Block subsampled randomized Hadamard transform for low-rank approximation on distributed architectures

Oleg Balabanov, Matthias Beaupère, Laura Grigori, Victor Lederer

Block subsampled randomized Hadamard transform for low-rank approximation on distributed architectures

Oleg Balabanov, Matthias Beaupère, Laura Grigori and Victor Lederer *

Abstract

This article introduces a novel structured random matrix composed blockwise from subsampled randomized Hadamard transforms (SRHTs). The block SRHT is expected to outperform well-known dimension reduction maps, including SRHT and Gaussian matrices, on distributed architectures with not too many cores compared to the dimension. We prove that a block SRHT with enough rows is an oblivious subspace embedding, i.e., an approximate isometry for an arbitrary low-dimensional subspace with high probability. Our estimate of the required number of rows is similar to that of the standard SRHT. This suggests that the two transforms should provide the same accuracy of approximation in the algorithms. The block SRHT can be readily incorporated into randomized methods, for instance to compute a low-rank approximation of a large-scale matrix. For completeness, we revisit some common randomized approaches for this problem such as Randomized Singular Value Decomposition and Nyström approximation, with a discussion of their accuracy and implementation on distributed architectures.

Key words — randomization, sketching, embedding, low-rank approximation, parallel computing.

1 Introduction

Randomization has become a powerful tool for tackling massive problems in numerical algebra and data science [1, 2, 3, 4]. Modern randomized methods can, in particular, provide solutions to problems of dimensions beyond the reach of deterministic methods, and allow effective use of computational resources. Recent significant development has made them very reliable, and not just used as a last resort, as it was not so long ago. Along with increased efficiency, they can now provide strong accuracy guarantees with a user-specified failure probability that can be chosen extremely low, say 10^{-10}, without much impact on computational costs.

This article is concerned with randomized methods that are based on a dimension reduction, called sketching [2], with oblivious ℓ_2-subspace embeddings (OSEs) defined below.

Definition 1.1. *Let $0 \leq \varepsilon < 1$ and $0 < \delta < 1$. A random matrix $\boldsymbol{\Omega} \in \mathbb{R}^{l \times n}$ is said to be a (ε, δ, d) OSE, if for any fixed d-dimensional subspace $V \subseteq \mathbb{R}^n$,*

$$\forall \mathbf{x} \in V, \quad |\|\mathbf{x}\|_2^2 - \|\boldsymbol{\Omega}\mathbf{x}\|_2^2| \leq \varepsilon \|\mathbf{x}\|_2^2 \tag{1.1}$$

holds with probability at least $1 - \delta$.

It is a consequence of the Johnson-Lindenstrauss lemma [5] that there exist (ε, δ, d) OSEs of sizes $l = \mathcal{O}(\varepsilon^{-2}(d + \log \frac{1}{\delta})$. The fact that n does not appear in the right-hand-side and the logarithmic dependence on the

*Sorbonne Université, Inria, CNRS, Université de Paris, Laboratoire Jacques-Louis Lions, Paris, France.

probability of failure δ shows the potential of a dimension reduction with such embeddings. There are several distributions that are known to satisfy the OSE property with the optimal or close to optimal l. The Gaussian, Rademacher distributions, sub-sampled randomized Hadamard transform (SRHT), sub-sampled randomized Fourier transform, and CountSketch matrix are ones of the most popular distributions. The random sketching matrix in the algorithm should be chosen depending on the computational architecture to yield the most benefit. For instance, the SRHT is a structured matrix that can be efficiently applied to a vector in a sequential computational environment, while the application of a Rademacher matrix is efficient in a highly parallel environment. In this paper we propose a novel OSE, called block SRHT, which should be superior to all currently existing ones on a distributed computational architecture, with not too many computational units.

The OSEs are used in a variety of randomized methods for machine learning, scientific computing, and signal processing. Perhaps one of the most representative applications is the linear regression problem. Suppose that we seek a vector $\mathbf{x} \in \mathbb{R}^d$ that minimizes $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2$, where $\mathbf{A} \in \mathbb{R}^{n\times d}$ is a large-scale dense matrix, $\mathbf{b} \in \mathbb{R}^n$ is a large-scale vector, and $d \ll n$. It follows that the solution to this problem can be approximated by a minimizer of $\|\mathbf{\Omega}(\mathbf{A}\mathbf{x} - \mathbf{b})\|_2$ requiring considerably lower computational cost. The accuracy of such an approximation is guaranteed, given that $\mathbf{\Omega}$ is $(\varepsilon, \delta, d + 1)$ OSE. Besides the linear regression problem, the sketching technique with OSEs has been successfully applied to the nearest neighbors problem [6], approximation of products of matrices [7], computation of low-rank approximations of matrices [8] and tensor decompositions [9], dictionary learning [10], solution of parametric equations [11], and solution of linear systems and eigenvalue problems [12].

In this paper the potential of the block SRHT is realized on the low-rank approximation problem. Such problems are ubiquitous, for instance, in the principal component analysis of large data sets and kernel ridge regression. A randomized low-rank approximation for machine learning tasks was addressed in e.g. [13, 14, 15]. In [16, 17, 18, 19, 20] a particular focus was given to make the methods suited to modern architectures. In details, given a large matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ with rapidly decaying spectrum, we seek a matrix $\mathbf{B}_k$ preferably in an SVD form, of rank at most $k \ll \min(m, n)$, that approximates well $\mathbf{A}$. The matrix $\mathbf{B}_k$ can be obtained by first restricting its range to a subspace that captures the most of the action of $\mathbf{A}$, and then minimizing the chosen error measure, say the spectral error $\|\mathbf{A} - \mathbf{B}_k\|_2$. As shown in [8], the most of $\mathbf{A}$'s action can be well captured by the range of $\mathbf{A}\mathbf{\Omega}^{\mathrm{T}}$, which constitutes the core of state-of-the-art Randomized Singular Value Decomposition (RSVD) algorithm. Over the past years more sophisticated randomized low-rank approximation methods have been developed such as the Nyström method for spd matrices [21, 22], and the single-view approximations for general matrices [23, 24, 25]. In this work, we outline some such methods and analyze them with a projection-based approach compatible with block SRHT.

The paper is organized as follows. The rest of Section 1 discusses contributions and relation to prior work. Section 2 introduces a block SRHT matrix and discusses its properties. In Section 3 we present randomized algorithms for computing a low-rank approximation based on oblivious embeddings. Section 4 contains some computational aspects and experimental results. The proof of the main theoretical result is given in Section 5. Section 6 concludes this work.

Contributions

The proposed block SRHT matrix has the potential to combine the benefits of structured oblivious embeddings, such as the SRHT, with the benefits of unstructured ones, such as Gaussian, from the complexity and performance standpoint. We are not aware of any other work in this direction. We prove that the block SRHT matrix of size $l = \mathcal{O}(\varepsilon^{-2}(d + \log \frac{n}{\delta}) \log \frac{d}{\delta})$ satisfies the (ε, δ, d) OSEs property. This result is similar to that for standard SRHT from the literature [26, 27] and implies that the two matrices should yield approximations of similar quality. This result does not simply follow from the analysis in [26, 27], and particularly requires incorporation of a useful technical trick that, to the best of our knowledge, was not employed before in the randomized numerical linear

algebra community.

In addition, we present low-rank approximation methods such as RSVD [8], Nyström approximation, and the single-view approximation from [24] in a new unified projection-based form that clearly shows their connection. We provide a rigorous characterization of their accuracy based solely on the OSE property, and therefore compatible with any types of sketching matrices that satisfy this property, including the block SRHT. For RSVD and Nyström approximation, this characterization follows almost directly from standard results from the literature. On the other hand, for the single-view approximation, our results are new. In particular, the novelty lies in the use of the projection-based interpretation of the single-view approximation to show that it is almost as accurate as RSVD if the embeddings for the "core sketch" are OSEs of sufficiently large sizes. Important aspects of implementation in distributed architectures are also discussed using the suitability of block SRHT for these architectures.

2 Block sub-sampled randomized Hadamard transform

For n being a power of two, an SRHT matrix can be defined as follows:

$$\boldsymbol{\Omega} = \sqrt{\tfrac{n}{l}}\mathbf{RHD}, \tag{2.1}$$

where $\mathbf{R}$ is a $l \times n$ uniform, with or without replacement, random sampling matrix, $\mathbf{H}$ is a $n \times n$ Walsh-Hadamard matrix rescaled by $\frac{1}{\sqrt{n}}$, and $\mathbf{D}$ is a diagonal matrix with i.i.d. Rademacher random variables ± 1 on the diagonal. The properties of SRHT were thoroughly described in [26] with a follow up analysis in [27]. SRHT matrices are commonly used in randomized algorithms as they can be applied to vectors using only $n \log_2 n$ flops, while general unstructured matrices require $2nl$ flops. At the same time, they satisfy the (ε, δ, d) OSE property, if [11]

$$l \geq 3\varepsilon^{-2}(\sqrt{d} + \sqrt{8 \log \tfrac{6n}{\delta}})^2 \tfrac{3d}{\delta}, \tag{2.2}$$

which is only by a logarithmic factor in δ and n larger than the optimal bound. For a general n, a partial SRHT can be used that is defined as the first n columns of an SRHT matrix. Using a partial SRHT is equivalent to padding the input data with zeros to make its dimension a power of two.

