}$$

2. *Bound on $\|X_j\|_2$.* We have that

$$\|X_j\|_2 \leq \frac{1}{n_1} \max\left\{ \left\| \nabla f(\mathbf{x}_j)(\nabla f(\mathbf{x}_j))^T \right\|_2,\ \|E\|_2 \right\} \leq \frac{1}{n_1} \max\{L^2, \|E\|_2\}.$$

   Now,

$$\begin{aligned} \|E\|_2 &= \left\| \int \nabla f(\mathbf{x}))(\nabla f(\mathbf{x}))^T \rho(\mathbf{x})\, d\mathbf{x} \right\|_2 \\ &\leq \int \left\| \nabla f(\mathbf{x})(\nabla f(\mathbf{x}))^T \right\| \rho(\mathbf{x})\, d\mathbf{x} \\ &\leq L^2. \end{aligned}$$

   Thus, $\|X_j\|_2 \leq L^2/n_1 \equiv \rho_1$.

3. *The matrix P.* We first calculate $\mathbf{E}\left[X_j^2\right]$. We have

$$\begin{aligned} \mathbf{E}\left[X_j^2\right] &= \frac{1}{{n_1}^2} \int \left(\nabla f(\mathbf{x})(\nabla f(\mathbf{x}))^T - E \right)^2 \rho(\mathbf{x}) d\mathbf{x} \\ &= \frac{1}{{n_1}^2} \left[\int (\nabla f(\mathbf{x})(\nabla f(\mathbf{x}))^T)^2 \rho(\mathbf{x}) d\mathbf{x} - \int \nabla f(\mathbf{x})(\nabla f(\mathbf{x}))^T E \rho(\mathbf{x}) d\mathbf{x} \right. \\ &\quad \left. - \int E \nabla f(\mathbf{x})(\nabla f(\mathbf{x}))^T \rho(\mathbf{x}) d\mathbf{x} + \int E^2 \rho(\mathbf{x}) d\mathbf{x} \right] \\ &= \frac{1}{{n_1}^2} \left[\int (\nabla f(\mathbf{x})(\nabla f(\mathbf{x}))^T)^2 \rho(\mathbf{x}) d\mathbf{x} - \int \nabla f(\mathbf{x})(\nabla f(\mathbf{x}))^T \rho(\mathbf{x}) d\mathbf{x}\, E \right. \end{aligned}$$

$$
\begin{aligned}
& -E \int \nabla f(\mathbf{x})(\nabla f(\mathbf{x}))^T \rho(\mathbf{x}) d\mathbf{x} + \int E^2 \rho(\mathbf{x}) d\mathbf{x} \Big] \\
= \; & \frac{1}{{n_1}^2} \left[\int (\nabla f(\mathbf{x})(\nabla f(\mathbf{x}))^T)^2 \rho(\mathbf{x}) d\mathbf{x} - E^2 - E^2 + E^2 \right] \\
= \; & \frac{1}{{n_1}^2} \left[\int (\nabla f(\mathbf{x})(\nabla f(\mathbf{x}))^T)^2 \rho(\mathbf{x}) d\mathbf{x} - E^2 \right].
\end{aligned}
$$

Thus, $\mathbf{E}\left[X_j^2\right] \preceq \frac{1}{{n_1}^2} \int (\nabla f(\mathbf{x})(\nabla f(\mathbf{x}))^T)^2 \rho(\mathbf{x}) d\mathbf{x}$. Now note that

$$
\int (\nabla f(\mathbf{x})(\nabla f(\mathbf{x}))^T)^2 \rho(\mathbf{x}) d\mathbf{x} = \int \|\nabla f(\mathbf{x})\|_2^2 \, \nabla f(\mathbf{x})(\nabla f(\mathbf{x}))^T \rho(\mathbf{x}) d\mathbf{x}.
$$

It follows that

$$
\frac{1}{{n_1}^2} \int \|\nabla f(\mathbf{x})\|_2^2 \, (\nabla f(\mathbf{x})(\nabla f(\mathbf{x}))^T)^2 \rho(\mathbf{x}) d\mathbf{x} \preceq \frac{L^2}{{n_1}^2} E.
$$

since $L^2 - \|\nabla f(\mathbf{x})\|_2^2 \geq 0$ for every $\mathbf{x} \in \mathrm{R}^m$.

We have shown that $\mathbf{E}\left[X_j^2\right] \preceq \frac{L^2}{{n_1}^2} E$. Thus, $\sum_{j=1}^{n_1} \mathbf{E}\left[X_j^2\right] \preceq \frac{L^2}{n_1} E$, and we set $P = \frac{L^2}{n_1} E$.

