

# Matrix Compression using the Nyström Method

Arik Nemtsov<sup>1</sup> Amir Averbuch<sup>1</sup> Alon Schclar<sup>2</sup>

<sup>1</sup>School of Computer Science

Tel Aviv University, Tel Aviv 69978

<sup>2</sup>School of Computer Science

The Academic College of Tel Aviv-Yaffo ,Tel Aviv, 61083

## Abstract

The Nyström method is routinely used for out-of-sample extension of kernel matrices. We describe how this method can be applied to find the singular value decomposition (SVD) of general matrices and the eigenvalue decomposition (EVD) of square matrices. We take as an input a matrix  $M \in \mathbb{R}^{m \times n}$ , a user defined integer  $s \leq \min(m, n)$  and  $A_M \in \mathbb{R}^{s \times s}$ , a matrix sampled from columns and rows of  $M$ . These are used to construct an approximate rank- $s$  SVD of  $M$  in  $O(s^2(m+n))$  operations. If  $M$  is square, the rank- $s$  EVD can be similarly constructed in  $O(s^2n)$  operations. In this sense,  $A_M$  is a compressed version of  $M$ .

We discuss theoretical considerations for the choice of  $A_M$  and how it relates to the approximation error. Finally, we propose an algorithm that selects a good initial sample for a pivoted version of  $M$ . The algorithm relies on a previously computed approximate rank- $s$  decomposition of  $M$ , termed  $M_{decomp}$ . We show that  $\|M_{decomp} - M\|_2$  is related to the Nyström approximation error when the selected sample is employed. Then, the user can choose a more computationally expensive algorithm for computing  $M_{decomp}$  when higher accuracy is required.

We present experimental results that evaluate our sample selection algorithm for general matrices and kernel matrices. The algorithm is shown to work well for matrices whose spectra exhibit fast decay.

Key words: {SVD, EVD, Nyström, out-of-sample extension}  
2000 MSC: {65F15, 65F30, 65F50}

## 1 Introduction

Low rank approximation of linear operators is an important problem in the areas of scientific computing and statistical analysis. Approximation reduces storage requirements for large datasets and improves the runtime complexity of algorithms operating on the matrix. When the matrix contains affinities between elements, low rank approximation can be used to reduce the dimension of the original problem ([29, 30, 32]) and to eliminate statistical noise ([31]).

Our approach involves the choice of a small sub-sample from the matrix, followed by the application of the Nyström method for out-of-sample extension. The Nyström method ([1]), which originates from the field of integral equations, is a way of discretizing an integral equation using a simple quadrature rule. When given an eigenfunction problem of the form

$$\lambda f(x) = \int_a^b M(x, y) f(y) dy,$$

the Nyström method employs a set of  $s$  sample points  $y_1, \dots, y_s$  that approximate  $f(x)$  as

$$\lambda \tilde{f}(x) \triangleq \frac{b-a}{s} \sum_{j=1}^s M(x, y_j) f(y_j).$$

In recent years, the Nyström method has gained widespread use in the field of spectral clustering. It was first popularized by [16] for sparsifying kernel matrices by approximating their entries. The matrix completion approach of [2] also enables the approximation of eigenvectors. It was now possible to use the Nyström method in order to speed up algorithms that require the spectrum of a kernel matrix. Over time, Nyström based out-of-sample extensions have been developed for a wide range of spectral methods, including Normalized-Cut ([17, 18]), Geometric Harmonics ([19]) and others ([20]).

Other noteworthy methods for speeding up kernel based algorithms, which are not applicable to the proposed setting of this paper, are based on sampling [21], convex optimization [22] and integral equations. ACA [23, 24] is an important example in the latter category. ACA

can be regarded as an efficient replacement of the SVD which is tailored to asymptotically smooth kernels. The kernel function itself is not required. ACA uses only few of the original entries for the approximation of the whole matrix and it was shown to have exponential convergence when used as part of the Nyström method.

In this paper, we present two extensions of the matrix completion approach of [2]. These allow us to form the SVD and EVD of a general matrix through the application of the Nyström method on a previously chosen sample.

In addition, we present a novel algorithm for selecting the initial sample to be used with the Nyström method. Our algorithm is applicable to general matrices whereas previous methods focused on kernel matrices. The algorithm uses a pre-existing low-rank decomposition of the input matrix. We show that our sample choice reduces the Nyström approximation error.

The paper is organized as follows: Section 2 describes the basic Nyström matrix form and the methods of [2] for finding the EVD of a Nyström approximated symmetric matrix. Section 3 outlines a Nyström-like method for out-of-sample extension of general matrices, starting with the SVD of a sample matrix. In section 4 we describe procedures that explicitly generate the canonical SVD and EVD forms for general matrices. Section 5 introduces the problem of sample choice and presents results that bound the accuracy of the algorithm in section 6. Section 6 presents our sample selection algorithm and analyzes its complexity. Experimental results on general and kernel matrices are presented in section 7.

## 2 Preliminaries

### 2.1 Square Nyström Matrix Form

Let  $M \in \mathbb{R}^{n \times n}$  be a square matrix. We assume that the  $M$  can be decomposed as

$$M = \begin{bmatrix} A_M & B_M \\ F_M & C_M \end{bmatrix} \quad (2.1)$$

where  $A_M \in \mathbb{R}^{s \times s}$ ,  $B_M \in \mathbb{R}^{s \times (n-s)}$ ,  $F_M \in \mathbb{R}^{(n-s) \times s}$  and  $C_M \in \mathbb{R}^{(n-s) \times (n-s)}$ . The matrix  $A_M$  is designated to be our sample matrix. The size of our sample is  $s$ , which is the size of  $A_M$ .

Let  $U\Lambda U^{-1}$  be the eigen-decomposition of  $A_M$ , where  $U \in \mathbb{R}^{s \times s}$  is the eigenvectors matrix and  $\Lambda \in \mathbb{R}^{s \times s}$  is the eigenvalues matrix. Let  $u^i \in \mathbb{R}^s$  be the column eigenvector belonging

to eigenvalue  $\lambda_i$ . We aim to extend the column eigenvector (the discrete form of an eigenfunction) to the rest of  $M$ . Let  $\hat{u}^i = \begin{bmatrix} u^i & \tilde{u}^i \end{bmatrix}^T \in \mathbb{R}^n$  be the extended eigenvector, where  $\tilde{u}^i \in \mathbb{R}^{n-s}$  is the extended part. By applying the Nyström method to  $u^i$ , we get the following form for the  $k^{\text{th}}$  coordinate in  $\hat{u}^i$ :

$$\lambda_i \hat{u}_k^i \simeq \frac{b-a}{s} \sum_{j=1}^s M_{kj} \cdot u_j^i. \quad (2.2)$$

By setting  $[a, b] = [0, 1]$  and presenting Eq. (2.2) in matrix product form we obtain

$$\lambda_i \tilde{u}^i = \frac{1}{s} F_M \cdot u^i. \quad (2.3)$$

This can be done for all the eigenvalues  $\{\lambda_i\}_{i=1}^s$  of  $A_M$ . Denote  $\tilde{U} = \begin{bmatrix} \tilde{u}^1 & \dots & \tilde{u}^s \end{bmatrix} \in \mathbb{R}^{(n-s) \times s}$ . By placing all expressions of the form Eq. (2.3) side by side we have  $\tilde{U}\Lambda = F_M U$ . Assuming the matrix  $A_M$  has non-zero eigenvalues (we return to this assumption in section 5.4), we obtain:

$$\tilde{U} = F_M U \Lambda^{-1}. \quad (2.4)$$

Analogously, we can derive a matrix representation for extending the left eigenvectors of  $M$ , denoted as  $\tilde{V} \in \mathbb{R}^{s \times (n-s)}$ :

$$\tilde{V} = \Lambda^{-1} U^{-1} B_M. \quad (2.5)$$

Combining Eqs. (2.4) and (2.5) with the eigenvectors of  $A_M$  yields the full left and right approximated eigenvectors:

$$\hat{U} = \begin{bmatrix} U \\ F_M U \Lambda^{-1} \end{bmatrix}, \quad \hat{V} = \begin{bmatrix} U^{-1} & \Lambda^{-1} U^{-1} B_M \end{bmatrix}. \quad (2.6)$$

The explicit ‘‘Nyström’’ representation of  $\hat{M}$  becomes:

$$\hat{M} = \hat{U} \Lambda \hat{V} = \begin{bmatrix} U \\ F_M U \Lambda^{-1} \end{bmatrix} \Lambda \begin{bmatrix} U^{-1} & \Lambda^{-1} U^{-1} B_M \end{bmatrix} = \begin{bmatrix} A_M & B_M \\ F_M & F_M A_M^+ B_M \end{bmatrix} = \begin{bmatrix} A_M \\ F_M \end{bmatrix} A_M^+ \begin{bmatrix} A_M & B_M \end{bmatrix} \quad (2.7)$$

where  $A_M^+$  denotes the pseudo-inverse of  $A_M$ .

Equation (2.7) shows that the Nyström extension does not modify  $A_M, B_M$  and  $F_M$ , and that it approximates  $C_M$  by  $F_M A_M^+ B_M$ .

## 2.2 Decomposition of Symmetric Matrices

The algorithm given in [2] is a commonly used method for SVD approximation of symmetric matrices. For a given matrix, it computes the SVD of its Nyström approximated form. The SVD and EVD of a symmetric matrix coincide up to the signs of the singular (eigen-) values. Therefore the SVD can approximate both simultaneously. We describe the method of [2] in section 2.2.2.

### 2.2.1 Symmetric Nyström Matrix Form

When  $M$  is symmetric, the matrix  $M$  has the decomposition

$$M = \begin{bmatrix} A_M & B_M \\ B_M^T & C_M \end{bmatrix} \quad (2.8)$$

where  $A_M \in \mathbb{R}^{s \times s}$ ,  $B_M \in \mathbb{R}^{s \times (n-s)}$  and  $C_M \in \mathbb{R}^{(n-s) \times (n-s)}$ . We replace  $F_M$  in Eq. (2.1) with  $B_M^T$ .

By using reasoning similar to section 2.1, we can express the right and left approximated eigenvectors as:

$$\hat{U} = \begin{bmatrix} U \\ B_M^T U \Lambda^{-1} \end{bmatrix}, \quad \hat{V} = \begin{bmatrix} U^{-1} & \Lambda^{-1} U^{-1} B_M \end{bmatrix}. \quad (2.9)$$

The explicit ‘‘Nyström’’ representation of  $\hat{M}$  becomes:

$$\hat{M} = \hat{U} \Lambda \hat{V} = \begin{bmatrix} U \\ B_M^T U \Lambda^{-1} \end{bmatrix} \Lambda \begin{bmatrix} U^{-1} & \Lambda^{-1} U^{-1} B_M \end{bmatrix} = \begin{bmatrix} A_M & B_M \\ B_M^T & B_M^T A_M^+ B_M \end{bmatrix} = \begin{bmatrix} A_M \\ B_M^T \end{bmatrix} A_M^+ \begin{bmatrix} A_M & B_M \end{bmatrix}. \quad (2.10)$$

### 2.2.2 Construction of SVD for Symmetric $\hat{M}$

Our goal is to find the  $s$  leading eigenvalues and eigenvectors of  $\hat{M}$  without explicitly forming the entire matrix.