Unfortunately, products with SRHT matrices are not well suited to distributed computing limiting the benefits of SRHT on modern architectures (see e.g. [28]). This happens majorly due to computing products with $\mathbf{H}$ in tensor form

$$\mathbf{H} = \tfrac{1}{\sqrt{n}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \cdots \otimes \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

requiring cumbersome reduction operator such as a sequence of arrays of butterflies, rather than a simple addition, which we have with Gaussian matrices. This article attempts to alleviate this problem by constructing $\boldsymbol{\Omega}$ block-wise as follows

$$\boldsymbol{\Omega} = [\boldsymbol{\Omega}^{(1)} \quad \boldsymbol{\Omega}^{(2)} \quad \dots \quad \boldsymbol{\Omega}^{(p)}], \tag{2.3}$$

where $\boldsymbol{\Omega}^{(i)} = \sqrt{\frac{r}{l}}\widetilde{\mathbf{D}}^{(i)}\mathbf{RHD}^{(i)}$ are $l \times r$ SRHT matrices related to a unique sampling matrix $\mathbf{R}$ and different (independent from each other) diagonal matrices $\mathbf{D}^{(i)}$ with i.i.d. Rademacher entries ± 1, multiplied from the left by another diagonal matrices $\widetilde{\mathbf{D}}^{(i)}$ with Rademacher entries, $r = \frac{n}{p}$, $1 \leq i \leq p$. As in the standard SRHT, the condition that r is a power of two can be achieved by zero padding of the input data. The advantage of $\boldsymbol{\Omega}$ defined by (2.3) is that it can be multiplied by an $n \times d$ matrix $\mathbf{V}$ distributed between p processors with rowwise partitioning, as

$$\boldsymbol{\Omega}\mathbf{V} = \sum_{1 \leq i \leq p} \boldsymbol{\Omega}^{(i)}\mathbf{V}^{(i)}, \tag{2.4}$$

where $\mathbf{V}^{(i)}$ are the corresponding local blocks of rows of $\mathbf{V}$. In this way, to obtain $\mathbf{\Omega V}$ one can compute the local contributions $\mathbf{\Omega}^{(i)}\mathbf{V}^{(i)}$ on each processor and then sum-reduce them to the master processor. This makes block SRHT matrices have the same application cost in terms of communication as Gaussian matrices. Thus, they should yield much better scalability of computations than standard SRHT [28]. The sum-reduce operation requires exchanging $\mathcal{O}(\log p)$ messages and $\mathcal{O}(dl \log p)$ per-processor communication volume that can be by a factor $\mathcal{O}(\frac{r}{l})$ less than the volume of communication used by standard SRHT (if $l \leq r$). At the same time, block SRHT require less flops per processor than Gaussian matrices. To be more specific, the application cost of block SRHT using (2.4) is only $\mathcal{O}(rd \log r + dl \log p)$ flops per processor, while Gaussian matrices require $\mathcal{O}(rdl + dl \log p)$ flops per processor. It is deduced that block SRHT matrices are both well-suited to distributed computing and efficient in terms of flops. They are expected to outperform all existing oblivious embedding when the local dimension r and the sampling dimension l are large enough.

The procedure for application of the block SRHT can be easily extended to the case when $\mathbf{V}$ is distributed with a 2D partitioning. Namely, to multiply $\mathbf{\Omega}$ by $n \times n$ matrix $\mathbf{V}$ distributed over a grid of $p \times p$ processors, we first compute the local contributions $\mathbf{X}^{(i,j)} = \mathbf{\Omega}^{(i)}\mathbf{V}^{(i,j)}$ on each processor, and then sum-reduce the contributions from the j-th column of blocks to the processor $(1, j)$, $1 \leq j \leq p$. Note that in this case the resulting matrix $\mathbf{Y}^{\mathrm{T}} = \mathbf{\Omega A}$ is distributed with rowwise partitioning over processors $(1, 1), (1, 2), \ldots, (1, p)$. This provides the ability to efficiently compute the sketch $\mathbf{\Omega Y}$, or to orthogonalize $\mathbf{Y}$ with a routine suited for distributed computing, with no need to reorganize $\mathbf{Y}$. This can be particularly handy in the low-rank approximation algorithm from Section 3.

We will assume that $\mathbf{R}$ in (2.3) samples rows uniformly at random and *with replacement*. Interestingly, in this case the block SRHT can be viewed as a generalization of the SRHT with replacement and the Rademacher embedding, as it reduces to these maps when $r = n$ and $r = 1$, respectively. Sampling with replacement can be important, for instance, when r is smaller than the dimension of the embedded subspace.

Theorem 2.1 is the main result of the article. It implies the compatibility of the block SRHT with all randomized methods that rely on OSEs, including the methods in Section 3. The estimate of l in Theorem 2.1 is similar to (2.2) for the standard SRHT matrix, and in particular depends only logarithmically on n and δ.

Theorem 2.1 (Main Theorem)**.** *Let $0 < \varepsilon < 1$ and $0 < \delta < 1$. Let $\mathbf{\Omega} \in \mathbb{R}^{l \times n}$ be defined by (2.3). If,*

$$n \geq l \geq 3.7\varepsilon^{-2}(\sqrt{d} + 4\sqrt{\log \tfrac{n}{\delta} + 6.3})^2 \log \tfrac{5d}{\delta},$$

then $\mathbf{\Omega}$ is an (ε, δ, d) OSE.

For better presentation the proof of Theorem 2.1 is deferred to the end of the article (see Section 5).

3 Randomized low-rank approximation

This section addresses the problem of efficient computation of a rank-k approximation of a large matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ with rapidly decaying spectrum. We provide and analyze some common randomized algorithms for this task. They are presented with a *projection-based* approach relying solely on the OSE property of the sketching matrix, and are compatible with the block SRHT thanks to Theorem 2.1. A particular focus is given to the scenario where $\mathbf{A}$ is uniformly distributed over a 2D grid of processors. For simplicity assume that $m \leq n$.

It is a well-known fact that the best rank-k approximation of $\mathbf{A}$ in terms of the spectral, trace and Frobenius error is given by $[\![\mathbf{A}]\!]_k := \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^{\mathrm{T}}$, where $\mathbf{\Sigma}_k$ is a diagonal matrix of k dominant singular values of $\mathbf{A}$, and $\mathbf{U}_k$ and $\mathbf{V}_k$ contain the associated left and right singular vectors. In other words, we have

$$[\![\mathbf{A}]\!]_k = \arg \min_{\mathrm{rank}(\mathbf{B})=k} \|\mathbf{A} - \mathbf{B}\|_\xi,$$

where $\xi = 2, *$ or F. Furthermore, $\|\mathbf{A} - [\![\mathbf{A}]\!]_k\|_\xi = \sigma_{k+1}$ if $\xi = 2$, $\sum_{i=k+1}^m \sigma_i$ if $\xi = *$, or $(\sum_{i=k+1}^m \sigma_i^2)^{\frac{1}{2}}$ if $\xi = \mathrm{F}$, where $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_m$ denote the singular values of $\mathbf{A}$.

Obtaining the best rank-k approximation can be computationally expensive and often becomes the bottleneck of the algorithm. In such case, one has to turn to alternative methods for computing a low-rank approximation, such as the randomized methods described below.

3.1 Randomized Singular Value Decomposition

A low-rank approximation of $\mathbf{A}$ can be interpreted as reduction of the range of $\mathbf{A}$ to a low-dimensional subspace Q capturing the most of $\mathbf{A}$'s action. The SVD approximation $[\![\mathbf{A}]\!]_k$ corresponds to taking Q as range($\mathbf{U}_k$). A more efficient way is to take $Q = \mathrm{range}(\mathbf{A}\boldsymbol{\Omega}^{\mathrm{T}})$, where $\boldsymbol{\Omega} \in \mathbb{R}^{l \times n}$ is an OSE [8, 2]. In this case, the optimal approximation is

$$[\![\mathbf{A}]\!]^{(\mathrm{RSVD})} := \arg \min_{\mathrm{range}(\mathbf{B}) \subseteq Q} \|\mathbf{A} - \mathbf{B}\|_\xi. \tag{3.1}$$

Then a rank-k approximation of $\mathbf{A}$ can be obtained by a truncated SVD of $[\![\mathbf{A}]\!]^{(\mathrm{RSVD})}$, which leads to the approximation $[\![\mathbf{A}]\!]_k^{(\mathrm{RSVD})} := [\![[\![\mathbf{A}]\!]^{(\mathrm{RSVD})}]\!]_k$. Notice that $\mathbf{Q}^{\mathrm{T}}[\![\mathbf{A}]\!]_k^{(\mathrm{RSVD})} = [\![\mathbf{Q}^{\mathrm{T}}\mathbf{A}]\!]_k$, where $\mathbf{Q}$ is an orthonormal basis for Q. This observation constitutes the core for the RSVD algorithm (see Algorithm 1) for the computation of $[\![\mathbf{A}]\!]_k^{(\mathrm{RSVD})}$. Algorithm 1 is same as [1, Algorithm 8], with specifying the way of computing the SVD of a tall and skinny matrix $\mathbf{Z}^{\mathrm{T}}$ by a QR factorization and the SVD of the R factor (see steps 4 and 5). The computation of $\mathbf{Y}$ can be effectively done with the procedure from Section 2 using the block structure of $\boldsymbol{\Omega}$. The QR factorizations in steps 2 and 4 should be performed with TSQR [29] or other methods having low communication cost. The computational cost of Algorithm 1 is dominated by computing $\mathbf{Q}^{\mathrm{T}}\mathbf{A}$ in step 3 and possibly $\mathbf{A}\boldsymbol{\Omega}^{\mathrm{T}}$ in step 1.

Algorithm 1 RSVD, based on [1, Algorithm 8]

Require: $m \times n$ matrix $\mathbf{A}$, $l \times n$ matrix $\boldsymbol{\Omega}$ with $l \ll n$, the target rank k.

1: Compute $\mathbf{Y} = \mathbf{A}\boldsymbol{\Omega}^{\mathrm{T}}$.
2: Orthogonalize $\mathbf{Y}$ with QR factorization, and get $\mathbf{Q}$.
3: Compute matrix $\mathbf{Z} = \mathbf{Q}^{\mathrm{T}}\mathbf{A}$.
4: Obtain a QR factorization $\mathbf{PR}$ of $\mathbf{Z}^{\mathrm{T}}$
5: Use SVD to compute the best rank-k approximation $\tilde{\mathbf{U}}_k\tilde{\boldsymbol{\Sigma}}_k\tilde{\mathbf{V}}_k^{\mathrm{T}}$ of $\mathbf{R}^{\mathrm{T}}$.
6: Output factorization $[\![\mathbf{A}]\!]_k^{(\mathrm{RSVD})} = (\mathbf{Q}\tilde{\mathbf{U}}_k)\tilde{\boldsymbol{\Sigma}}_k(\mathbf{P}\tilde{\mathbf{V}}_k)^{\mathrm{T}}$.

