Fast Length-Squared Sampling for Positive-Semidefinite Matrix Approximation

by
Aaron Tian

submitted in partial fulfillment
of the requirements for the degree of
Bachelor of Science
at the
University of Massachusetts Amherst
June 2026

Abstract

We describe an algorithm to perform *length-squared sampling* on a positive semidefinite matrix: that is, to sample a column with probability proportional to its squared ℓ_2-norm, using only $O(n)$ entries of the input in expectation. With this result in mind, we revisit a number of sublinear matrix approximation algorithms that previously assumed access to the row norms of the input matrix. We use our length-squared sampling approach to derive new sublinear algorithms for positive semidefinite matrix approximation that do not require access to the squared row or column norms of the input.

Contents

Chapter 1

Introduction

In modern data analysis, some naturally-arising matrices are too large to compute and store efficiently. A canonical example comes from *kernel methods* in machine learning. Given n data points $x_1, \ldots, x_n$ and a positive semidefinite kernel function $k(\cdot, \cdot)$, an associated kernel matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is defined entrywise by

$$\mathbf{A}_{ij} = k(x_i, x_j).$$

Although we can compute any single entry $\mathbf{A}_{ij}$ on demand, computing and storing all n^2 entries is prohibitively expensive when n, the size of the dataset, is large. Thus: *how can we perform matrix computations when the input is available only through limited entry queries?*

An effective approach is to work with a small sketch of the matrix obtained by sampling a subset of rows or columns. Among the most natural and widely studied sampling rules is *length-squared sampling*, in which a row or column is sampled with probability proportional to its squared ℓ_2 norm. This rule appears, for example, in the randomized Kaczmarz method for solving linear systems [NSW15] and in additive-error algorithms for low-rank approximation [FKV04]. However, a direct implementation of length-squared sampling must have access to the relevant row or column norms. If these norms are not already available, then forming the sampling distribution requires a preprocessing pass over the entire input matrix. For dense $n \times n$ matrices, this takes $\Theta(n^2)$ time.

1.1 Our Results

We consider the problem of sampling a column with probability proportional to its squared ℓ_2 norm, i.e., sample column i with probability

$$p_i = \frac{\|\mathbf{A}_{*,i}\|_2^2}{\|\mathbf{A}\|_F^2}$$

where $\mathbf{A}_{*,i}$ denotes the ith column of $\mathbf{A}$. We show that for a psd matrix, we can draw such a sample without having to read all entries of the input. Our algorithm queries $O(n)$ entries of $\mathbf{A}$ and runs in $O(n)$ time in expectation. If the number of needed samples is small compared to n, our algorithm offers significant speedup over pre-computing all the column norms.

We next show how this sampling primitive can be used in several low-rank and spectral approximation problems for positive semidefinite matrices. We consider eigenvalue approximation, additive-error low-rank approximation, and robust low-rank approximation. In each application, our sampler replaces a length-squared column sampling step that would otherwise require computing all the column norms.

Finally, we consider the stronger goal of relative-error low-rank approximation. This part of the thesis studies RPCHOLESKY, a randomized Nyström method with a relative trace-error guarantee, as a natural candidate for relative Frobenius error approximation. Although it was previously known that RPCHOLESKY itself does not have a relative Frobenius error guarantee [Epp25], we consider whether a modified sampling primitive might achieve the desired relative error guarantee. We conclude with a query lower bound that shows any Nyström method requires $\Omega(n^{1.5})$ samples to achieve a relative Frobenius error guarantee in the worst case. Thus we cannot hope to use Nyström approximation for sublinear low-rank approximation *with only linear dependence* on n, in contrast to the results of [MW19, BCW20].

1.2 Outline

Chapter 2 collects the background material used throughout the thesis. We review low-rank approximation, positive semidefinite matrices, Nyström approximation, the entry-query model, and several sampling primitives that arise in randomized matrix approximation.

Chapter 3 develops the main sampling primitive. We show that, for a psd matrix, one can sample a column with probability proportional to its squared ℓ_2 norm using only $O(n)$ entry queries in expectation. The chapter then applies this sampler to several matrix approximation tasks, including Frobenius norm estimation, additive-error eigenvalue approximation, additive-error low-rank approximation, and robust low-rank approximation.

Chapter 4 investigates relative-error low-rank approximation. We focus on RPCHOLESKY, an adaptive Nyström method with a known relative trace-error guarantee, and ask whether a modified sampling technique can also yield a Frobenius relative-error guarantee. We present empirical results suggesting good performance on kernel matrices, followed by a worst-case construction showing that this approach fails in general.

Chapter 5 concludes by summarizing the main contributions and discussing several open questions.

Chapter 2

Background

2.1 Preliminaries

This section collects some relevant mathematical knowledge. Throughout, bold uppercase letters denote real-valued matrices and bold lowercase letters denote real-valued vectors. For a given matrix $\mathbf{M}$, we write $\mathbf{M}_{ij}$ for the entry at row i and column j of $\mathbf{M}$. Moreover, we use $\mathbf{M}_{i,*}$ to denote the ith row vector and $\mathbf{M}_{*,i}$ to denote the ith column vector of $\mathbf{M}$.

Low-Rank Approximation

Given a matrix $\mathbf{M} \in \mathbb{R}^{n \times d}$ and target rank k, the low-rank approximation problem is to find a matrix $\widehat{\mathbf{M}}$ of rank at most k that approximates $\mathbf{M}$. To measure the quality of the approximation, we consider the size of the *residual matrix* $\mathbf{M} - \widehat{\mathbf{M}}$ with respect to some suitable matrix norm. We focus primarily on the *Frobenius norm*, $\|\cdot\|_F$. A good approximation to $\mathbf{M}$ should make $\|\mathbf{M} - \widehat{\mathbf{M}}\|_F$ small.

The *singular value decomposition (svd)* yields an optimal rank-k approximation to $\mathbf{M}$ in the following manner. Let $r = \text{rank}(\mathbf{M})$ and write

$$\mathbf{M} = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$$

for the singular value decomposition of $\mathbf{M}$, where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0$ are the singular values of $\mathbf{M}$, and the $\mathbf{u}_i$ and $\mathbf{v}_i$ are corresponding left and right singular vectors of $\mathbf{M}$, respectively. The matrix $\mathbf{M}_k$ is defined as

$$\mathbf{M}_k := \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top.$$

Fact 1. Let $\mathbf{M} \in \mathbb{R}^{n \times d}$ and $\mathbf{M}_k$ be as defined above. Then

$$\|\mathbf{M} - \mathbf{M}_k\|_F \leq \|\mathbf{M} - \mathbf{X}\|_F$$

for all matrices $\mathbf{X}$ of rank at most k. In particular, $\mathbf{M}_k$ is an optimal rank-k approximation to $\mathbf{M}$ with respect to $\|\cdot\|_F$.

Positive Semidefinite Matrices

A symmetric matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is *positive semidefinite (psd)* if $\mathbf{x}^\top \mathbf{A}\mathbf{x} \geq 0$ for all $\mathbf{x} \in \mathbb{R}^n$. For symmetric matrices $\mathbf{A}$, we have the following standard set of equivalences:

1. $\mathbf{A}$ is psd
2. the eigenvalues of $\mathbf{A}$ are nonnegative
3. $\mathbf{A} = \mathbf{U}^\top \mathbf{U}$ for some matrix $\mathbf{U}$.

Fact 2. Let $\mathbf{A}$ be an $n \times n$ psd matrix. For all $i, j \in [n]$, we have the inequality

$$\mathbf{A}_{ij}^2 \leq \mathbf{A}_{ii}\mathbf{A}_{jj}.$$

Nyström Approximation

Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be a psd matrix and $S \subset [n]$. Define

$$\mathbf{C} := \mathbf{A}_{*,S} \qquad \mathbf{W} := \mathbf{A}_{S,S}$$

We write $\mathbf{A}\langle S \rangle$ to denote the *Nyström approximation* of $\mathbf{A}$ with index set S, where

$$\mathbf{A}\langle S \rangle = \mathbf{C}\mathbf{W}^{-1}\mathbf{C}^\top.$$

Both the Nyström approximation and its residual, $\mathbf{A} - \mathbf{A}\langle S \rangle$, are psd. Moreover, $\operatorname{rank}(\mathbf{A}\langle S \rangle) \leq |S|$.

2.2 Sublinear Matrix Approximation

As in the kernel matrix example from Chapter 1, we consider large matrices to which we may access individual entries without forming the entire matrix explicitly. This motivates the *entry-query model*: from indices (i, j), an algorithm may query and receive the entry $\mathbf{A}_{ij}$. For a dense $n \times n$ matrix, reading the full input requires n^2 entry queries. We call an algorithm *sublinear* if it uses $o(n^2)$ queries and $o(n^2)$ additional arithmetic in the entry query model.

This thesis was motivated by sublinear low-rank approximation. Given a matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, the goal is to output a matrix $\widehat{\mathbf{A}}$ of small rank such that the residual $\mathbf{A} - \widehat{\mathbf{A}}$ is small. Note that even forming all entries of $\widehat{\mathbf{A}}$ is too expensive in the sublinear setting, so instead we output approximation factors $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n\times k}$, where k is the rank of $\widehat{\mathbf{A}}$, so that $\mathbf{X}\mathbf{Y}^\top = \widehat{\mathbf{A}}$.

There are two common types of error guarantees for low-rank approximation. A *relative-error* guarantee compares the algorithm's error to the best possible rank-r approximation:

$$\|\mathbf{A} - \widehat{\mathbf{A}}\|_F^2 \le (1+\epsilon)\|\mathbf{A} - \mathbf{A}_r\|_F^2,$$

where $\mathbf{A}_r$ denotes the optimal rank-r approximation to $\mathbf{A}$. An *additive-error* guarantee instead bounds the error by

$$\|\mathbf{A} - \widehat{\mathbf{A}}\|_F^2 \le \|\mathbf{A} - \mathbf{A}_r\|_F^2 + \epsilon\|\mathbf{A}\|_F^2,$$

Additive error is generally weaker than relative error since $\|\mathbf{A} - \mathbf{A}_r\|_F^2 \le \|\mathbf{A}\|_F^2$.

For general matrices, sublinear-time low-rank approximation is impossible: Clarkson and Woodruff give an $\Omega(\text{nnz}\mathbf{A})$ query lower bound, where $\text{nnz}\mathbf{A}$ denotes the number of nonzero entries of $\mathbf{A}$ [CW13]. At the same time, this barrier can be avoided on structured classes of matrices. A large body of work focuses on positive semidefinite matrices; for example, Musco and Woodruff show that for any psd matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, one can compute a relative-error low-rank approximation with time and query complexity $\widetilde{O}\big(n \cdot \text{poly}(k/\epsilon)\big)$[1]. Sublinear-time algorithms have also been developed for distance matrices [BW18], Toeplitz matrices [MS24], and Hankel matrices [KMS25].

[1] we use $\widetilde{O}$ to hide polylogarithmic factors in n, k, and $1/\epsilon$.

2.3 Sampling Primitives for Matrix Approximation

A common technique used in query-efficient matrix algorithms is to construct a small sketch of the input via a subset of sampled rows or columns. Generally speaking, the quality and cost of the resulting approximation depend on how this subset is chosen. In this section, we review several sampling rules and their related results.

Uniform Sampling

The uniform sampling method picks k rows or columns uniformly at random, either with or without replacement. A clear advantage of uniform sampling is that it is data agnostic: it generates a sample without using any information about the matrix. Moreover, uniform sampling is often effective in practice [KMT12, GM13].

Uniform sampling appears in the seminal work of Williams and Seeger [WS00] on the Nyström method. They show that a Nyström approximation with uniformly sampled index set can substantially reduce the cost of kernel computations without a significant loss in accuracy in practice.

Subsequent work shows that the performance of uniform sampling depends on the coherence of the relevant singular or eigenspaces [TR10, Git11]. Informally, coherence measures whether the important directions of the matrix are spread evenly or concentrated on a small number of coordinates. When the coherence is small, important directions are well-spread among the rows or columns, so uniform sampling is likely to capture them. When the coherence is large, a small uniform sample may miss important directions with high probability. Consequently, strong worst-case guarantees for uniform sampling typically require incoherence assumptions or a large sample size [KMT12].

Length-Squared Sampling

In contrast to uniform sampling, *importance sampling* assigns sampling probabilities according to a prescribed measure of "importance" of a row or column. When a suitable notion of importance is available, this can lead to stronger approximation guarantees than uniform sampling. However, importance sampling probabilities are data-dependent and thus require a preprocessing step to compute exactly.

Length-squared sampling is an importance sampling rule in which a row or column is picked with probability proportional to its squared ℓ_2 norm. To distinguish between the two cases, we may refer to them as squared column norm sampling and squared row norm sampling, respectively.

For a matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, the squared column norm distribution is

$$p_i = \frac{\|\mathbf{A}_{*,i}\|_2^2}{\|\mathbf{A}\|_F^2}.$$

This distribution appears in several matrix approximation algorithms, including randomized algorithms for additive error low-rank approximation [FKV04], eigenvalue approximation [BDD+22, SW24], and randomized Kaczmarz-type methods for solving linear systems [NSW15].

The main difficulty with length-squared sampling is that the sampling distribution is not usually available for free. To sample from the distribution exactly by direct preprocessing, one must compute all squared row or column norms. In the entry query model, this would require reading every nonzero entry of the matrix, precluding a sublinear-time algorithm.

The central observation of this thesis is that positive semidefinite structure makes length-squared sampling easier. For a psd matrix $\mathbf{A}$, the inequality

$$\mathbf{A}_{ij}^2 \leq \mathbf{A}_{ii}\mathbf{A}_{jj}$$

allows the diagonal distribution to serve as a proposal distribution for rejection sampling. This leads to an exact length-squared sampler using only $O(n)$ entry queries in expectation, avoiding the quadratic preprocessing step needed to compute all column norms. This sampler is the main primitive we develop in Chapter 3.

Leverage Score Sampling

Leverage score sampling is a more refined importance sampling method. For a matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$ of rank r, let $\mathbf{U} \in \mathbb{R}^{m \times r}$ be an orthonormal basis for the columns of $\mathbf{M}$. The statistical leverage scores of the rows of $\mathbf{M}$ are given by

$$\ell(i) := \|\mathbf{U}_{i,*}\|_2^2$$

Sampling proportional to row leverage scores occurs in randomized algorithms for relative error low-rank approximation [DMM07] and least-squares regression [DMMS07].

A closely related notion is ridge leverage score sampling. Given a matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$ and regularization parameter $\lambda > 0$, the ridge leverage scores are

$$\ell_\lambda(i) = \left(\mathbf{M}(\mathbf{M} + \lambda \mathbf{I})^{-1}\right)_{ii}.$$

Ridge leverage scores have led to strong guarantees for Nyström approximation and related low-rank approximation problems for psd matrices [MM17, CW17, MW19]. In particular, Musco and Woodruff [MW19] show that the ridge leverage scores of a psd matrix $\mathbf{A}$ can be approximated in sublinear time to sufficient accuracy for relative error low-rank approximation.

Adaptive Sampling

Adaptive sampling is a sequential column selection method in which the sampling distribution is updated after each selected column. Given a matrix $\mathbf{B}$ and a current set of selected columns S, let $\mathbf{P}_{\mathbf{B}_S}$ denote the orthogonal projection onto the span of the selected columns. Adaptive sampling chooses the next column j with probability proportional to the squared norm of the residual column:

$$p_j = \frac{\|(\mathbf{B} - \mathbf{P}_{\mathbf{B}_S}\mathbf{B})_{*,j}\|_2^2}{\|\mathbf{B} - \mathbf{P}_{\mathbf{B}_S}\mathbf{B}\|_F^2}.$$

Thus, columns that are already well-approximated by the current span become less likely to be sampled, while columns that are poorly approximated become more likely.

Adaptive sampling has strong theoretical guarantees. In particular, it can be used to obtain a relative-error Frobenius norm approximation after sampling $\text{poly}(k, 1/\epsilon)$ columns [DV06].

Chapter 3

Fast Length-Squared Sampling

In this chapter, we demonstrate how to sample columns from a psd matrix $\mathbf{A}$ such that any column i is picked with probability

$$p_i = \frac{\|\mathbf{A}_{*,i}\|_2^2}{\|\mathbf{A}\|_F^2}$$

while reading only $O(n)$ entries of $\mathbf{A}$ in expectation.

Our algorithm applies a well-known technique called *rejection sampling*. One related application of this technique is due to Needell, Srebro, and Ward [NSW15], who use rejection sampling to obtain length-squared samples from an arbitrary matrix assuming a known upper bound B on the row norms, i.e.,

$$\|a_i\|_2^2 \leq B \qquad \text{for all } i.$$

Given such a bound, one can obtain a length-squared sample using $O(n^2 B/\|\mathbf{A}\|_F^2)$ entry queries in expectation. However, the cost of this approach can still match that of reading the full input, for example when $B = \Omega(\|\mathbf{A}\|_F^2)$. In the psd setting, we avoid the need for an external row-norm upper bound and obtain a sampler whose expected query complexity depends only linearly on n. We present the algorithm below.