4. *Compute* $\mathsf{intdim}\,(P)$. Since $\mathsf{trace}\,()$ is a linear function

$$
\mathsf{intdim}\,(P) = \frac{\mathsf{trace}\left(\frac{L^2}{n_1} E\right)}{\left\| \frac{L^2}{n_1} E \right\|_2} = \frac{\mathsf{trace}\,(E)}{\|E\|_2}
$$

5. *Application of Theorem 4.6.* Substitute the quantities we computed into Theorem 4.6. We see that

$$
Pr\left[\left\| \widehat{E} - E \right\|_2 \geq \epsilon \right] \leq 4 \, \frac{\mathsf{trace}\,(E)}{\|E\|_2} \, \exp\left(\frac{-\epsilon^2/2}{L^2/n_1 \, \|E\|_2 + L^2 \epsilon/(3n_1)} \right).
$$

Set the right hand side of the above equation equal to δ and solve for ϵ to obtain

$$
\epsilon = \gamma + \sqrt{\gamma(\gamma + 6 \, \|E\|_2)}, \text{ where } \gamma = \frac{L^2}{3n_1} \ln\left(\frac{4\mathsf{trace}\,(E)}{\|E\|_2 \, \delta} \right)
$$

6. *Check condition.* We need to verify that

$$
\epsilon \geq \|P\|_2^{1/2} + p_1/3,
$$

for the quantities computed above. In other words, we need to check that

$$\gamma + \sqrt{\gamma(\gamma + 6\left\|E\right\|_2)} \geq \frac{L}{\sqrt{n_1}} \left\|E\right\|_2^{1/2} + \frac{L^2}{3n_1}.$$

Expanding the terms on the left hand side, notice that

$$\frac{L^2}{3n_1} \ln\left(\frac{4\mathsf{trace}\,(E)}{\left\|E\right\|_2 \delta}\right) + \sqrt{\frac{L^2}{3n_1} \ln\left(\frac{4\mathsf{trace}\,(E)}{\left\|E\right\|_2 \delta}\right)\left(\frac{L^2}{3n_1} \ln\left(\frac{4\mathsf{trace}\,(E)}{\left\|E\right\|_2 \delta}\right) + 6\left\|E\right\|_2\right)}$$
$$\geq \frac{L^2}{3n_1} \ln\left(\frac{4\mathsf{trace}\,(E)}{\left\|E\right\|_2 \delta}\right) + \frac{\sqrt{2}L\left\|E\right\|_2^{1/2}}{\sqrt{n_1}}.$$

Now,

$$\frac{L^2}{3n_1} \ln\left(\frac{4\mathsf{trace}\,(E)}{\left\|E\right\|_2 \delta}\right) + \frac{\sqrt{2}L}{\sqrt{n_1}} \left\|E\right\|_2^{1/2} \geq \frac{L}{\sqrt{n_1}} \left\|E\right\|_2^{1/2} + \frac{L^2}{3n_1}$$

provided that $\ln\left(4\mathsf{trace}\,(E)/(\left\|E\right\|_2 \delta)\right) > 1$, which is true for any $0 < \delta < 1$. We made this assumption in the statement of the theorem.

□

We can use the previous result to get a bound on the number of samples (n_1) to obtain an error of at most ϵ.

Corollary 4.1. *Let* $0 < \epsilon < 1$ *and* $0 < \delta < 1$. *If*

$$n_1 \geq \left(2 + \frac{2\epsilon}{3}\right) \frac{1}{\epsilon^2} L^2 \ln\left(\frac{4\mathsf{trace}\,(E)}{\left\|E\right\|_2 \delta}\right),$$

then

$$\left\|\widehat{E} - E\right\|_2 \leq \epsilon.$$

Proof. We want to determine n_1 such that $\gamma + \sqrt{\gamma(\gamma + 6\left\|E\right\|_2)} \leq \epsilon$, where

$$\gamma = \frac{L^2}{3n_1} \ln(4\mathsf{trace}\,(E)/(\left\|E\right\|_2 \delta)).$$

Set $\gamma = t/(3n_1)$ and $n_1 = \alpha t/\epsilon^2$. Our goal now is to determine an α such that

$$\frac{\epsilon^2}{3\alpha} + \sqrt{\frac{\epsilon^2}{3\alpha}\left(\frac{\epsilon^2}{3\alpha} + 6\left\|E\right\|_2\right)} \leq \epsilon.$$

Multiply both sides by α/ϵ and simplify to obtain

$$\sqrt{\epsilon^2 + 18\alpha} \leq 3\alpha - \epsilon.$$

Square both sides and solve for α to find

$$\alpha \geq 2 + \frac{2}{3}\epsilon.$$

□

We will also need one final result, which bounds $\left\|V_1 V_1^T - \widehat{V}_1 \widehat{V}_1^T\right\|_2$ in terms of $\widehat{E} - E$, to prove our theorem.