Let us now characterize the accuracy of RSVD approximation. It can be measured for instance by the quasi-optimality constant $\frac{\|\mathbf{A}-[\![\mathbf{A}]\!]_k^{(\mathrm{RSVD})}\|_\xi}{\|\mathbf{A}-[\![\mathbf{A}]\!]_k\|_\xi} - 1$. It is first shown that the accuracy of $[\![\mathbf{A}]\!]_k^{(\mathrm{RSVD})}$ is guaranteed if $Q = \mathrm{range}(\mathbf{A}\boldsymbol{\Omega}^{\mathrm{T}})$ captures well the range of $\mathbf{A}$, i.e., for some $d \geq k$ and $\varepsilon^* \leq \frac{1}{2}$ we have

$$\|\mathbf{A} - [\![\mathbf{A}]\!]^{(\mathrm{RSVD})}\|_{\mathrm{F}}^2 \leq (1 + \varepsilon^*)\|\mathbf{A} - [\![\mathbf{A}]\!]_d\|_{\mathrm{F}}^2. \tag{3.2}$$

Then, by the triangle inequality,

$$\|\mathbf{A} - [\![\mathbf{A}]\!]_k^{(\mathrm{RSVD})}\|_\xi \leq \|\mathbf{A} - [\![\mathbf{A}]\!]_k\|_\xi + \|\mathbf{A} - [\![\mathbf{A}]\!]^{(\mathrm{RSVD})}\|_\xi, \tag{3.3}$$

where $\xi = 2, *$ or F, we obtain

$$\|\mathbf{A} - [\![\mathbf{A}]\!]_k^{(\mathrm{RSVD})}\|_\xi \leq \|\mathbf{A} - [\![\mathbf{A}]\!]_k\|_\xi + 2.5\|\mathbf{A} - [\![\mathbf{A}]\!]_d\|_{\mathrm{F}}. \tag{3.4}$$

This result guarantees, under the condition (3.2), the quasi-optimality of $[\![\mathbf{A}]\!]_k^{(\mathrm{RSVD})}$ with respect to the Frobenius norm. Furthermore, if $\mathbf{A}$ has a fast enough singular value decay and d is large enough, so that the tail after d-th singular value of $\mathbf{A}$, i.e. $(\sum_{i=d+1}^{m}\sigma_i^2)^{\frac{1}{2}} = \|\mathbf{A}-[\![\mathbf{A}]\!]_d\|_{\mathrm{F}}$ is small compared to the $k+1$-th singular value $\sigma_{k+1} = \|\mathbf{A}-[\![\mathbf{A}]\!]_k\|_2$ then $[\![\mathbf{A}]\!]_k^{(\mathrm{RSVD})}$ is almost as accurate as $[\![\mathbf{A}]\!]_k$ with respect to all three norms. Moreover, increasing d can make the quasi-optimality constant arbitrary close to zero.

It remains to obtain the conditions on $\boldsymbol{\Omega}$ such that (3.2) holds with high probability. This can be done for instance with the results from [2]. Take $\varepsilon = \frac{4}{9}\varepsilon^*$. It follows from [Lemma 45][2] and its proof that (3.2) holds with probability at least $1-\delta$ if $\boldsymbol{\Omega}$ is an $(\frac{1}{3},\delta,d)$ OSE, and

$$\|\mathbf{V}_d^{\mathrm{T}}\boldsymbol{\Omega}^{\mathrm{T}}\boldsymbol{\Omega}(\mathbf{A}-[\![\mathbf{A}]\!]_d)^{\mathrm{T}}\|_{\mathrm{F}}^2 \leq \varepsilon\|\mathbf{A}-[\![\mathbf{A}]\!]_d\|_{\mathrm{F}}^2.$$

In turn the latter condition is satisfied with probability at least $1-\delta$ if $\boldsymbol{\Omega}$ is an $(\sqrt{\frac{\varepsilon}{d}},\frac{\delta}{N},1)$ OSE, where $N = 2md+m+d$, as shown below. The OSE property of $\boldsymbol{\Omega}$ and the union bound argument guarantee that for given N fixed vectors $\mathbf{z}_i$, we have

$$(1-\sqrt{\tfrac{\varepsilon}{d}})\|\mathbf{z}_i\|_2^2 \leq \|\boldsymbol{\Omega}\mathbf{z}_i\|_2^2 \leq (1+\sqrt{\tfrac{\varepsilon}{d}})\|\mathbf{z}_i\|_2^2, \text{ for } 1\leq i\leq N \tag{3.5}$$

with probability at least $1-\delta$. Take set $\{\mathbf{z}_i\}$ composed of the columns of $\mathbf{V}_d$ denoted by $\mathbf{x}_i$, the columns of $(\mathbf{A}-[\![\mathbf{A}]\!]_d)^{\mathrm{T}}$ denoted by $\mathbf{y}_i$, and all the pairs $\mathbf{x}_i+\mathbf{y}_j$ and $\mathbf{x}_i-\mathbf{y}_j$. Then the relation (3.5), the parallelogram identity and the fact that $\mathbf{x}_i^{\mathrm{T}}\mathbf{y}_j = 0$, imply that

$$|\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Omega}^{\mathrm{T}}\boldsymbol{\Omega}\mathbf{y}_j| \leq \sqrt{\tfrac{\varepsilon}{d}}\|\mathbf{x}_i\|_2\|\mathbf{y}_j\|_2, \text{ for } 1\leq i\leq d,\ 1\leq j\leq m.$$

Consequently, we have

$$\|\mathbf{V}_d^{\mathrm{T}}\boldsymbol{\Omega}^{\mathrm{T}}\boldsymbol{\Omega}(\mathbf{A}-[\![\mathbf{A}]\!]_d)^{\mathrm{T}}\|_{\mathrm{F}}^2 = \sum_{i=1}^{d}\sum_{j=1}^{m}|\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Omega}^{\mathrm{T}}\boldsymbol{\Omega}\mathbf{y}_j|^2 \leq \tfrac{\varepsilon}{d}\sum_{i=1}^{d}\sum_{j=1}^{m}\|\mathbf{x}_i\|_2^2\|\mathbf{y}_j\|_2^2 = \tfrac{\varepsilon}{d}\|\mathbf{V}_d\|_{\mathrm{F}}^2\|\mathbf{A}-[\![\mathbf{A}]\!]_d\|_{\mathrm{F}}^2 \tag{3.6}$$

with probability at least $1-\delta$. The proof is finished by noting that $\|\mathbf{V}_d\|_{\mathrm{F}}^2 = d$.

It is concluded that (3.2) and as a consequence (3.10) are satisfied with probability at least $1-2\delta$ if $\boldsymbol{\Omega}$ is an $(\frac{1}{3},\delta,d)$ OSE and $(\sqrt{\frac{\varepsilon}{d}},\frac{\delta}{N},1)$ OSE. In turn, according to Theorem 2.1 and (2.2), this condition is satisfied by the block as well as the standard SRHT with $l = \mathcal{O}(d\log\frac{n}{\delta}\log\frac{m}{\delta})$ rows (taking $\varepsilon^* = \frac{1}{2}$, $\varepsilon = \frac{2}{9}$). Whereas for Gaussian matrices the required number of rows to satisfy the aforementioned OSEs properties is somewhat lower: $l = \mathcal{O}(d\log\frac{m}{\delta})$. Although it has to be said that SRHT matrices in practice give similar results as Gaussian matrices [8]. As can be seen from our experiments, this should also be the case for block SRHT. Moreover, we note that the condition $l = \mathcal{O}(d\log\frac{n}{\delta})$ for Gaussian matrices is still pessimistic. This overestimation is an artifact due to the use of a general analysis based solely on the OSE property. In reality, a Gaussian $\boldsymbol{\Omega}$ should satisfy (3.2) with high probability if it has size $l = \mathcal{O}(d)$ with a small constant (say 2 or 4) [8, 21].

3.2 Nyström approximation

Although being more efficient than the deterministic SVD, the RSVD still can be computationally heavy, especially on distributed architectures, as it requires two passes over $\mathbf{A}$, and a multiplication of $\mathbf{A}$ by $\mathbf{Q}$, which is a large dense matrix. Next we discuss improved algorithm that can circumvent these drawbacks. Assume that $\mathbf{A}$ is positive semi-definite matrix.

Notice that it can be computationally beneficial to change the norm $\|\cdot\|_\xi$ in (3.1) to its sketched estimate $\|\boldsymbol{\Omega}\cdot\boldsymbol{\Omega}^{\mathrm{T}}\|_\xi$. The accuracy of such an estimation can be guaranteed thanks to the fact that $\boldsymbol{\Omega}$ is an OSE. This leads to Nyström approximation $[\![\mathbf{A}]\!]^{(\mathrm{Nyst})}$ given below

$$[\![\mathbf{A}]\!]^{(\mathrm{Nyst})} := \arg\min_{\mathrm{range}(\mathbf{B})\subseteq Q}\|\boldsymbol{\Omega}(\mathbf{A}-\mathbf{B})\boldsymbol{\Omega}^{\mathrm{T}}\|_\xi, \tag{3.7}$$

or in a more usual form [13, 21, 22, 30, 31]:

$$[\![\mathbf{A}]\!]^{(\mathrm{Nyst})} = (\mathbf{\Omega A})^{\mathrm{T}}(\mathbf{\Omega A \Omega}^{\mathrm{T}})^{\dagger}(\mathbf{\Omega A}),$$

where $(\mathbf{\Omega A \Omega}^{\mathrm{T}})^{\dagger}$ denotes the pseudo-inverse of $\mathbf{\Omega A \Omega}^{\mathrm{T}}$. Then a rank-$k$ approximation of $\mathbf{A}$ can be obtained by an SVD of $[\![\mathbf{A}]\!]^{(\mathrm{Nyst})}$, which leads to the approximation $[\![\mathbf{A}]\!]_k^{(\mathrm{Nyst})} := [\![[\![\mathbf{A}]\!]^{(\mathrm{Nyst})}]\!]_k$. This way of obtaining a rank-k approximation from $[\![\mathbf{A}]\!]^{(\mathrm{Nyst})}$ is referred to as the modified fixed-rank Nyström via QR [32, 33, 21] Algorithm 2 describes a way for computing $[\![\mathbf{A}]\!]_k^{(\mathrm{Nyst})}$ suited for distributed computing under 2D partitioning of $\mathbf{A}$. The matrices $\mathbf{Y}$ and $\mathbf{\Omega Y}$ can be computed with the procedure from Section 2 using the block structure of $\mathbf{\Omega}$. The QR factorization $\mathbf{Z} = \tilde{\mathbf{Q}}\mathbf{R}$ in step 4 can be computed with TSQR. Note that in step 6, instead of computing $\hat{\mathbf{U}}_k$ as $(\mathbf{Y}\tilde{\mathbf{V}}_k)\tilde{\mathbf{\Sigma}}_k^{-1}$ we could use $\hat{\mathbf{U}}_k = \tilde{\mathbf{Q}}\tilde{\mathbf{U}}_k$, which would provide more numerical stability but entail a larger computational cost. Algorithm 2 needs only one pass over the matrix $\mathbf{A}$, and does not involve any high-dimensional operations on $\mathbf{A}$ except the computation of the sketch $\mathbf{Y} = \mathbf{A\Omega}^{\mathrm{T}}$, which implies its superiority over the standard SVD as well as randomized SVD [8, 21]. In fact, the dominant computational cost of Algorithm 2 is associated with computing $\mathbf{Y}$ and $\mathbf{\Omega Y}$ in steps 1 and 2, when r is sufficiently large.