Algorithm 1: Fast Length-Squared Sampling

Input: psd matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$.

Let d be the diagonal distribution given by $d_i = \frac{\mathbf{A}_{ii}}{\mathrm{tr}(\mathbf{A})}$. Repeat the following in rounds until an index is accepted:

1. Sample indices $i, j \overset{\text{i.i.d.}}{\sim} d$.
2. Accept and return i with probability $\frac{\mathbf{A}_{ij}^2}{\mathbf{A}_{ii}\mathbf{A}_{jj}}$. Otherwise, reject.

Here, we use the diagonal as a proposal distribution for rejection sampling. Indeed, we note

$$\begin{aligned} p_i = \frac{\|\mathbf{A}_{*,i}\|_2^2}{\|\mathbf{A}\|_F^2} &= \frac{1}{\|\mathbf{A}\|_F^2} \sum_{j=1}^n \mathbf{A}_{ij}^2 \\ &\le \frac{1}{\|\mathbf{A}\|_F^2} \sum_{j=1}^n \mathbf{A}_{ii}\mathbf{A}_{jj} \qquad \text{by Fact 2.} \\ &= \frac{1}{\|\mathbf{A}\|_F^2} \mathbf{A}_{ii} \,\mathrm{tr}(\mathbf{A}) \\ &= \frac{\mathrm{tr}(\mathbf{A})^2}{\|\mathbf{A}\|_F^2} \cdot \frac{\mathbf{A}_{ii}}{\mathrm{tr}(\mathbf{A})} = \frac{\mathrm{tr}(\mathbf{A})^2}{\|\mathbf{A}\|_F^2} d_i \end{aligned} \tag{3.1}$$

so the diagonal distribution d is a valid proposal distribution with constant $M := \mathrm{tr}(\mathbf{A})^2 / \|\mathbf{A}\|_F^2$. At this point, we would like to sample $i \sim d$ and accept with probability

$$\frac{p_i}{M d_i} = \frac{\|\mathbf{A}_{*,i}\|_2^2}{\mathbf{A}_{ii}\,\mathrm{tr}(\mathbf{A})},$$

but computing this quantity would require reading the entire column i at the cost of n entry queries per round. Our main observation is that we can implement the accept probability without reading $\mathbf{A}_{*,i}$ in the following manner.

Let $J \sim d$ be a random index sampled from the diagonal distribution d, and consider the random variable $X_i := \frac{\mathbf{A}_{iJ}^2}{\mathbf{A}_{ii}\mathbf{A}_{JJ}}$. Since $\mathbf{A}$ is psd, X_i takes on values in the range $[0,1]$. Thus, after observing $X_i = x$, the variable

$$Y_i := \begin{cases} 1 & \text{with probability } x \\ 0 & \text{otherwise} \end{cases}$$

is well-defined. Now, we have

$$\begin{aligned}\Pr[Y_i = 1] &= \sum_x \Pr[Y_i = 1 \mid X_i = x] \cdot \Pr[X_i = x] \\ &= \mathbb{E}[X_i] = \sum_{J \in [n]} \frac{\mathbf{A}_{JJ}}{\operatorname{tr}(\mathbf{A})} \frac{\mathbf{A}_{iJ}^2}{\mathbf{A}_{ii}\mathbf{A}_{JJ}} \\ &= \frac{\|\mathbf{A}_{*,i}\|_2^2}{\mathbf{A}_{ii} \operatorname{tr}(\mathbf{A})}\end{aligned} \tag{3.2}$$

which is exactly the probability we need. Moreover, observing Y_i only requires us to read the entries $\mathbf{A}_{iJ}, \mathbf{A}_{ii}$, and $\mathbf{A}_{JJ}$. With this technique in mind, we may view Algorithm 1 as a standard application of rejection sampling. Nonetheless, we give a self-contained theorem statement and proof of correctness below.

Theorem 1. *Algorithm 1 outputs column i with probability*

$$p_i = \frac{\|\mathbf{A}_{*,i}\|_2^2}{\|\mathbf{A}\|_F^2}$$

for all i. Moreover, the expected number of rounds before acceptance is at most n.

Proof. The acceptance probability in each round is well-defined because $\mathbf{A}_{ij}^2 \le \mathbf{A}_{ii}\mathbf{A}_{jj}$ holds for all $i, j \in [n]$ when $\mathbf{A}$ is psd.

In any given round, the probability of outputting i is

$$\frac{\mathbf{A}_{ii}}{\operatorname{tr}(\mathbf{A})} \sum_{j \in [n]} \frac{\mathbf{A}_{jj}}{\operatorname{tr}(\mathbf{A})} \cdot \frac{\mathbf{A}_{ij}^2}{\mathbf{A}_{ii}\mathbf{A}_{jj}} = \sum_{j \in [n]} \frac{\mathbf{A}_{ij}^2}{\operatorname{tr}(\mathbf{A})^2} = \frac{\|\mathbf{A}_{*,i}\|_2^2}{\operatorname{tr}(\mathbf{A})^2}.$$

Moreover, summing over all indices gives the success probability

$$\beta := \frac{\|\mathbf{A}\|_F^2}{\operatorname{tr}(\mathbf{A})^2}$$

for one round. Conditioning on success, the probability of outputting i is thus

$$p_i = \frac{\|\mathbf{A}_{*,i}\|_2^2}{\|\mathbf{A}\|_F^2}$$

as desired. Finally, expand β to obtain a lower bound on the acceptance probability:

$$\begin{aligned}
\beta &= \frac{\|\mathbf{A}\|_F^2}{\operatorname{tr}(\mathbf{A})^2} \\
&= \frac{\sum_{k\in[n]} \lambda_k(\mathbf{A})^2}{\left(\sum_{k\in[n]} \lambda_k(\mathbf{A})\right)^2} \\
&\geq \frac{\sum_{k\in[n]} \lambda_k(\mathbf{A})^2}{\left(\sum_{k\in[n]} 1^2\right)\left(\sum_{k\in[n]} \lambda_k(\mathbf{A})^2\right)} \qquad \text{by Cauchy-Schwarz} \\
&= \frac{1}{n}.
\end{aligned}$$

In particular, the expected number of rounds before the first acceptance is $1/\beta \leq n$. □

Theorem 1 gives an exact length-squared sampler whose query complexity is bounded in expectation. In some settings, however, it may be preferable to have a fixed query budget that does not depend on the random number of rejection sampling rounds. The next corollary shows that using exactly $2n$ entry queries, we can sample from a distribution that is at most a constant factor from the length-squared distribution.

Corollary 1 (Fixed-query length-squared sampling)**.** Suppose we modify Algorithm 1 as follows:

1. Run exactly n rounds.
2. Return the first accepted index if any round accepts.
3. Otherwise, return a uniformly random index.

Then the modified sampling procedure outputs index i with probability

$$p_i \geq \left(1 - e^{-1}\right) \cdot \frac{\|\mathbf{A}_{*,i}\|_2^2}{\|\mathbf{A}\|_F^2}$$

while reading exactly $2n$ entries of $\mathbf{A}$.

Proof. Let

$$\beta = \frac{\|\mathbf{A}\|_F^2}{\operatorname{tr}(\mathbf{A})^2}$$

be the success probability for one round, as in the proof of Theorem 1.

Suppose we run k rounds of the sampling algorithm. The probability of at least one success is $1-(1-\beta)^k$, so the probability of outputting i is at least

$$\left[1-(1-\beta)^k\right]\cdot\frac{\|\mathbf{A}_{*,i}\|_2^2}{\|\mathbf{A}\|_F^2}.$$

In particular, in n rounds the probability of outputting i is

$$\begin{aligned} p_i &\geq \left[1-\left(1-\frac{1}{n}\right)^n\right]\cdot\frac{\|\mathbf{A}_{*,i}\|_2^2}{\|\mathbf{A}\|_F^2} \\ &\geq \left(1-e^{-1}\right)\cdot\frac{\|\mathbf{A}_{*,i}\|_2^2}{\|\mathbf{A}\|_F^2}. \end{aligned}$$

We query the n diagonal entries upfront, followed by one entry per round for a total of $2n$ entry queries as required. □

3.1 Frobenius Norm Approximation

In this section, we extend our techniques to estimate the Frobenius norm of a psd matrix. Recall that in Algorithm 1, we proceed in rounds of rejection sampling to obtain a length-squared sample. In the proof of Theorem 1, we show that the probability of acceptance for each round is $\|\mathbf{A}\|_F^2/\operatorname{tr}(\mathbf{A})^2$. Thus, to estimate $\|\mathbf{A}\|_F^2$, we simply estimate the acceptance probability and multiply by $\operatorname{tr}(\mathbf{A})^2$.

In Theorem 3, we will show that using only $O(n/\epsilon^2)$ entries of $\mathbf{A}$, one may compute an estimate $\widehat{F}$ that is within $(1\pm\epsilon)$ of $\|\mathbf{A}\|_F^2$ with constant probability. If $1/\epsilon^2$ is small enough compared to n, our algorithm does not need to read all entries of $\mathbf{A}$. In Section 3.2, we apply this result to give a sublinear algorithm for approximating the full spectrum of a psd matrix.

We begin with the following calculation to determine the number of rejection sampling rounds required to obtain an accurate estimate of the acceptance probability, and hence $\|\mathbf{A}\|_F^2$.

Lemma 2. *Fix $\epsilon,\delta\in(0,1)$. Suppose we run r rounds of Algorithm 1. Let X be the number of accepted rounds and compute*

$$\widehat{F} := \frac{X}{r}\operatorname{tr}(\mathbf{A})^2.$$

If $r \geq 3n\log(2/\delta)/\epsilon^2$ then with probability at least $1-\delta$,

$$(1-\epsilon)\|\mathbf{A}\|_F^2 \leq \widehat{F} \leq (1+\epsilon)\|\mathbf{A}\|_F^2.$$

Proof. Let X_i be an indicator random variable where $X_i = 1$ if round i is accepted and $X_i = 0$ otherwise. Recall from the proof of Theorem 1 that the acceptance probability for any round is

$$\beta = \frac{\|\mathbf{A}\|_F^2}{\operatorname{tr}(\mathbf{A})^2}$$

so $\mathbb{E}X_i = \beta$. Since the rounds are independent, $X = \sum_{i=1}^{r} X_i$ is a sum of independent indicator random variables with $\mathbb{E}X = r\beta$. By the Chernoff bound,

$$\Pr\left[|X - r\beta| \geq \epsilon r\beta\right] \leq 2\exp\left(-\frac{\epsilon^2 r\beta}{3}\right).$$

From the proof of Theorem 1, we also have $\beta \geq 1/n$. Therefore, if $r \geq 3n\log(2/\delta)/\epsilon^2$, then

$$2\exp\left(-\frac{\epsilon^2 r\beta}{3}\right) \leq 2\exp\left(-\frac{\epsilon^2 r}{3n}\right) \leq \delta.$$

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

$$(1-\epsilon)\beta \leq \frac{X}{r} \leq (1+\epsilon)\beta.$$

Multiplying through by $\operatorname{tr}(\mathbf{A})^2$ gives the desired result. □

We are now ready to state the algorithm.

Algorithm 2: Frobenius Norm Approximation

Let d be the diagonal distribution with $d_i = \mathbf{A}_{ii}/\operatorname{tr}(\mathbf{A})$.
Set $r = \lceil 3n\log(200)/\epsilon^2 \rceil$. Run the following procedure for r independent rounds:

1. Sample indices $i, j \overset{\text{i.i.d.}}{\sim} d$.
2. Accept with probability $\frac{\mathbf{A}_{ij}^2}{\mathbf{A}_{ii}\mathbf{A}_{jj}}$.

Let X be the number of accepted rounds. Return

$$\widehat{F} := \frac{X}{r}\operatorname{tr}(\mathbf{A})^2.$$

Theorem 3. *Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be psd, and fix $\epsilon \in (0,1)$. With probability at least 0.99, the estimate $\widehat{F}$ returned by Algorithm 2 satisfies*

$$(1-\epsilon)\|\mathbf{A}\|_F^2 \le \widehat{F} \le (1+\epsilon)\|\mathbf{A}\|_F^2.$$

Moreover, the algorithm reads only $O(n/\epsilon^2)$ entries of $\mathbf{A}$.

Proof. The error guarantee on $\widehat{F}$ follows from Lemma 2 applied to $\delta = 0.01$. The query complexity is also immediate from the choice of r. □

3.2 Eigenvalue Approximation

We now give a sublinear algorithm for additive error eigenvalue approximation of psd matrices. The basic algorithmic template is due to Bhattacharjee et al. [BDD+22], who sample and rescale a small principal submatrix according to squared row norms, zero out certain entries to control variance, then return its eigenvalues together with additional zeros. Swartworth and Woodruff [SW24] improve on the analysis of this procedure, showing that

$$O\left(\frac{1}{\epsilon^2}\log^4 n \log^2 \frac{1}{\epsilon}\right)$$

squared row-norm samples suffice for additive $\epsilon\|\mathbf{A}\|_F$ eigenvalue approximation.

The algorithms of [BDD+22, SW24] apply more broadly to symmetric matrices, but they assume access to the exact row norms, as well as the Frobenius norm, of the input. These quantities are not available to us in the entry query model unless we read the entire input. Instead, we specialize to psd matrices, which allows us to use Algorithm 1 to sample rows by squared row norm, as well as Algorithm 2 to estimate the Frobenius norm. The comparison is summarized in Table 3.1.

Result	Setting and assumptions	Guarantee	Query complexity
[BDD+22]	Symmetric matrices. Assumes access to exact row norms.	Additive $\epsilon\|\mathbf{A}\|_F$ approximation to the spectrum.	$\widetilde{O}\left(\frac{n}{\epsilon^8}\right)$
[SW24]	Symmetric matrices. Assumes access to exact row norms.	Additive $\epsilon\|\mathbf{A}\|_F$ approximation to the spectrum.	$\widetilde{O}\left(\frac{n}{\epsilon^2}\right)$
This work	Psd matrices. Entry-wise access only.	Additive $\epsilon\|\mathbf{A}\|_F$ approximation to the spectrum.	$\widetilde{O}\left(\frac{n}{\epsilon^2}\right)$ in expectation

Table 3.1: Comparison of additive-error eigenvalue approximation guarantees. Here, $\widetilde{O}$ hides polylogarithmic factors in n and $1/\epsilon$.

We first recall the approximation guarantee that we seek.

Definition 1 (Additive eigenvalue approximation)**.** Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be symmetric. We say that a sequence $\widehat{\lambda}_1 \geq \widehat{\lambda}_2 \geq \cdots \geq \widehat{\lambda}_n$ is an additive ϵ-approximation to the spectrum of $\mathbf{A}$ if

$$|\widehat{\lambda}_i - \lambda_i(\mathbf{A})| \leq \epsilon \qquad \text{for every } i \in [n].$$

The sampling theorem below applies to a modified version of $\mathbf{A}$ in which the diagonal is set to zero and certain off-diagonal entries are also set to zero when their row norms are small relative to their magnitude. Following [SW24], we make this operation explicit in the following definition. We introduce a new parameter F_0 in order to accommodate for the $(1\pm\epsilon)$ estimate that we use in place of $\|\mathbf{A}\|_F^2$.

Definition 2. Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be symmetric, let $\epsilon \in (0,1)$, and let $F_0 > 0$. We define

$$\mathsf{Z}_{\epsilon,F_0}(\mathbf{A}) \in \mathbb{R}^{n\times n}$$

entrywise by

$$\big(\mathsf{Z}_{\epsilon,F_0}(\mathbf{A})\big)_{ij} = \begin{cases} 0, & i = j, \\ 0, & i \neq j \text{ and } \|\mathbf{A}_{i,*}\|_2^2 \|\mathbf{A}_{j,*}\|_2^2 \leq \dfrac{\epsilon^2 F_0 |\mathbf{A}_{ij}|^2}{c \log^4 n}, \\ \mathbf{A}_{ij}, & \text{otherwise}, \end{cases}$$

where $c > 0$ is a fixed absolute constant. When $F_0 = \|\mathbf{A}\|_F^2$, we abbreviate

$$\mathsf{Z}_\epsilon(\mathbf{A}) := \mathsf{Z}_{\epsilon,\|\mathbf{A}\|_F^2}(\mathbf{A}).$$

We are now ready to state the following theorem of Swartworth and Woodruff:

Theorem 4 ([SW24], Theorem 7.8)**.** *Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be a symmetric matrix, and let $\epsilon \in (0,1)$. Let q be the squared row norm distribution on $\mathbf{A}$, where $q_i = \|\mathbf{A}_{i,*}\|_2^2 / \|\mathbf{A}\|_F^2$. Suppose $i_1, \ldots, i_{m_0}$ are independent samples from q, where*

$$m_0 \geq \frac{c}{\epsilon^2} \log^4 n \log^2 \frac{1}{\epsilon}$$

and c is the same absolute constant as in Definition 2. Define the sampling matrix $\mathbf{S} \in \mathbb{R}^{m\times n}$ by

$$\mathbf{S}_{r,*} = \frac{\mathbf{e}_{i_r}^\top}{\sqrt{m_0 q_{i_r}}}.$$

Then, with probability at least $5/6$, the eigenvalues of $\mathbf{S}\mathsf{Z}_\epsilon(\mathbf{A})\mathbf{S}^\top$, together with $n - m_0$ additional zeros, form an additive $\epsilon\|\mathbf{A}\|_F$-approximation to the spectrum of $\mathbf{A}$.