We begin with the decomposition of  $M$  as in Eq. (2.8). The approximation technique in [2] uses the standard Nyström method in Eq. (2.9) to obtain  $\hat{U}$ . Then, the algorithm forms

the matrix  $Z = \hat{U}\Lambda^{1/2}$  such that  $\hat{M} = ZZ^T = \hat{U}\Lambda\hat{U}^T$ . The symmetric  $s \times s$  matrix  $Z^T Z$  is diagonalized as  $F\Sigma F^T$ . The eigenvectors of  $\hat{M}$  are given by  $U_o = ZF\Sigma^{-1/2}$  and the eigenvalues are given by  $\Sigma$ . To qualify for use in the SVD,  $U_o$  and  $\Sigma$  must meet the following requirements:

1. The columns of  $U_o$  must be orthogonal. Namely,  $U_o^T U_o = I$ .
2. The SVD form of  $U_o$  and  $\Sigma$  must form  $\hat{M}$ . Formally,  $\hat{M} = U_o \Sigma U_o^T$ .

The following identities can be readily verified using our expressions for  $U_o$  and  $\Sigma$ :

1. Bi-orthogonality:  $U_o^T U_o = \Sigma^{-1/2} F^T Z^T Z F \Sigma^{-1/2} = \Sigma^{-1/2} F^T (F \Sigma F^T) F \Sigma^{-1/2} = I$ ;
2. SVD form:  $U_o \Sigma U_o^T = Z F \Sigma^{-1/2} \cdot \Sigma \cdot \Sigma^{-1/2} F^T Z^T = Z Z^T = \hat{M}$ .

The computational complexity of the algorithm is  $O(s^2 n)$ , where  $s$  is the sample size and  $n$  is the number of rows and columns of  $M$ . The bottleneck is in the computation of the matrix product  $Z^T Z$ .

### 2.2.3 A Single-Step Solution for the SVD of $\hat{M}$

The “one-shot” solution in [2] assumes that  $A_M$  has a square root matrix  $A_M^{1/2}$ . This assumption is true if the matrix is positive definite. Otherwise, it imposes some limitations on  $A_M$ . These will be discussed in section 4.3.

Let  $A_M^{-1/2}$  be the pseudo-inverse of the square root matrix of  $A_M$ . Denote  $G^T = A_M^{-1/2} \begin{bmatrix} A_M & B_M \end{bmatrix}$ .

From this definition we have  $\hat{M} = GG^T$ . The matrix  $S \in \mathbb{R}^{s \times s}$  was defined in [2], where  $S = G^T G = A_M + A_M^{-1/2} B_M B_M^T A_M^{-1/2}$ .  $S$  is fully decomposed as  $U_S \Lambda_S U_S^T$ . The orthogonal eigenvectors of  $\hat{M}$  are formed as  $U_o = G U_S \Lambda_S^{-1/2}$  and the eigenvalues are given in  $\Lambda_S$ .

The following required identities, as in section 2.2.2, can again be verified as follows:

1. Bi-orthogonality:
 
$$U_o^T U_o = \Lambda_S^{-1/2} U_S^T G^T G U_S \Lambda_S^{-1/2} = \Lambda_S^{-1/2} U_S^T S U_S \Lambda_S^{-1/2} = \Lambda_S^{-1/2} U_S^T \cdot U_S \Lambda_S U_S^T \cdot U_S \Lambda_S^{-1/2} = I.$$
2. SVD form:  $U_o \Lambda_S U_o^T = G U_S \Lambda_S^{-1/2} \cdot \Lambda_S \cdot \Lambda_S^{-1/2} U_S^T G^T = GG^T = \hat{M}$ .

The computational complexity remains the same (the bottleneck of the algorithm is the formation of  $B_M B_M^T$ ). However this version is numerically more accurate. According to [2], the extra calculations in the general method of solution lead to an increase in the loss of significant digits.

### 3 Nyström-like SVD approximation

The SVD of a matrix can also be approximated via the basic quadrature technique of the Nyström method. In this case, we do not require an eigen-decomposition. Therefore,  $M$  does not necessarily have to be square. Let  $M \in \mathbb{R}^{m \times n}$  be a matrix with the decomposition given in Eq. (2.1). We begin with the SVD form  $A_M = U\Lambda H$  where  $U, H \in \mathbb{R}^{s \times s}$  are unitary matrices and  $\Lambda \in \mathbb{R}^{s \times s}$  is diagonal. We assume that zero is not a singular value of  $A_M$ . Accordingly,  $U$  can be formulated as:

$$U = A_M H \Lambda^{-1}. \quad (3.1)$$

Let  $u^i, h^i \in \mathbb{R}^s$  be the  $i^{\text{th}}$  columns in  $U$  and  $H$ , respectively. Let  $u^i = \{u_l^i\}_{l=1}^s$  be the partition of  $u^i$  into elements. By using Eq. (3.1), each element  $u_l^i$  can be presented as the sum  $u_l^i = \frac{1}{\lambda_i} \sum_{j=1}^n M_{lj} \cdot h_j^i$ .

We can use the entries of  $F_M$  as interpolation weights for extending the singular vector  $u^i$  to the  $k^{\text{th}}$  row of  $M$ , where  $s + 1 \leq k \leq n$ . Let  $\tilde{u}^i = \{\tilde{u}_{k-s}^i\}_{k=s+1}^n \in \mathbb{R}^{n-s}$  be a column vector that contains all the approximated entries. Each element  $\tilde{u}_{k-s}^i$  will be calculated as  $\tilde{u}_{k-s}^i = \frac{1}{\lambda_i} \sum_{j=1}^n M_{kj} \cdot h_j^i$ . Therefore, the matrix form of  $\tilde{u}^i$  becomes  $\tilde{u}^i = \frac{1}{\lambda_i} F_M \cdot h^i$ .

Putting together all the  $\tilde{u}^i$ 's as  $\tilde{U} = \begin{bmatrix} \tilde{u}^1 & \tilde{u}^2 & \dots & \tilde{u}^s \end{bmatrix} \in \mathbb{R}^{(n-s) \times s}$ , we get  $\tilde{U} = F_M H \Lambda^{-1}$ .

The basic SVD equation of  $A_M$  can also be written as  $H = A_M^T U \Lambda^{-1}$ . We approximate the right singular vectors of the out-of-sample columns by employing a symmetric argument. We obtain  $\hat{H} = B_M^T U \Lambda^{-1}$ .

The full approximations of the left and right singular vectors of  $\hat{M}$ , denoted by  $\hat{U}$  and  $\hat{H}$ , respectively, are

$$\hat{U} = \begin{bmatrix} U \\ F_M H \Lambda^{-1} \end{bmatrix}, \quad \hat{H} = \begin{bmatrix} H \\ B_M^T U \Lambda^{-1} \end{bmatrix}. \quad (3.2)$$

The explicit ‘‘Nyström’’ form of  $\hat{M}$  becomes

$$\hat{M} = \hat{U}\Lambda\hat{H}^T = \begin{bmatrix} U \\ F_M H \Lambda^{-1} \end{bmatrix} \Lambda \begin{bmatrix} H^T & \Lambda^{-1} U^T B_M \end{bmatrix} = \begin{bmatrix} A_M & B_M \\ F_M & F_M A_M^+ B_M \end{bmatrix} = \begin{bmatrix} A_M \\ F_M \end{bmatrix} A_M^+ \begin{bmatrix} A_M & B_M \end{bmatrix} \quad (3.3)$$

where  $A_M^+$  denotes the pseudo-inverse of  $A_M$ .  $\hat{M}$  does not modify  $A_M, B_M$  and  $F_M$  but approximates  $C_M$  by  $F_M A_M^+ B_M$ . Note that the Nyström matrix form of the SVD is similar to Eq. (2.7), which is the Nyström form of the EVD matrix.

## 4 Decomposition of General Matrices

We will refer to a decomposition of  $M$  given in Eq. (2.1) with the corresponding decomposition into  $A_M, B_M, F_M$  and  $C_M$ .  $\hat{M}$  denotes the approximated Nyström matrix.

This section presents procedures for explicit orthogonalization of the singular-vectors and eigenvectors of  $\hat{M}$ . Starting with  $\hat{M}$  in the form of Eqs. (2.7) and (3.3), we find its canonical SVD and EVD form, respectively. Constructing these representations takes time and space that are linear in the dimensions of  $M$ .

### 4.1 Construction of EVD for $\hat{M}$

Let  $M$  be a square matrix. We will approximate the eigenvalue decomposition of  $\hat{M}$  without explicitly forming  $\hat{M}$ .

We begin with a matrix  $M$  that is partitioned as in Eq. (2.1). By explicitly employing the Nyström method, we construct  $\hat{U}$  and  $\hat{V}$  as defined in Eq. (2.6). Then, we proceed by defining the matrices  $G_U = \hat{U}\Lambda^{1/2}$  and  $G_V = \Lambda^{1/2}\hat{V}$ . We directly compute the EVD of  $G_V G_U$  as  $F\Sigma F^{-1}$ . The eigenvalues of  $\hat{M}$  are given by  $\Sigma$  and the right and left eigenvectors are  $U_o = G_U F \Sigma^{-1/2}$  and  $V_o = \Sigma^{-1/2} F^{-1} G_V$ , respectively.

The left and right eigenvectors are mutually orthogonal since

$$V_o U_o = \Sigma^{-1/2} F^{-1} G_V \cdot G_U F \Sigma^{-1/2} = \Sigma^{-1/2} F^{-1} \cdot F \Sigma F^{-1} \cdot F \Sigma^{-1/2} = I.$$

The EVD form of  $U_o, V_o$  and  $\Sigma$  gives  $\hat{M}$ , as we see from

$$U_o \Sigma V_o = G_U F \Sigma^{-1/2} \cdot \Sigma \cdot \Sigma^{-1/2} F^{-1} G_V = G_U G_V = \hat{U} \Lambda^{1/2} \cdot \Lambda^{1/2} \hat{V} = \hat{M}.$$

These two properties qualify  $U_o \Sigma V_o$  as the EVD of  $\hat{M}$ .

When  $M$  is symmetric, the matrix  $G_V$  is simply  $G_U^T$ . By using the terminology in section 2.2.2, we denote  $G_V = Z$  and the matrix  $G_V G_U$  is transformed into  $Z Z^T$ . From here on the method of solution in section 2.2.2 coincides with the current section. Hence, this form of EVD approximation generalizes the symmetric case.

The computational complexity is  $O(s^2 n)$ , where  $s$  is the sample size (the size of  $A_M$ ) and  $n$  is the size of  $M$ . The computational bottleneck is in the formation of  $G_V G_U$ .

#### 4.1.1 A Single-Step Solution for the EVD for $\hat{M}$

This solution method assumes that  $A_M$  has a square root matrix  $A_M^{1/2}$ . From this assumption, we can modify the algorithm in section 4.1 to construct the EVD of  $\hat{M}$  with fewer steps.