Algorithm 2 Randomized Nyström approximation

Require: $n \times n$ matrix $\mathbf{A}$, $l \times n$ matrix $\mathbf{\Omega}$ with $l \ll n$, the target rank k.

1: Compute $\mathbf{Y} = \mathbf{A\Omega}^{\mathrm{T}}$.
2: Obtain a Cholesky factor $\mathbf{C}$ of $\mathbf{\Omega Y}$.
3: Compute $\mathbf{Z} = \mathbf{YC}^{-1}$ with backward substitution.
4: Obtain the R factor $\mathbf{R}$ of $\mathbf{Z}$ (with TSQR or similar algorithm).
5: Use SVD to compute the best rank-k approximation $\tilde{\mathbf{U}}_k\tilde{\mathbf{\Sigma}}_k\tilde{\mathbf{V}}_k^{\mathrm{T}}$ of $\mathbf{R}$.
6: Compute $\hat{\mathbf{U}}_k = (\mathbf{Y}\tilde{\mathbf{V}}_k)\tilde{\mathbf{\Sigma}}_k^{-1}$.
7: Output factorization $[\![\mathbf{A}]\!]_k^{(\mathrm{Nyst})} = \hat{\mathbf{U}}_k\tilde{\mathbf{\Sigma}}_k^2\hat{\mathbf{U}}_k^{\mathrm{T}}$.

Remark 3.1. *The matrix $\mathbf{\Omega A \Omega}^{\mathrm{T}}$ can be rank-deficient, for instance, if $\mathbf{A}$ or $\mathbf{\Omega}$ have lower rank than l, which will cause a problem for obtaining a Cholesky factorization in step 2. In this case, a remedy can be to compute an SVD instead of the Cholesky factorization, and take $\mathbf{C}$ as a square root of $\mathbf{\Omega Y}$ in SVD form, that then can be used for the pseudo-inversion in step 3. Another possibility is to make $\mathbf{A}$ full-rank by using shifting as in [34].*

Let us now characterize the accuracy of Nyström approximation. Notice the following identity [35]:

$$\mathbf{A} - [\![\mathbf{A}]\!]^{(\mathrm{Nyst})} = (\mathbf{A}^{\frac{1}{2}} - [\![\mathbf{A}^{\frac{1}{2}}]\!]^{(\mathrm{RSVD})})^{\mathrm{T}}(\mathbf{A}^{\frac{1}{2}} - [\![\mathbf{A}^{\frac{1}{2}}]\!]^{(\mathrm{RSVD})}).$$

By combining this observation with the derived earlier results on RSVD with $\mathbf{A} \leftarrow \mathbf{A}^{\frac{1}{2}}$, we obtain that

$$\|\mathbf{A} - [\![\mathbf{A}]\!]^{(\mathrm{Nyst})}\|_* = \|\mathbf{A}^{\frac{1}{2}} - [\![\mathbf{A}^{\frac{1}{2}}]\!]^{(\mathrm{RSVD})}\|_{\mathrm{F}}^2 \leq (1+\varepsilon^*)\|\mathbf{A}^{\frac{1}{2}} - [\![\mathbf{A}^{\frac{1}{2}}]\!]_d\|_{\mathrm{F}}^2 = (1+\varepsilon^*)\|\mathbf{A} - [\![\mathbf{A}]\!]_d\|_* \tag{3.8}$$

holds with probability at least $1-2\delta$ if $\mathbf{\Omega}$ is $(\frac{1}{3}, \delta, d)$ OSE and $(\sqrt{\frac{\varepsilon}{d}}, \frac{\delta}{N}, 1)$ OSE. It is then noticed that (3.8) also implies the accuracy of the truncated approximation $[\![\mathbf{A}]\!]_k^{(\mathrm{Nyst})}$ due to the following consequence of the triangle inequality (see for instance [36, Proposition A.6]):

$$\|\mathbf{A} - [\![\mathbf{A}]\!]_k^{(\mathrm{Nyst})}\|_\xi \leq \|\mathbf{A} - [\![\mathbf{A}]\!]_k\|_\xi + 2\|\mathbf{A} - [\![\mathbf{A}]\!]^{(\mathrm{Nyst})}\|_\xi, \tag{3.9}$$

where $\xi = 2$ or $*$, so that we have by (3.8),

$$\|\mathbf{A} - [\![\mathbf{A}]\!]_k^{(\mathrm{Nyst})}\|_\xi \leq \|\mathbf{A} - [\![\mathbf{A}]\!]_k\|_\xi + 3\|\mathbf{A} - [\![\mathbf{A}]\!]_d\|_*. \tag{3.10}$$

Similarly to RSVD approximation, this relation guarantees the quasi-optimality of $[\![\mathbf{A}]\!]_k^{(\mathrm{Nyst})}$ with respect to the trace norm. Furthermore, $[\![\mathbf{A}]\!]_k^{(\mathrm{Nyst})}$ is almost as accurate as $[\![\mathbf{A}]\!]_k$ with respect to both the trace norm and the spectral norm, if the tail after d-th singular value of $\mathbf{A}$, i.e. $\sum_{i=d+1}^{n}\sigma_i$ is small compared to the $k+1$-th singular value σ_{k+1}.

3.3 Single-view approximation of non-psd matrix

The Nyström method is applicable only when $\mathbf{A}$ is positive semi-definite. Next, we describe a single-view algorithm that works with general matrices. In recent years several such algorithms have been proposed [8, 23, 36, 24, 25]. The single-view approximation from [24] involves four sketching matrices. Two of them, $\mathbf{\Omega}\in\mathbb{R}^{l\times n}$, and $\mathbf{\Gamma}\in\mathbb{R}^{l\times m}$, are used to construct the approximation subspaces $Q=\mathrm{range}(\mathbf{A}\mathbf{\Omega}^{\mathrm{T}})$ and $P=\mathrm{range}(\mathbf{A}^{\mathrm{T}}\mathbf{\Gamma}^{\mathrm{T}})$ capturing the actions of $\mathbf{A}$ and $\mathbf{A}^{\mathrm{T}}$. Whereas the other two, $\mathbf{\Phi}\in\mathbb{R}^{s\times m}$, and $\mathbf{\Psi}\in\mathbb{R}^{s\times n}$, with $l\le s$, provide the "core sketch" of $\mathbf{A}$. In our projection-based interpretation, the "core sketch" corresponds to estimation of the norm $\|\cdot\|_\xi$ of the residual by $\|\mathbf{\Phi}\cdot\mathbf{\Psi}^{\mathrm{T}}\|_\xi$. Thus, we arrive to the following low-rank approximation of $\mathbf{A}$:

$$[\![\mathbf{A}]\!]^{(\mathrm{sRSVD})}=\arg\min_{\substack{\mathrm{range}(\mathbf{B})\subseteq Q,\\ \mathrm{range}(\mathbf{B}^{\mathrm{T}})\subseteq P}}\|\mathbf{\Phi}(\mathbf{A}-\mathbf{B})\mathbf{\Psi}^{\mathrm{T}}\|_\xi. \tag{3.11}$$

Notice that $[\![\mathbf{A}]\!]^{(\mathrm{sRSVD})}$ is given as $\mathbf{Q}(\mathbf{\Phi}\mathbf{Q})^\dagger(\mathbf{\Phi}\mathbf{A}\mathbf{\Psi}^{\mathrm{T}})(\mathbf{\Psi}\mathbf{P})^\dagger\mathbf{P}^{\mathrm{T}}$, where $\mathbf{Q}$ denotes an orthogonal basis for Q and $\mathbf{P}$ denotes an orthogonal basis for P. Similarly as in the case of RSVD and Nyström approximations, the rank-k approximation of $\mathbf{A}$ can be obtained by truncating $[\![\mathbf{A}]\!]^{(\mathrm{sRSVD})}$ with SVD, which provides $[\![\mathbf{A}]\!]_k^{(\mathrm{sRSVD})}=[\![[\![\mathbf{A}]\!]^{(\mathrm{sRSVD})}]\!]_k$ [24]. Notice that $[\![\mathbf{A}]\!]_k^{(\mathrm{sRSVD})}$ is now given as $\mathbf{Q}[\![\mathbf{C}]\!]_k\mathbf{P}^{\mathrm{T}}$, where $\mathbf{C}=(\mathbf{\Phi}\mathbf{Q})^\dagger(\mathbf{\Phi}\mathbf{A}\mathbf{\Psi}^{\mathrm{T}})(\mathbf{\Psi}\mathbf{P})^\dagger$. This observation leads to Algorithm 3 for the computation of $[\![\mathbf{A}]\!]_k^{(\mathrm{RSVD})}$. Again, in step 2 one can use TSQR algorithm for the efficient QR factorization on distributed architectures. As in Nyström method, the computational cost of Algorithm 3 is dominated by the applications of sketching matrices.

Algorithm 3 Single-view RSVD [1, Algorithm 17]

Require: $m\times n$ matrix $\mathbf{A}$, matrices $\mathbf{\Omega},\mathbf{\Gamma}$ with l rows, and $\mathbf{\Phi},\mathbf{\Psi}$ with s rows, with $l\le s\ll m$, the target rank k.

1: Compute $\mathbf{X}=\mathbf{A}^{\mathrm{T}}\mathbf{\Gamma}^{\mathrm{T}}$, $\mathbf{Y}=\mathbf{A}\mathbf{\Omega}^{\mathrm{T}}$ and $\mathbf{Z}=\mathbf{\Phi}\mathbf{A}\mathbf{\Psi}^{\mathrm{T}}$.
2: Orthogonalize $\mathbf{X}$ and $\mathbf{Y}$ with a QR factorization and obtain $\mathbf{Q}$ and $\mathbf{P}$.
3: Compute $\mathbf{\Phi}\mathbf{Q}$ and $\mathbf{\Psi}\mathbf{P}$.
4: Compute the core matrix $\mathbf{C}=(\mathbf{\Phi}\mathbf{Q})^\dagger\mathbf{Z}(\mathbf{\Psi}\mathbf{P})^\dagger$ with least-squares solves.
5: Use SVD to compute the best rank-k approximation $\tilde{\mathbf{U}}_k\tilde{\mathbf{\Sigma}}_k\tilde{\mathbf{V}}_k^{\mathrm{T}}$ of $\mathbf{C}$.
6: Output factorization $[\![\mathbf{A}]\!]_k^{(\mathrm{sRSVD})}=(\mathbf{Q}\tilde{\mathbf{U}}_k)\tilde{\mathbf{\Sigma}}_k(\mathbf{P}\tilde{\mathbf{V}}_k)$.