The preceding theorem assumes access to quantities that are not directly available in the entry-query model. First, the zeroing rule defining $\mathsf{Z}_\epsilon(\mathbf{A})$ requires access to $\|\mathbf{A}\|_F^2$. We replace this quantity by an estimate $\widehat{F}$ given by Algorithm 2. After sampling an index i_r, we query the full row $\mathbf{A}_{i_r,*}$ and compute $\|\mathbf{A}_{i_r,*}\|_2^2$. In place of the exact sampling probability $q_{i_r} = \|\mathbf{A}_{i_r,*}\|_2^2 / \|\mathbf{A}\|_F^2$, we use the estimated probability $\widehat{q}_{i_r} = \|\mathbf{A}_{i_r,*}\|_2^2 / \widehat{F}$. The algorithm below is the resulting implementable version of Theorem 4.

Algorithm 3: Sublinear PSD Eigenvalue Approximation

Input: nonzero psd matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, accuracy parameter $\epsilon \in (0,1)$.
Let $C > 0$ be a sufficiently large absolute constant, and set

$$m = \left\lceil \frac{C}{\epsilon^2} \log^4 n \log^2 \frac{1}{\epsilon} \right\rceil.$$

Run the following procedure:

1. Run Algorithm 2 with accuracy parameter $\epsilon/4$, producing an estimate $\widehat{F}$ for $\|\mathbf{A}\|_F^2$.
2. Run Algorithm 1 m times on $\mathbf{A}$, producing $i_1, \ldots, i_m$ sampled by squared row norms.
3. Query the full rows $\mathbf{A}_{i_1,*}, \ldots, \mathbf{A}_{i_m,*}$ and, for each $r \in [m]$, set

$$\widehat{q}_{i_r} := \frac{\|\mathbf{A}_{i_r,*}\|_2^2}{\widehat{F}}.$$

   Define the sampling matrix $\widehat{\mathbf{S}} \in \mathbb{R}^{m\times n}$ by

$$\widehat{\mathbf{S}}_{r,*} := \frac{\mathbf{e}_{i_r}^\top}{\sqrt{m\widehat{q}_{i_r}}} \qquad \text{for } r = 1, \ldots, m.$$

4. Compute the eigenvalues $\widehat{\lambda}_1 \geq \widehat{\lambda}_2 \geq \cdots \geq \widehat{\lambda}_m$ of $\widehat{\mathbf{S}}\mathsf{Z}_{\epsilon/4,\widehat{F}}(\mathbf{A})\widehat{\mathbf{S}}^\top$.
5. Return $\widehat{\lambda}_1 \geq \widehat{\lambda}_2 \geq \cdots \geq \widehat{\lambda}_m$ together with $n - m$ additional zeros.

Theorem 5. *Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be a psd matrix, and let $\epsilon \in (0,1)$. There is an absolute constant $C > 0$ such that Algorithm 3, run with*

$$m = \left\lceil \frac{C}{\epsilon^2} \log^4 n \log^2 \frac{1}{\epsilon} \right\rceil,$$

returns an $\epsilon\|\mathbf{A}\|_F$ additive approximation to the spectrum of $\mathbf{A}$ with probability at least $2/3$. The algorithm queries

$$O\left(\frac{n}{\epsilon^2} \log^4 n \log^2 \frac{1}{\epsilon}\right)$$

entries of $\mathbf{A}$ in expectation, and uses

$$O\left(\frac{n}{\epsilon^2} \log^4 n \log^2 \frac{1}{\epsilon} + \frac{1}{\epsilon^6} \log^{12} n \log^6 \frac{1}{\epsilon}\right)$$

arithmetic operations in expectation.

Proof. If $m \geq n$, we may query all entries of $\mathbf{A}$ and compute its eigenvalues exactly. This satisfies the desired error guarantee. Moreover, in this case $n^2 \leq nm$ and $n^3 \leq m^3$, so the query and arithmetic bounds are also consistent with the claimed bounds. Thus, for the remainder of the proof, assume that $m < n$.

We first prove correctness. Condition on $\widehat{F}$ produced by Step 1 of the algorithm and the indices $i_1, \ldots, i_m$ sampled in Step 2. Let $\mathbf{S}$ be the sampling matrix from Theorem 4 formed using the same sampled indices and the exact sampling probabilities $q_i = \|\mathbf{A}_{i,*}\|_2^2 / \|\mathbf{A}\|_F^2$, where

$$\mathbf{S}_{r,*} = \frac{\mathbf{e}_{i_r}^\top}{\sqrt{m q_{i_r}}}.$$

Let $\alpha := \frac{\epsilon}{4}\sqrt{\frac{\widehat{F}}{\|\mathbf{A}\|_F^2}}$. We make the following connection between $\widehat{\mathbf{S}}\mathsf{Z}_{\epsilon/4,\widehat{F}}(\mathbf{A})\widehat{\mathbf{S}}^\top$ and $\mathbf{S}\mathsf{Z}_\alpha(\mathbf{A})\mathbf{S}^\top$. First,

$$\widehat{\mathbf{S}}_{r,*} = \frac{\mathbf{e}_{i_r}^\top}{\sqrt{m\widehat{q}_{i_r}}} = \sqrt{\frac{\widehat{F}}{\|\mathbf{A}\|_F^2}}\,\frac{\mathbf{e}_{i_r}^\top}{\sqrt{m q_{i_r}}} = \sqrt{\frac{\widehat{F}}{\|\mathbf{A}\|_F^2}}\,\mathbf{S}_{r,*},$$

and hence $\widehat{\mathbf{S}} = \sqrt{\widehat{F}/\|\mathbf{A}\|_F^2}\,\mathbf{S}$. We also have $\mathsf{Z}_{\epsilon/4,\widehat{F}}(\mathbf{A}) = \mathsf{Z}_\alpha(\mathbf{A})$ because

$$\frac{\epsilon^2}{16}\widehat{F} = \left(\frac{\epsilon}{4}\sqrt{\frac{\widehat{F}}{\|\mathbf{A}\|_F^2}}\right)^2 \|\mathbf{A}\|_F^2,$$

so the thresholds in Definition 2 agree. Consequently,

$$\widehat{\mathbf{S}}\mathsf{Z}_{\epsilon/4,\widehat{F}}(\mathbf{A})\widehat{\mathbf{S}}^\top = \frac{\widehat{F}}{\|\mathbf{A}\|_F^2}\,\mathbf{S}\mathsf{Z}_\alpha(\mathbf{A})\mathbf{S}^\top.$$

Now, condition on the event that Step 1 satisfies

$$\left(1 - \frac{\epsilon}{4}\right)\|\mathbf{A}\|_F^2 \leq \widehat{F} \leq \left(1 + \frac{\epsilon}{4}\right)\|\mathbf{A}\|_F^2.$$

In this case, we readily check that

$$\frac{\epsilon}{8} \leq \alpha = \frac{\epsilon}{4}\sqrt{\frac{\widehat{F}}{\|\mathbf{A}\|_F^2}} \leq \frac{\epsilon}{2}.$$

In particular, we may pick the constant C sufficiently large compared to c of Theorem 4 in order to apply the theorem with m sampled indices and accuracy parameter α. It follows that, with

probability at least 5/6, if $\nu_1 \geq \cdots \geq \nu_m$ are the eigenvalues of $\mathbf{S}\mathsf{Z}_\alpha(\mathbf{A})\mathbf{S}^\top$, together with $n-m$ additional zeros, then

$$|\nu_i - \lambda_i(\mathbf{A})| \leq \frac{\epsilon}{2}\|\mathbf{A}\|_F \qquad \text{for every } i \in [n].$$

Finally, let $\widehat{\lambda}_1 \geq \cdots \geq \widehat{\lambda}_m$ be the eigenvalues of $\widehat{\mathbf{S}}\mathsf{Z}_{\epsilon/4,\widehat{F}}(\mathbf{A})\widehat{\mathbf{S}}^\top$, again together with $n-m$ additional zeros. Since $\widehat{\mathbf{S}}\mathsf{Z}_{\epsilon/4,\widehat{F}}(\mathbf{A})\widehat{\mathbf{S}}^\top = (\widehat{F}/\|\mathbf{A}\|_F^2)\cdot\mathbf{S}\mathsf{Z}_\alpha(\mathbf{A})\mathbf{S}^\top$ and $\widehat{F} > 0$, we have $\widehat{\lambda}_i = (\widehat{F}/\|\mathbf{A}\|_F^2)\nu_i$ for all $i \in [n]$. It follows that

$$\begin{aligned}
|\widehat{\lambda}_i - \lambda_i(\mathbf{A})| &= \left|\frac{\widehat{F}}{\|\mathbf{A}\|_F^2}\nu_i - \lambda_i(\mathbf{A})\right| \\
&\leq \frac{\widehat{F}}{\|\mathbf{A}\|_F^2}|\nu_i - \lambda_i(\mathbf{A})| + \left|\frac{\widehat{F}}{\|\mathbf{A}\|_F^2} - 1\right||\lambda_i(\mathbf{A})| \\
&\leq \left(1+\frac{\epsilon}{4}\right)\frac{\epsilon}{2}\|\mathbf{A}\|_F + \frac{\epsilon}{4}\|\mathbf{A}\|_F \\
&\leq \epsilon\|\mathbf{A}\|_F.
\end{aligned}$$

Here we used $|\lambda_i(\mathbf{A})| \leq \|\mathbf{A}\|_F$ and $\epsilon \in (0,1)$. Thus the returned sequence is an $\epsilon\|\mathbf{A}\|_F$ additive approximation to the spectrum of $\mathbf{A}$.

Taking a union bound over the failure of the Frobenius norm estimate and the failure of Theorem 4, the total success probability is at least

$$1 - \frac{1}{100} - \frac{1}{6} > \frac{2}{3}.$$

It remains to bound the query and arithmetic complexity. Step 1 runs Algorithm 2 with accuracy parameter $\epsilon/4$. By Theorem 3, this costs $O\left(n/\epsilon^2\right)$ entry queries and arithmetic operations, which is dominated by the final bound.

By Theorem 1, each call to Algorithm 1 uses $O(n)$ entry queries and $O(n)$ arithmetic operations in expectation. Since Step 2 makes

$$m = O\left(\frac{1}{\epsilon^2}\log^4 n \log^2\frac{1}{\epsilon}\right)$$

independent calls, Step 2 uses

$$O(nm) = O\left(\frac{n}{\epsilon^2}\log^4 n \log^2\frac{1}{\epsilon}\right)$$

entry queries and arithmetic operations in expectation.

Step 3 queries m full rows of $\mathbf{A}$, for a total of nm entry queries. Computing the row norms also costs $O(nm)$ arithmetic operations.

In Step 4, we form the $m \times m$ matrix $\widehat{\mathbf{S}}\mathrm{Z}_{\epsilon/4,\widehat{F}}(\mathbf{A})\widehat{\mathbf{S}}^\top$. No additional entry queries are needed since every entry $\mathbf{A}_{i_r i_t}$ is contained in the sampled row $\mathbf{A}_{i_r,*}$. Each entry can be computed with constant arithmetic from $\widehat{F}$, $\widehat{q}_{i_r}$, $\widehat{q}_{i_t}$, and $\mathbf{A}_{i_r i_t}$, so forming this matrix costs $O(m^2)$ arithmetic operations. Since $m < n$, this is bounded by $O(nm)$. Computing its eigenvalues costs $O(m^3)$ arithmetic operations. Step 5 adds $n - m$ zeros to the returned sequence, costing at most $O(n)$ additional arithmetic operations, which is dominated by the preceding terms.

Therefore the total expected query complexity is

$$O(nm) = O\left(\frac{n}{\epsilon^2}\log^4 n \log^2 \frac{1}{\epsilon}\right).$$

The total expected arithmetic complexity is $O(nm + m^3)$. Substituting the value of m, we obtain

$$O\left(\frac{n}{\epsilon^2}\log^4 n \log^2 \frac{1}{\epsilon} + \frac{1}{\epsilon^6}\log^{12} n \log^6 \frac{1}{\epsilon}\right).$$

This completes the proof. □

3.3 Additive Error Low-Rank Approximation

In this section, we derive a simple algorithm for computing additive error low-rank approximations to a psd matrix using our fast length-squared sampling procedure. Applying a result of Drineas, Kannan, and Mahoney [DKM06b], we compute an approximate basis for the top k-dimensional left singular subspace of $\mathbf{A}$ from $O(k/\epsilon^2)$ columns sampled by squared column norms. Then, we use a well-known leverage score sampling guarantee to approximately project $\mathbf{A}$ onto this subspace. Given entry-wise access to $\mathbf{A} \in \mathbb{R}^{n\times n}$, our algorithm outputs matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n\times k}$ satisfying

$$\|\mathbf{A} - \mathbf{X}\mathbf{Y}^\top\|_F^2 \leq \|\mathbf{A} - \mathbf{A}_k\|_F^2 + \epsilon\|\mathbf{A}\|_F^2$$

with constant probability, using $O(nk/\epsilon^2)$ entries of $\mathbf{A}$ and $O(nk^2/\epsilon^4)$ arithmetic in expectation.

Our sample complexity incurs an additional $1/\epsilon$ factor in comparison to the best known algorithm for additive error low-rank approximation: Bakshi, Chepurko, and Woodruff [BCW20, Corollary 5.19] achieve an additive error low-rank approximation using only $\tilde{O}(nk/\epsilon)$ entries, where $\tilde{O}$ hides polylogarithmic factors in n, k, and ϵ. Their work also achieves relative error approximation using $\tilde{O}(nk/\epsilon)$ entries, matching a $\Omega(nk/\epsilon)$ query lower bound of Musco and Woodruff [MW19] up to polylogarithmic factors. The relative error guarantee from $\tilde{O}(nk/\epsilon)$ entries is a strictly stronger result from a theoretical perspective. Nonetheless, the algorithm presented here admits a simple analysis. We pose as an open question whether it is enough to sample only $O(k/\epsilon)$ columns by squared column norms, thus leading to an $O(nk/\epsilon)$ algorithm. The comparison is summarized in Table 3.2.

Result	Assumptions	Guarantee	Query complexity	Arithmetic complexity
[BCW20]	Psd matrix with entry-wise access.	$\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n\times k}$ such that $\|\mathbf{A}-\mathbf{X}\mathbf{Y}^\top\|_F^2 \le \|\mathbf{A}-\mathbf{A}_k\|_F^2 + \epsilon\|\mathbf{A}\|_F^2$.	$\widetilde{O}\left(\frac{nk}{\epsilon}\right)$	$\widetilde{O}\left(n\left(\frac{k}{\epsilon}\right)^{\omega-1}\right)$
This work	Psd matrix with entry-wise access.	$\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n\times k}$ such that $\|\mathbf{A}-\mathbf{X}\mathbf{Y}^\top\|_F^2 \le \|\mathbf{A}-\mathbf{A}_k\|_F^2 + \epsilon\|\mathbf{A}\|_F^2$.	$O\left(\frac{nk}{\epsilon^2}\right)$ in expectation	$O\left(\frac{nk^2}{\epsilon^4}\right)$ in expectation

Table 3.2: Comparison of additive-error low-rank approximation guarantees for positive semidefinite matrices. Here ω is the matrix multiplication exponent, and $\widetilde{O}$ hides polylogarithmic factors in n, k, and $1/\epsilon$.

We begin with the following theorem due to Drineas, Kannan, and Mahoney [DKM06b], which shows that one can compute a good approximation to the top k-dimensional left singular subspace of $\mathbf{A}$ from $O(k/\epsilon^2)$ columns sampled by squared column norms.