We define the matrices  $G_U$  and  $G_V$  to be

$$G_U = \begin{bmatrix} A_M \\ F_M \end{bmatrix} A_M^{-1/2}, \quad G_V = A_M^{-1/2} \begin{bmatrix} A_M & B_M \end{bmatrix}.$$

We proceed to explicitly compute the eigen-decomposition of  $G_V G_U \in \mathbb{R}^{s \times s}$  as  $G_V G_U = F \Sigma F^{-1}$ . The eigenvalues of  $\hat{M}$  are given by  $\Sigma$  and the right and left eigenvectors of  $\hat{M}$  are formed by  $U_o = G_U F \Sigma^{-1/2}$  and  $V_o = \Sigma^{-1/2} F^{-1} G_V$ , respectively. Again, we can verify the eigenvectors are mutually orthogonal:

$$V_o U_o = \Sigma^{-1/2} F^{-1} G_V \cdot G_U F \Sigma^{-1/2} = \Sigma^{-1/2} F^{-1} \cdot G_V G_U \cdot F \Sigma^{-1/2} = \Sigma^{-1/2} F^{-1} \cdot F \Sigma F^{-1} \cdot F \Sigma^{-1/2} = I,$$

and the matrices  $U_o, V_o$  and  $\Lambda_S$  form  $\hat{M}$  as

$$U_o \Lambda_S V_o = G_U F \Sigma^{-1/2} \cdot \Sigma \cdot \Sigma^{-1/2} F^{-1} G_V = G_U G_V = \begin{bmatrix} A_M \\ F_M \end{bmatrix} A_M^{-1/2} \cdot A_M^{-1/2} \begin{bmatrix} A_M & B_M \end{bmatrix} = \hat{M}.$$

The reduction to the symmetric case is straightforward here as well. We have  $G_V = G_U^T$  when  $M$  is symmetric. By using the terms of section 2.2.3, we have  $G_U^T = G_V = G$ . The expression  $G_V G_U$  turns into  $G^T G$ . After that point the methods of solution coincide.

Again, the algorithm takes  $O(s^2n)$  operations due to the need to calculate  $G_V G_U$ . Compared to the solution given in section 4.1, the single-step solution performs fewer matrix operations. Therefore, it achieves better numerical accuracy.

## 4.2 Construction of SVD for $\hat{M}$

Let  $M$  be a general  $m \times n$  matrix with the decomposition in Eq. (2.1). Given an initial sample  $A_M$ , we present an algorithm that efficiently computes the SVD of  $\hat{M}$  (defined by Eq. (2.7)).

We explicitly compute the SVD of  $A_M$  and use the technique outlined in section 3 to obtain  $\hat{U}$  and  $\hat{H}$  as in Eq. (3.2). We form the matrices  $Z_U = \hat{U}\Lambda^{1/2}$  and  $Z_H = \hat{H}\Lambda^{1/2}$ . We proceed by forming the symmetric  $s \times s$  matrices  $Z_U^T Z_U$  and  $Z_H^T Z_H$  and compute their SVD as  $Z_U^T Z_U = F_U \Sigma_U F_U^T$  and  $Z_H^T Z_H = F_H \Sigma_H F_H^T$ , respectively. The next stage derives an SVD form for the  $s \times s$  matrix  $D = \Sigma_U^{1/2} F_U^T F_H \Sigma_H^{1/2}$ . This is given explicitly by computing  $D = U_D \Lambda_D H_D^T$ . The singular values of  $\hat{M}$  are given in  $\Lambda_D$  and the leading left and right singular vectors of  $\hat{M}$  are  $U_o = Z_U F_U \Sigma_U^{-1/2} U_D$  and  $H_o = Z_H F_H \Sigma_H^{-1/2} H_D$ , respectively. The columns of  $U_o$  and  $H_o$  are orthogonal since

$$U_o^T U_o = U_D^T \Sigma_U^{-1/2} F_U^T Z_U^T \cdot Z_U F_U \Sigma_U^{-1/2} U_D = U_D^T \Sigma_U^{-1/2} F_U^T \cdot F_U \Sigma_U F_U^T \cdot F_U \Sigma_U^{-1/2} U_D = U_D^T U_D = I,$$

$$H_o^T H_o = H_D^T \Sigma_H^{-1/2} F_H^T Z_H^T \cdot Z_H F_H \Sigma_H^{-1/2} H_D = H_D^T \Sigma_H^{-1/2} F_H^T \cdot F_H \Sigma_H F_H^T \cdot F_H \Sigma_H^{-1/2} H_D = H_D^T H_D = I.$$

The SVD of  $\hat{M}$  is formed by using  $U_o, H_o$  and  $V_D$

$$\begin{aligned} U_o \Lambda_D H_o^T &= Z_U F_U \Sigma_U^{-1/2} U_D \cdot \Lambda_D \cdot H_D^T \Sigma_H^{-1/2} F_H^T Z_H^T = Z_U F_U \Sigma_U^{-1/2} \cdot D \cdot \Sigma_H^{-1/2} F_H^T Z_H^T = \\ &= Z_U F_U \Sigma_U^{-1/2} \cdot \Sigma_U^{1/2} F_U^T F_H \Sigma_H^{1/2} \cdot \Sigma_H^{-1/2} F_H^T Z_H^T = Z_U Z_H^T = \hat{U} \Lambda^{1/2} \cdot \Lambda^{1/2} \hat{H}^T = \hat{M}. \end{aligned}$$

When  $M$  is symmetric, this solution method coincides with the method in section 2.2.2. The matrices  $Z_U$  and  $Z_H$  correspond to  $Z$  in section 2.2.2. The matrix  $D$  becomes the diagonal matrix  $\Sigma$  of the symmetric case. The computational complexity of the procedure is  $O(s^2(m+n))$ . The bottleneck is the computation of  $Z_U^T Z_U$  and  $Z_H^T Z_H$ .

### 4.2.1 A Single-Step Solution for the SVD of $\hat{M}$

This solution method assumes that  $A_M$  has a square root matrix  $A_M^{1/2}$ . Similar to section 4.1.1, this assumption allows us to modify the algorithm of the general case to achieve the

same result in fewer steps.

Let  $A_M^{-1/2}$  be the pseudo-inverse of the square root matrix of  $A_M$ . We begin by forming the matrices  $G_U$  and  $G_H$  such that

$$G_U = \begin{bmatrix} A_M \\ F_M \end{bmatrix} A_M^{-1/2}, \quad G_H = \left( A_M^{-1/2} \begin{bmatrix} A_M & B_M \end{bmatrix} \right)^T.$$

The symmetric matrices  $G_U^T G_U$  and  $G_H^T G_H$  are diagonalized by  $G_U^T G_U = F_U \Sigma_U F_U^T$  and  $G_H^T G_H = F_H \Sigma_H F_H^T$ . From these parts we form  $D = \Sigma_U^{1/2} F_U^T F_H \Sigma_H^{1/2}$  which is explicitly diagonalized as  $D = U_D \Lambda_D H_D^T$ . The singular values of  $\hat{M}$  are given by  $\Lambda_D$  and the left and right singular vectors are given by  $U_o = G_U F_U \Sigma_U^{-1/2} U_D$  and  $H_o = G_H F_H \Sigma_H^{-1/2} H_D$ , respectively.

As in section 4.2, we can verify the identities that make this decomposition a valid SVD. The singular vectors are orthogonal:

$$U_o^T U_o = U_D^T \Sigma_U^{-1/2} F_U^T G_U^T \cdot G_U F_U \Sigma_U^{-1/2} U_D = U_D^T \Sigma_U^{-1/2} F_U^T \cdot F_U \Sigma_U F_U^T \cdot F_U \Sigma_U^{-1/2} U_D = U_D^T U_D = I,$$

$$H_o^T H_o = H_D^T \Sigma_H^{-1/2} F_H^T G_H^T \cdot G_H F_H \Sigma_H^{-1/2} H_D = H_D^T \Sigma_H^{-1/2} F_H^T \cdot F_H \Sigma_H F_H^T \cdot F_H \Sigma_H^{-1/2} H_D = H_D^T H_D = I.$$

The SVD is formed by  $U_o, H_o$  and  $\Lambda_D$ :

$$\begin{aligned} U_o \Lambda_D H_o^T &= G_U F_U \Sigma_U^{-1/2} U_D \cdot \Lambda_D \cdot H_D^T \Sigma_H^{-1/2} F_H^T G_H^T = G_U F_U \Sigma_U^{-1/2} \cdot D \cdot \Sigma_H^{-1/2} F_H^T G_H^T = \\ &= G_U F_U \Sigma_U^{-1/2} \cdot \Sigma_U^{1/2} F_U^T F_H \Sigma_H^{1/2} \cdot \Sigma_H^{-1/2} F_H^T G_H^T = G_U G_H^T = \begin{bmatrix} A_M \\ F_M \end{bmatrix} A_M^{-1/2} \cdot A_M^{-1/2} \begin{bmatrix} A_M & B_M \end{bmatrix} = \hat{M}. \end{aligned}$$

If  $M$  is symmetric, this method reduces to the single-step solution described in section 2.2.3.

The matrices  $G_U$  and  $G_H$  correspond to  $G$  in the symmetric case. The matrix  $D$  becomes  $\Lambda_S$ .

The computational complexity of the procedure remains  $O(s^2(m+n))$ . The computational bottleneck of the algorithm is in the formation of  $G_U^T G_U$ .

### 4.3 Prerequisite for the Single-Step method

The single-step methods, described in sections 2.2.3, 4.1.1 and 4.2.1, require that  $A_M$  have a square root matrix.

When a matrix is positive semi-definite, a square root can be found via the Cholesky factorization algorithm ([5] chapter 4.2.3). But positive-definiteness is not a necessary prerequisite. For example, the square root of a diagonalizable matrix can be found via its diagonalization. If  $A_M = U\Lambda U^{-1}$ , then,  $A_M^{1/2} = U\Lambda^{1/2}U^{-1}$ . In this case, the matrix does not need to be invertible.

It can be shown that under a complex realm, every non-singular matrix has a square root. An algorithm for calculating the square root for a given non-singular matrix is given in [15]. This suggests a way of assuring the existence of a square root matrix. We can make  $A_M$  non-singular, or equivalently, a full rank matrix.

The rank of  $A_M$  will also have a role in bounding the approximation error of the Nyström procedure. This will be elaborated in section 5.4.

## 5 Choice of Sub-Sample

The choice of initial sample for performing the Nyström extension is an important part in the approximation procedure. The sample matrix  $A_M$  is determined by permutation of the rows and columns of  $M$  (as given in Eq. (2.1)). Our goal is to choose a (possibly constrained) permutation of  $M$  such that the resulting matrix can be approximated more accurately by the Nyström method. Here accuracy is measured by  $L_2$  distance between the pivoted version of  $M$  and the Nyström approximated version. This notion is made precise in section 5.4.

We allow for complete pivoting in the choice of a permutation for  $M$ . This means that both columns and rows can be independently permuted. This kind of pivoting does not generally preserve the eigenvalues and eigenvectors of the matrix. However, the singular values of the matrix remain unchanged and the singular vectors are permuted. Formally, let  $E_r$  and  $E_c$  be the row and column permutation matrices, respectively. Using the SVD of  $M$ , the pivoted matrix is decomposed as  $E_r M E_c = E_r U \Sigma V^T E_c = (E_r U) \Sigma (V^T E_c)$ . Row and column permutations leave  $U$  and  $V^T$  unitary. Therefore  $(E_r U) \Sigma (V^T E_c)$  is the SVD of  $E_r M E_c$ . The singular vectors of  $M$  can be easily regenerated by permuting the left and right singular vectors of  $E_r M E_c$  by  $E_r^{-1}$  and  $E_c^{-1}$  respectively.

Section 5.4 shows the choice of  $A_M$  determines the Nyström approximation error. Hence, the problem of choosing a sample is equivalent to choosing the rows and columns of  $M$  whose

intersection forms  $A_M$ . Therefore, it makes sense to use the size  $s$  of  $A_M$  as our sample size. This size largely determines the time and space complexity of the presented approximation procedures. The complexities are  $O(s^2(m+n))$  and  $O(s(m+n))$ , respectively.