To characterize the accuracy of $[\![\mathbf{A}]\!]_k^{(\mathrm{sRSVD})}$ we shall assume that $\mathbf{\Phi},\mathbf{\Psi}$ satisfy both (ε,δ,l) OSE and $(\varepsilon,\frac{\delta}{n},1)$ OSE properties with $\varepsilon\le\frac{2}{9}$. Then these sketching matrices are ε-embeddings for Q and P, i.e. they satisfy (1.1) taking $\mathbf{\Omega}\leftarrow\mathbf{\Phi},V\leftarrow Q$ or $\mathbf{\Omega}\leftarrow\mathbf{\Psi},V\leftarrow P$, simultaneously, with probability at least $1-2\delta$. Moreover, let $\mathbf{\Pi}_Q$ and $\mathbf{\Pi}_P$ denote the orthogonal projectors onto Q and P. Then $\mathbf{\Psi}$ satisfies the ε-embedding property for every subspace spanned by a row of $\mathbf{A}-\mathbf{\Pi}_Q\mathbf{A}\mathbf{\Pi}_P$ simultaneously with probability at least $1-\delta$, and $\mathbf{\Phi}$ satisfies the ε-embedding property for every subspace spanned by a column of $(\mathbf{A}-\mathbf{\Pi}_Q\mathbf{A}\mathbf{\Pi}_P)\mathbf{\Psi}^{\mathrm{T}}$ simultaneously with probability at least $1-\delta$, so that

$$\|\mathbf{\Phi}(\mathbf{A}-\mathbf{\Pi}_Q\mathbf{A}\mathbf{\Pi}_P)\mathbf{\Psi}^{\mathrm{T}}\|_{\mathrm{F}}\le\sqrt{1+\varepsilon}\|(\mathbf{A}-\mathbf{\Pi}_Q\mathbf{A}\mathbf{\Pi}_P)\mathbf{\Psi}^{\mathrm{T}}\|_{\mathrm{F}}\le(1+\varepsilon)\|\mathbf{A}-\mathbf{\Pi}_Q\mathbf{A}\mathbf{\Pi}_P\|_{\mathrm{F}}$$

holds with probability at least $1-2\delta$. Thus, we have with probability at least $1-4\delta$,

$$\begin{aligned}
&\|\mathbf{A}-[\![\mathbf{A}]\!]^{(\mathrm{sRSVD})}\|_{\mathrm{F}} \leq \|\boldsymbol{\Pi}_Q\mathbf{A}\boldsymbol{\Pi}_P-[\![\mathbf{A}]\!]^{(\mathrm{sRSVD})}\|_{\mathrm{F}}+\|\mathbf{A}-\boldsymbol{\Pi}_Q\mathbf{A}\boldsymbol{\Pi}_P\|_{\mathrm{F}}\\
&\leq \frac{1}{1-\varepsilon}\|\boldsymbol{\Phi}(\boldsymbol{\Pi}_Q\mathbf{A}\boldsymbol{\Pi}_P-[\![\mathbf{A}]\!]^{(\mathrm{sRSVD})})\boldsymbol{\Psi}^{\mathrm{T}}\|_{\mathrm{F}}+\|\mathbf{A}-\boldsymbol{\Pi}_Q\mathbf{A}\boldsymbol{\Pi}_P\|_{\mathrm{F}}\\
&\leq \frac{1}{1-\varepsilon}\|\boldsymbol{\Phi}(\mathbf{A}-[\![\mathbf{A}]\!]^{(\mathrm{sRSVD})})\boldsymbol{\Psi}^{\mathrm{T}}\|_{\mathrm{F}}+2.6\|\mathbf{A}-\boldsymbol{\Pi}_Q\mathbf{A}\boldsymbol{\Pi}_P\|_{\mathrm{F}}\\
&\leq \frac{1}{1-\varepsilon}\|\boldsymbol{\Phi}(\mathbf{A}-\boldsymbol{\Pi}_Q\mathbf{A}\boldsymbol{\Pi}_P)\boldsymbol{\Psi}^{\mathrm{T}}\|_{\mathrm{F}}+2.6\|\mathbf{A}-\boldsymbol{\Pi}_Q\mathbf{A}\boldsymbol{\Pi}_P\|_{\mathrm{F}}\\
&\leq 4.2\|\mathbf{A}-\boldsymbol{\Pi}_Q\mathbf{A}\boldsymbol{\Pi}_P\|_{\mathrm{F}} \leq 4.2(\|\mathbf{A}-\boldsymbol{\Pi}_Q\mathbf{A}\|_{\mathrm{F}}+\|\mathbf{A}-\mathbf{A}\boldsymbol{\Pi}_P\|_{\mathrm{F}}),
\end{aligned}$$

which in turn guarantees the accuracy of $[\![\mathbf{A}]\!]^{(\mathrm{sRSVD})}$ if Q and P capture well the actions of $\mathbf{A}$ and $\mathbf{A}^{\mathrm{T}}$.

Clearly, $\boldsymbol{\Pi}_Q\mathbf{A}=[\![\mathbf{A}]\!]^{(\mathrm{RSVD})}$ and $\mathbf{A}\boldsymbol{\Pi}_P=([\![\mathbf{A}^{\mathrm{T}}]\!]^{(\mathrm{RSVD})})^{\mathrm{T}}$, where $[\![\mathbf{A}^{\mathrm{T}}]\!]^{(\mathrm{RSVD})}$ is the RSVD approximation of $\mathbf{A}^{\mathrm{T}}$, associated with the sketching matrix $\boldsymbol{\Gamma}$. This suggests that the single-view approximation (3.11) should have a similar quality as the RSVD approximation (3.1). By combining the above consideration with the results on RSVD from Section 3.1 and the triangle inequality we obtain that

$$\begin{aligned}
\|\mathbf{A}-[\![\mathbf{A}]\!]_k^{(\mathrm{sRSVD})}\|_\xi &\leq \|\mathbf{A}-[\![\mathbf{A}]\!]_k^{(\mathrm{sRSVD})}\|_\xi+2\|\mathbf{A}-[\![\mathbf{A}]\!]^{(\mathrm{sRSVD})}\|_{\mathrm{F}}\\
&\leq \|\mathbf{A}-[\![\mathbf{A}]\!]_k^{(\mathrm{sRSVD})}\|_\xi+16.8\sqrt{1+2.25\varepsilon}\|\mathbf{A}-[\![\mathbf{A}]\!]_d\|_{\mathrm{F}}
\end{aligned}$$

holds with probability at least $1-6\delta$ for $\xi=2,*$ or F, given that $\boldsymbol{\Phi},\boldsymbol{\Psi}$ satisfy both (ε,δ,l) OSE and $(\varepsilon,\frac{\delta}{n},1)$ OSE properties, and $\boldsymbol{\Omega},\boldsymbol{\Gamma}$ satisfy both $(\frac{1}{3},\delta,d)$ OSE and $(\sqrt{\frac{\varepsilon}{d}},\frac{\delta}{N},1)$ OSE properties, with $\varepsilon\leq\frac{2}{9}$ and $N=2nd+n+d$. According to Theorem 2.1, these conditions can be satisfied by $\boldsymbol{\Omega},\boldsymbol{\Gamma}$ that are block SRHT matrices of size $l=\mathcal{O}(d\log^2\frac{n}{\delta})$, and $\boldsymbol{\Phi},\boldsymbol{\Psi}$ that are block SRHT matrices of size $s=\mathcal{O}(d\log^3\frac{n}{\delta})$.

4 Numerical experiments

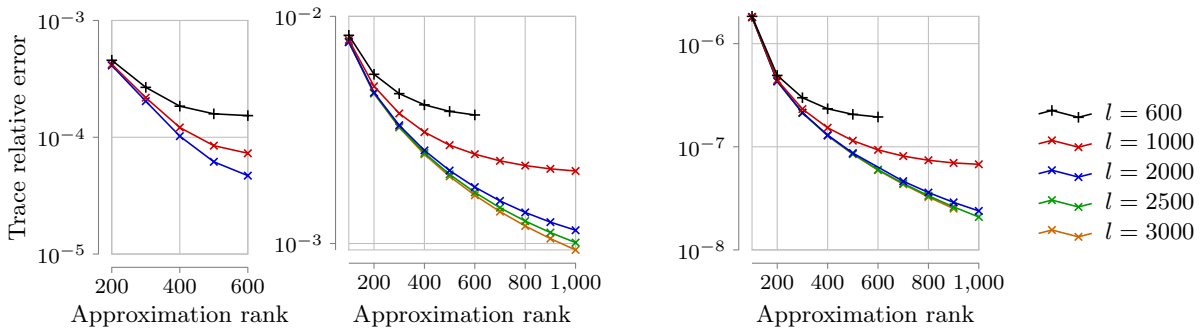
For numerical experiments, we chose the Nyström approximation as a representative application. The validation of block SRHT is done through comparison with Gaussian embeddings. In the plots, BSRHT refers to block SRHT. The comparison with standard SRHT is impertinent since SRHT matrices are not that well scalable as Gaussian matrices and have no better accuracy [28].

4.1 Nyström approximation

This experiment was executed with Julia programming language version 1.7.2 along with the Distributed.jl and DistributedArrays.jl packages for parallelism. We used 2 nodes Intel Skylake 2.7GHz (AVX512) having 48 available cores and 180 MB of RAM each. In this experiment we used only 32 cores on each node. As input data we took the MNIST or YearPredictionMSD datasets [37, 38]. The radial basis function $e^{-\|x_i-x_j\|^2/\sigma^2}$ was used to build a dense positive definite matrix $\mathbf{A}$ of size $n\times n$ from n rows of the input data. The parameter σ was chosen as 100 for the MNIST dataset and 10^4 as well as 10^5 for the YearPredictionMSD dataset. The dimension n was taken as 65536. The matrix $\mathbf{A}$ has been uniformly distributed on a square grid of 8×8 processors. In all the experiments, the local matrices $\boldsymbol{\Omega}^{(i)}$ on each processor were generated with a seeded random number generator with a low communication cost.

Figure 1 depicts the convergence of the error of the low-rank approximation obtained with Algorithm 2 taking $\boldsymbol{\Omega}$ as a block SRHT. The results for Gaussian $\boldsymbol{\Omega}$ are practically identical and therefore are not displayed.