Lemma 6 ([DKM06b], Theorem 4)**.** *Let $\mathbf{A} \in \mathbb{R}^{m\times n}$ be a nonzero matrix and $1 \le k \le c \le n$. Let $p_1, \ldots, p_n$ satisfy*

$$p_i = \frac{\|\mathbf{A}_{*,i}\|_2^2}{\|\mathbf{A}\|_F^2}.$$

Now, suppose $i_1, i_2, \ldots, i_c$ are c independent samples from the squared column-norm distribution on $\mathbf{A}$, and let $\mathbf{C} \in \mathbb{R}^{m \times c}$ be the matrix whose columns are given by

$$\mathbf{C}_{*,j} := \frac{1}{\sqrt{c p_{i_j}}} \mathbf{A}_{*,i_j}.$$

Finally, let $\mathbf{H} \in \mathbb{R}^{m \times k}$ be an orthonormal basis for the subspace spanned by the top-k left singular vectors of $\mathbf{C}$. We have the following:

(a) Fix $\delta \in (0,1)$ and let $\eta = 1 + \sqrt{8 \log(1/\delta)}$. If $c \geq 4k\eta^2/\epsilon^2$, then with probability at least $1-\delta$,

$$\|\mathbf{A} - \mathbf{H}\mathbf{H}^\top \mathbf{A}\|_F^2 \leq \|\mathbf{A} - \mathbf{A}_k\|_F^2 + \epsilon \|\mathbf{A}\|_F^2.$$

(b) If $c \geq 4k/\epsilon^2$, then

$$\mathbb{E}\|\mathbf{A} - \mathbf{H}\mathbf{H}^\top \mathbf{A}\|_F^2 \leq \|\mathbf{A} - \mathbf{A}_k\|_F^2 + \epsilon \|\mathbf{A}\|_F^2.$$

(c) If $c \geq 4/\epsilon^2$, then

$$\mathbb{E}\|\mathbf{A} - \mathbf{H}\mathbf{H}^\top \mathbf{A}\|_2^2 \leq \|\mathbf{A} - \mathbf{A}_k\|_2^2 + \epsilon \|\mathbf{A}\|_F^2.$$

With constant probability, Lemma 6 produces a matrix $\mathbf{H}$ with orthonormal columns such that $\mathbf{H}\mathbf{H}^\top \mathbf{A}$, the projection of $\mathbf{A}$ onto the subspace spanned by $\mathbf{H}$, yields an additive error approximation to $\mathbf{A}$. To produce an explicit factorization, we would like to set $\mathbf{X} = \mathbf{H}$ and $\mathbf{Y}^\top = \mathbf{H}^\top \mathbf{A}$ so that $\mathbf{X}\mathbf{Y}^\top$ yields an additive error approximation. However, computing $\mathbf{H}^\top \mathbf{A}$ requires us to read all of $\mathbf{A}$, so we instead apply a standard matrix multiplication result to approximate $\mathbf{H}^\top \mathbf{A}$. We state and prove the needed guarantee below for completeness. A closely related result appears in Lemma 8 of [DKM06a].

Lemma 7. *Let $\mathbf{A} \in \mathbb{R}^{n \times n}$, and let $\mathbf{H} \in \mathbb{R}^{n \times k}$ have orthonormal columns. Let q be the leverage score distribution on $\mathbf{H}$, where*

$$q_i = \frac{\ell(i)}{k}, \qquad \ell(i) = \|\mathbf{H}_{i,*}\|_2^2.$$

Suppose $i_1, \ldots, i_t$ are independent samples from q, and construct the matrix $\widehat{\mathbf{Z}} \in \mathbb{R}^{k \times n}$ given by

$$\widehat{\mathbf{Z}} := \frac{1}{t} \sum_{r=1}^{t} \frac{\mathbf{H}_{i_r,*}^\top \mathbf{A}_{i_r,*}}{q_{i_r}}.$$

Then for every $\delta \in (0,1)$, with probability at least $1-\delta$,

$$\|\mathbf{A} - \mathbf{H}\widehat{\mathbf{Z}}\|_F^2 \le \|\mathbf{A} - \mathbf{H}\mathbf{H}^\top \mathbf{A}\|_F^2 + \frac{k}{\delta t}\|\mathbf{A}\|_F^2.$$

Proof. For a sample $i \sim q$, define the random matrix

$$\mathbf{R} = \frac{\mathbf{H}_{i,*}^\top \mathbf{A}_{i,*}}{q_i}.$$

Then

$$\mathbb{E}[\mathbf{R}] = \sum_{i:\ell_i>0} q_i \frac{\mathbf{H}_{i,*}^\top \mathbf{A}_{i,*}}{q_i} = \mathbf{H}^\top \mathbf{A}.$$

Let $\mathbf{R}_1, \ldots, \mathbf{R}_t$ be the independent copies of $\mathbf{R}$ used to form $\widehat{\mathbf{Z}}$. Since

$$\widehat{\mathbf{Z}} - \mathbf{H}^\top \mathbf{A} = \frac{1}{t}\sum_{r=1}^{t} \left(\mathbf{R}_r - \mathbf{H}^\top \mathbf{A}\right),$$

we have

$$\begin{aligned}
\mathbb{E}\|\widehat{\mathbf{Z}} - \mathbf{H}^\top \mathbf{A}\|_F^2 &= \mathbb{E}\left\| \frac{1}{t}\sum_{r=1}^{t} \left(\mathbf{R}_r - \mathbf{H}^\top \mathbf{A}\right)\right\|_F^2 \\
&= \frac{1}{t^2}\sum_{r=1}^{t}\sum_{s=1}^{t} \mathbb{E}\,\mathrm{tr}\left[\left(\mathbf{R}_r - \mathbf{H}^\top \mathbf{A}\right)^\top \left(\mathbf{R}_s - \mathbf{H}^\top \mathbf{A}\right)\right].
\end{aligned}$$

For $r \neq s$, independence and $\mathbb{E}[\mathbf{R}_r - \mathbf{H}^\top \mathbf{A}] = 0$ imply $\mathbb{E}\,\mathrm{tr}\left[\left(\mathbf{R}_r - \mathbf{H}^\top \mathbf{A}\right)^\top \left(\mathbf{R}_s - \mathbf{H}^\top \mathbf{A}\right)\right] = 0$ and so

$$\begin{aligned}
\mathbb{E}\|\widehat{\mathbf{Z}} - \mathbf{H}^\top \mathbf{A}\|_F^2 &= \frac{1}{t^2}\sum_{r=1}^{t} \mathbb{E}\|\mathbf{R}_r - \mathbf{H}^\top \mathbf{A}\|_F^2 \\
&= \frac{1}{t}\mathbb{E}\|\mathbf{R} - \mathbf{H}^\top \mathbf{A}\|_F^2 \\
&\le \frac{1}{t}\mathbb{E}\|\mathbf{R}\|_F^2,
\end{aligned}$$

Moreover,

$$\begin{aligned}
\mathbb{E}\|\mathbf{R}\|_F^2 &= \sum_{i:\ell_i>0} q_i \left\| \frac{\mathbf{H}_{i,*}^\top \mathbf{A}_{i,*}}{q_i}\right\|_F^2 \\
&= \sum_{i:\ell_i>0} \frac{\|\mathbf{H}_{i,*}^\top \mathbf{A}_{i,*}\|_F^2}{q_i} \\
&= \sum_{i:\ell_i>0} \frac{\|\mathbf{H}_{i,*}\|_2^2 \|\mathbf{A}_{i,*}\|_2^2}{q_i} \\
&= \sum_{i:\ell_i>0} k\|\mathbf{A}_{i,*}\|_2^2 \qquad \text{using } q_i = \ell(i)/k \\
&\le k\|\mathbf{A}\|_F^2.
\end{aligned}$$

Thus,

$$\mathbb{E}\|\widehat{\mathbf{Z}} - \mathbf{H}^\top \mathbf{A}\|_F^2 \le \frac{k}{t}\|\mathbf{A}\|_F^2.$$

By Markov's inequality,

$$\|\widehat{\mathbf{Z}} - \mathbf{H}^\top \mathbf{A}\|_F^2 \le \frac{k}{\delta t}\|\mathbf{A}\|_F^2 \qquad \text{with probability at least } 1-\delta. \tag{$*$}$$

Finally, let $\mathbf{P} = \mathbf{H}\mathbf{H}^\top$. Since $\mathbf{P}$ is the orthogonal projector onto range($\mathbf{H}$), we have the orthogonal decomposition

$$\mathbf{A} - \mathbf{H}\widehat{\mathbf{Z}} = (\mathbf{I}-\mathbf{P})\mathbf{A} + \mathbf{H}(\mathbf{H}^\top \mathbf{A} - \widehat{\mathbf{Z}}).$$

Thus,

$$\|\mathbf{A} - \mathbf{H}\widehat{\mathbf{Z}}\|_F^2 = \|\mathbf{A} - \mathbf{H}\mathbf{H}^\top \mathbf{A}\|_F^2 + \|\mathbf{H}(\widehat{\mathbf{Z}} - \mathbf{H}^\top \mathbf{A})\|_F^2.$$

Since $\mathbf{H}$ has orthonormal columns, $\|\mathbf{H}(\widehat{\mathbf{Z}} - \mathbf{H}^\top \mathbf{A})\|_F = \|\widehat{\mathbf{Z}} - \mathbf{H}^\top \mathbf{A}\|_F$. Recalling $(*)$ completes the proof. □

We are now ready to state the main algorithm of this section.

Algorithm 4: Sublinear PSD Low-Rank Approximation

Input: psd matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, target rank $k \in [n]$, accuracy parameter $\epsilon \in (0,1)$.
Let $s = \lceil Ck/\epsilon^2 \rceil$ and $t = \lceil C'k/\epsilon \rceil$ for some absolute constants C, C'.
Run the following procedure:

1. Run Algorithm 1 s times on $\mathbf{A}$, producing $i_1, \ldots, i_s$ sampled by squared column norms.
2. Read the full columns $\mathbf{A}_{*,i_1}, \ldots, \mathbf{A}_{*,i_s}$ and form the matrix $\mathbf{B} \in \mathbb{R}^{n\times s}$ whose r-th column is

$$\mathbf{B}_{*,r} = \frac{\mathbf{A}_{*,i_r}}{\|\mathbf{A}_{*,i_r}\|_2}.$$

3. Let $\mathbf{H} \in \mathbb{R}^{n\times k}$ have orthonormal columns spanning a top-k left singular subspace of $\mathbf{B}$.
4. Let q be the leverage score distribution on $\mathbf{H}$ with $q_i = \frac{\|\mathbf{H}_{i,*}\|_2^2}{k}$ Sample indices $j_1, \ldots, j_t \overset{\text{i.i.d.}}{\sim} q$.
5. Query the full rows $\mathbf{A}_{j_1,*}, \ldots, \mathbf{A}_{j_t,*}$. and set

$$\widehat{\mathbf{Y}} = \frac{1}{t}\sum_{r=1}^{t} \frac{\mathbf{A}_{j_r,*}^\top \mathbf{H}_{j_r,*}}{q_{j_r}} \in \mathbb{R}^{n\times k}.$$

6. Return $\mathbf{X} = \mathbf{H}, \mathbf{Y} = \widehat{\mathbf{Y}}$.

Theorem 8. *Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be a nonzero psd matrix, let $1 \le k \le n$, and let $\epsilon \in (0,1)$. There are absolute constants $C, C' > 0$ such that Algorithm 4, run with*

$$s = \left\lceil \frac{Ck}{\epsilon^2} \right\rceil, \qquad t = \left\lceil \frac{C'k}{\epsilon} \right\rceil,$$

returns matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n\times k}$ satisfying

$$\|\mathbf{A} - \mathbf{X}\mathbf{Y}^\top\|_F^2 \le \|\mathbf{A} - \mathbf{A}_k\|_F^2 + \epsilon\|\mathbf{A}\|_F^2$$

with probability at least $2/3$. The algorithm queries $O(nk/\epsilon^2)$ entries of $\mathbf{A}$ and uses $O(nk^2/\epsilon^4)$ arithmetic operations in expectation.

Proof. Let $\eta = 1 + \sqrt{8\log 6}$ and fix constants

$$C \ge 16\eta^2, \qquad C' \ge 12.$$

We first prove the approximation guarantee.

Let $i_1, \ldots, i_s$ be the column indices sampled in Step 1 of Algorithm 4. Let $\widetilde{\mathbf{C}}$ denote the matrix of sampled columns appearing in Lemma 6. We have

$$\begin{aligned} \widetilde{\mathbf{C}}_{*,r} &= \frac{1}{\sqrt{s p_{i_r}}} \mathbf{A}_{*,i_r} \\ &= \frac{\|\mathbf{A}\|_F}{\sqrt{s}} \frac{\mathbf{A}_{*,i_r}}{\|\mathbf{A}_{*,i_r}\|_2} \qquad \text{since } p_i = \|\mathbf{A}_{*,i}\|_2^2 / \|\mathbf{A}\|_F^2 \\ &= \frac{\|\mathbf{A}\|_F}{\sqrt{s}} \mathbf{B}_{*,r}. \end{aligned}$$

Thus $\widetilde{\mathbf{C}} = \frac{\|\mathbf{A}\|_F}{\sqrt{s}}\mathbf{B}$, is a scalar multiple of $\mathbf{B}$, so $\widetilde{\mathbf{C}}$ and $\mathbf{B}$ have the same top k left singular subspaces. In particular, the matrix $\mathbf{H}$ computed in Step 3 is also a valid choice of $\mathbf{H}$ for Lemma 6.

Apply Lemma 6(a) with accuracy parameter $\epsilon/2$ and failure probability $\delta = 1/6$. Since

$$s \ge \frac{16k\eta^2}{\epsilon^2} = \frac{4k\eta^2}{(\epsilon/2)^2},$$

we have, with probability at least $5/6$,

$$\|\mathbf{A} - \mathbf{H}\mathbf{H}^\top\mathbf{A}\|_F^2 \le \|\mathbf{A} - \mathbf{A}_k\|_F^2 + \frac{\epsilon}{2}\|\mathbf{A}\|_F^2. \tag{1}$$

Now condition on $\mathbf{H}$ produced by the algorithm. Let $\widehat{\mathbf{Z}} = \widehat{\mathbf{Y}}^\top$. Applying Lemma 7 with $\delta = 1/6$, we obtain, with probability at least $5/6$,

$$\begin{aligned}\|\mathbf{A} - \mathbf{H}\widehat{\mathbf{Z}}\|_F^2 &\le \|\mathbf{A} - \mathbf{H}\mathbf{H}^\top\mathbf{A}\|_F^2 + \frac{6k}{t}\|\mathbf{A}\|_F^2 \\ &\le \|\mathbf{A} - \mathbf{H}\mathbf{H}^\top\mathbf{A}\|_F^2 + \frac{\epsilon}{2}\|\mathbf{A}\|_F^2. \qquad \text{since } t \ge 12k/\epsilon. \end{aligned} \tag{2}$$

By a union bound, (1) and (2) occur simultaneously with probability at least $2/3$, in which case we have

$$\begin{aligned}\|\mathbf{A} - \mathbf{X}\mathbf{Y}^\top\|_F^2 &= \|\mathbf{A} - \mathbf{H}\widehat{\mathbf{Y}}^\top\|_F^2 \\ &= \|\mathbf{A} - \mathbf{H}\widehat{\mathbf{Z}}\|_F^2 \\ &\le \|\mathbf{A} - \mathbf{H}\mathbf{H}^\top\mathbf{A}\|_F^2 + \frac{\epsilon}{2}\|\mathbf{A}\|_F^2 \\ &\le \|\mathbf{A} - \mathbf{A}_k\|_F^2 + \epsilon\|\mathbf{A}\|_F^2.\end{aligned}$$

This proves the claimed additive error guarantee.

It remains to bound the query complexity and time complexity. The s calls to Algorithm 1 use $O(ns)$ entry queries in expectation. Step 2 reads s full columns of $\mathbf{A}$, using ns further entry queries. Step 5 reads t full rows of $\mathbf{A}$, using nt entry queries. Therefore the total expected number of entry queries to $\mathbf{A}$ is

$$O(ns + nt) = O\left(n\frac{k}{\epsilon^2} + n\frac{k}{\epsilon}\right) = O\left(\frac{nk}{\epsilon^2}\right),$$

since $\epsilon \in (0,1)$.