## 5.1 Related Work on Sub-Sample Selection

Previous works on sub-sample selection focused on kernel matrices. These were done for symmetric matrices where the entries represent affinities. In these settings, we can use a single permutation for the columns and rows without changing the original meaning of the matrix. This pivoting variant is called symmetric pivoting. Sample selection algorithms for kernel matrices try to find a permutation matrix  $E_p$  such that  $E_p^T M E_p$  is most accurately approximated by the Nyström method.

The simplest sample selection method is based on random sampling. It works well for dense image data ([2]). Random sampling is also used in [27] while employing a greedy criterion that helps to determine the quality of the sample. A different greedy approach for sample selection is used in [25], where a new point is added to the sample based on its distance from a constrained linear combination of previously selected points.

In [26], the  $k$ -means clustering algorithm is used for selecting the sub-sample. The  $k$ -means cluster centers are shown to minimize an error criterion related to the Nyström approximation error. Finally, Incomplete Cholesky Decomposition (ICD) ([12]) employs the pivoted Choleksy algorithm and uses a greedy stopping criterion to determine the required sample size for a given approximation accuracy.

The Cholesky decomposition of a matrix factors it into  $Z^T Z$ , where  $Z$  is an upper triangular matrix. Initially,  $Z = 0$ . The ICD algorithm applies the Cholesky decomposition to  $M$  while symmetrically pivoting the columns and rows of  $M$  according to a greedy criterion. The algorithm has an outer loop that scans the columns of  $M$  according to a pivoting order. The results for each column determine the next column to scan. This loop is terminated early after  $s$  columns were scanned by using a heuristic on the trace of the residual  $Z^T Z - M$ . This algorithm ([12]) approximates  $M$ . This is equivalent to a Nyström approximation where the initial sample is taken as the intersection of the pivoted columns and rows.

When  $M$  is a Gram matrix, it can be expressed as the product of two matrices. Let  $M$  be

decomposed into  $M = X^T X$  where  $X \in \mathbb{R}^{n \times n}$ . The special properties of  $M$  were exploited differently in [13]. Specifically, the fact that  $M_{ii}$  is the norm of the column  $X_i$  is used. A non-Gram matrix requires  $O(n^2)$  additional operations to compute  $X_i^T X_i$ , which is impractical for large matrices. Once the norms of the columns in  $X$  are known, a method similar to [6] is used to choose a good column sample from  $X$ . The intersection in  $M$  of the pivoted columns and the corresponding rows is a good choice for  $A_M$ . The Nyström procedure is then performed similarly to what was described in section 2.2.2. The runtime complexity of the algorithm in [13] is  $O(n)$ .

## 5.2 Preliminaries

**Definition 5.1. Approximate ‘thin’ Matrix Decomposition.** *Given a matrix  $M \in \mathbb{R}^{m \times n}$ . A “thin” matrix decomposition is an approximation of the form  $M = GS$  where  $G \in \mathbb{R}^{m \times k}$ ,  $S \in \mathbb{R}^{k \times n}$  and  $k \leq \min(m, n)$ .*

This form effectively approximates  $M$  using a rank- $k$  matrix product. A good example for such an approximation is the truncated rank- $k$  SVD. It approximates a  $m \times n$  matrix as  $U\Lambda V^T$ , where  $U \in \mathbb{R}^{m \times k}$ ,  $\Lambda \in \mathbb{R}^{k \times k}$  and  $V \in \mathbb{R}^{n \times k}$ . When this decomposition is employed, we can choose, for example,  $G = U, S = \Lambda V^T$ . Many algorithms ([6, 9, 10, 11]) exist for approximating the rank- $k$  SVD with a runtime close to  $O(mn)$ .

Truncated SVD is a popular choice, but it is by no means the only one. Other examples include truncated pivoted QR ([7]) or the interpolative decomposition (ID) as outlined in [8].

**Definition 5.2. Numerical Rank.** *A matrix  $A$  has numerical rank  $r$  with respect to a threshold  $\epsilon$  if  $\sigma_{r+1}(A)$  is the first singular value such that*

$$\frac{\sigma_1(A)}{\sigma_{r+1}(A)} > \epsilon.$$

This definition generalizes the  $L_2$  condition number ( $\kappa_2(A)$ ), since it also applies to non-invertible and non-square matrices.

**Definition 5.3. Rank Revealing QR Decomposition (RRQR).** *Let  $A \in \mathbb{R}^{m \times n}$  be a matrix and let  $k$  be a user defined threshold. A RRQR algorithm finds a permutation*

matrix  $E$  such that  $AE$  has a QR decomposition with special properties. Formally, we write  $AE = QR$  such that  $Q$  is an orthogonal matrix and  $R$  is upper triangular. Let  $R$  have the following decomposition:

$$R = \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix} \quad (5.1)$$

where  $R_{11} \in \mathbb{R}^{k \times k}$ ,  $R_{12} \in \mathbb{R}^{k \times (n-k)}$  and  $R_{22} \in \mathbb{R}^{(n-k) \times (n-k)}$ . Let  $p(k, n)$  be a fixed non-negative function bounded by a low degree polynomial in  $k$  and  $n$ . A RRQR algorithm tries to permute the columns of  $A$  such that

$$\sigma_k(R_{11}) \geq \frac{\sigma_k(A)}{p(k, n)}, \quad \sigma_1(R_{22}) \leq \sigma_{k+1}(A) \cdot p(k, n).$$

An overview on this topic is given in [3].

The relation between  $A$  and  $R$  can shed some light on the rank-revealing properties of RRQR. Let  $AE = \begin{bmatrix} A_1 & A_2 \end{bmatrix}$  be a partitioning of  $AE$  such that  $A_1$  contains the first  $k$  columns. The RRQR decomposition is rank-revealing in the sense that it tries to put a set of  $k$  maximally independent columns of  $A$  into  $A_1$ . We formalize this statement with Lemma 5.4.

**Lemma 5.4.** *Assume that the RRQR algorithm found a pivoting of  $A$  such that  $\sigma_k(R_{11}) \geq \sigma_k(A)/\beta$ , where  $\beta \geq 1$ . If  $A$  has numerical rank of at least  $k$  with respect to the threshold  $\epsilon$ , then, the numerical rank of  $A_1$  (the first  $k$  columns of  $AE$ ) is  $k$  with respect to the threshold  $\beta \cdot \epsilon$ .*

*Proof.* The RRQR algorithm yields  $A_1 = Q \begin{bmatrix} R_{11} & 0 \end{bmatrix}^T$ . Since  $Q$  is orthogonal, it does not modify singular values. Therefore, we have  $\sigma_k(A_1) = \sigma_k \begin{bmatrix} R_{11} & 0 \end{bmatrix}^T = \sigma_k(R_{11})$ . By combining the above with our assumption on the RRQR algorithm, we get

$$\beta \cdot \sigma_k(A_1) \geq \sigma_k(A). \quad (5.2)$$

The interlacing property of singular values (Corollary 8.6.3 in [5]) gives us

$$\sigma_1(A) \geq \sigma_1(A_1). \quad (5.3)$$

By employing definition 5.2 for  $A$  and incorporating Eqs. (5.2) and (5.3), we get

$$\epsilon \geq \frac{\sigma_1(A)}{\sigma_k(A)} \geq \frac{\sigma_1(A_1)}{\sigma_k(A)} \geq \frac{\sigma_1(A_1)}{\beta \cdot \sigma_k(A_1)}.$$

By rearranging terms, we get

$$\frac{\sigma_1(A_1)}{\sigma_k(A_1)} \leq \beta \cdot \epsilon.$$

Therefore the numerical rank of  $A_1$  is at least  $k$  with respect to the threshold  $\beta \cdot \epsilon$ . Since  $A_1$  has only  $k$  columns, it has precisely this rank.  $\square$

### 5.3 Algorithm Description and Rationale

Initially, our algorithm decomposes the matrix  $M$  into  $G \cdot S$ . Then, a RRQR algorithm chooses the  $s$  *most* non-singular columns of  $G^T$  and  $S$  and insert them into  $G_A^T$  and  $S_A$ , respectively. We use a variant of RRQR that measures non-singularity according to the magnitude of the last singular value (see the proof of Corollary 5.8). The non-singularity of  $G_A$  and  $S_A$  will bound the non-singularity of  $G_A S_A$  (see Eq. (5.7)).

On a higher level observation, the algorithm will try to perform an exhaustive search for the  $s \times s$  most non-singular square in  $GS$ . However, since  $GS$  approximates  $M$ , choosing  $A_M$  from the same rows and columns of  $M$  amounts to choosing one of its most non-singular squares. These notions are formalized in Theorem 5.6.

The magnitude of the last singular-value in  $A_M$ , denoted by  $\sigma_s(A_M)$ , will be used as a measure for the singularity of  $A_M$ . This quantity is instrumental in defining the bound of the approximation error given in Theorem 5.12. We show in the experimental results section (section 7) that empirically,  $\sigma_s(A_M)$  is strongly related to the approximation error of the Nyström procedure.

### 5.4 Analysis of Nyström Error

Let  $M$  be a matrix with the decomposition given by Eq. (2.1). This partitioning corresponds to sampling  $s$  columns and rows from  $M$  to form the matrix  $A_M$ . Our error analysis depends on an approximate decomposition of  $M$  into a product of two ‘thin’ matrices. Let  $M \simeq GS$  be a decomposition of  $M$  where  $G \in \mathbb{R}^{m \times s}$  and  $S \in \mathbb{R}^{s \times n}$ . The approximation error of  $M$  by  $GS$  is denoted by  $e_s$ . Formally,  $\|M - GS\|_2 \leq e_s$ . Let  $G = \begin{bmatrix} G_A & G_B \end{bmatrix}^T$  be a row partitioning of  $G$  where  $G_A \in \mathbb{R}^{s \times r}$  and  $G_B \in \mathbb{R}^{(m-s) \times r}$ . Let  $S = \begin{bmatrix} S_A & S_B \end{bmatrix}$  be a column partitioning of  $S$  where  $S_A \in \mathbb{R}^{r \times s}$ ,  $S_B \in \mathbb{R}^{r \times (n-s)}$ . This notation yields the following forms

for the sub-matrices of  $M$ :

$$A_M \simeq G_A S_A, \quad B_M \simeq G_A S_B, \quad F_M \simeq G_B S_A, \quad C_M \simeq G_B S_B. \quad (5.4)$$

where  $A_M, B_M, F_M$  and  $C_M$  were defined in Eq. 2.1.