Theorem 4.8 (Corollary 8.1.11 in [53])**.** *Let E and $\widehat{E}$ be $m \times m$ symmetric matrices. Let $V \Lambda V^T$ be the eigenvalue decomposition E, where $V = [V_1 \; V_2]$ and V_1 is $m \times k$. Then, if $\lambda_k - \lambda_{k+1} > 0$ and*

$$\left\|\widehat{E} - E\right\|_2 \leq \frac{\lambda_k - \lambda_{k+1}}{5}$$

we have that

$$\left\|V_1 V_1^T - \widehat{V}_1 \widehat{V}_1^T\right\|_2 \leq \frac{4}{(\lambda_k - \lambda_{k+1})} \left\|V_2^T (\widehat{E} - E) V_1\right\|_2.$$

Proof of Theorem 4.5: We assumed in the statement of the theorem that $0 < \epsilon < 1$ and $0 < \delta < 1$. Set

$$\widehat{\epsilon} = \frac{\min\{\epsilon\lambda_1, \epsilon\}(\lambda_k - \lambda_{k+1})}{5\lambda_1}.$$

By Corollary 4.1, we know that, since $0 < \widehat{\epsilon} < 1$, that if

$$n_1 \geq \left(2 + \frac{2\epsilon}{3}\right) \frac{25\lambda_1^2 L^2}{\min\{\epsilon\lambda_1, \epsilon\}^2 (\lambda_k - \lambda_{k+1})^2} \ln\left(\frac{4\mathsf{trace}\,(E)}{\|E\|_2 \, \delta}\right)$$

then, with probability at least $1 - \delta$, $\left\|\widehat{E} - E\right\|_2 \leq \widehat{\epsilon}$.

We now want to apply Theorem 4.8. Since it is also true that $\widehat{\epsilon} < \frac{\lambda_k - \lambda_{k+1}}{5}$, we have that, with probability at least $1 - \delta$,

$$\begin{aligned}
\left\|V_1 V_1^T - \widehat{V}_1 \widehat{V}_1^T\right\|_2 &\leq \frac{4}{\lambda_k - \lambda_{k+1}} \left\|V_2^T (\widehat{E} - E) V_1\right\|_2 \\
&\leq \frac{4}{\lambda_k - \lambda_{k+1}} \left\|\widehat{E} - E\right\|_2 \\
&\leq \frac{4}{\lambda_k - \lambda_{k+1}} \widehat{\epsilon} \\
&= \frac{4}{5} \frac{\min\{\epsilon\lambda_1, \epsilon\}}{\lambda_1} \\
&\leq \epsilon.
\end{aligned}$$

4.7 Conditional probability

Let $\rho(\mathbf{x})$ be the probability density function of $\mathbf{x}$ and $\rho(\mathbf{x}) > 0$ for all $\mathbf{x}$. Also let $V = [V_1\ V_2]$ be an $m \times m$ orthogonal matrix, where V_1 is $m \times k$ and V_2 is $m \times (m-k)$ and define $\mathbf{y} = V_1^T\mathbf{x}$, and $\mathbf{z} = V_2^T\mathbf{x}$.

Let $\rho_{\mathbf{y},\mathbf{z}}(\mathbf{y},\mathbf{z}) \equiv \rho(V_1\mathbf{y} + V_2\mathbf{z})$ be the joint probability density function of $\mathbf{y}$ and $\mathbf{z}$ and recall that the marginal density of $\mathbf{y}$ is

$$\rho_{\mathbf{y}}(\mathbf{y}) \equiv \int_{\mathbb{R}^{m-k}} \rho_{\mathbf{y},\mathbf{z}}(\mathbf{y},\mathbf{z})d\mathbf{z}.$$

Notice that since we assumed that $\rho(\mathbf{x})$ is strictly positive, the marginal density of $\mathbf{y}$ is also strictly positive. Finally, the conditional density function of $\mathbf{z}$ given $\mathbf{y} = \mathbf{y}^*$ is

$$\rho_{\mathbf{z}|\mathbf{y}}(\mathbf{y},\mathbf{z}) \equiv \frac{\rho_{\mathbf{y},\mathbf{z}}(\mathbf{y},\mathbf{z})}{\rho_{\mathbf{y}}(\mathbf{y})}.$$

Since the marginal density of $\mathbf{y}$ is strictly positive, we do not divide by zero.

Similarly, one can also define the conditional density function of $\mathbf{y}$ given $\mathbf{z} = \mathbf{z}^*$ using the marginal density of $\mathbf{z}$.

4.8 Random Fields

A Gaussian random field $a(\mathbf{s})$ [1, Definition 1.3] over $\mathbb{R}^2$ is a function such that, for any integer $k > 0$, and any k fixed points $\mathbf{s}_1, \ldots, \mathbf{s}_k \in \mathbb{R}^2$, the vector

$$\begin{bmatrix} a(\mathbf{s}_1) & \ldots & a(\mathbf{s}_k) \end{bmatrix}$$

has a multivariate Gaussian distribution. We can thus describe a Gaussian random field on $\mathbb{R}^2$ with a mean function $m(\mathbf{s})$ and a covariance function $\mathcal{C}(\mathbf{s},\mathbf{s}^*)$.

Similarly, a log-Gaussian random field is a random field such that the natural log of the field is a Gaussian random field. Thus, if $a(\mathbf{s})$ is a log-Gaussian random field, then $\ln(a(\mathbf{s}))$ is a Gaussian random field. Using our description of Gaussian random fields above, we can also say that, for any integer $k > 0$, and any k fixed points $\mathbf{s}_1, \ldots, \mathbf{s}_k \in \mathbb{R}^2$ the vector

$$\begin{bmatrix} \ln(a(\mathbf{s}_1)) & \ldots & \ln(a(\mathbf{s}_k)) \end{bmatrix}$$

has a multivariate Gaussian distribution. Log-Gaussian fields are described by specifying the mean and covariance function of the underlying Gaussian random field.