For the arithmetic bound, forming and rescaling $\mathbf{B} \in \mathbb{R}^{n\times s}$ costs $O(ns)$ operations after the sampled columns have been read. To compute a top k left singular subspace for $\mathbf{B}$ we compute $\mathbf{B}^\top\mathbf{B} \in \mathbb{R}^{s\times s}$ and its svd, which requires $O(ns^2 + s^3)$ arithmetic. For $s \le n$, this is $O(ns^2)$, hence $O\left(nk^2/\epsilon^4\right)$. Computing all leverage scores of $\mathbf{H}$ costs $O(nk)$. Finally, forming $\widehat{\mathbf{Y}}$ from the t sampled rows costs $O(ntk)$ arithmetic operations, since each sampled row contributes an $n \times k$ outer product. This is $O\left(nk^2/\epsilon\right)$, which is dominated by $O(nk^2/\epsilon^4)$ for $\epsilon \in (0,1)$. The expected arithmetic cost of the sampling procedure contributes $O(ns)$, which is also dominated. Thus the total expected arithmetic complexity is $O\left(nk^2/\epsilon^4\right)$. □

3.4 Robust Low-Rank Approximation

In the previous section, we used Algorithm 1 to obtain additive-error low-rank approximations to psd matrices in sublinear time. In this section, we show that the same approach is robust to a natural class of entry-wise perturbations. Our starting point is the robust low-rank approximation model of Bakshi, Chepurko, and Woodruff [BCW20]. In this model, there is an underlying psd matrix $\mathbf{A}$, but we are only given entry-wise access to a corrupted matrix

$$\mathbf{B} = \mathbf{A} + \mathbf{N},$$

where $\mathbf{N}$ is an arbitrary noise matrix with bounded Frobenius norm. Our goal is to compute a low-rank approximation to $\mathbf{A}$, despite only querying entries of $\mathbf{B}$.

Bakshi, Chepurko, and Woodruff [BCW20] consider this problem under three assumptions on the corruption. First, they assume that the noise is bounded by $\|\mathbf{A}\|_F^2$ in the sense that

$$\|\mathbf{N}\|_F^2 \leq \eta \|\mathbf{A}\|_F^2. \tag{A1}$$

Second, they assume that the noise is well-spread across the rows, in the sense that

$$\|\mathbf{N}_{i,*}\|_2^2 \leq c\|\mathbf{A}_{i,*}\|_2^2 \qquad \text{for every } i \in [n], \tag{A2}$$

where $c > 0$ is an absolute constant. Third, they assume access to the diagonal corruption parameter

$$\phi_{\max} := \max_{i\in[n]} \frac{\mathbf{A}_{ii}}{|\mathbf{B}_{ii}|}. \tag{A3}$$

Their algorithm returns rank-k approximation factors $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n\times k}$ satisfying

$$\|\mathbf{A} - \mathbf{X}\mathbf{Y}^\top\|_F^2 \leq \|\mathbf{A} - \mathbf{A}_k\|_F^2 + \left(\epsilon + O(\sqrt{\eta})\right)\|\mathbf{A}\|_F^2$$

with constant probability using $\widetilde{O}\left(\phi_{\max}^2 nk/\epsilon\right)$ entries of $\mathbf{B}$, where $\widetilde{O}$ hides polylogarithmic factors in n, k, and $1/\epsilon$.

The main theorem of this section obtains the same form of additive-error guarantee under assumptions $(A1)$ and $(A3)$, but it does not need the row-wise noise assumption $(A2)$. Our

approach is to modify our sampling algorithm to obtain length-squared samples from $\mathbf{B}$ at the cost of $O(\phi_{\max}^2 n)$ expected queries per sample. Then we apply the additive error algorithm of Section 3.3 to obtain a low rank approximation to $\mathbf{B}$, and thus to $\mathbf{A}$. Our algorithm queries $O\left(\phi_{\max}^2 nk/\epsilon^2\right)$ entries of $\mathbf{B}$ in expectation, so we incur an additional factor of $1/\epsilon$ in the query complexity compared to [BCW20]. The comparison is summarized in Table 3.3.

Result	Assumptions	Guarantee	Query complexity
[BCW20]	$(A1)$, $(A2)$, and $(A3)$	$\mathbf{X},\mathbf{Y} \in \mathbb{R}^{n\times k}$ satisfying $\|\mathbf{A}-\mathbf{X}\mathbf{Y}^\top\|_F^2 \le \|\mathbf{A}-\mathbf{A}_k\|_F^2 + \left(\epsilon + O(\sqrt{\eta})\right)\|\mathbf{A}\|_F^2$ with constant probability.	$\widetilde{O}\left(\frac{\phi_{\max}^2 nk}{\epsilon}\right)$
This work	$(A1)$ and $(A3)$.	$\mathbf{X},\mathbf{Y} \in \mathbb{R}^{n\times k}$ satisfying $\|\mathbf{A}-\mathbf{X}\mathbf{Y}^\top\|_F^2 \le \|\mathbf{A}-\mathbf{A}_k\|_F^2 + O(\epsilon + \sqrt{\eta})\|\mathbf{A}\|_F^2$ with constant probability.	$O\left(\frac{\phi_{\max}^2 nk}{\epsilon^2}\right)$ in expectation

Table 3.3: Comparison of robust additive-error low-rank approximation guarantees. Here $\widetilde{O}$ hides polylogarithmic factors in n, k, and $1/\epsilon$.

The main issue is that $\mathbf{B}$ need not be positive semidefinite, so Algorithm 1 cannot be applied directly. Fortunately, we can apply similar techniques after clipping the entries of $\mathbf{B}$ according to the procedure of [BCW20], which we recall here.

Definition 3. Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be psd and $\mathbf{B} = \mathbf{A} + \mathbf{N}$. Define

$$\phi_{\max} := \max\left\{1, \max_{i\in[n]} \frac{\mathbf{A}_{ii}}{|\mathbf{B}_{ii}|}\right\}.$$

For each $i, j \in [n]$, define

$$\widetilde{\mathbf{B}}_{ij} = \operatorname{clip}\left(\mathbf{B}_{ij}, -\phi_{\max}\sqrt{|\mathbf{B}_{ii}|\,|\mathbf{B}_{jj}|}, \phi_{\max}\sqrt{|\mathbf{B}_{ii}|\,|\mathbf{B}_{jj}|}\right),$$

where

$$\operatorname{clip}(x, -a, a) := \begin{cases} -a & x < -a, \\ x & -a \le x \le a, \\ a & x > a. \end{cases}$$

The following lemma, due to Bakshi, Chepurko, and Woodruff [BCW20], shows that clipping does not increase corruption to $\mathbf{A}$. We prove it here for completeness.

Lemma 9 ([BCW20])**.** *Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be psd, and let $\mathbf{B} = \mathbf{A} + \mathbf{N}$. Let $\widetilde{\mathbf{B}}$ be as in Definition 3. Then we may write*

$$\widetilde{\mathbf{B}} = \mathbf{A} + \widetilde{\mathbf{N}}$$

such that $\|\widetilde{\mathbf{N}}\|_F \le \|\mathbf{N}\|_F$.

Proof. We first note that $\mathbf{A}_{ii} \le \phi_{\max}|\mathbf{B}_{ii}|$ and $\mathbf{A}_{jj} \le \phi_{\max}|\mathbf{B}_{jj}|$. by the definition of $\phi_{\max}$. Moreover, $\mathbf{A}_{ij}^2 \le \mathbf{A}_{ii}\mathbf{A}_{jj}$ since $\mathbf{A}$ is psd, so

$$|\mathbf{A}_{ij}| \le \phi_{\max}\sqrt{|\mathbf{B}_{ii}|\,|\mathbf{B}_{jj}|}.$$

In particular, $\mathbf{A}_{ij}$ lies in the clipping interval for entry ij, and clipping at entry ij does not increase distance to $\mathbf{A}_{ij}$. Thus,

$$(\widetilde{\mathbf{B}}_{ij} - \mathbf{A}_{ij})^2 \le \left(\mathbf{B}_{ij} - \mathbf{A}_{ij}\right)^2 = \mathbf{N}_{ij}^2.$$

Summing over all entries gives $\|\widetilde{\mathbf{B}} - \mathbf{A}\|_F^2 \le \|\mathbf{N}\|_F^2$, which is the desired result. □

Although $\widetilde{\mathbf{B}}$ is not necessarily positive semidefinite, the clipping threshold allows us to write

$$|\widetilde{\mathbf{B}}_{ij}|^2 \le \phi_{\max}^2|\mathbf{B}_{ii}|\,|\mathbf{B}_{jj}| \qquad \text{for all } i, j \in [n].$$

In Algorithm 5, we use this identity to sample by squared column norms of $\widetilde{\mathbf{B}}$.

Algorithm 5: Fast Length-Squared Sampling (Robust LRA)

Let $D = \sum_i |\mathbf{B}_{ii}|$, and let d be the distribution given by $d_i = \frac{|\mathbf{B}_{ii}|}{D}$.
Repeat the following in rounds until an index is accepted:

1. Sample indices $i, j \overset{\text{i.i.d.}}{\sim} d$.
2. Accept and return i with probability $\frac{\widetilde{\mathbf{B}}_{ij}^2}{\phi_{\max}^2|\mathbf{B}_{ii}||\mathbf{B}_{jj}|}$. Otherwise, reject.

Lemma 10. *Algorithm 5 outputs i with probability*

$$p_i = \frac{\|\widetilde{\mathbf{B}}_{i,*}\|_2^2}{\|\widetilde{\mathbf{B}}\|_F^2}.$$

for all $i \in [n]$ and queries $O(\phi_{\max}^2 n)$ entries of $\mathbf{B}$ in expectation.

Proof. The acceptance probability is well-defined since $\widetilde{\mathbf{B}}_{ij}^2 \le \phi_{\max}^2 |\mathbf{B}_{ii}||\mathbf{B}_{jj}|$.

In one trial, the probability of outputting i is

$$\frac{|\mathbf{B}_{ii}|}{D}\sum_{j=1}^{n}\frac{|\mathbf{B}_{jj}|}{D}\cdot\frac{\widetilde{\mathbf{B}}_{ij}^2}{\phi_{\max}^2|\mathbf{B}_{ii}||\mathbf{B}_{jj}|} = \frac{\left\|\widetilde{\mathbf{B}}_{i,*}\right\|_2^2}{\phi_{\max}^2 D^2}.$$

Summing over all indices gives total acceptance probability

$$\frac{\left\|\widetilde{\mathbf{B}}\right\|_F^2}{\phi_{\max}^2 D^2}$$

and conditioning on acceptance, the probability of outputting i is

$$\frac{\left\|\widetilde{\mathbf{B}}_{i,*}\right\|_2^2}{\left\|\widetilde{\mathbf{B}}\right\|_F^2}.$$

It remains to bound the acceptance probability. Since $\phi_{\max} \ge 1$, the diagonal entries are not clipped, so

$$D^2 = \left(\sum_{i=1}^{n}|\mathbf{B}_{ii}|\right)^2 \le n\sum_{i=1}^{n}\mathbf{B}_{ii}^2 = n\sum_{i=1}^{n}\widetilde{\mathbf{B}}_{ii}^2 \le n\left\|\widetilde{\mathbf{B}}\right\|_F^2.$$

Thus the acceptance probability satisfies

$$\frac{\left\|\widetilde{\mathbf{B}}\right\|_F^2}{\phi_{\max}^2 D^2} \ge \frac{1}{\phi_{\max}^2 n}.$$

This gives $\phi_{\max}^2 n$ rounds in expectation. In each round, we use one entry from $\widetilde{\mathbf{B}}$, which is computed from the corresponding entry in $\mathbf{B}$, along with two additional diagonal entries of $\mathbf{B}$. This gives $O(\phi_{\max}^2 n)$ entry queries to $\mathbf{B}$ in expectation. □

We now use the robust sampler in the additive-error low-rank approximation algorithm of Section 3.3. The algorithm is identical to Algorithm 4, except that we run it on the clipped matrix $\widetilde{\mathbf{B}}$ and generate each length-squared column sample using Algorithm 5.

Algorithm 6: Robust Additive-Error Low-Rank Approximation

Input: Entry-wise access to $\mathbf{B} = \mathbf{A} + \mathbf{N}$, target rank k, accuracy parameter $\epsilon \in (0,1)$, and $\phi_{\max}$.

Output: Matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n \times k}$.

Let $\widetilde{\mathbf{B}}$ be the clipped matrix of Definition 3.

1. Use Algorithm 5 to draw $c = O(k/\epsilon^2)$ independent length-squared column samples from $\widetilde{\mathbf{B}}$.
2. Query these columns and let $\mathbf{C}$ be the rescaled sampled column matrix as defined in Step 2 of Algorithm 4.
3. Let $\mathbf{H} \in \mathbb{R}^{n \times k}$ be an orthonormal basis for the top k-dimensional left singular subspace of $\mathbf{C}$.
4. Use the sampling procedure from Steps 4-5 of Algorithm 4 to compute an approximation $\widehat{\mathbf{Z}} \in \mathbb{R}^{k \times n}$ to $\mathbf{H}^\top \widetilde{\mathbf{B}}$.
5. Return $\mathbf{X} = \mathbf{H}$ and $\mathbf{Y} = \widehat{\mathbf{Z}}^\top$.

Theorem 11. *Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be psd, let $\mathbf{B} = \mathbf{A} + \mathbf{N}$, and suppose that $\|\mathbf{N}\|_F^2 \le \eta \|\mathbf{A}\|_F^2$ for some $\eta \in [0,1]$. Then Algorithm 6 returns matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n \times k}$ such that, with probability at least $2/3$,*

$$\|\mathbf{A} - \mathbf{X}\mathbf{Y}^\top\|_F^2 \le \|\mathbf{A} - \mathbf{A}_k\|_F^2 + O(\epsilon + \sqrt{\eta})\|\mathbf{A}\|_F^2.$$

The algorithm queries

$$O\left(\frac{\phi_{\max}^2 nk}{\epsilon^2}\right)$$

entries of $\mathbf{B}$ in expectation and uses

$$O\left(\frac{nk^2}{\epsilon^4} + \frac{\phi_{\max}^2 nk}{\epsilon^2}\right)$$

arithmetic operations in expectation.

Proof. The proof of Theorem 8 applies verbatim to $\widetilde{\mathbf{B}}$, once length-squared column samples from $\widetilde{\mathbf{B}}$ are supplied by Algorithm 5. The positive semidefinite assumption in Theorem 8 is used only to obtain such samples efficiently. Thus, with probability at least $2/3$, Algorithm 6 returns $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n \times k}$ satisfying

$$\|\widetilde{\mathbf{B}} - \mathbf{X}\mathbf{Y}^\top\|_F^2 \le \|\widetilde{\mathbf{B}} - \widetilde{\mathbf{B}}_k\|_F^2 + \epsilon\|\widetilde{\mathbf{B}}\|_F^2. \qquad (*)$$

Moreover, by the same query and time complexity analysis, together with Lemma 10, the algorithm uses

$$O\left(\frac{\phi_{\max}^2 nk}{\epsilon^2}\right)$$

queries to $\mathbf{B}$ in expectation and

$$O\left(\frac{nk^2}{\epsilon^4} + \frac{\phi_{\max}^2 nk}{\epsilon^2}\right)$$

arithmetic operations in expectation.

Finally, we transfer this bound from $\widetilde{\mathbf{B}}$ back to $\mathbf{A}$. If $\eta = 0$, then $\widetilde{\mathbf{B}} = \mathbf{A}$, and the desired guarantee follows immediately from $(*)$. Thus assume $\eta > 0$. We first bound

$$\begin{aligned}\|\mathbf{A} - \mathbf{X}\mathbf{Y}^\top\|_F^2 &= \|(\widetilde{\mathbf{B}} - \mathbf{X}\mathbf{Y}^\top) - (\widetilde{\mathbf{B}} - \mathbf{A})\|_F^2 \\ &\le (1+\sqrt{\eta})\|\widetilde{\mathbf{B}} - \mathbf{X}\mathbf{Y}^\top\|_F^2 + \left(1 + \frac{1}{\sqrt{\eta}}\right)\|\widetilde{\mathbf{B}} - \mathbf{A}\|_F^2.\end{aligned}$$

Here we used the triangle inequality and bounded the cross term by

$$2\|\widetilde{\mathbf{B}} - \mathbf{X}\mathbf{Y}^\top\|_F\|\widetilde{\mathbf{B}} - \mathbf{A}\|_F \le \sqrt{\eta}\|\widetilde{\mathbf{B}} - \mathbf{X}\mathbf{Y}^\top\|_F^2 + \frac{1}{\sqrt{\eta}}\|\widetilde{\mathbf{B}} - \mathbf{A}\|_F^2.$$

Now, we recall $\|\widetilde{\mathbf{B}} - \mathbf{A}\|_F^2 \le \|\mathbf{N}\|_F^2 \le \eta\|\mathbf{A}\|_F^2$ due to Lemma 9. We have

$$\begin{aligned}\|\mathbf{A} - \mathbf{X}\mathbf{Y}^\top\|_F^2 &\le (1+\sqrt{\eta})\|\widetilde{\mathbf{B}} - \mathbf{X}\mathbf{Y}^\top\|_F^2 + O(\sqrt{\eta})\|\mathbf{A}\|_F^2 \\ &\le (1+\sqrt{\eta})\left(\|\widetilde{\mathbf{B}} - \widetilde{\mathbf{B}}_k\|_F^2 + \epsilon\|\widetilde{\mathbf{B}}\|_F^2\right) + O(\sqrt{\eta})\|\mathbf{A}\|_F^2 \\ &\le (1+\sqrt{\eta})\left(\|\widetilde{\mathbf{B}} - \mathbf{A}_k\|_F^2 + \epsilon\|\widetilde{\mathbf{B}}\|_F^2\right) + O(\sqrt{\eta})\|\mathbf{A}\|_F^2.\end{aligned}$$

For the second inequality, we use $(*)$. For the third inequality, we view $\mathbf{A}_k$ as a rank-k approximation to $\widetilde{\mathbf{B}}$ and note the optimality of $\widetilde{\mathbf{B}}_k$.