**Lemma 5.5.** *(based on Corollary 8.6.2 in [5]) If  $A$  and  $A + E$  are in  $\mathbb{R}^{m \times n}$  then for  $k \leq \min(m, n)$  we have  $|\sigma_k(A + E) - \sigma_k(A)| \leq \sigma_1(E) = \|E\|_2$ .*

*Proof.* Corollary 8.6.2 in [5] states the same lemma with the requirement  $m \geq n$ . If  $m < n$ , we can use the original version of the lemma to get  $|\sigma_k(A^T + E^T) - \sigma_k(A^T)| \leq \|E^T\|_2$ . Transposition neither modifies the singular values nor the norm of a matrix.  $\square$

**Theorem 5.6.** *Assuming that*

1.  $\sigma_s(GS) > 0$ ;

*(This means that  $GS$  is of rank at least  $s$ . Otherwise, a non-singular  $A_M$  cannot be found)*

2.  $\sigma_s(G) \sigma_s(S) = \sigma_s(GS) / \gamma$  for some constant  $\gamma \geq 1$ ;

*(It will allow us to use the non-singularity of  $G_A$  and  $S_A$  as a bound for the non-singularity of  $G_A S_A$ . This demands the initial decomposition to be reasonably well conditioned. See Corollary 5.7 for details)*

3.  $\sigma_s(G_A) \geq \sigma_s(G) / \beta$  and  $\sigma_s(S_A) \geq \sigma_s(S) / \beta$  for some constant  $\beta \geq 1$ ;

*(This will allow us to use  $\sigma_s(A_M)$  as a bound for  $\sigma_s(GS)$ . The RRQR algorithm will fulfill this assumption in its choice of  $G_A$  and  $S_A$ )*

4.  $e_s < (\sigma_s(M) - e_s) / \beta^2 \gamma$ , where  $e_s$  is the error given by the rank- $s$  approximation of  $M$  by  $GS$ .

*(The initial rank- $s$  approximation should be good enough)*

Then,  $A_M$  is non-singular.

*Proof.* Lemma 5.5 yields  $|\sigma_s(M) - \sigma_s(G_S)| \leq \|M - GS\|_2 = e_s$ , or

$$\sigma_s(M) - e_s \leq \sigma_s(G_S). \quad (5.5)$$

From assumptions 2 and 3 we obtain

$$\sigma_s(G_S) / \beta^2 \gamma \leq \sigma_s(G) \sigma_s(S) / \beta^2 \leq \sigma_s(G_A) \sigma_s(S_A). \quad (5.6)$$

$G_A$  and  $S_A$  are  $s \times s$  matrices. Assumptions 1, 2 and 3 show that  $\sigma_s(G_A)$  and  $\sigma_s(S_A)$  are non-zero. Thus,  $G_A$  and  $S_A$  are non-singular and we obtain

$$\sigma_s(G_A) \sigma_s(S_A) = \frac{1}{\|G_A^{-1}\| \|S_A^{-1}\|} \leq \frac{1}{\|S_A^{-1} G_A^{-1}\|} = \frac{1}{\|(G_A S_A)^{-1}\|} = \sigma_s(G_A S_A). \quad (5.7)$$

By combining Eqs. (5.5), (5.6) and (5.7) we get

$$(\sigma_s(M) - e_s) / \beta^2 \gamma \leq \sigma_s(G_A S_A). \quad (5.8)$$

$A_M$  and  $G_A S_A$  are the top left  $s \times s$  corners of  $M$  and  $GS$ , respectively. Hence, we can write  $\|A_M - G_A S_A\|_2 \leq \|M - GS\|_2 = e_s$ . By combining this expression with Eq. (5.8) and using assumption 4, we have  $\|A_M - G_A S_A\|_2 \leq \sigma_s(G_A S_A)$ . Equivalently,

$$\frac{\|A_M - G_A S_A\|_2}{\|G_A S_A\|_2} < \frac{1}{\kappa(G_A S_A)}. \quad (5.9)$$

The matrix  $G_A S_A$  is non-singular since it is the product of the non-singular matrices  $G_A$  and  $S_A$ . Equation 2.7.6 in [5] states that for any matrix  $A$  and perturbation matrix  $\Delta A$  we have

$$\frac{1}{\kappa_2(A)} = \min_{A + \Delta A \text{ singular}} \frac{\|\Delta A\|_2}{\|A\|_2}.$$

This equation in effect gauges the minimal  $L_2$  distance from  $A$  to a singular matrix. By setting  $G_A S_A = A$  in Eq. (5.9) we conclude that  $A_M$  is non-singular.  $\square$

Assumption 2 can be verified for different types of rank- $s$  approximations of  $M$ . For the approximated SVD we have Corollary 5.7.

**Corollary 5.7.** *When the approximated SVD is used to form  $GS$ , we have  $\gamma = 1$  (where  $\gamma$  is defined by assumption 2 in Theorem 5.6).*

*Proof.* Let  $M \simeq U\Sigma V^T$  be the approximated SVD of  $M$ . We can choose  $G = U\Sigma$  and  $S = V^T$ . From the properties of the SVD, we have  $\sigma_s(G) = \sigma_s(U\Sigma) = \Sigma_{ss} = \sigma_s(GS)$  and  $\sigma_s(S) = 1$ . It follows that  $\sigma_s(G)\sigma_s(S) = \sigma_s(GS)$ .  $\square$

Similarly, the  $\beta$  in assumption 3 depends on the algorithm that is used to pick  $G_A$  and  $S_A$  from within  $G$  and  $S$ , respectively. When a state-of-the-art RRQR algorithm is used, we derive Corollary 5.8.

**Corollary 5.8.** *When the RRQR version given in Algorithm 1 in [4] is used to choose  $G_A$  and  $S_A$ , we have  $\beta \leq \sqrt{s(\min(m, n) - s) + 1}$ , where  $\beta$  is defined by assumption 3 in Theorem 5.6.*

*Proof.* Let  $A \in \mathbb{R}^{n \times k}$  be a matrix where  $k \leq n$  and let  $A = \begin{bmatrix} A_1 & A_2 \end{bmatrix}$  be a partition of  $A$  where  $A_1 \in \mathbb{R}^{k \times k}$ . The concept of local  $\mu$ -maximum volume was used in [4] to find a pivoting scheme such that  $\sigma_{\min}(A_1)$  is bounded from below. Formally, Lemma 3.5 in [4] states that when  $A_1$  is a local  $\mu$ -maximum volume in  $A$ , we have  $\sigma_{\min}(A_1) \geq \sigma_k(A) / \sqrt{k(n-k)\mu^2 + 1}$ .  $\mu$  is a user-controlled parameter that has negligible effect in this bound. For instance, [4] suggests setting  $\mu = 1 + \mathbf{u}$ , where  $\mathbf{u}$  is the machine precision. Therefore, we omit  $\mu$  in subsequent references of this bound.

Algorithm 1 in [4] describes how a local  $\mu$ -maximum volume can be found for a given matrix  $A$ . This algorithm can be applied to the choice of  $G_A$  and  $S_A^T$  from the rows of  $G$  and  $S^T$ , respectively. It follows from Lemma 3.5 in [4] that  $\sigma_s(G_A) \geq \sigma_s(G) / \sqrt{s(m-s) + 1}$  and  $\sigma_s(S_A) = \sigma_s(S_A^T) \geq \sigma_s(S^T) / \sqrt{s(n-s) + 1} = \sigma_s(S) / \sqrt{s(n-s) + 1}$ . The definition of  $\beta$  yields the required expression.  $\square$

Later the RRQR algorithm will be used to select  $G_A^T$  and  $S_A$  as columns from  $G^T$  and  $S$ , respectively. This is equivalent to choosing rows from  $G$  and  $S^T$ . The latter form was used for compatibility with the notation of [4].

Theorem 5.6 states that if our rank- $s$  approximation of  $M$  is sufficiently accurate and our RRQR algorithm managed to pick  $s$  non-singular columns from  $G^T$  and  $S$ , then our sample matrix  $A_M$  is non-singular.

We bring a few definitions in order to bound the error of the Nyström approximation procedure. We will decompose the matrix  $M$  into a sum of two matrices:  $M_{lg}$  that contains the en-

ergy of the top  $s$  singular values and  $M_{sm}$  that contains the residual. If  $M_{lg}$  and  $M_{sm}$  are given in SVD outer product form, then we have  $M_{lg} = \sum_{i=1}^s \sigma_i u_i v_i$  and  $M_{sm} = \sum_{i=s+1}^{\min(m,n)} \sigma_i u_i v_i$ , respectively. Based on this decomposition, we define the following decompositions of  $M_{lg}$  and  $M_{sm}$ :

$$M = M_{lg} + M_{sm} = \begin{bmatrix} A_M & B_M \\ F_M & C_M \end{bmatrix} = \begin{bmatrix} A_{lg} & B_{lg} \\ F_{lg} & C_{lg} \end{bmatrix} + \begin{bmatrix} A_{sm} & B_{sm} \\ F_{sm} & C_{sm} \end{bmatrix}. \quad (5.10)$$

**Lemma 5.9.** *If all the assumptions of Theorem 5.6 hold and if we have*

$$\sigma_{s+1}(M) < \frac{\sigma_s(M) - e_s}{\beta^2 \gamma} - e_s \quad (5.11)$$

(where  $e_s$  is defined by assumption 4 in Theorem 5.6), then  $A_{lg}$  is non-singular.

*Proof.* We employ Lemma 5.5 to bound  $|\sigma_s(A_M) - \sigma_s(G_A S_A)|$ . Formally, we have

$$|\sigma_s(A_M) - \sigma_s(G_A S_A)| \leq \|A_M - G_A S_A\|_2 \leq \|M - G_S\|_2 = e_s.$$

By rearranging terms, we obtain  $\sigma_s(G_A S_A) - e_s \leq \sigma_s(A_M)$ . Combining this expression with Eq. (5.8) from the proof of Theorem 5.6 yields

$$\frac{\sigma_s(M) - e_s}{\beta^2 \gamma} - e_s \leq \sigma_s(A_M). \quad (5.12)$$

The quantity  $\|A_M - A_{lg}\|_2$  can be bounded by  $\|A_M - A_{lg}\|_2 \leq \|M - M_{lg}\|_2 = \sigma_{s+1}(M)$ . Combining the above with Eqs. (5.11) and (5.12) yields

$$\|A_M - A_{lg}\|_2 \leq \sigma_{s+1}(M) < \frac{\sigma_s(M) - e_s}{\beta^2 \gamma} - e_s \leq \sigma_s(A_M).$$

The terms are rearranged to get

$$\|A_M - A_{lg}\|_2 / \|A_M\|_2 < 1/\kappa(A_M), \quad (5.13)$$

where  $\kappa$  is the standard  $L_2$ -norm condition number. This expression is similar to Eq. (5.9) in the proof of Theorem 5.6. As before, if  $A_M$  is non-singular, then Eq. (5.13) implies that  $A_{lg}$  is non-singular.  $\square$

We define the rank- $s$  approximation of  $M$  that is based on the truncated SVD form of  $M_{lg}$ . Let  $M_{lg} = U_s \Sigma_s V_s^T$  be the truncated SVD of  $M$ . Denote  $X = U_s \Sigma_s$  and  $Y = V_s^T$  such that  $M_{lg} = XY$ . We define  $X = \begin{bmatrix} X_A & X_B \end{bmatrix}^T$  and  $Y = \begin{bmatrix} Y_A & Y_B \end{bmatrix}$  where  $X_A, Y_A \in \mathbb{R}^{s \times s}$ . We get the following forms for the components of  $M_{lg}$ :  $A_{lg} = X_A Y_A$ ,  $B_{lg} = X_A Y_B$ ,  $F_{lg} = X_B Y_A$  and  $C_{lg} = X_B Y_B$ .

The Nyström approximation error can now be formulated.