If $a(\mathbf{s})$ is a second order random field, meaning that $\mathbf{E}\left[a(\mathbf{s})^2\right] \leq \infty$ [87, Definition 4.43], then $a(\mathbf{s})$ can be expressed with a Karhunen-Loéve expansion (see [87, (5.5)]). Let μ_i and $\phi_i(\mathbf{s})$, $1 \leq i \leq \infty$, be the eigenvalues and orthonormal eigenfunctions of $\mathcal{C}(\mathbf{s}, \mathbf{s}')$. Then,

$$a(\mathbf{s}) = \sum_{i=1}^{\infty} \sqrt{\mu_i}\phi_i(\mathbf{s})x_i,$$

where x_i are random variables. If $a(\mathbf{s})$ is a Gaussian random field, then x_i are standard Gaussian random variables [87, pg 110].

4.9 Piecewise multilinear interpolation on sparse grids

We borrow the notation of Barthelmann et. al [5] to describe piecewise multilinear interpolation over a sparse grid. See also [68, Section 3.3] and [48]. The main advantage of sparse grids is that the accuracy of a base interpolation method (for us, piecewise linear interpolation) in one dimension is preserved in d dimensions, using many fewer points than are contained in a full grid in d dimensions.

Suppose we want to interpolate a function f in d dimensions on $[0,1]^d$ and that for each dimension we have a set of

$$m_i = \begin{cases} 1 & : i = 1 \\ 2^{i-1} + 1 & : i > 1 \end{cases}$$

equally spaced nodes $x^i_{j_1}, \ldots, x^i_{j_{m_i}}$ where

$$x^i_{j_k} = \begin{cases} \frac{1}{2} & : k = 1,\ m_i = 1 \\ \frac{k-1}{m_i - 1} & : k = 1, \ldots, m_i,\ m_i > 1; \end{cases}$$

the standard hat functions $h^i_{j_1}(x^i), \ldots, h^i_{j_{m_i}}(x^i)$ (see [68, pg. 38 (c)]) centered at the nodes; and a corresponding one-dimensional interpolation formula

$$U^i(f) = \sum_{j=1}^{m_i} f(x^i_j)h^i_j.$$

To construct the interpolant over all d dimensions, we simply tensor product the individual

approximations

$$(U^{i_1} \otimes \ldots \otimes U^{i_d})(f) = \sum_{j_1=1}^{m_{i_1}} \ldots \sum_{j_d=1}^{m_{i_d}} f(x_{j_1}^{i_1}, \ldots, x_{j_d}^{i_d})(h_{j_1}^{i_1} \otimes \ldots \otimes h_{j_d}^{i_d}),$$

where $h_{j_1}^{i_1} \otimes \ldots \otimes h_{j_d}^{i_d} \equiv \prod_{k=1}^{d} h_{j_k}^{i_k}$. This approximation is over the full grid and is very expensive to compute.

The sparse grid and the approximation over the sparse grid are defined as follows. Let $\Delta^i = U^i - U^{i-1}$ where $U^0 = 0$. Then the sparse grid interpolation, for $\mathbf{i} = (i_1, \ldots, i_d)^T$ [4] and any integer $q \geq d$, is

$$\sum_{\|\mathbf{i}\|_1 \leq q} (\Delta^{i_1} \otimes \ldots \otimes \Delta^{i_d})(f).$$

The level ℓ of a sparse grid is defined to be $\ell \equiv q - d$.

The sparse grid we have described is called the Clenshaw-Curtis grid. Other sparse grids are possible (see [68, Section 3.3], and not all sparse grids have equally spaced nodes.

As an example, let $d = 2$ and choose $q = 3$ so that we construct the sparse grid of level $\ell = 1$. Then we have the sparse grid interpolation formula

$$(\Delta^1 \otimes \Delta^1)(f) + (\Delta^1 \otimes \Delta^2)(f) + (\Delta^2 \otimes \Delta^1)(f).$$

The node associated with $\Delta^1 = U^1 - U^0 = U^1$ is $\{1/2\}$. The nodes associated with $\Delta^2 = U^2 - U^1$ are $\{0, 1\}$. The nodes associated with each tensor product are shown in Figure 4.7, along with the complete level one sparse grid. In Figure 4.8, we show the level two sparse grid, which consists of the level one sparse grid, as well as the nodes associated with $\Delta^2 \otimes \Delta^2$, $\Delta^3 \otimes \Delta^1$, and $\Delta^1 \otimes \Delta^3$.

[4] Each i_j is a positive integer.