It remains to compare $\|\widetilde{\mathbf{B}} - \mathbf{A}_k\|_F^2$ and $\|\mathbf{A} - \mathbf{A}_k\|_F^2$. Again using the cross-term bound,

$$\begin{aligned}\|\widetilde{\mathbf{B}} - \mathbf{A}_k\|_F^2 &= \|(\mathbf{A} - \mathbf{A}_k) + (\widetilde{\mathbf{B}} - \mathbf{A})\|_F^2 \\ &\le (1+\sqrt{\eta})\|\mathbf{A} - \mathbf{A}_k\|_F^2 + \left(1 + \frac{1}{\sqrt{\eta}}\right)\|\widetilde{\mathbf{B}} - \mathbf{A}\|_F^2 \\ &\le (1+\sqrt{\eta})\|\mathbf{A} - \mathbf{A}_k\|_F^2 + O(\sqrt{\eta})\|\mathbf{A}\|_F^2.\end{aligned}$$

Since $\|\mathbf{A} - \mathbf{A}_k\|_F \le \|\mathbf{A}\|_F$, this implies

$$(1+\sqrt{\eta})\|\widetilde{\mathbf{B}} - \mathbf{A}_k\|_F^2 \le \|\mathbf{A} - \mathbf{A}_k\|_F^2 + O(\sqrt{\eta})\|\mathbf{A}\|_F^2.$$

Also,

$$\|\widetilde{\mathbf{B}}\|_F^2 \le \left(\|\mathbf{A}\|_F + \|\widetilde{\mathbf{B}} - \mathbf{A}\|_F\right)^2 \le (1+\sqrt{\eta})^2\|\mathbf{A}\|_F^2 \le 4\|\mathbf{A}\|_F^2,$$

where we used $\eta \leq 1$. Combining our various bounds gives

$$\|\mathbf{A} - \mathbf{X}\mathbf{Y}^\top\|_F^2 \leq \|\mathbf{A} - \mathbf{A}_k\|_F^2 + O(\epsilon + \sqrt{\eta})\|\mathbf{A}\|_F^2.$$

This completes the proof. □

Chapter 4

Relative Error Low-Rank Approximation

In the previous chapter, we used fast length-squared sampling to derive a sublinear algorithm for additive error low-rank approximation of psd matrices. This chapter returns to the stronger goal that originally motivated the search for such sampling primitives: relative-error low-rank approximation. Given entrywise access to a psd matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, a target rank r, and an accuracy parameter $\varepsilon > 0$, the desired guarantee is

$$\|\mathbf{A} - \widehat{\mathbf{A}}\|_F^2 \leq (1+\varepsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2,$$

where $\widehat{\mathbf{A}}$ is the low-rank approximation and $\mathbf{A}_k$ is the optimal rank-k approximation to $\mathbf{A}$. This chapter documents why a natural Nyström approximation-based route to this guarantee does not succeed.

Recall that for an index set $S \subset [n]$, the column Nyström approximation is

$$\mathbf{A}\langle S\rangle = \mathbf{A}_{*,S}\mathbf{A}_{S,S}^{-1}\mathbf{A}_{S,*}.$$

Both $\mathbf{A}\langle S\rangle$ and the residual matrix $\mathbf{R}_S := \mathbf{A} - \mathbf{A}\langle S\rangle$ are psd. Randomly Pivoted Cholesky, abbreviated RPCholesky, is an adaptive rule for choosing the column set S. Starting from $S_0 = \emptyset$, at step t it samples a pivot s_t according to the diagonal of the current Nyström residual:

$$\Pr[s_t = j] = \frac{(\mathbf{A} - \mathbf{A}\langle S_{t-1}\rangle)_{jj}}{\operatorname{tr}(\mathbf{A} - \mathbf{A}\langle S_{t-1}\rangle)}.$$

The new set is $S_t = S_{t-1} \cup \{s_t\}$, and after k steps the algorithm returns $\mathbf{A}\langle S_k \rangle$.

Algorithm 7: Randomly Pivoted Cholesky

Input: psd matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, approximation rank k
Initially, $S \leftarrow \varnothing$.
Do, for $i = 1, \ldots, k$:

1. sample a column $s_i \sim \dfrac{\operatorname{diag}(\mathbf{A} - \mathbf{A}\langle S \rangle)}{\operatorname{tr}(\mathbf{A} - \mathbf{A}\langle S \rangle)}$.
2. $S \leftarrow s_i$

In order to avoid evaluating $\mathbf{A} - \mathbf{A}\langle S \rangle$ in every step, we can compute the approximation for $\mathbf{A}$ in factored form using the pivoted partial Cholesky algorithm [CETW25], as demonstrated in Algorithm 8.

Algorithm 8: Pivoted partial Cholesky

Input: psd matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, approximation rank k
Output: pivot set $S = \{s_1, \ldots, s_k\}$; matrix $\mathbf{F} \in \mathbb{R}^{n \times k}$ satisfying $\mathbf{F}\mathbf{F}^\top = \mathbf{A}\langle S \rangle$
Initialize $\mathbf{F} \leftarrow \mathbf{0}_{n \times k}$
Do, for $i = 1 \ldots k$:

1. Select a pivot $s_i \in [n]$
2. $\mathbf{g} \leftarrow \mathbf{A}(:, s_i)$
3. $\mathbf{g} \leftarrow \mathbf{g} - \mathbf{F}(*, [i-1])\mathbf{F}(s_i, [i-1])^\top$
4. $\mathbf{F}(*, i) \leftarrow \mathbf{g} \big/ \sqrt{\mathbf{g}(s_i)}$

Return $\mathbf{F}$

The main theoretical guarantee for RPCHOLESKY is a relative trace-error bound. Let

$$\eta_r = \frac{\operatorname{tr}(\mathbf{A} - \mathbf{A}_r)}{\operatorname{tr}(\mathbf{A})}.$$

Chen et al. [CETW25] show that if

$$k \geq \frac{r}{\varepsilon} + r \log\left(\frac{1}{\varepsilon \eta_r}\right),$$

then the Nyström approximation produced by RPCHOLESKY satisfies

$$\mathbb{E}\left[\operatorname{tr}(\mathbf{A} - \mathbf{A}\langle S_k \rangle)\right] \leq (1 + \varepsilon) \operatorname{tr}(\mathbf{A} - \mathbf{A}_r).$$

Since $\mathbf{A} - \mathbf{A}\langle S_k \rangle$ is psd, this is a trace-norm guarantee for the residual. The question considered here is whether the same simple column-selection method, or any similarly small column Nyström approximation, can also achieve the relative Frobenius error guarantee above.

We begin with experiments on Gaussian kernel matrices. Given a dataset $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, the associated Gaussian kernel matrix is

$$\mathbf{K}(i, j) = \exp\left(-\frac{1}{2\sigma}\|\mathbf{x}_i - \mathbf{x}_j\|^2\right),$$

where $\sigma > 0$ is the kernel bandwidth. To serve as a baseline, we will also consider the greedy pivot selection method:

$$\text{select pivot} \quad s_i \in \operatorname*{arg\,max}_{1 \le j \le n} \mathbf{R}^{(i)}(j, j) \quad \text{with ties broken arbitrarily.}$$

The experiments below use our independent Python implementation. The source code is available at https://github.com/a2tian/matrices.

We first reproduce a subset of rows from Table 1 of [CETW25], which compares RPCHOLESKY against greedy pivot selection and the optimal rank-k approximation. We then repeat the same comparison for the Frobenius norm. For a low-rank approximation $\widehat{\mathbf{A}}$, define the relative trace and Frobenius errors by

$$\eta_* = \frac{\operatorname{tr}\left(\mathbf{A} - \widehat{\mathbf{A}}\right)}{\operatorname{tr}(\mathbf{A})} \qquad \eta_F = \frac{\|\mathbf{A} - \widehat{\mathbf{A}}\|_F}{\|\mathbf{A}\|_F}.$$

Table 4.1 reports these errors for rank 1000 approximations. The “Opt” columns represent the error of the optimal rank-1000 approximation given by the truncated svd.

dataset	η_*			η_F		
	Greedy	RPCHOLESKY	Opt	Greedy	RPCHOLESKY	Opt
yolanda	2.07e-1	**1.40e-1**	8.71e-2	4.64e-2	**6.30e-3**	3.15e-3
mnist_784	1.70e-1	**1.09e-1**	5.96e-2	2.53e-2	**4.95e-3**	1.97e-3
jannis	1.27e-1	**1.09e-1**	5.42e-2	1.16e-2	**6.52e-3**	2.16e-3
volkert	6.09e-2	**4.31e-2**	2.06e-2	8.15e-3	**2.26e-3**	7.16e-4
creditcard	4.36e-2	**2.32e-2**	9.87e-3	1.04e-2	**1.60e-3**	5.04e-4
hls4ml_lhc_jets_hlf	6.62e-5	**4.47e-5**	1.07e-5	1.27e-5	**4.77e-6**	7.97e-7

Table 4.1: Relative trace and Frobenius errors of rank-1000 greedy and RPCHOLESKY methods when used to approximate a Gaussian kernel matrix. Each dataset is pruned to 10^4 points and standardized, and we report the median across 10 trials. The error of the optimal rank-1000 approximation is included for reference.

The table suggests that RPCHOLESKY performs well on these kernel matrices in both norms. It consistently improves over greedy pivot selection and is often within a small constant factor of the optimal rank-1000 error. This empirical behavior was the original motivation for asking whether RPCHOLESKY might satisfy a Frobenius relative-error guarantee analogous to its trace-error guarantee.

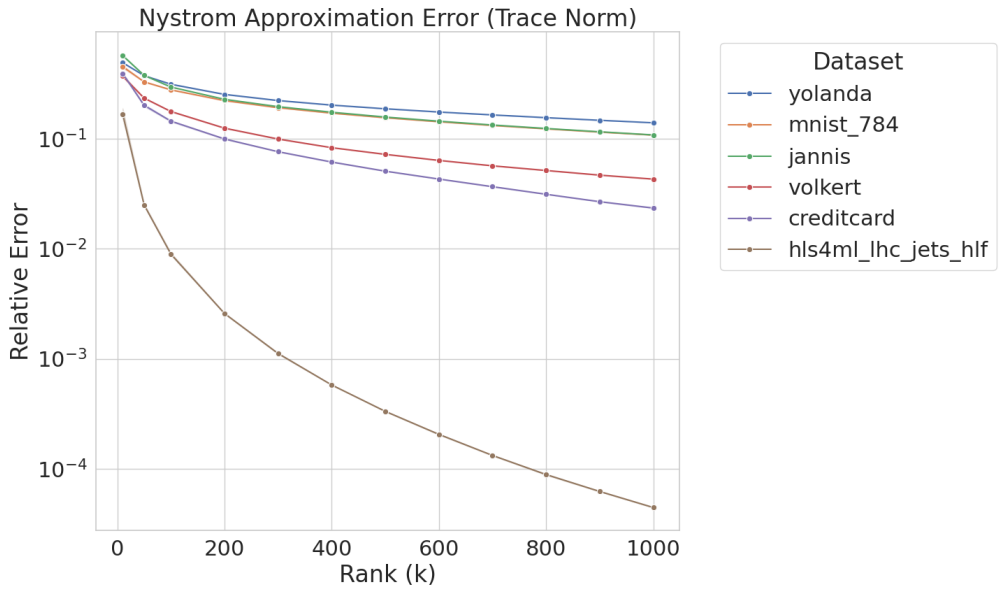

Figure 4.1: Relative trace error vs approximation rank when RPCHOLESKY is used to approximate a Gaussian kernel matrix. Each dataset is trimmed to 10^4 points and standardized. The bandwidth is set to $\sigma = \sqrt{d}$ for each dataset, where d is the number of features. The plot depicts the mean and 95% confidence interval across 10 trials for each value of k.

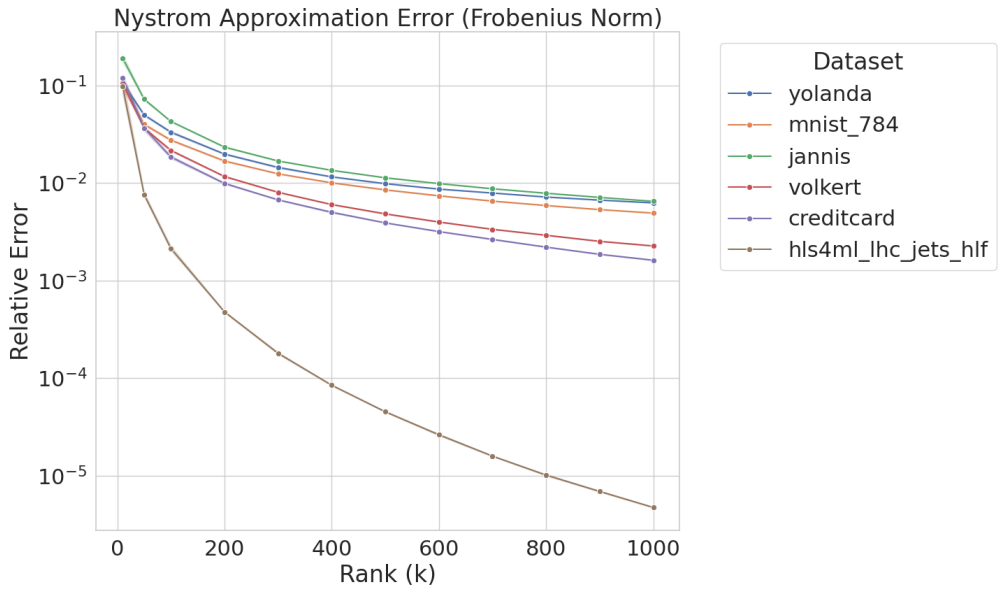

Figure 4.2: Relative Frobenius error vs approximation rank when RPCHOLESKY is used to approximate a Gaussian kernel matrix. Each dataset is trimmed to 10^4 points and standardized. The bandwidth is set to $\sigma = \sqrt{d}$ for each dataset, where d is the number of features. The plot depicts the mean and 95% confidence interval across 10 trials for each value of k.

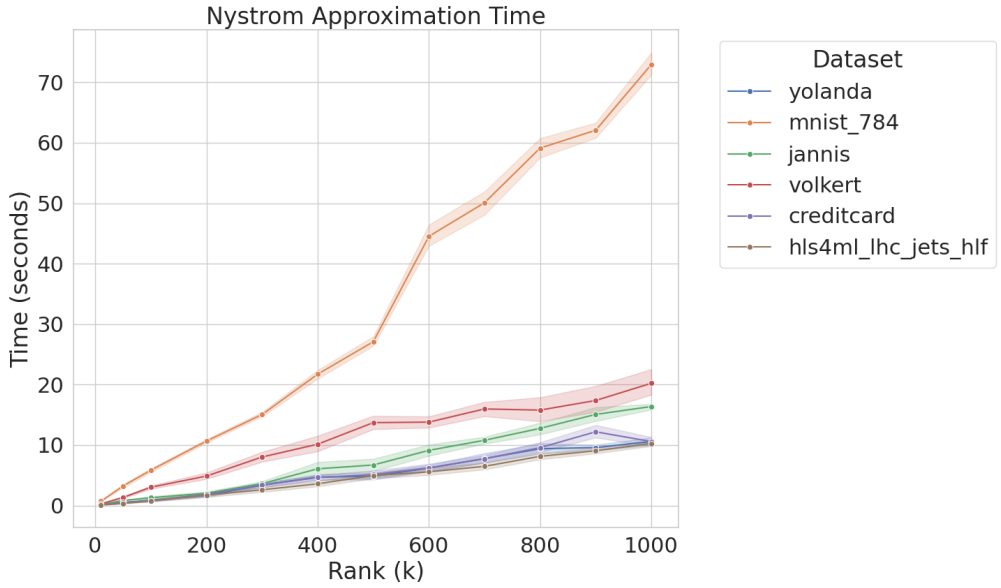

Figure 4.3: Computation time vs approximation rank when RPCHOLESKY is used to approximate a Gaussian kernel matrix. Each dataset is trimmed to 10^4 points and standardized. The bandwidth is set to $\sigma = \sqrt{d}$ for each dataset, where d is the number of features. The plot depicts the mean and 95% confidence interval across 10 trials for each value of k.