**Lemma 5.10.** *Assume that  $A_M$  and  $A_{lg}$  are non-singular. Then, the error of the Nyström approximation procedure is bounded by*

$$\frac{\sigma_{s+1}(M)}{\sigma_s(A_M)} \left( \frac{\sigma_1(M)^2}{\sigma_s(A_{lg})} + 2\sigma_1(M) + \sigma_{s+1}(M) \right). \quad (5.14)$$

*Proof.* As seen from Eq. (3.3), the matrices  $A_M, B_M$  and  $F_M$  are not modified by the Nyström extension.  $C_M$  is approximated as  $F_M A_M^+ B$ . Assuming that  $A$  is non-singular, then  $F_M A_M^+ B$  is equivalent to  $F_M A_M^{-1} B$ . The latter can be decomposed using the partitioning in Eq. (5.10):

$$\begin{aligned} F_M A_M^{-1} B &= (F_{lg} + F_{sm}) A_M^{-1} (B_{lg} + B_{sm}) = \\ &= F_{lg} A^{-1} B_{lg} + F_{lg} A^{-1} B_{sm} + F_{sm} A^{-1} B_{lg} + F_{sm} A^{-1} B_{sm}. \end{aligned} \quad (5.15)$$

Since  $A_M$  and  $A_{lg}$  are non-singular, we have  $A_M^{-1} - A_{lg}^{-1} = -A_{lg}^{-1} (A - A_{lg}) A_M^{-1} = -A_{lg}^{-1} A_{sm} A_M^{-1}$ .

The first term of Eq. (5.15) can be written as

$$F_{lg} A^{-1} B_{lg} = F_{lg} (A_{lg}^{-1} - A_{lg}^{-1} A_{sm} A_M^{-1}) B_{lg} = F_{lg} A_{lg}^{-1} B_{lg} - F_{lg} A_{lg}^{-1} A_{sm} A_M^{-1} B_{lg}. \quad (5.16)$$

By our assumption, the matrices  $X_A$  and  $Y_A$  are non-singular since  $A_{lg} = X_A Y_A$  is non-singular. The first term of Eq. (5.16) becomes:

$$F_{lg} A_{lg}^{-1} B_{lg} = X_B Y_A (X_A Y_A)^{-1} X_A Y_B = X_B Y_A Y_A^{-1} X_A^{-1} X_A Y_B = X_B Y_B = C_{lg}.$$

This means that  $F_{lg} A_{lg}^{-1} B_{lg}$  is the best rank- $s$  approximation to  $C_M$ , as given by the truncated SVD of  $M$ . We can bound the error by collecting all the other terms in Eqs. (5.15) and (5.16):

$$E_{nys} = -F_{lg} A_{lg}^{-1} A_{sm} A_M^{-1} B_{lg} + F_{lg} A^{-1} B_{sm} + F_{sm} A^{-1} B_{lg} + F_{sm} A^{-1} B_{sm}.$$

By the definition of  $M_{sm}$  in Eq. (5.10), we have  $\|M_{sm}\|_2 \leq \sigma_{s+1}(M)$ . Therefore, we can bound  $\|A_{sm}\|_2$ ,  $\|B_{sm}\|_2$  and  $\|F_{sm}\|_2$  by  $\sigma_{s+1}(M)$ . Similarly,  $\|B_{lg}\|_2$  and  $\|F_{lg}\|_2$  are bounded by  $\sigma_1(M)$ . The overall bound on  $\|E_{nys}\|_2$  is

$$\begin{aligned} \|E_{nys}\|_2 &= \left\| -F_{lg}A_{lg}^{-1}A_{sm}A_M^{-1}B_{lg} + F_{lg}A^{-1}B_{sm} + F_{sm}A^{-1}B_{lg} + F_{sm}A^{-1}B_{sm} \right\|_2 \leq \\ &\left\| F_{lg}A_{lg}^{-1}A_{sm}A_M^{-1}B_{lg} \right\|_2 + \|F_{lg}A^{-1}B_{sm}\|_2 + \|F_{sm}A^{-1}B_{lg}\|_2 + \|F_{sm}A^{-1}B_{sm}\|_2 \leq \\ &\frac{\sigma_1(M)^2\sigma_{s+1}(M)}{\sigma_s(A_M)\sigma_s(A_{lg})} + \frac{\sigma_1(M)\sigma_{s+1}(M)}{\sigma_s(A_M)} + \frac{\sigma_1(M)\sigma_{s+1}(M)}{\sigma_s(A_M)} + \frac{\sigma_{s+1}(M)^2}{\sigma_s(A_M)} = \\ &\frac{\sigma_{s+1}(M)}{\sigma_s(A_M)} \left( \frac{\sigma_1(M)^2}{\sigma_s(A_{lg})} + 2\sigma_1(M) + \sigma_{s+1}(M) \right). \end{aligned}$$

□

Corollary 5.11 is derived straightforwardly:

**Corollary 5.11.** *If  $A_M$  is non-singular and the matrix  $M$  is rank- $s$ , then, the Nyström extension approximates  $M$  perfectly.*

*Proof.* If  $M$  is rank- $s$  then  $A_{lg} = A_M$  and the conditions in Lemma 5.10 hold. We obtain the result by setting  $\sigma_{s+1}(M) = 0$  in Eq. (5.14). □

We proceed to express the Nyström approximation error in relation to the parameters  $\beta, \gamma$  and  $e_s$ , as defined by the assumptions in Theorem 5.6.

**Theorem 5.12.** *Assume that the assumptions of Theorem 5.6 hold as well as the assumptions of Lemmas 5.9 and 5.10. The error term of the Nyström procedure is bounded by:*

$$\frac{\sigma_{s+1}(M)\beta^2\gamma}{\sigma_s(M) - (1 + \beta^2\gamma)e_s} \left( \frac{\sigma_1(M)^2\beta^2\gamma}{\sigma_s(M) - (1 + \beta^2\gamma)e_s - \sigma_{s+1}(M)\beta^2\gamma} + 2\sigma_1(M) + \sigma_{s+1}(M) \right). \quad (5.17)$$

*Proof.* We use Lemma 5.5 to obtain:

$$|\sigma_s(A_M) - \sigma_s(A_{lg})| \leq \|A_M - A_{lg}\|_2 \leq \|M - M_{lg}\|_2 = \sigma_{s+1}(M).$$

Equivalently,  $\sigma_s(A_M) - \sigma_{s+1}(M) \leq \sigma_s(A_{lg})$ . We substitute  $\sigma_s(A_M)$  with the left side of Eq. (5.12) to get

$$\frac{\sigma_s(M) - e_s}{\beta^2\gamma} - e_s - \sigma_{s+1}(M) \leq \sigma_s(A_{lg}). \quad (5.18)$$

The result follows when the expressions for  $\sigma_s(A_M)$  and  $\sigma_s(A_{lg})$  in Eq. (5.14) are replaced with the left sides of Eqs. (5.12) and (5.18), respectively. □

When  $A_M$  is non-singular, the eigengap in the  $s^{\text{th}}$  singular value governs the approximation error. This can be seen from Eq. (5.17), where the eigengap appears in the expression  $\frac{\sigma_{s+1}(M)\beta^2\gamma}{\sigma_s(M)-(1+\beta^2\gamma)e_s}$ . Theorem 5.12 bounds the general case. Corollary 5.11 shows what happens in the limit case when the eigengap is infinite.

## 6 Sample Selection Algorithm

Our algorithm is based on Theorem 5.6 and Corollaries 5.7 and 5.8. It receives as its input a matrix  $M \in \mathbb{R}^{m \times n}$  and a parameter  $s$  that determines the sample size. It returns  $A_M$  - a “good” sub-sample of  $M$ . If the algorithm succeeds, we can use Theorem 5.12 to bound the approximation error. The algorithm is described in Algorithm 1.

---

### Algorithm 1 ( $M, s$ )

---

1. Form a rank- $s$  decomposition of  $M$ . Formally  $M \simeq GS$ , where  $G \in \mathbb{R}^{m \times s}$  and  $S \in \mathbb{R}^{s \times n}$ .
  2. Apply the RRQR algorithm to  $G^T$  to find a column pivoting matrix  $E_G$  such that  $\begin{bmatrix} G_A^T & G_B^T \end{bmatrix} = G^T E_G = Q_G R_G$ , where  $G_A \in \mathbb{R}^{s \times s}$  and  $G_B \in \mathbb{R}^{s \times m-s}$ . Let  $I_s$  be the group of indices in  $M$  that correspond to the first  $s$  columns of  $E_G$ .
  3. Apply the RRQR algorithm to  $S$  to find a column pivoting matrix  $E_S$  such that  $\begin{bmatrix} S_A & S_B \end{bmatrix} = S E_S = Q_S R_S$ , where  $S_A \in \mathbb{R}^{s \times s}$  and  $S_B \in \mathbb{R}^{s \times n-s}$ . Let  $J_s$  be the group of indices in  $M$  that correspond to the first  $s$  columns of  $E_S$ .
  4. **if**  $\text{rank}(G_A) \neq s$  or  $\text{rank}(S_A) \neq s$  **then**  
     return “Algorithm failed. Please pick a different value for  $s$ .”  
   **end if**
  5. Form the matrix  $A_M \in \mathbb{R}^{s \times s}$  such that  $A_M = [M_{ij}]_{i \in I_s, j \in J_s}$ . Returns  $A_M$  as the sub-sample matrix.
-

## 6.1 Algorithm Complexity Analysis

Step 1 is the computational bottleneck of the algorithm and can take up to  $O(\min(mn^2, nm^2))$  operations if full SVD is used. Approximate SVD algorithms are typically faster. For example, the algorithm in [10] runs in  $O(mn)$  time, which is linear in the number of elements in the matrix. If we have some prior knowledge about the structure of the matrix, it can take even less time. For example, if an approximation of the norms of the columns is known, we can use *LinearTimeSvd* [6] to achieve a sub-linear runtime complexity of  $O(s^2m + s^3)$ . We denote the runtime complexity of this step by  $T_{approx}$ . Using the *RRQR* algorithm in [3], steps 2 and 3 in Algorithm 1 take  $O(ms^2)$  and  $O(ns^2)$  operations, respectively. Finally, the formation of  $A_M$  takes  $O(s^2)$  time. The total runtime complexity becomes  $O(T_{approx} + (m+n)s^2)$  and it is usually dominated by  $O(T_{approx})$ .

Denote the space requirements of step 1 in Algorithm 1 by  $S_{approx}$ . Then, the total space complexity becomes  $O(S_{approx} + s(m+n))$ . Typically, a total of  $O((m+n)s^{O(1)})$  space is used.

## 6.2 Relation to ICD

Let  $M$  be decomposed into  $M = X^T X$  where  $X \in \mathbb{R}^{n \times n}$ . In this case, the  $R$  factor in the QR decomposition of  $X$  is the Cholesky factor of  $M$  since  $X = QR$  means that  $M = X^T X = R^T Q^T QR = R^T R$ . Similarly, the Cholesky decomposition of a symmetrically pivoted  $M$  corresponds to a column pivoted QR of  $X$ . The pivoting strategy used by the Cholesky algorithm in the ICD algorithm is the greedy scheme of the classical pivoted-QR algorithm in [14]. Applying ICD to  $M$  gives the  $R$  factor of the pivoted QR on  $X$ , and vice versa. The special structure of the matrix enables the ICD to unite steps 1,2 and 3 in Algorithm 1, creating a rank- $s$  approximation to  $M$  while at the same time choosing pivots according to a greedy QR criterion. This allows the ICD to achieve a runtime complexity of  $O(s^2n)$ .