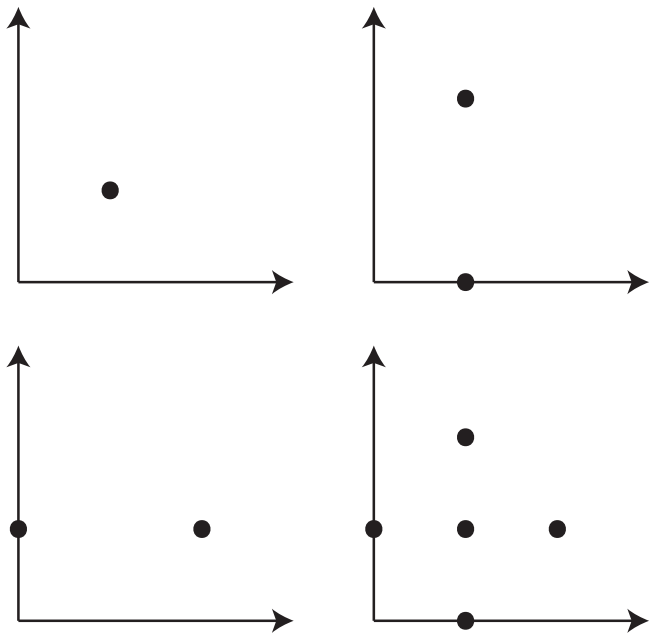


Figure 4.7: Top left: $\Delta^1 \otimes \Delta^1$. Top right: $\Delta^1 \otimes \Delta^2$. Bottom left: $\Delta^2 \otimes \Delta^1$. Bottom right: level one sparse grid.

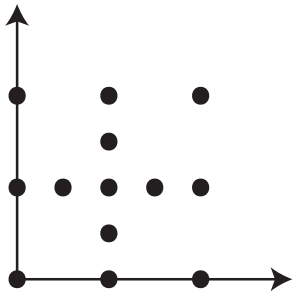


Figure 4.8: Level two sparse grid.

REFERENCES

[1] P. Abrahamsen. *A review of Gaussian random fields and correlation functions*. Norwegian Computing Center, 2nd edition, 1997.

[2] H. Avron, P. Maymounkov, and S. Toledo. Blendenpik: Supercharging LAPACK's least-squares solver. *SIAM J. Sci. Comput.*, 32(3):1217, 2010.

[3] K. Bache and M. Lichman. UCI machine learning repository. http://archive.ics.uci.edu/ml, 2013.

[4] Y. Bang, H. S. Abdel-Khalik, and J. M. Hite. Hybrid reduced order modeling applied to nonlinear models. *Internat. J. Numer. Methods Engrg.*, 91(9):929–949, 2012.

[5] V. Barthelmann, E. Novak, and K. Ritter. High dimensional polynomial interpolation on sparse grids. *Adv. Comput. Math.*, 12(4):273–288, 2000.

[6] J. Batson, D. A. Spielman, and N. Srivastava. Twice-Ramanujan sparsifiers. *SIAM J. Comput.*, 41(6):1704–1721, 2012.

[7] J. D. Batson, D. A. Spielman, and N. Srivastava. Twice-Ramanujan sparsifiers. In *STOC'09—Proceedings of the 2009 ACM International Symposium on Theory of Computing*, pages 255–262. ACM, New York, 2009.

[8] M. Bauerheim, A. Ndiaye, P. Constantine, G. Iaccarino, S. Moreau, and F. Nicoud. Uncertainty quantification of thermo-acoustic instabilities in annular combustors. Technical report, Center for Turbulence Research, Stanford University, 2014.

[9] M.-A. Belabbas and P. J. Wolfe. On sparse representations of linear operators and the approximation of matrix products. In *Proc. 42nd Ann. Conf. Information Sciences and Systems*, pages 258–263, 2008.

[10] R. Bhatia and K. Mukherjea. Variation of the unitary part of a matrix. *SIAM J. Matrix Anal. Appl.*, 15(3):1007–1014, 1994.

[11] C. Boutsidis. *Topics in matrix sampling algorithms.* PhD thesis, Rensselaer Polytechnic Institute, 2011.

[12] C. Boutsidis, P. Drineas, and M. Magdon-Ismail. Near-optimal column-based matrix reconstruction. In *2011 IEEE 52nd Ann. Symp. on Foundations of Computer Science (FOCS)*, pages 305–314. IEEE Comput. Soc. Press, Los Alamitos, CA, 2011.

[13] C. Boutsidis and A. Gittens. Improved matrix algorithms via the subsampled randomized Hadamard transform. *SIAM J. Matrix Anal. Appl.*, 34(3):1301–1340, 2013.

[14] C. Boutsidis, M. W. Mahoney, and P. Drineas. An improved approximation algorithm for the column subset selection problem. In *Proc. 19th Ann. ACM-SIAM Symp. Discrete Algorithms*, pages 968–977, Philadelphia, 2009. SIAM.

[15] C. Boutsidis, M. W. Mahoney, and P. Drineas. An improved approximation algorithm for the column subset selection problem, 2010. arXiv:0812.4293.

[16] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Found. Comput. Math.*, 9(6):717–772, 2009.

[17] Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Found. Comput. Math.*, 9(6):717–772, 2009.