Figures 4.1 and 4.2 show that the relative error decays steadily as the approximation rank

increases. The rate of decay varies substantially across datasets, which reflects differences in the spectra of the corresponding kernel matrices. Figure 4.3 shows approximately increasing runtime as a function of k. The cost of each entry query depends on the ambient feature dimension of the dataset; hence, the running time varies between datasets.

The preceding comparison measures error relative to $\|\mathbf{A}\|_F^2$ or $\mathrm{tr}(\mathbf{A})$ rather than relative to the optimal residual. To assess relative-error behavior more directly, fix a target rank r and an index set S, and define the trace and Frobenius residual ratios

$$\gamma_{r,*}(S) = \frac{\mathrm{tr}(\mathbf{A} - \mathbf{A}\langle S\rangle)}{\mathrm{tr}(\mathbf{A} - \mathbf{A}_r)}, \qquad \gamma_{r,F}(S) = \frac{\|\mathbf{A} - \mathbf{A}\langle S\rangle\|_F^2}{\|\mathbf{A} - \mathbf{A}_r\|_F^2}.$$

For each r, each error measure, and each column-selection rule, we estimate the smallest Nyström rank k needed to match the optimal rank-r error:

$$\min\{k \geq 1 : \mathbb{E}[\gamma_{r,\bullet}(S)] \leq 1\},$$

where $\bullet \in \{*, F\}$. The expectation is over the randomized column-selection procedure. Figure 4.4 reports this quantity for RPCHOLESKY and greedy selection on the `yolanda` dataset. For RPCHOLESKY, a larger approximation rank k is required to match the optimal rank-r approximation in the Frobenius norm compared to the trace norm.

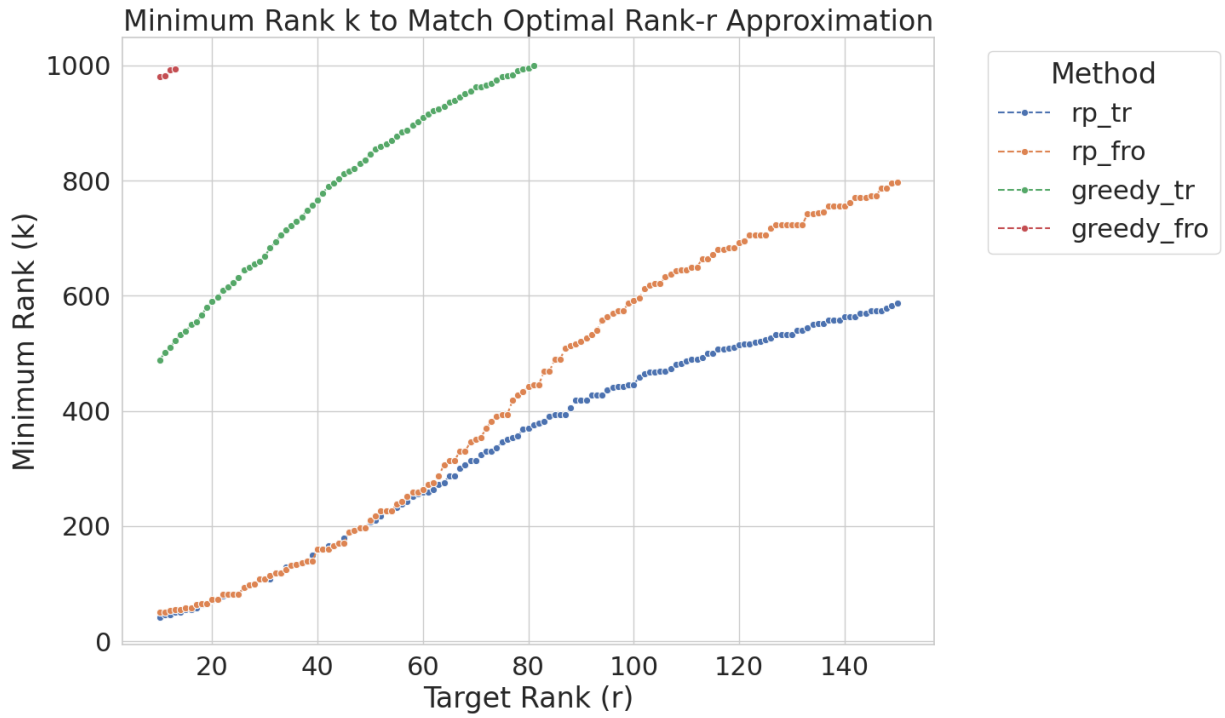

Figure 4.4: Smallest Nyström approximation rank k required to match optimal rank-r approximation error on the `yolanda` dataset. The plotted methods compare RPCHOLESKY and greedy pivot selection under trace and Frobenius error.

The empirical results make RPCHOLESKY a natural first candidate for Frobenius relative-error approximation. Nevertheless, this direction fails in the worst case. The following construction shows that RPCHOLESKY can miss columns with large Frobenius mass with high probability. A related result appears in [Epp25, Section 11.2], but we present the proof below to document a specific hard matrix for RPCHOLESKY.

Theorem 12. *Fix $r \in \mathbb{N}$ and $c \geq 1$. For all sufficiently large n, there is a psd matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ for which the Nyström approximation $\mathbf{A}\langle S \rangle$ produced by* RPCHOLESKY *satisfies*

$$\mathbb{E}\left[\|\mathbf{A} - \mathbf{A}\langle S \rangle\|_F^2\right] > c \cdot \|\mathbf{A} - \mathbf{A}_r\|_F^2$$

for all $|S| < \sqrt{n}/(4c)$.

Proof. We produce $\mathbf{A}$ directly. Consider the matrix

$$\mathbf{A} = \begin{pmatrix} \underbrace{\begin{matrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{matrix}}_{m} & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{pmatrix} \tag{4.1}$$

The optimal rank-1 approximation is spanned by any of the first m columns of $\mathbf{A}$, with error

$$\delta := \|\mathbf{A} - \mathbf{A}_1\|_F^2 = n - m.$$

Suppose RPCHOLESKY selects k columns. The diagonal entries of $\mathbf{A}$ have equal weight, so RPCHOLESKY chooses each column with equal probability. The number of optimal columns selected by RPCHOLESKY follows a hypergeometric distribution,

$$X \sim \text{Hypergeometric}(n, m, k).$$

In particular, the probability that none of the optimal columns are chosen is

$$p := \Pr[X = 0] = \prod_{i=0}^{k-1} \left(1 - \frac{m}{n-i}\right).$$

We obtain a lower bound for p as follows:

$$\begin{aligned} p &= \prod_{i=0}^{k-1} \left(1 - \frac{m}{n-i}\right) \\ &= \exp\left(\sum_{i=0}^{k-1} \log(1 - x_i)\right) \qquad x_i = \frac{m}{n-i} \in (0,1) \\ &\geq \exp\left(-\sum_{i=0}^{k-1} \frac{m}{n-m-i}\right) \qquad \text{using } \log(1-x_i) \geq -\frac{x_i}{1-x_i} \\ &\geq \exp\left(-\frac{km}{n-m-k}\right). \end{aligned}$$

If in addition we have $k < \sqrt{n}/(4c)$ and $m = c\sqrt{n}$, then

$$\begin{aligned} n - m - k &\geq n - 2m \\ &\geq \frac{n}{2} \qquad \text{for } n \geq 16c^2 \end{aligned}$$

and hence

$$\begin{aligned} p &\geq \exp\left(-\frac{km}{n-m-k}\right) \\ &\geq \exp\left(-1/2\right) \\ &> 1/2. \end{aligned}$$

By inspection, we see that the error of the rank-k RPCHOLESKY approximation satisfies

$$\|\mathbf{A}-\mathbf{A}\langle S\rangle\|_F^2 \geq \begin{cases} m^2+\delta-k & \text{with probability } p \\ \delta-k+1 & \text{with probability } 1-p. \end{cases}$$

Thus,

$$\begin{aligned} \mathbb{E}\left[\|\mathbf{A}-\mathbf{A}\langle S\rangle\|_F^2\right] &\geq p(m^2+\delta-k) \\ &> \frac{1}{2}\cdot\left(c^2 n-k+\delta\right) \\ &\geq \frac{1}{2}\cdot\delta(c^2+1) \\ &\geq c\cdot\delta \qquad \text{for all } c\geq 1 \\ &\geq c\cdot\|\mathbf{A}-\mathbf{A}_r\|_F^2. \end{aligned}$$

Choosing n large enough so that

$$n \geq 16c^2$$

completes the proof. □

The matrix in (4.1) has a simple geometric interpretation. Form a dataset with N points in dimension d by taking

1. $c\sqrt{N}$ points drawn from a spherical Gaussian distribution centered at the origin with covariance matrix $p\mathbf{I}$, and
2. $N-c\sqrt{N}$ points corresponding to vertices of the d-dimensional hypercube of width 2 centered at the origin.

The Gaussian kernel matrix of this dataset resembles the hard matrix above: the clustered points form a dense all-ones-like block, while the separated hypercube points contribute the diagonal block. Figure 4.5 illustrates this structure.

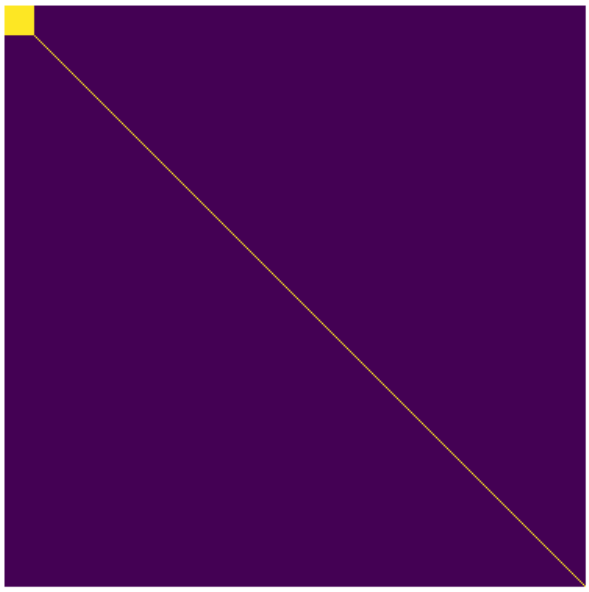

Figure 4.5: Heatmap of the Gaussian kernel matrix for the geometric hard dataset at $N = 10^4$, $d = 100$, and $c = 500$.

The preceding example suggests a natural modification of the pivot rule. The failure occurs because diagonal-proportional sampling treats the dense block and the isolated diagonal columns as equally important, even though the dense block contains much more Frobenius mass. A possible remedy is to sample a moderately large candidate principal submatrix, reweight it by the sampling probabilities, and then choose the candidate whose normalized sampled column appears largest. Taking $s \asymp \sqrt{n}$ candidates gives a constant probability of seeing a hidden block of size $\asymp \sqrt{n}$, after which the largest-column rule is designed to favor that block over the isolated columns.

Algorithm 9: RPCHOLESKY2 pivot selection

Input: residual matrix $\mathbf{R} = \mathbf{A} - \widehat{\mathbf{A}}$, parameter $c > 0$.

1. Define $p_i = \mathbf{R}_{ii} / \operatorname{tr}(\mathbf{R})$ and set $s = \lceil c\sqrt{n} \rceil$.
2. Draw candidate indices $J = (j_1, \ldots, j_s)$ independently from the distribution p.
3. Form
$$\mathbf{D} = \operatorname{diag}(p_{j_1}^{-1/2}, \ldots, p_{j_s}^{-1/2}), \qquad \mathbf{M} = \mathbf{D}\mathbf{R}(J, J)\mathbf{D}.$$
4. Return $j_{\ell^\star}$, where
$$\ell^\star \in \operatorname*{arg\,max}_{\ell \in [s]} \|\mathbf{M}\bar{\mathbf{m}}_\ell\|_2^2, \qquad \bar{\mathbf{m}}_\ell = \frac{\mathbf{M}_{:,\ell}}{\|\mathbf{M}_{:,\ell}\|_2}.$$

The experiments below test this modified pivot rule on the same `yolanda` kernel matrix used in Figure 4.4. Figure 4.6 compares the smallest Nyström rank k required to match the optimal rank-r error for RPCHOLESKY and RPCHOLESKY2, again under both trace and Frobenius errors. The modified rule reduces the required rank over much of the plotted range, but the improvement is empirical and does not by itself suggest a worst-case guarantee.

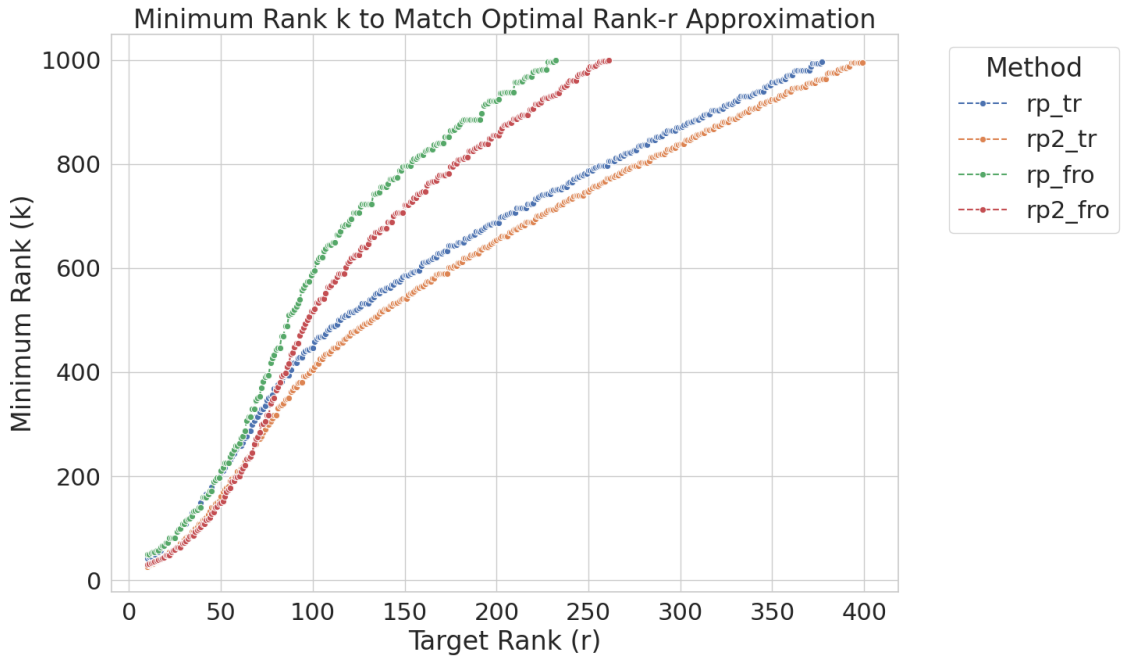

Figure 4.6: Smallest Nyström approximation rank k required to match the optimal rank-r error on the `yolanda` dataset. The figure compares RPCHOLESKY and RPCHOLESKY2 under trace and Frobenius error.

Table 4.2 compares rank-1000 RPCHOLESKY2 and RPCHOLESKY approximations on the same benchmark datasets as Table 4.1. The modified rule gives modest but consistent improvements in these tests. This is useful as a diagnostic, but the gains are not large enough to overcome the theoretical obstruction below.

dataset	η_*			η_F		
	RPCHOLESKY2	RPCHOLESKY	Opt	RPCHOLESKY2	RPCHOLESKY	Opt
yolanda	**1.38e-1**	1.40e-1	8.71e-2	**6.02e-3**	6.30e-3	3.15e-3
mnist_784	**1.04e-1**	1.09e-1	5.96e-2	**4.67e-3**	4.95e-3	1.97e-3
jannis	**1.07e-1**	1.09e-1	5.42e-2	**6.27e-3**	6.52e-3	2.16e-3
volkert	**4.07e-2**	4.31e-2	2.06e-2	**2.09e-3**	2.26e-3	7.16e-4
creditcard	**2.08e-2**	2.32e-2	9.87e-3	**1.39e-3**	1.60e-3	5.04e-4
hls4ml_lhc_jets_hlf	**3.98e-5**	4.47e-5	1.07e-5	**4.01e-6**	4.77e-6	7.97e-7

Table 4.2: Relative trace and Frobenius error comparison between rank-1000 RPCHOLESKY and RPCHOLESKY2 methods when used to approximate a Gaussian kernel matrix. Each dataset is pruned to 10^4 points and standardized, and we report the median across 10 trials. The error of the optimal rank-1000 approximation is included for reference.