## 7 Experimental Results

In our experiments, we employ a fast but inaccurate sub-linear SVD approximation for step 1 in Algorithm 1. This approximated SVD first randomly samples the columns of the

matrix. Then, it uses these columns in the *LinearTimeSVD* algorithm of [6] to compute an SVD approximation in  $O(s^2m + s^3)$  operations. For this SVD algorithm, the total runtime complexity of Algorithm 1 is  $O(s^2(m + n) + s^3)$  which is dominated by  $O(s^2(m + n))$ .

## 7.1 Kernel Matrices

First, we compare between the performance of Algorithm 1 and the state-of-the-art sample selection algorithms for kernel matrices. We construct a kernel matrix for a given dataset, then each algorithm is used to choose a fixed sized sample. From the notation of Eqs. (2.1) and (2.7), the error is displayed as  $\|\hat{M} - M\|$ .

The following algorithms were compared: 1. The ICD algorithm presented in section 5.1; 2. The  $k$ -means based algorithm presented in section 5.1; 3. Random choice of sub-sample as given in [2]; 4. *LinearTimeSVD* of [6]; 5. Algorithm 1; 6. SVD. The SVD algorithm is used as a benchmark, since it provides rank- $s$  approximation with the lowest Frobenius norm error. The empirical gain of our procedure can be measured by the difference between the approximation errors of *LinearTimeSVD* and Algorithm 1, since *LinearTimeSVD* is used in Step 1 of Algorithm 1.

We use a Gaussian kernel of the form  $k(x, y) = \exp(-\|x - y\|^2 / \epsilon)$  where  $\epsilon$  is the average squared distance between data points and the means of each dataset. Results for methods which contain probabilistic components are presented as the averages over 20 trials. These include methods 2, 3, 4 and 5. The sample size is gradually increased from 1% to 10% of the total data and the error is measured in terms of the Frobenius norm. The benchmark datasets, summarized in Table 7.1, were taken from the LIBSVM archive [33]. The overall experimental parameters were chosen to allow for comparison with Fig. 1 in [26].

The results are presented in Fig. 7.1. Algorithm 1 generally outperforms the random sample selection algorithm, particularly on datasets with fast spectrum decay such as *german*, *segment* and *svmguide1a*. In these datasets, our algorithm approaches and sometimes even surpasses the state-of-the-art  $k$ -means based algorithm of [26]. This fits our derivation for the approximation error given by Theorem 5.12.

It should be noted that the algorithm in [26] has a runtime complexity of  $O(sn)$  compared to our  $O(s^2n)$  for this setting. This difference has no real-world consequences when  $s$  is very

small or even constant, as typical for these problems.

In some cases, Algorithm 1 actually performs worse than *LinearTimeSVD*. We use a greedy RRQR algorithm which sometimes does not properly sort the singular-vectors according to their importance (namely, the absolute value of the singular-value). This can happen for instance when the spectrum decays slowly, which means leading singular values are close in magnitude. In Algorithm 1, we always choose the top  $s$  indices as found by the RRQR algorithm, so we might get things wrong.

dataset	<b>german</b>	<b>splice</b>	<b>adult1a</b>	<b>dna</b>	<b>segment</b>	<b>w1a</b>	<b>svmgd1a</b>	<b>satimage</b>
sample count	1000	1000	1605	2000	2310	2477	3089	4435
dimension	24	60	123	180	19	300	4	36

Table 7.1: Summary of benchmark datasets (taken from [33])

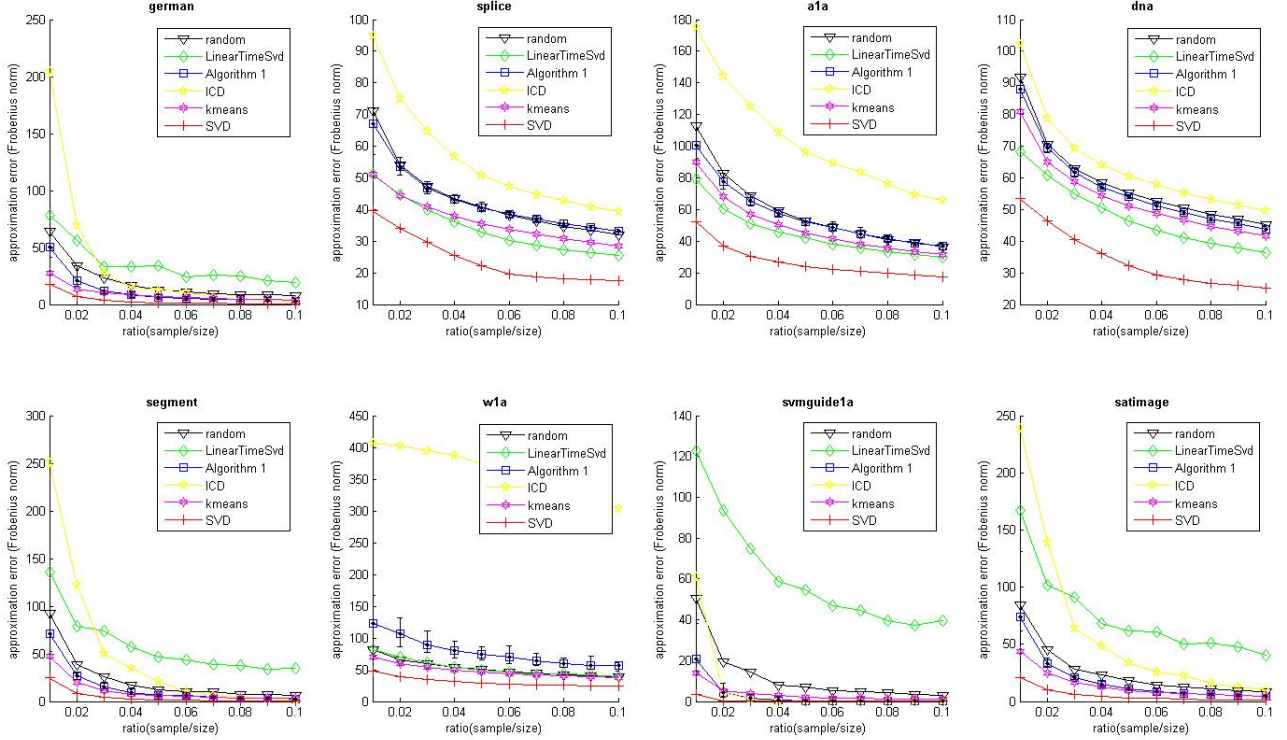


Figure 7.1: Nyström approximation errors for kernel matrices. The X-axis is the sampling ratio given as sample size divided by the matrix size. The Y-axis is the approximation error given in Frobenius norm. The tested algorithms are: random, LinearTimeSVD, Algorithm 1, ICD, k-means and SVD

## 7.2 General Matrices

We evaluate the performance of Algorithm 1 on general matrices by comparing it to a random choice of sub-sample. We use the full SVD as a benchmark that theoretically achieves the best accuracy. The approximation error is measured by  $\left\| \hat{M} - M \right\|_2$ .

The testing matrices in this section were chosen to have non-random spectra with random singular subspaces. Initially, a non-random diagonal matrix  $L$  is chosen with non-increasing diagonal entries.  $L$  will serve as the spectrum of our testing matrix. Then, two random unitary matrices  $U$  and  $V$  are generated. Our testing matrix is formed by  $ULV^T$ . We examine two degrees of spectrum decay: linear decay (slow) and exponential decay (fast).

The error is presented in  $L_2$  norm and we vary the sample size to be between 1%-10% of

the matrix size. The presented results are from an averaging of 20 iterations to reduce the statistical variability. For simplicity, we produce results only for  $500 \times 500$  square matrices. The results are presented in Fig. 7.2. When the spectrum decays slowly, Algorithm 1 has no advantage over random sample selection. It produces overall pretty bad results. But the situation is much different in the presence of a fast spectrum decay. Algorithm 1 displays good results when the sample size allows it to capture most of the significant singular values of the data (at a sample rate of about 3%). It is interesting to note that random sample selection does not lag far behind. This hints that, on average, any sample is a good sample as long as it captures more data than the numeric rank of the matrix.

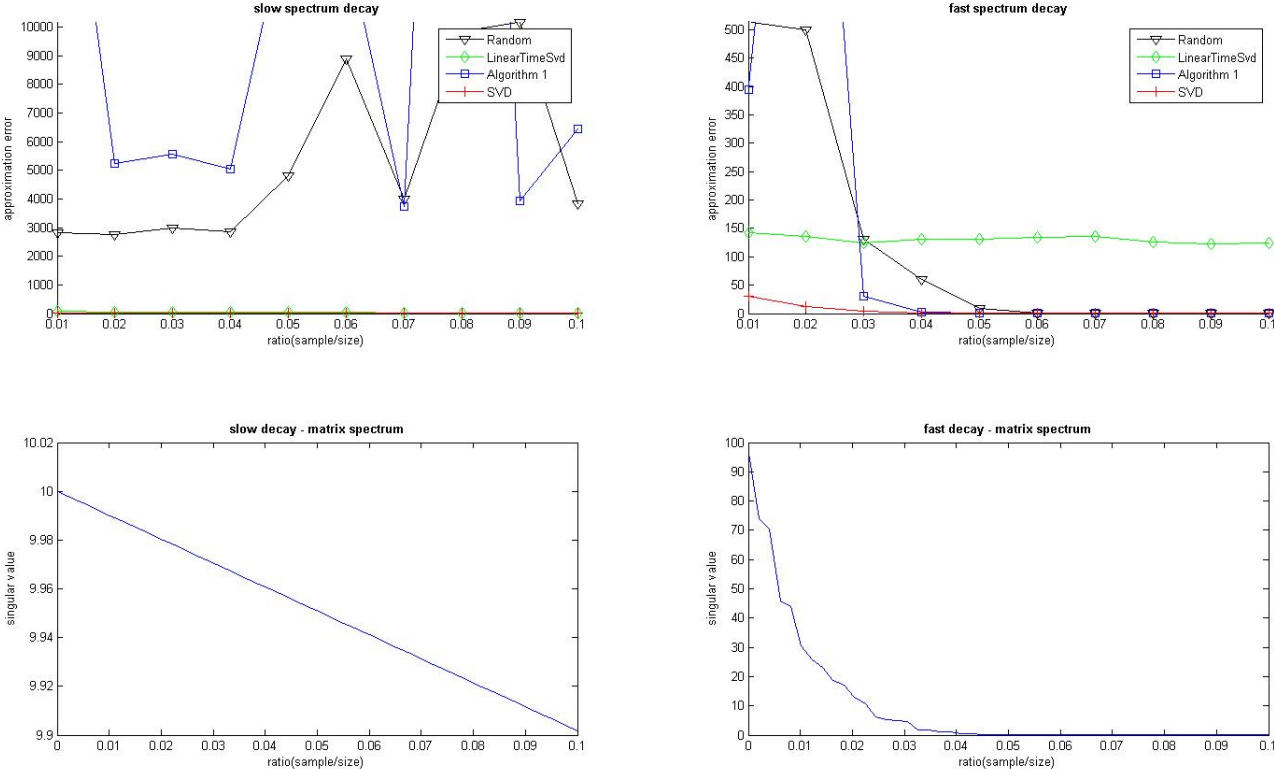


Figure 7.2: Nyström approximation errors for random matrices. The X-axis is the sampling ratio given as sample size divided by the matrix size. The Y-axis is the approximation error given in  $L_2$  norm. The tested algorithms are Random, LinearTimeSVD, Algorithm 1 and SVD.