[18] S. Chandrasekaran and I. C. F. Ipsen. On rank-revealing QR factorisations. *SIAM J. Matrix Anal. Appl.*, 15:592–622, 1994.

[19] X.-W. Chang. On the perturbation of the Q-factor of the QR factorization. *Numer. Linear Algebra Appl.*, 19(3):607–619, 2012.

[20] X-W. Chang and C. C. Paige. Componentwise perturbation analyses for the QR factorization. *Numer. Math.*, 88(2):319–345, 2001.

[21] X.-W. Chang, C. C. Paige, and G. W. Stewart. Perturbation analyses for the QR factorization. *SIAM J. Matrix Anal. Appl.*, 18(3):775–791, 1997.

[22] X.-W. Chang and D. Stehlé. Rigorous perturbation bounds of some matrix factorizations. *SIAM J. Matrix Anal. Appl.*, 31(5):2841–2859, 2010.

[23] S. Chatterjee and A. S. Hadi. Influential observations, high leverage points, and outliers in linear regression. *Statist. Sci.*, 1(3):379–416, 1986. With discussion.

[24] H. Chen, Wang Q., R. Hu, and Constantine P. G. Conditional sampling and experiment design for quantifying manufacturing error of transonic airfoil. In *Proceedings of the 49th AIAA Aerospace Sciences Meeting*, 2011.

[25] E. Cohen and D. D. Lewis. Approximating matrix multiplication for pattern recognition tasks. In *Proc. 8th Ann. ACM-SIAM Symp. on Discrete Algorithms*, pages 682–691, 1997.

[26] E. Cohen and D. D. Lewis. Approximating matrix multiplication for pattern recognition tasks. *J. Algorithms*, 30:211–252, 1999.

[27] P. G. Constantine, A. Doostan, Q. Wang, and G. Iaccarino. A surrogate accelerated Bayesian inverse analysis of the HyShot ii flight data. In *Proceedings of the 52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Material Conference*, 2011.

[28] P. G. Constantine, E. Dow, and Q Wang. Active subspace methods in theory and practice: applications to kriging surfaces. *SIAM J. Sci. Comput.*, 36(4):A1500–A1524, 2014.

[29] P. G. Constantine, E. Dow, and Q. Wang. Erratum: Active subspace methods in theory and practice: applications to kriging surfaces. *SIAM J. Sci. Comput.*, 36(6):A3030–A3031, 2014.

[30] P. G. Constantine and D. Gleich. Computing active subspaces, 2014. arXiv:1408.0545.

[31] P. G. Constantine, Q. Wang, and G. Iaccarino. A method for spatial sensitivity analysis. Technical report, Center for Turbulence Research, Stanford University, 2012.

[32] P. G. Constantine, B. Zaharatos, and M. Campanelli. Discovering an active subspace in a single-diode solar cell model, 2014. arXiv:1406.7607.

[33] T. A. Davis and Y. Hu. The University of Florida Sparse Matrix Collection. *ACM Trans. Math. Software*, 38:1–25, 2011.

[34] L. Dieci and T. Eirola. On smooth decompositions of matrices. *SIAM J. Matrix Anal. Appl.*, 20(3):800–819, 1999.

[35] P. Drineas and R. Kannan. Fast Monte-Carlo algrithms for approximate matrix multiplication. In *Proc. 42nd IEEE Symp. Foundations of Computer Science (FOCS)*, pages 452–459, Los Alamitos, CA, 2001. IEEE Comput. Soc. Press.

[36] P. Drineas, R. Kannan, and M. W. Mahoney. Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication. *SIAM J. Comput.*, 36(1):132–157, 2006.

[37] P. Drineas, M. Magdon-Ismail, M. W. Mahoney, and D. P. Woodruff. Fast approximation of matrix coherence and statistical leverage. *J. Mach. Learn. Res.*, 13:3475–3506, 2012.

[38] P. Drineas, M. W. Mahoney, and S. Muthukrishnan. Sampling algorithms for l_2 regression and applications. In *Proc. 17th Ann. ACM-SIAM Symp. on Discrete Algorithms*, SODA '06, pages 1127–1136, New York, NY, 2006. ACM.

[39] P. Drineas, M. W. Mahoney, and S. Muthukrishnan. Sampling algorithms for l_2 regression and applications. In *Proc. 17th Ann. ACM-SIAM Symp. Discrete Algorithms*, pages 1127–1136, New York, 2006. ACM.

[40] P. Drineas, M. W. Mahoney, and S. Muthukrishnan. Subspace sampling and relative-error matrix approximation: Column-based methods. In *Approximation, randomization and combinatorial optimization*, volume 4110 of *Lecture Notes in Comput. Sci.*, pages 316–326. Springer, Berlin, 2006.

[41] P. Drineas, M. W. Mahoney, and S. Muthukrishnan. Relative-error CUR matrix decompositions. *SIAM J. Matrix Anal. Appl.*, 30(2):844–881, 2008.

[42] P. Drineas, M. W. Mahoney, and S. Muthukrishnan. Relative-error CUR matrix decompositions. *SIAM J. Matrix Anal. Appl.*, 30(2):844–881, 2008.