In this numerical experiment, the error is measured with the trace norm. Different sketching sizes l were tested. For each pair of parameters (l,k) 20 different approximations were computed for each type of $\boldsymbol{\Omega}$, in order to have



(a) Dataset `mnist` **(b)** Dataset `year` with $\sigma = 10^4$ **(c)** Dataset `year` with $\sigma = 10^5$

Figure 1: Trace error $\|\mathbf{A} - [\![\mathbf{A}]\!]_k^{(\mathrm{Nyst})}\|_*/\|\mathbf{A}\|_*$ using BSRHT.

the 95% confidence interval. Nevertheless this interval is not displayed as it is too small to be visible. Figure 2 gives runtime characterization. In particular we depict the runtime spent on computing $\mathbf{Y} = \mathbf{A}\mathbf{\Omega}^{\mathrm{T}}$ and $\mathbf{\Omega}\mathbf{Y}$ in steps 1 and 2 of Algorithm 2. These operations will dominate the overall computational cost, when the block size is large enough. Nevertheless the reader should be aware that TSQR and the SVD of $\mathbf{R}$ (step 4 and 5) are also important, especially when the sampling size is close to the block size. The parameter k is not involved in steps 1 and 3 hence not mentioned in Figure 2.

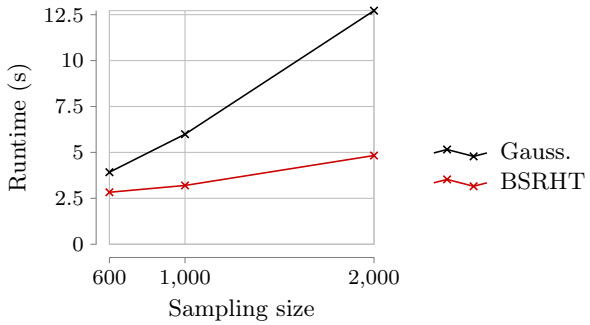


Figure 2: Runtimes of computing $\mathbf{Y} = \mathbf{A}\mathbf{\Omega}^{\mathrm{T}}$ and $\mathbf{\Omega}\mathbf{Y}$ in Algorithm 2 for different sampling sizes.

According to Figure 2, the runtime of the Gaussian sampling is up to 2.5 times higher and grows faster with l than the runtime taken by block SRHT. Note that for block SRHT the local computation cost is independent of l, hence the slope comes only from the reductions in steps 1 and 2. On the other hand, the Gaussian sampling involves local computations with linear dependency in l, in addition to these reductions.

4.2 Cost of application to tall-and-skinny matrix

Next we investigate the performance of block SRHT on larger scale. For this we consider a product of $\mathbf{\Omega}$ with a tall-and-skinny matrix $\mathbf{V}$, for instance in the context of solving an overdetermined least-squares problem. The same computing environment is used as in the previous experiment, involving now up to 32 nodes and using C99/MPI

instead of Julia. The code was compiled using IntelMPI C compiler version 20.0.2 and sequential MKL 20.0.2 with option ILP64. The library FFTW3 used has Intel-specific routines. There is therefore up to 1536 cores available. In this way we generated a random matrix $\mathbf{V}$ with $d = 200$ columns and a variable number n of rows. This matrix was distributed among a variable number p of processors with block rowwise partitioning. Then $\mathbf{V}$ was multiplied by either a Gaussian or block SRHT matrix $\mathbf{\Omega}$ with $l = 2000$ rows using (2.4). In all experiments, the local $\mathbf{\Omega}^{(i)}$ matrices on each processor were generated with a seeded random number generator with negligible communication cost. Figure 3 presents a strong scalability test for $n = 10^7$. We see that the block SRHT provides an overall speedup by a factor of more than 2.5 over the Gaussian matrices, while demonstrating as good scalability when $p \leq 384$. For larger p, however, the reduction operation starts to dominate, which reduces the gain in efficiency. We observe a variability in the MPI_Allreduce operation on larger number of processors for both Gaussian and block SRHT algorithms. However the compute times for both algorithms scale well when increasing the number of processors up to $p = 1536$. Figure 4 shows a strong scalability test for a higher dimension $n = 10^8$. Again we see

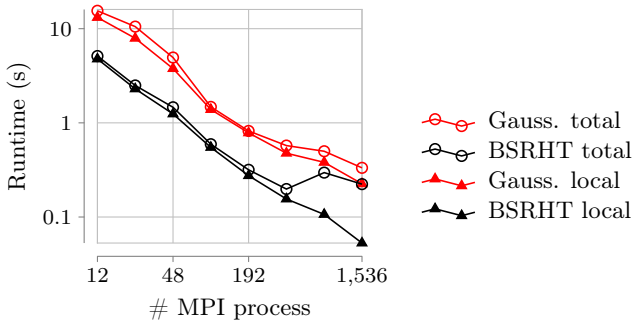


Figure 3: Strong scalability runtimes associated with computing $\mathbf{\Omega V}$ with $n = 10^7$ and $l = 2000$, versus p. "Gauss. total" and "BSRHT total" correspond to the overall runtimes, whereas "Gauss. local" and "BSRHT local" stand for the max per-processor runtimes taken by local multiplications.

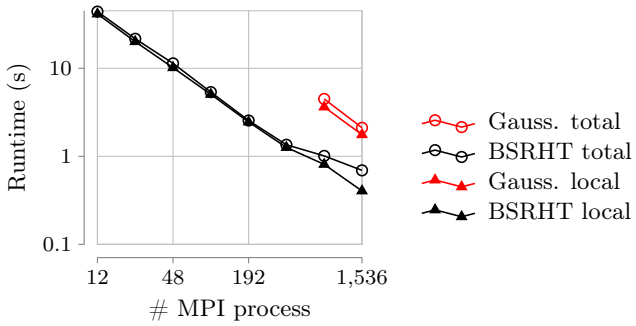


Figure 4: Strong scalability runtimes associated with computing $\mathbf{\Omega V}$ with $n = 10^8$ and $l = 2000$, versus p. "Gauss. total" and "BSRHT total" correspond to the overall runtimes, whereas "Gauss. local" and "BSRHT local" stand for the max per-processor runtimes taken by local multiplications.

a great scalability of block SRHT for $p \leq 384$. For Gaussian matrices, on the other hand, we revealed issues with reaching the memory limit needed to store $\mathbf{\Omega}^{(i)}$ which made its application on $p \leq 384$ processors infeasible. In principle, this problem can be overcome by generating $\mathbf{\Omega}^{(i)}$ blockwise and applying the blocks to $\mathbf{V}^{(i)}$ "on the fly".

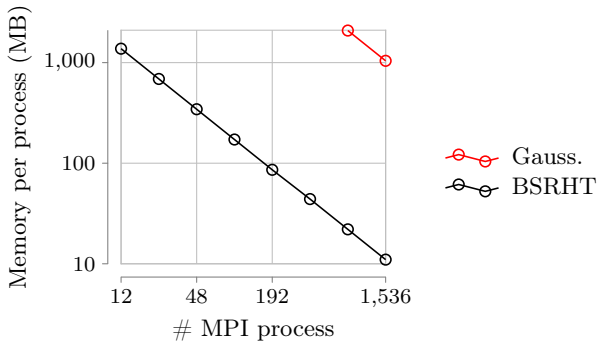


Figure 5: Max per-processor memory needed for computing $\mathbf{\Omega V}$ with $n = 10^8$ and $l = 2000$, versus p.

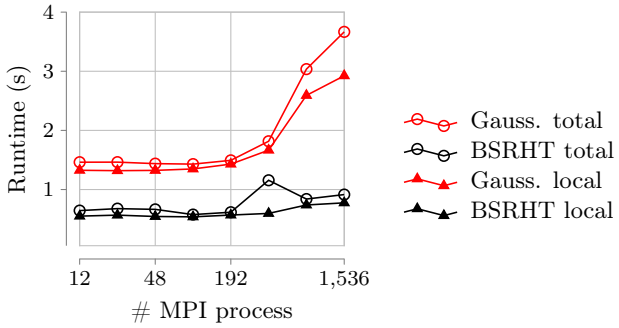


Figure 6: Weak scalability runtimes associated with computing $\mathbf{\Omega V}$ with $n = 10^5 \times p$ and $l = 2000$, versus p. "Gauss. total" and "BSRHT total" correspond to the overall runtimes, whereas "Gauss. local" and "BSRHT local" stand for the max per-processor runtimes taken by local multiplications.

This however entails a dramatic increase in runtime and therefore is omitted in comparison. On the other hand, for block SRHT we do not have any memory problems[1]. To quantify the advantage of block SRHT in such context, in Figure 5 we provide the memory consumption of the Gaussian and block SRHT matrix. We see that in this sense the reduction in computational cost is indeed drastic. Finally, Figure 6 provides a weak scalability test using $n = rp$ where $r = 10^5$. We again see a reduction in runtime of about 2.5 and good scalability for block SRHT up to using $p = 1536$ processors, similar as in the strong scalability test.

5 Proof of the main theorem

Before providing the proof for Theorem 2.1, let us first motivate the chosen proof path. Le $\mathbf{V}$ be a fixed $n \times d$ matrix with orthonormal columns, partitioned using block rowwise partitioning with p blocks $\mathbf{V}^{(i)}$ of size $r \times d$. The statement of the theorem can then be proven by showing that the singular values of $\mathbf{\Omega V}$ belong to the interval $[\sqrt{1-\varepsilon}, \sqrt{1+\varepsilon}]$ with probability at least $1 - \delta$.

Assume for a moment, that $\mathbf{R}$ in (2.3) is a uniform sampling matrix *without replacement*. Notice that the random sampling of rows without replacement and then flipping their signs is equivalent to first flipping the signs

[1]To reduce the memory consumption, local matrices $\mathbf{V}^{(i)}$ are multiplied by $\mathbf{\Omega}^{(i)}$ in blocks of 20 columns.

and then sampling. By using this consideration, the expression (2.4) can be developed further as

$$\begin{aligned}\mathbf{\Omega V} &= \sqrt{\frac{r}{l}}\sum_{i=1}^{p}\left(\widetilde{\mathbf{D}}^{(i)}\mathbf{R}\mathbf{H}\mathbf{D}^{(i)}\mathbf{V}^{(i)}\right) = \sqrt{\frac{r}{l}}\sum_{i=1}^{p}\left(\mathbf{R}\widehat{\mathbf{D}}^{(i)}\mathbf{H}\mathbf{D}^{(i)}\mathbf{V}^{(i)}\right)\\ &= \sqrt{\frac{r}{l}}\mathbf{R}\sum_{i=1}^{p}\left(\widehat{\mathbf{D}}^{(i)}\mathbf{H}\mathbf{D}^{(i)}\mathbf{V}^{(i)}\right) = \sqrt{\frac{r}{l}}\mathbf{R}\mathbf{W}\mathbf{V},\end{aligned} \tag{5.1}$$

where $\widehat{\mathbf{D}}^{(i)}$ are $r \times r$ diagonal matrices with Rademacher random variables ± 1 on the diagonal, and $\mathbf{W} = \left[\widehat{\mathbf{D}}^{(1)}\mathbf{H}\mathbf{D}^{(1)}, \widehat{\mathbf{D}}^{(2)}\mathbf{H}\mathbf{D}^{(2)}, \ldots, \widehat{\mathbf{D}}^{(l)}\mathbf{H}\mathbf{D}^{(l)}\right]$.