### 7.3 Non-Singularity of Sample Matrix

We empirically examine the relationship between the Nyström approximation error and the non-singularity of the sub-sample matrix. The approximation error is measured in  $L_2$ -norm and the non-singularity of  $A_M \in \mathbb{R}^{s \times s}$  is measured by the magnitude of  $\sigma_s(A_M)$ . We employ testing matrices similar to those in section 7.2. These feature a non-random spectrum and random singular subspaces. The sample was chosen to be 5% of the data of the matrix. In this test, we compare between the random sample selection algorithm and Algorithm 1. Each algorithm ran 100 times on each matrix. The results of each run were recorded. Figure 7.3 features a log-log scale plot of the approximation error as a function of  $\sigma_s(A_M)$ . The performance of the different algorithm versions is compared. We arrive at similar conclusions to those in section 7.2. Our algorithms do no better than random sampling when the spectrum decay is slow, but consistently outperforms the random selection in the presence of fast spectrum decay. Figure 7.3 also shows a strong negative correlation between the variables in all the examined matrices. Hence, a large  $\sigma_s(A_M)$  implies a small approximation error. The linear shape of the graphs, drawn in a log-log scale, suggests that this relationship is exponential. The results hint at a possible extension of the Nyström procedure to a Monte-Carlo method: Algorithm 1 can be run many times. In the end, we choose the sample for which  $\sigma_s(A_M)$  is maximal.

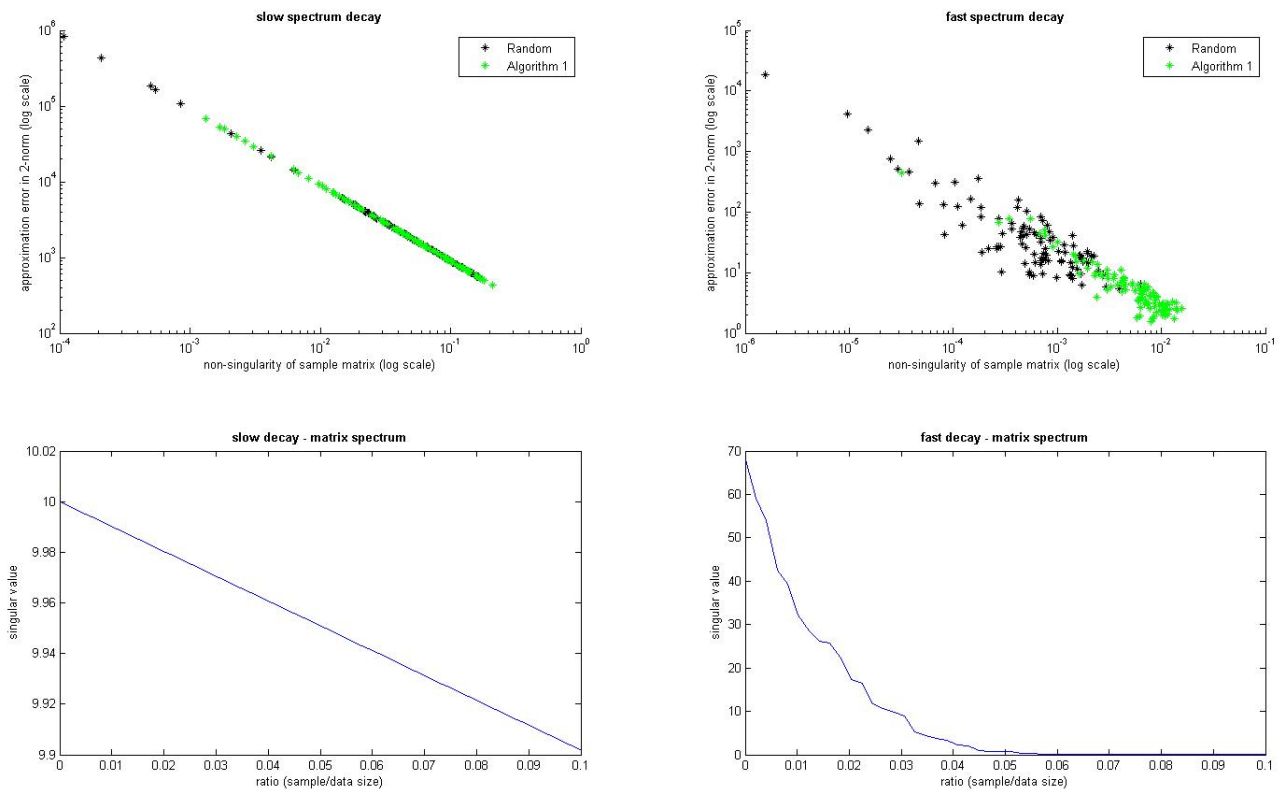


Figure 7.3: Errors in Nyström approximation as a function of  $\sigma_s(A_M)$

## 8 Conclusion and Future Research

In this paper, we showed how the Nyström approximation method can be used to find the canonical SVD and EVD of a general matrix. In addition, we developed a sample selection algorithm that operates on general matrices. Experiments have been performed on real-world kernels and random general matrices. These show that the algorithm performs well when the spectrum of the matrix decays quickly and the sample is sufficiently large to capture most of the energy of the matrix (the number of non-zero singular values). Another experiment showed that the non-singularity of the sample matrix (as measured by the magnitude of the smallest singular value) is exponentially inversely related to the approximation error. This shows that our theoretical reasoning in Lemma 5.10 is qualitatively on par with empirical evidence.

Future research should focus on additional formalization of the relationship between the smallest singular value of the sample matrix and the Nyström approximation error. Another interesting possibility is to find a constrained class of matrices and develop a sample selection algorithm to take advantage of the constraint. Some classes of matrices may be easier to sub-sample with respect to the Nyström method.

ACKNOWLEDGMENT: The second author in this research was supported by the Israel Science Foundation (Grant No. 1041/10).

## References

- [1] C.T.H. Baker, *The Numerical Treatment of Integral Equations*, Oxford: Clarendon Press, 1977.
- [2] C. Fowlkes, S. Belongie, F. Chung, J. Malik, Spectral Grouping Using the Nyström Method, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 214-225, February, 2004.
- [3] M. Gu, S.C. Eisenstat, An efficient algorithm for computing a strong rank revealing QR factorization, *SIAM J. Sci. Comput.*, 17 (1996), pp. 848-869.
- [4] C. T. Pan. On the existence and computation of rank-revealing LU factorizations. *Linear Algebra Appl*, 316:199-222, 2000.
- [5] G.H. Golub and C.F. Van Loan, *Matrix Computations* (3rd Ed.), Johns Hopkins University Press, 1996.
- [6] P. Drineas, R. Kannan and M. W. Mahoney, Fast Monte Carlo Algorithms for Matrices II: Computing a Low-Rank Approximation to a Matrix, *SIAM J. Comput.* 36(1), 158-183, 2006.
- [7] G. W. Stewart, Four algorithms for the efficient computation of truncated QR approximations to a sparse matrix, *Numer. Math.*, 83, pp. 313-323, 1999.

- [8] E. Liberty, F. Woolfe, P.-G. Martinsson, V. Rokhlin, and M. Tygert. Randomized algorithms for the low-rank approximation of matrices. *Proc. Natl. Acad. Sci. USA*, 104(51): 20167-20172, 2007.
- [9] A. Deshpande and S. Vempala, Adaptive sampling and fast low-rank matrix approximation, Technical report TR06-042, Electronic Colloquium on Computational Complexity, 2006.
- [10] S. Har-Peled. Low rank matrix approximation in linear time. Manuscript. January 2006.
- [11] T. Sarlós, Improved approximation algorithms for large matrices via random projections, in *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pp. 143-152, 2006
- [12] S. Fine and K. Scheinberg. Efficient SVM training using low-rank kernel representations. *J. Mach. Learn. Res.*, 2:243-264, 2001.
- [13] P. Drineas and M. W. Mahoney, On the Nyström method for approximating a Gram matrix for improved kernel-based learning, *J. Machine Learning*, 6, pp. 2153-2175, 2005.
- [14] P. A. Businger and G. H. Golub, Linear least squares solution by Householder transformation, *Numerische Mathematik*, 7 (1965), pp. 269-276.
- [15] A. Bjorck and S. Hammarling, A Schur method for the square root of a matrix, *Linear Algebra and Appl.*, 52/53 (1983) pp. 127-140.
- [16] C. K. I. Williams and M. Seeger, Using the Nyström method to speed up kernel machines, *Advances in Neural Information Processing Systems 2000*, MIT Press, 2001.
- [17] C. Fowlkes, S. Belongie, and J. Malik, Efficient Spatiotemporal Grouping Using the Nyström Method, *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Dec. 2001.
- [18] S. Belongie, C. Fowlkes, F. Chung, and J. Malik, Spectral Partitioning with Indefinite Kernels Using the Nyström Extension, *Proc. European Conf. Computer Vision*, 2002.

- [19] R.R. Coifman and S. Lafon. Geometric harmonics: a novel tool for multiscale out-of-sample extension of empirical functions. *Appl. Comp. Harm. Anal.*, 21(1):31-52, 2006.
- [20] Y. Bengio, O. Delalleau, N. Roux, J. Paiement, P. Vincent and M. Ouimet, Learning eigenfunctions links spectral embedding and kernel PCA. *Neural Computation*, 16, 2197-2219, 2004.
- [21] D. Achlioptas et. al., Sampling Techniques for Kernel Methods, *Annual Advances in Neural Information Processing Systems 14*, 2001.
- [22] E. J. Candes and T. Tao, The Power of Convex Relaxation: Near-Optimal Matrix Completion. *IEEE Transactions on Information Theory*, 56 (5). pp. 2053-2080.
- [23] M. Bebendorf and R. Grzhibovskis, Accelerating Galerkin BEM for linear elasticity using adaptive cross approximation, *Mathematical Methods in the Applied Sciences*, *Math. Meth. Appl. Sci.*, 29:1721-1747, 2006.
- [24] M. Bebendorf and S. Rjasanow, Adaptive Low-Rank Approximation of Collocation Matrices, *Computing* 70, 1-24, 2003.
- [25] M. Ouimet and Y. Bengio. Greedy spectral embedding. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, 2005.
- [26] K. Zhang, I.W. Tsang, and J.T. Kwok, Improved Nyström low-rank approximation and error analysis. In *Proceedings of the 25th international Conference on Machine Learning (Helsinki, Finland, July 05 - 09, 2008)*. ICML '08, vol. 307. ACM, New York, NY, 1232-1239, 2008.
- [27] A. Smola and B. Schölkopf, Sparse greedy matrix approximation for machine learning, *Proceedings of the 17th international conference on machine learning*, pp 911-918. June, 2000.
- [28] V. Rokhlin, A. Szlam, and M. Tygert, A randomized algorithm for principal component analysis, *Tech. Rep. 0809.2274*, arXiv, 2008. Available at <http://arxiv.org>.

- [29] S.T. Roweis and L.K. Saul. Nonlinear dimensionality reduction by Locally Linear Embedding. *Science*, 290(5500):2323-2326, 2000.
- [30] T. Cox and M. Cox. *Multidimensional scaling*. Chapman & Hall, London, UK, 1994.
- [31] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417-441, 1933.
- [32] S. Lafon and A.B. Lee. Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1393-1403, 2006.
- [33] C. C. Chang and C.J. Lin, LIBSVM : a library for support vector machines, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/>