[43] P. Drineas, M. W. Mahoney, S. Muthukrishnan, and T. Sarlós. Faster least squares approximation. *Numer. Math.*, 117(2):219–249, 2010.

[44] S. Eriksson-Bique, M. Solbrig, M. Stefanelli, S. Warkentin, R. Abbey, and I. C. F. Ipsen. Importance sampling for a Monte Carlo matrix multiplication algorithm, with application to information retrieval. *SIAM J. Sci. Comput.*, 33(4):1689–1706, 2011.

[45] S. Friedland and A. Torokhti. Generalized rank-constrained matrix approximations. *SIAM J. Matrix Anal. Appl.*, 29(2):656–659, 2007.

[46] A. Frieze, R. Kannan, and S. Vempala. Fast monte-carlo algorithms for finding low-rank approximations. In *Proc. 39th Ann. Symp. Foundations of Computer Science (FOCS)*, pages 370–378, Los Alamitos, CA, 1998. IEEE Comput. Soc. Press.

[47] A. Frieze, R. Kannan, and S. Vempala. Fast Monte-Carlo algorithms for finding low-rank approximations. *J. ACM*, 51(6):1025–1041, November 2004.

[48] T. Gerstner and M. Griebel. Sparse grids. In *Encyclopedia of Quantitative Finance*. Wiley, 2008.

[49] A. Gittens. The spectral norm error of the naïve Nyström extension. arxiv:1110.5305v1, 2011.

[50] A. Gittens. *Topics in randomized numerical linear algebra*. PhD thesis, California Institute of Technology, 2013.

[51] G. Golub. Numerical methods for solving linear least squares problems. *Numer. Math.*, 7(3):206–216, 1965.

[52] G. Golub, V. Klema, and G. Stewart. Rank degeneracy and least squares problems. Technical Report STAN-CS-76-559, Computer Science Department, Stanford University, 1976.

[53] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, third edition, 1996.

[54] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, fourth edition, 2013.

[55] G. R. Grimmett and D. R. Stirzaker. *Probability and random processes*. Oxford University Press, New York, third edition, 2001.

[56] M. Gu and S. C. Eisenstat. Efficient algorithms for computing a strong rank-revealing qr factorization. *SIAM J. Sci. Comput.*, 17(4):848–869, July 1996.

[57] N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.*, 53(2):217–288, 2011.

[58] N. J. Higham. Computing the polar decomposition—with applications. *SIAM J. Sci. Statist. Comput.*, 7(4):1160–1174, 1986.

[59] N. J. Higham. *Accuracy and stability of numerical algorithms*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second edition, 2002.

[60] D. C. Hoaglin and R. E. Welsch. The Hat matrix in regression and ANOVA. *Amer. Statist.*, 32(1):17–22, 1978.

[61] J. T. Holodnak and I. C. F. Ipsen. Randomized Approximation of the Gram Matrix: Exact Computation and Probabilistic Bounds. *SIAM J. Matrix Anal. Appl.*, 36(1):110–137, 2015.

[62] J. T. Holodnak, I. C. F. Ipsen, and T. Wentworth. Conditioning of leverage scores and computation by QR decomposition. arXiv:1402.0957, 2014.

[63] H. Hong and C. Pan. The rank-revealing QR decomposition and SVD. *Math. Comp.*, 58:213–232, 1992.

[64] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge University Press, Cambridge, second edition, 2013.

[65] D. Hsu, S. M. Kakade, and T. Zhang. Tail inequalities for sums of random matrices that depend on the intrinsic matrix dimension. *Electron. Commun. Probab.*, 17(14):1–13, 2012.

[66] I. C. F. Ipsen and T. Wentworth. The effect of coherence on sampling from matrices with orthonormal columns, and preconditioned least squares problems. *SIAM J. Matrix Anal. Appl.*, 35(4):1490–1520, 2014.

[67] I. T. Jolliffe. *Principal component analysis*. Springer Series in Statistics. Springer-Verlag, New York, second edition, 2002.

[68] A. Klimke. *Uncertainty Modeling using Fuzzy Arithmetic and Sparse Grids*. PhD thesis, University of Stuttgart, 2006.

[69] A. Klimke. Sparse Grid Interpolation Toolbox – User's guide. Technical Report IANS report 2007/017, University of Stuttgart, 2007.

[70] A. Klimke and B. Wohlmuth. Algorithm 847: spinterp: Piecewise multilinear hierarchical sparse grid interpolation in MATLAB. *ACM Transactions on Mathematical Software*, 31(4), 2005.

[71] S. Kumar, M. Mohri, and A. Talwalkar. Sampling techniques for the Nyström method. In *Proc. 12th Int. Conf. Artificial Intelligence and Statistics*, volume 5, pages 304–311, 2009.

[72] M. Li, G. L. Miller, and R. Peng. Iterative row sampling. In *Proc. 54th IEEE Symp. Foundations of Computer Science (FOCS)*, pages 127–136, Los Alamitos, CA, 2013. IEEE Computer Society.