Looking at (2.4), one can detect many similarities of $\mathbf{\Omega}$ with standard SRHT matrix. Consequently, in order to argue that $\mathbf{\Omega V}$ is approximately orthonormal, the first thing to try should be to follow the steps from [26] in the analysis of the original SRHT. In this case the proof recipe would be as follows. First, it could be shown that the matrix $\mathbf{W}$ with high probability homogenizes the rows of $\mathbf{V}$. This result then would allow the Matrix Chernoff concentration inequality from [26] to be applied to show that $\mathbf{WV}$ and $\sqrt{\frac{r}{l}}\mathbf{RWV}$ have approximately equal minimal and maximal singular values. With these results, it would remain to show that with high probability $\mathbf{WV}$ is approximately orthonormal. This, however, can be cumbersome or even impossible in some situations. Think, for example, of the situation when $r < d$. Therefore, we will assume that $\mathbf{R}$ is a uniform sampling matrix *with replacement* and use the following trick. For better presentation define parameters $\varepsilon^* = \frac{15}{16}\varepsilon$ and $\delta^* = \frac{\delta}{5}$.

Recall that the sampling matrix $\mathbf{R}$ restricts a vector $\mathbf{x} = (x_1, \ldots, x_r)$ to l coordinates, i.e., we have

$$\mathbf{R}\mathbf{x} = (x_{i_1}, \ldots, x_{i_l}), \text{ with } 1 \le i_1, \ldots, i_l \le r.$$

The (multi-)set of indices $\{i_1, \ldots, i_l\}$ is a uniform random sample of $\{1, \ldots, r\}$ with replacement. Notice that such sampling of indices is equivalent to the sampling uniformly at random with replacement from $\{1, \ldots, 1, 2, \ldots 2, \ldots, r, \ldots, r\}$ containing $K = \lceil 10^4 \frac{n^2}{r\delta^*} \rceil$ copies of each index. This observation implies that the sampling matrix $\mathbf{R}$ satisfies the identity $\mathbf{RH} = \widehat{\mathbf{R}}[\mathbf{H}\ \mathbf{H} \ldots \mathbf{H}]^{\mathrm{T}} = \widehat{\mathbf{R}}\widehat{\mathbf{H}}$, where $\widehat{\mathbf{R}}$ is uniform sampling, with replacement, matrix of size $l \times rK$, and $\widehat{\mathbf{H}}$ is a block matrix with K blocks of rows, each being equal to $\mathbf{H}$. For a vector $\mathbf{x} = (x_1, \ldots, x_{rK})$, matrix $\widehat{\mathbf{R}}$ satisfies

$$\widehat{\mathbf{R}}\mathbf{x} = (x_{i_1}, \ldots, x_{i_l}), \text{ with } 1 \le i_1, \ldots, i_l \le rK, \tag{5.2}$$

where the indices $\{i_1, \ldots, i_l\}$ are drawn uniformly at random *with replacement* from $\{1, \ldots, rK\}$. Let $\mathcal{S}$ denote the event when $i_1, \ldots, i_l$ in (5.2) are all disjoint indices.

Lemma 5.1. *$\mathcal{S}$ occurs with probability at least $1 - \delta^*$.*

Proof. There are in total $\frac{(rK)^l}{l!}$ ways to select l elements from a (rK)-element set and $\binom{rK}{l} = \frac{rK(rK-1)\ldots(rK-l+1)}{l!}$ ways to select l disjoint elements. Consequently, we have

$$\mathbb{P}(S) = \prod_{i=1}^{l}(1 - \frac{i-1}{rK}) \ge (1 - \frac{l}{rK})^l \ge 1 - \frac{l^2}{rK} \ge 1 - \delta^*.$$

□

The goal will be to bound the singular values of $\mathbf{\Omega V}$ under the condition $\mathcal{S}$. The overall probability of success, then will follow by the union bound argument. Next is assumed that $\mathcal{S}$ is occurring. Notice that, in this case, matrix $\widehat{\mathbf{R}}$ is equivalent to the matrix that samples the entries uniformly at random and *without replacement*. Then, using the same arguments as in (5.1), we have the following expression for the product $\mathbf{\Omega V}$:

$$\mathbf{\Omega V} = \sqrt{\frac{r}{l}}\sum_{i=1}^{p}\left(\widetilde{\mathbf{D}}^{(i)}\widehat{\mathbf{R}}\widehat{\mathbf{H}}\mathbf{D}^{(i)}\mathbf{V}^{(i)}\right) = \sqrt{\frac{r}{l}}\widehat{\mathbf{R}}\widehat{\mathbf{W}}\mathbf{V}, \tag{5.3}$$

where $\widehat{\mathbf{W}}$ is a block matrix composed of $K \times p$ blocks, with the (j,i)-th block being $\widehat{\mathbf{D}}^{(i,j)}\mathbf{H}\mathbf{D}^{(i)}$, where $\widehat{\mathbf{D}}^{(i,j)}$ are diagonal matrices with entries i.i.d. Rademacher random variables ± 1. Unlike $\mathbf{W}\mathbf{V}$, the matrix $\widehat{\mathbf{W}}\mathbf{V}$ (rescaled by $1/\sqrt{K}$) for sufficiently large K can be proven to be approximately orthonormal with high probability. We are ready to establish the proof of Theorem 2.1.

Notice that the condition in Theorem 2.1 implies that

$$n \geq l \geq 3.2{\varepsilon^*}^{-2}(\sqrt{d} + \sqrt{8\log(rK/\delta^*)})^2 \log(d/\delta^*). \tag{5.4}$$

In Proposition 5.2 is shown that, given $\mathcal{S}$, the matrix $\widehat{\mathbf{W}}\mathbf{V}$ has rows with equilibrated norms.

Proposition 5.2. *Given $\mathcal{S}$. The rows $\varphi^{(j)}$ of $\widehat{\mathbf{W}}\mathbf{V}$ satisfy*

$$\mathbb{P}\left(\max_{j=1,\ldots,rK} \|\varphi^{(j)}\|_2 \leq \sqrt{\tfrac{d}{r}} + \sqrt{\tfrac{8\log(rK/\delta^*)}{r}}\right) \geq 1 - \delta^*.$$

Proof. Notice that, each row of $\widehat{\mathbf{W}}$ has entries that are i.i.d Rademacher random variables rescaled by $1/\sqrt{r}$. Consequently, we have

$$\varphi^{(j)} = {\xi^{(j)}}^{\mathrm{T}}\mathbf{V}/\sqrt{r},$$

where $\xi^{(j)}$ is a Rademacher vector. Define convex function $f(\mathbf{x}) = \|\mathbf{x}^{\mathrm{T}}\mathbf{V}/\sqrt{r}\|_2$. Observe that $f(\mathbf{x})$ satisfies the Lipschitz bound:

$$\forall \mathbf{x}, \mathbf{y},\ |f(\mathbf{x}) - f(\mathbf{y})| \leq \|\mathbf{x}-\mathbf{y}\|_2 \|\mathbf{V}/\sqrt{r}\|_2 = \|\mathbf{x}-\mathbf{y}\|_2/\sqrt{r}.$$

This allows to apply the Rademacher tail bound

$$\mathbb{P}\left(f(\xi^{(j)}) \geq \mathbb{E}f(\xi^{(j)}) + t/\sqrt{r}\right) \leq \exp(-t^2/8),\ \forall t \geq 0. \tag{5.5}$$

Observe that $\mathbb{E}(f(\xi^{(j)})) \leq (\mathbb{E}(f(\xi^{(j)})^2))^{\frac{1}{2}} = \|\mathbf{V}/\sqrt{r}\|_{\mathrm{F}} \leq \sqrt{d/r}$. The statement of the lemma follows by combining this relation with (5.5) with $t = \sqrt{8\log\frac{1}{\delta}}$ and using the union bound argument. □

In Proposition 5.3 is proven that $\frac{1}{\sqrt{K}}\widehat{\mathbf{W}}\mathbf{V}$ with high probability has singular values close to 1.

Proposition 5.3. *Given $\mathcal{S}$. The singular values of $\frac{1}{\sqrt{K}}\widehat{\mathbf{W}}\mathbf{V}$ with probability at least $1-\delta^*$ lie inside the interval $[\sqrt{1-\varepsilon^*/30}, \sqrt{1+\varepsilon^*/30}]$.*

Proof. Define $\tau = \varepsilon^*/30$. Notice that

$$K \geq 10^4 l \geq 7.87\tau^{-2}(6.9d + \log(r/\delta^*)).$$

We have, for any $\mathbf{x} \in \mathbb{R}^d$,

$$\|\widehat{\mathbf{W}}\mathbf{V}\mathbf{x}\|_2^2 = \sum_{j=1}^{K} \|\sum_{i=1}^{r} \widehat{\mathbf{D}}^{(i,j)}\mathbf{H}\mathbf{D}^{(i)}\mathbf{V}^{(i)}\mathbf{x}\|_2^2. \tag{5.6}$$

Denote by $\mathbf{d}^{(k,j)}$ a vector with i-th entry equal to the (k,k)-th entry of matrix $\widehat{\mathbf{D}}^{(i,j)}$, $1 \leq k \leq r$. Denote by $\mathbf{Z}^{(k)}$ the matrix with i-th row equal to the k-th row of matrix $\mathbf{H}\mathbf{D}^{(i)}\mathbf{V}^{(i)}$, $1 \leq k \leq r$. Notice the following relations:

$$\|\sum_{i=1}^{r} \widehat{\mathbf{D}}^{(i,j)}\mathbf{H}\mathbf{D}^{(i)}\mathbf{V}^{(i)}\mathbf{x}\|_2^2 = \sum_{k=1}^{r} \langle \mathbf{d}^{(k,j)}, \mathbf{Z}^{(k)}\mathbf{x}\rangle^2,\ 1 \leq j \leq K, \tag{5.7}$$

and

$$\sum_{k=1}^{r} \|\mathbf{Z}^{(k)}\mathbf{x}\|_2^2 = \|\mathbf{V}\mathbf{x}\|_2^2. \tag{5.8}$$