Figure 4.7 shows the effect of the oversampling parameter c on the `yolanda` dataset. Larger values of c inspect larger candidate principal submatrices at each step, increasing the cost per pivot but improving the chance of identifying a high-Frobenius-mass column.

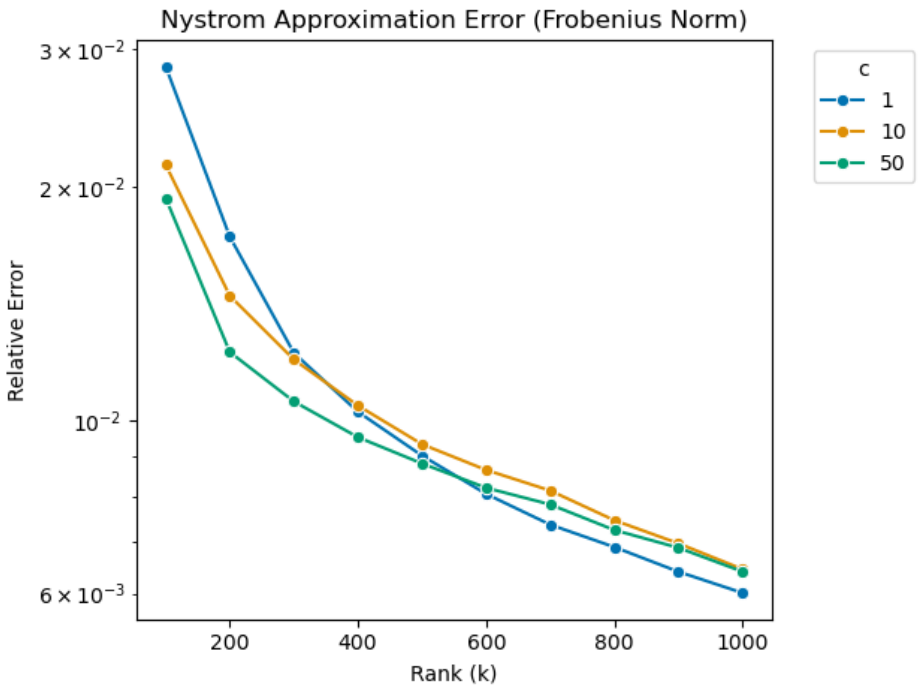

Figure 4.7: Relative Frobenius error of RPCHOLESKY2 on the `yolanda` dataset with varied choices of c, where $c\sqrt{n}$ is the number of candidate indices used to form the sampled principal submatrix at each pivot step.

We now introduce the main negative result of this section, which shows that $\Omega(\sqrt{n})$ sampled columns are needed in the worst case to achieve a relative Frobenius error guarantee using the Nyström approximation. We note that a similar lower bound appears in [WZ13].

After querying the sampled columns, $\Omega(n^{1.5})$ entry queries are needed for a relative Frobenius error approximation. In particular, we cannot hope to achieve linear dependence on n, matching the results of [MW19, BCW20], using a Nyström approximation.

Theorem 13 (Nyström lower bound)**.** *For all $\alpha \geq 1$, there exist PSD matrices $\mathbf{A} \in \mathbb{R}^{n\times n}$ such that **any** rank-k column Nyström approximation with*

$$k \leq \sqrt{n/(2\alpha)}$$

has error $\|\mathbf{A} - \mathbf{A}\langle S\rangle\|_F^2 > \alpha \cdot \|\mathbf{A} - \mathbf{A}_1\|_F^2$.

Proof. Let $\mathbf{A} = \mathbf{I}_n + \delta\mathbf{J}_n$ for some $\delta \geq 1$, where $\mathbf{I}_n$ is the $n \times n$ identity matrix and $\mathbf{J}_n$ is the $n \times n$

all-ones matrix. The eigenvalues of $\mathbf{A}$ are

$$\lambda_1 = 1 + n\delta \qquad \lambda_2 = 1 \qquad \cdots \qquad \lambda_n = 1.$$

The first eigenvector of $\mathbf{A}$ is $\mathbf{u}_1 = (1/\sqrt{n})\mathbf{1}$, and the best rank-1 approximation to $\mathbf{A}$ is therefore given by

$$\mathbf{A}_1 = (1 + n\delta)\mathbf{u}_1\mathbf{u}_1^\top = \left(\frac{1}{n} + \delta\right)\mathbf{J}_n \qquad 1 + n\delta \geq 1$$

with approximation error

$$\|\mathbf{A} - \mathbf{A}_1\|_F^2 = \sum_{i=2}^{n} \lambda_i^2 = n - 1. \tag{$*$}$$

On the other hand, suppose we choose k columns of $\mathbf{A}$ to compute the Nyström approximation $\mathbf{A}\langle S\rangle$. Without loss of generality, we can take the first k columns, i.e., set $S = [k]$. With $d = n - k$, partition $\mathbf{A}$ into blocks as follows:

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_{kk} & \mathbf{A}_{kd} \\ \mathbf{A}_{dk} & \mathbf{A}_{dd} \end{pmatrix}$$

where

$$\mathbf{A}_{kk} = \mathbf{I}_k + \delta\mathbf{J}_k \qquad \mathbf{A}_{dk} = \mathbf{A}_{kd}^\top = \delta\mathbf{1}_{d\times k} \qquad \mathbf{A}_{dd} = \mathbf{I}_d + \delta\mathbf{J}_d.$$

The Nyström approximation in block form is given by

$$\mathbf{A}\langle S\rangle = \begin{pmatrix} \mathbf{A}_{kk} \\ \mathbf{A}_{dk} \end{pmatrix} \mathbf{A}_{kk}^{-1} \begin{pmatrix} \mathbf{A}_{kk} \\ \mathbf{A}_{dk} \end{pmatrix}^\top = \begin{pmatrix} \mathbf{A}_{kk} & \mathbf{A}_{kd} \\ \mathbf{A}_{dk} & \mathbf{A}_{dk}\mathbf{A}_{kk}^{-1}\mathbf{A}_{kd} \end{pmatrix}.$$

In particular,

$$\mathbf{A} - \mathbf{A}\langle S\rangle = \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{A}_{dd} - \mathbf{A}_{dk}\mathbf{A}_{kk}^{-1}\mathbf{A}_{kd} \end{pmatrix}$$

so the error of the Nyström approximation is

$$\|\mathbf{A} - \mathbf{A}\langle S\rangle\|_F^2 = \|\mathbf{A}_{dd} - \mathbf{A}_{dk}\mathbf{A}_{kk}^{-1}\mathbf{A}_{kd}\|_F^2.$$

Now, set $\mathbf{R} := \mathbf{A}_{dd} - \mathbf{A}_{dk}\mathbf{A}_{kk}^{-1}\mathbf{A}_{kd}$ and compute directly:

$$\begin{aligned}
\mathbf{R} &= \mathbf{A}_{dd} - \mathbf{A}_{dk}\left(\mathbf{I}_k + \delta \mathbf{1}_k \mathbf{1}_k^\top\right)^{-1}\mathbf{A}_{kd} \\
&= \mathbf{A}_{dd} - \mathbf{A}_{dk}\left(\mathbf{I}_k - \frac{\delta \mathbf{1}_k \mathbf{1}_k^\top}{1+\delta \mathbf{1}_k^\top \mathbf{1}_k}\right)\mathbf{A}_{kd} \qquad \text{(Sherman-Morrison)} \\
&= \mathbf{A}_{dd} - \delta^2 \mathbf{1}_{d\times k}\left(\mathbf{I}_k - \frac{\delta}{1+k\delta}\mathbf{1}_{k\times k}\right)\mathbf{1}_{k\times d} \\
&= \mathbf{A}_{dd} - \delta^2 \mathbf{1}_{d\times k}\left(\mathbf{1}_{k\times d} - \frac{k\delta}{1+k\delta}\mathbf{1}_{k\times d}\right) \\
&= \mathbf{A}_{dd} - \delta^2 \mathbf{1}_{d\times k}\left(\frac{1}{1+k\delta}\mathbf{1}_{k\times d}\right) \\
&= \mathbf{A}_{dd} - \frac{k\delta^2}{1+k\delta}\mathbf{1}_{d\times d} \\
&= \mathbf{I}_d + \delta \mathbf{1}_{d\times d} - \frac{k\delta^2}{1+k\delta}\mathbf{1}_{d\times d} \\
&= \mathbf{I}_d + \frac{\delta}{1+k\delta}\mathbf{1}_{d\times d}.
\end{aligned}$$

Based on the above characterization, the eigenvalues of $\mathbf{R}$ are

$$\lambda_1 = 1 + \frac{d\delta}{1+k\delta} = \frac{1+n\delta}{1+k\delta} \qquad \lambda_2 = 1 \qquad \cdots \qquad \lambda_d = 1$$

Thus, the approximation error satisfies

$$\begin{aligned}
\|\mathbf{A} - \mathbf{A}\langle S\rangle\|_F^2 &= \sum_{i=1}^{d} \lambda_i^2 \\
&\geq \lambda_1^2 = \left(\frac{1+n\delta}{1+k\delta}\right)^2 \\
&\geq \frac{n^2\delta^2}{2k^2\delta^2} \qquad \text{for } \delta \text{ sufficiently large} \\
&\geq \alpha \cdot n \qquad \text{for } k \leq \sqrt{n/(2\alpha)} \\
&> \alpha \cdot (n-1) \\
&= \alpha \cdot \|\mathbf{A} - \mathbf{A}_1\|_F^2
\end{aligned}$$

as required. □

For constant α, Theorem 13 implies that some matrices require $k = \Omega(\sqrt{n})$ selected columns before a column Nyström approximation can achieve constant-factor Frobenius relative error. After reading the full columns, this gives an $\Omega(n^{1.5})$ query lower bound, so linear dependence on n is not possible.

Chapter 5

Discussion

This thesis studied the problem of performing matrix approximation when a large positive semidefinite matrix is available only through entry queries. The main technical contribution is a fast procedure for length-squared sampling from a psd matrix. Given a psd matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, the algorithm in Chapter 3 outputs index i with probability

$$p_i = \frac{\|\mathbf{A}_{*,i}\|_2^2}{\|\mathbf{A}\|_F^2},$$

while reading only $O(n)$ entries of $\mathbf{A}$ in expectation. This avoids the direct approach of computing all column norms.

We then explored how this sampling primitive can be used in matrix approximation algorithms that previously assumed access to squared row or column norm distributions. In particular, we considered Frobenius norm approximation, additive-error eigenvalue approximation, additive-error low-rank approximation, and robust low-rank approximation.

Finally, we turned to the stronger goal of relative-error low-rank approximation. We studied RPCHOLESKY, a randomized Nyström method with a relative trace-error guarantee, as a natural candidate for Frobenius relative-error approximation. We attempted to modify the RPCHOLESKY sampling procedure to achieve a relative Frobenius error guarantee, but we ultimately show that any Nyström approximation requires $\Omega(\sqrt{n})$ sampled columns to achieve a relative Frobenius error approximation in the worst case.

5.1 Open Questions

The most direct open question is whether the additive-error low-rank approximation algorithm in Section 3.3 can be improved from $O(nk/\epsilon^2)$ queries to $O(nk/\epsilon)$, matching the optimal dependence on n, k, and ϵ. The current analysis uses $O(k/\epsilon^2)$ length-squared column samples. It is natural to ask whether $O(k/\epsilon)$ length-squared samples are sufficient for the same additive-error guarantee in the psd setting.

A second open question is whether length-squared sampling can be used more directly to obtain relative-error low-rank approximations of psd matrices. One possible approach is through adaptive sampling: if our techniques could be extended to sample from the residual matrix, then one could apply the guarantee of [DV06]. However, it is not clear whether the residual retains enough structure for this approach to work.

A third direction is to further develop the robust sampling algorithm. At present, the algorithm assumes access to the diagonal corruption parameter $\phi_{\max}$. It remains open whether $\phi_{\max}$ can itself be estimated efficiently from entry queries.

Bibliography

[BCW20] Ainesh Bakshi, Nadiia Chepurko, and David P. Woodruff. Robust and Sample Optimal Algorithms for PSD Low Rank Approximation. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 506–516, Durham, NC, USA, November 2020. IEEE.

[BDD+22] Rajarshi Bhattacharjee, Gregory Dexter, Petros Drineas, Cameron Musco, and Archan Ray. Sublinear Time Eigenvalue Approximation via Random Sampling, July 2022. arXiv:2109.07647 [cs].

[BW18] Ainesh Bakshi and David Woodruff. Sublinear Time Low-Rank Approximation of Distance Matrices. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[CETW25] Yifan Chen, Ethan N. Epperly, Joel A. Tropp, and Robert J. Webber. Randomly pivoted Cholesky: Practical approximation of a kernel matrix with few entry evaluations. *Communications on Pure and Applied Mathematics*, 78(5):995–1041, 2025.

[CW13] Kenneth L. Clarkson and David P. Woodruff. Low Rank Approximation and Regression in Input Sparsity Time, April 2013. arXiv:1207.6365 [cs].

[CW17] Kenneth L. Clarkson and David P. Woodruff. Low-Rank PSD Approximation in Input-Sparsity Time. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Sympo-*

sium on Discrete Algorithms, pages 2061–2072. Society for Industrial and Applied Mathematics, January 2017.

[DKM06a] Petros Drineas, Ravi Kannan, and Michael W. Mahoney. Fast Monte Carlo Algorithms for Matrices I: Approximating Matrix Multiplication. *SIAM Journal on Computing*, 36(1):132–157, January 2006.

[DKM06b] Petros Drineas, Ravi Kannan, and Michael W. Mahoney. Fast Monte Carlo Algorithms for Matrices II: Computing a Low-Rank Approximation to a Matrix. *SIAM Journal on Computing*, 36(1):158–183, January 2006.

[DMM07] Petros Drineas, Michael W. Mahoney, and S. Muthukrishnan. Relative-Error CUR Matrix Decompositions, 2007. Version Number: 1.

[DMMS07] Petros Drineas, Michael W. Mahoney, S. Muthukrishnan, and Tamas Sarlos. Faster Least Squares Approximation, 2007. Version Number: 4.

[DV06] Amit Deshpande and Santosh Vempala. Adaptive Sampling and Fast Low-Rank Matrix Approximation. In Josep Díaz, Klaus Jansen, José D. P. Rolim, and Uri Zwick, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 292–303, Berlin, Heidelberg, 2006. Springer.

[Epp25] Ethan N. Epperly. *Make the most of what you have: Resource-efficient randomized algorithms for matrix computations*. PhD thesis, June 2025. arXiv:2512.15929 [math.NA].

[FKV04] Alan Frieze, Ravi Kannan, and Santosh Vempala. Fast monte-carlo algorithms for finding low-rank approximations. *Journal of the ACM*, 51(6):1025–1041, November 2004.

[Git11] Alex Gittens. The spectral norm error of the naive Nystrom extension, 2011. Version Number: 1.

[GM13] Alex Gittens and Michael W. Mahoney. Revisiting the Nystrom Method for Improved Large-Scale Machine Learning, 2013. Version Number: 2.

[KMS25] Michael Kapralov, Cameron Musco, and Kshiteej Sheth. Sublinear Time Low-Rank Approximation of Hankel Matrices, November 2025. arXiv:2511.21418 [cs].

[KMT12] Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. Sampling Methods for the Nyström Method. *Journal of Machine Learning Research*, 13(34):981–1006, 2012.

[MM17] Cameron Musco and Christopher Musco. Recursive Sampling for the Nyström Method, November 2017. arXiv:1605.07583 [cs].

[MS24] Cameron Musco and Kshiteej Sheth. Sublinear Time Low-Rank Approximation of Toeplitz Matrices. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 5084–5117. 2024.

[MW19] Cameron Musco and David P. Woodruff. Sublinear Time Low-Rank Approximation of Positive Semidefinite Matrices, January 2019. arXiv:1704.03371 [cs].

[NSW15] Deanna Needell, Nathan Srebro, and Rachel Ward. Stochastic Gradient Descent, Weighted Sampling, and the Randomized Kaczmarz algorithm, January 2015. arXiv:1310.5715 [math].

[SW24] William Swartworth and David P. Woodruff. Tight Sampling Bounds for Eigenvalue Approximation, November 2024. arXiv:2411.03227 [cs.DS].

[TR10] Ameet Talwalkar and Afshin Rostamizadeh. Matrix Coherence and the Nystrom Method, 2010. Version Number: 1.

[WS00] Christopher K. I. Williams and Matthias Seeger. Using the Nyström method to speed up kernel machines. In *Proceedings of the 14th International Conference on Neural Information Processing Systems*, NIPS'00, pages 661–667, Cambridge, MA, USA, January 2000. MIT Press.

[WZ13] Shusen Wang and Zhihua Zhang. Improving CUR Matrix Decomposition and the Nyström Approximation via Adaptive Sampling, October 2013. arXiv:1303.4207 [cs].