[73] E. Liberty. Simple and deterministic matrix sketching. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 581–588, New York, NY, USA, 2013. ACM.

[74] H. Madrid, V. Guerra, and M. Rojas. Sampling techniques for Monte Carlo matrix multiplication with applications to image processing. In *Proc. 4th Mexican Conference on Pattern Recognition*, pages 45–54, 2012.

[75] M. Magdon-Ismail. Row sampling for matrix algorithms via a non-commutative Bernstein bound. arXiv:1008.0587, 2010.

[76] M. Magdon-Ismail. Using a non-commutative Bernstein bound to approximate some matrix algorithms in the spectral norm. arXiv1103.5453v1, 2011.

[77] A. Magen and A. Zouzias. Low rank matrix-valued Chernoff bounds and approximate matrix multiplication. In *Proc. 22nd Ann. ACM-SIAM Symp. Discrete Algorithms*, pages 1422–1436, Philadelphia, 2011. SIAM.

[78] M. W. Mahoney. Randomized algorithms for matrices and data. *Foundations and Trends in Machine Learning*, 3(2):123–224, 2011.

[79] M. W. Mahoney. *Randomized Algorithms for Matrices and Data.* Now Publishers Inc., 2011.

[80] N. Namura, K. Shimoyama, and S. Obayashi. Kriging surrogate model enhanced by coordinate transformation of design space based on eigenvalue decomposition. In *Evolutionary Multi-Criterion Optimization*, Lecture Notes in Computer Science. Springer International Publishing, 2015.

[81] R. Pagh. Compressed matrix multiplication. *ACM Trans. Comput. Theory*, 5(3):Art. 9, 17, 2013.

[82] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, 2006.

[83] S. Ross. *Introduction to Probability Models*. Elsevier, Burlington, MA, 10th edition, 2010.

[84] M. Rudelson and R. Vershynin. Sampling from large matrices: an approach through geometric functional analysis. *J. ACM*, 54(4):Art. 21, 19 pp. (electronic), 2007.

[85] T. M. Russi. *Uncertainty Quantification with experimental data and complex system models*. PhD thesis, University of California, Berkeley, 2010.

[86] T. Sarlós. Improved approximation for large matrices via random projections. In *Proc. 47th Ann. IEEE Symp. Foundations of Computer Science (FOCS)*, pages 143–152, Los Alamitos, CA, 2006. IEEE Comput. Soc. Press.

[87] R. C. Smith. *Uncertainty quantification: Theory, implementation, and applications*. SIAM, Philadelphia, PA, 2014.

[88] I. M. Sobol' and S. Kucherenko. Derivative based global sensitivity measures and their link with global sensitivity indices. *Math. Comput. Simulation*, 79(10):3009–3017, 2009.

[89] N. Srivastava. *Spectral sparsification and restricted invertibility*. PhD thesis, Yale University, 2010.

[90] G. W. Stewart. Perturbation bounds for the QR factorization of a matrix. *SIAM J. Numer. Anal.*, 14(3):509–518, 1977.

[91] G. W. Stewart. On the perturbation of LU, Cholesky, and QR factorizations. *SIAM J. Matrix Anal. Appl.*, 14(4):1141–1145, 1993.

[92] M. Stoyanov and C. G. Webster. A gradient-based sampling approach for dimension reduction of partial differential equations with stochastic coefficients. *Int. J. Uncertainty Quantification*, 2014.

[93] J.-G. Sun. Perturbation bounds for the Cholesky and QR factorizations. *BIT*, 31(2):341–352, 1991.

[94] J.-G. Sun. Componentwise perturbation bounds for some matrix decompositions. *BIT*, 32(4):702–714, 1992.

[95] J.-G. Sun. On perturbation bounds for the QR factorization. *Linear Algebra Appl.*, 215:95–111, 1995.

[96] A. Talwalkar and A. Rostamizadeh. Matrix coherence and the Nyström method. In *Proc. 26th Conf. Uncertainty in Artificial Intelligence (UAI-10)*, pages 572–579. AUAI Press, Corvalis, Oregon, 2010.

[97] J. Tropp. User-friendly tools for random matrices: An introduction. users.cms.caltech.edu /~jtropp/pubs.html, 2012.

[98] J. A. Tropp. Improved analysis of the subsampled Hadamard transform. *Adv. Adapt. Data Anal.*, 3(1-2):115–126, 2011.

[99] J. A. Tropp. User-friendly tail bounds for sums of random matrices. *Found. Comput. Math.*, pages 1–46, 2011.

[100] P. F. Velleman and R. E. Welsch. Efficient computing of regression diagnostics. *Amer. Statist.*, 35(4):234–242, 1981.

[101] H. Y. Zha. A componentwise perturbation analysis of the QR decomposition. *SIAM J. Matrix Anal. Appl.*, 14(4):1124–1131, 1993.

[102] X. Zhan. Singular values of differences of positive semidefinite matrices. *SIAM J. Matrix Anal. Appl.*, 22(3):819–823, 2000.

[103] A. Zouzias. *Randomized primitives for linear algebra and applications*. PhD thesis, University of Toronto, 2013.