We have $\frac{1}{K}\sum_{j=1}^{K}\langle \mathbf{d}^{(k,j)}, \mathbf{Z}^{(k)}\mathbf{x}\rangle^2 = \|\mathbf{\Theta}\mathbf{Z}^{(k)}\mathbf{x}\|_2^2$, where $\mathbf{\Theta}$ is a $K \times l$ rescaled Rademacher matrix. By [11, Proposition 3.7], $\mathbf{\Theta}$ is an $(\tau, \delta^*/r, d)$ OSE, which implies that

$$\forall \mathbf{x} \in \mathbb{R}^d,\ \frac{1}{K}\sum_{j=1}^{K}\langle \mathbf{d}^{(k,j)}, \mathbf{Z}^{(k)}\mathbf{x}\rangle^2 = (1 \pm \tau)\|\mathbf{Z}^{(k)}\mathbf{x}\|_2^2,$$

holds with probability at least $1 - \delta^*/r$. By the summation over k and the union bound argument we conclude that

$$\forall \mathbf{x} \in \mathbb{R}^d,\ \frac{1}{K}\sum_{j=1}^{K}\sum_{k=1}^{r}\langle \mathbf{d}^{(k,j)}, \mathbf{Z}^{(k)}\mathbf{x}\rangle^2 = (1 \pm \tau)\sum_{k=1}^{r}\|\mathbf{Z}^{(k)}\mathbf{x}\|_2^2, \tag{5.9}$$

holds with probability at least $1 - \delta^*$. By straightforward substitution of the expressions (5.7) and (5.8) into (5.9), and using (5.6), we conclude that with probability at least $1 - \delta^*$,

$$\forall \mathbf{x} \in \mathbb{R}^d,\ \frac{1}{K}\|\widehat{\mathbf{W}}\mathbf{V}\mathbf{x}\|_2^2 = (1 \pm \tau)\|\mathbf{V}\mathbf{x}\|_2^2,$$

which is equivalent to the statement of the proposition. □

Proposition 5.4 presents a corollary of the Matrix Chernoff inequality from [26], used to show that $\mathbf{M} = \frac{1}{\sqrt{K}}\widehat{\mathbf{W}}\mathbf{V}$ and $\sqrt{\frac{rK}{l}}\widehat{\mathbf{R}}\mathbf{M} = \mathbf{\Omega}\mathbf{V}$ have approximately equal maximal and minimal singular values.

Proposition 5.4 (Corollary of Theorem 2.2 in [26])**.** *Let* $\mathbf{M}$ *be some* $rK \times d$ *matrix. Let* $0 < \varepsilon^* < 1$ *and* $0 < \delta^* < 1$. *Let* $\mathbf{m}^{(j)}$ *denote the rows of* $\mathbf{M}$ *and let* $M := rK \max_{j=1,\dots,rK}\|\mathbf{m}^{(j)}\|_2^2$ *and* $N \geq \sigma_{min}(\mathbf{M})^{-2}$. *Draw at random a sampling matrix* $\widehat{\mathbf{R}}$ *in* (5.2) *with*

$$l \geq 2(\varepsilon^{*2} - \varepsilon^{*3}/3)^{-1} MN \log(d/\delta^*).$$

Given $\mathcal{S}$*, then with probability at least* $1 - 2\delta^*$*,*

$$\sqrt{1-\varepsilon^*}\sigma_{min}(\mathbf{M}) \leq \sigma_{min}(\sqrt{\frac{rK}{l}}\widehat{\mathbf{R}}\mathbf{M}) \leq \sigma_{max}(\sqrt{\frac{rK}{l}}\widehat{\mathbf{R}}\mathbf{M}) \leq \sqrt{1+\varepsilon^*}\sigma_{max}(\mathbf{M}). \tag{5.10}$$

Proof. For any symmetric matrix $\mathbf{X}$, let $\lambda_{min}(\mathbf{X})$ and $\lambda_{max}(\mathbf{X})$ denote the minimal and the maximal eigenvalues of $\mathbf{X}$. To prove Proposition 5.4 we will use the matrix Chernoff tail bounds from [26] presented in Theorem 5.5.

Define $X := \{\mathbf{m}^{(j)}(\mathbf{m}^{(j)})^{\mathrm{T}}\}_{j=1}^{n}$. Consider the matrix

$$\mathbf{X} := (\widehat{\mathbf{R}}\mathbf{M})^{\mathrm{T}}\widehat{\mathbf{R}}\mathbf{M} = \sum_{j \in T}\mathbf{m}^{(j)}(\mathbf{m}^{(j)})^{\mathrm{T}},$$

where T is a set, with $\#T = l$, of elements of $\{1, 2, \dots, rK\}$ drawn uniformly and without replacement. The matrix $\mathbf{X}$ can be written as $\mathbf{X} = \sum_{i=1}^{l}\mathbf{X}_i$, where $\{\mathbf{X}_i\}_{i=1}^{l}$ is a uniformly drawn, without replacement, random subset of X. We have $\mathbb{E}(\mathbf{X}_1) = \frac{1}{rK}\mathbf{M}^{\mathrm{T}}\mathbf{M}$. Furthermore,

$$\lambda_{\max}(\mathbf{m}^{(j)}(\mathbf{m}^{(j)})^{\mathrm{T}}) = \|\mathbf{m}^{(j)}\|^2 \leq \frac{M}{rK},\ 1 \leq j \leq rK.$$

By applying Theorem 5.5 and some algebraic operations, we obtain

$$\begin{aligned}\mathbb{P}(\lambda_{\min}(\mathbf{X}) \leq (1-\varepsilon^*)\lambda_{min}(\mathbf{M}^{\mathrm{T}}\mathbf{M})\tfrac{l}{rK}) &\leq d\left(\tfrac{e^{-\varepsilon^*}}{(1-\varepsilon^*)^{1-\varepsilon^*}}\right)^{\lambda_{min}(\mathbf{M}^{\mathrm{T}}\mathbf{M})l/M} \\ &\leq d\, e^{-(\varepsilon^{*2}/2-\varepsilon^{*3}/6)(MN)^{-1}l} \leq \delta,\end{aligned}$$

$$\begin{aligned}\mathbb{P}(\lambda_{\max}(\mathbf{X}) \geq (1+\varepsilon^*)\lambda_{max}(\mathbf{M}^{\mathrm{T}}\mathbf{M})\tfrac{l}{rK}) &\leq d\left(\tfrac{e^{\varepsilon^*}}{(1+\varepsilon^*)^{1+\varepsilon^*}}\right)^{\lambda_{max}(\mathbf{M}^{\mathrm{T}}\mathbf{M})l/M} \\ &\leq d\, e^{-(\varepsilon^{*2}/2-\varepsilon^{*3}/6)(MN)^{-1}l} \leq \delta.\end{aligned}$$

The statement of the lemma follows by a union bound argument. □

Theorem 5.5 (Matrix Chernoff tail bounds from [26]). *Consider a finite set X of symmetric positive semi-definite matrices of size $d \times d$. Define the constant $L := \max_{\mathbf{X}_j \in X} \lambda_{max}(\mathbf{X}_j)$. Let $\{\mathbf{X}_i\}_{i=1}^{l}$ be a uniformly sampled, without replacement, random subset of X and $\mathbf{X} := \sum_{i=1}^{l} \mathbf{X}_i$. Then*

$$\mathbb{P}\left(\lambda_{min}(\mathbf{X}) \leq (1-\varepsilon)\mu_{min}\right) \leq d\left(\tfrac{e^{-\varepsilon}}{(1-\varepsilon)^{1-\varepsilon}}\right)^{\mu_{min}/L}$$

$$\mathbb{P}\left(\lambda_{max}(\mathbf{X}) \geq (1+\varepsilon)\mu_{max}\right) \leq d\left(\tfrac{e^{\varepsilon}}{(1+\varepsilon)^{1+\varepsilon}}\right)^{\mu_{max}/L}$$

where $\mu_{min} = l\lambda_{min}(\mathbb{E}\mathbf{X}_1)$ and $\mu_{max} = l\lambda_{max}(\mathbb{E}\mathbf{X}_1)$.

By plugging Proposition 5.2 and the result of Proposition 5.3 into Proposition 5.4 and taking $\mathbf{M} = \frac{1}{\sqrt{K}}\widehat{\mathbf{W}}\mathbf{V}$, $M = (\sqrt{d} + \sqrt{8\log rK/\delta^*})^2$, $N = 1.07$, along with the union bound argument, we deduce that,

$$\sqrt{1-\varepsilon^*}\sqrt{1-\varepsilon^*/30} \leq \sigma_{min}(\mathbf{\Omega V}) \leq \sigma_{max}(\mathbf{\Omega V}) \leq \sqrt{1+\varepsilon^*}\sqrt{1+\varepsilon^*/30}$$

holds with probability at least $1-4\delta^*$ under the condition $\mathcal{S}$. Finally by few algebraic operations, we conclude that, given $\mathcal{S}$, the singular values of $\mathbf{\Omega V}$ belong to $[\sqrt{1-\varepsilon}, \sqrt{1+\varepsilon}]$ with probability at least $4\delta/5$. The proof of the main theorem is finished by reminding that $\mathcal{S}$ occurs with probability at least $1-\delta^*$, the union bound argument and few additional algebraic operations. □

6 Conclusion

The proposed block SRHT can combine the advantages of structured and unstructured matrices, such as low application complexity and suitability for distributed computing. It should outperform all known embeddings in a distributed architecture with not too large number of processors. At the same time it yields the same approximation guarantees as standard SRHT. We have chosen the low-rank approximation problem as a representative application. We revised popular randomized methods for this problem with implementation aspects on distributed architectures, and then presented their quasi-optimality characterizations from a projection-based point of view, compatible with block SRHT. Numerical validation of the methodology showed that the block SRHT in practice provides solutions of the same quality as Gaussian embeddings. Yet, the block SRHT was up to a factor of 2.5 faster to apply. Moreover, even greater gains in runtime are expected for larger problems and sampling dimensions.

7 Acknowledgments

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No 810367).

References

[1] Per-Gunnar Martinsson and Joel A Tropp. Randomized numerical linear algebra: Foundations and algorithms. *Acta Numerica*, 29:403–572, 2020.

[2] David P Woodruff et al. Sketching as a tool for numerical linear algebra. *Foundations and Trends® in Theoretical Computer Science*, 10(1–2):1–157, 2014.

[3] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

[4] Michael W Mahoney. Randomized algorithms for matrices and data. *arXiv preprint arXiv:1104.5557*, 2011.

[5] William B Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space 26. *Contemporary mathematics*, 26:28, 1984.

[6] Nir Ailon and Bernard Chazelle. Approximate nearest neighbors and the fast johnson-lindenstrauss transform. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 557–563, 2006.

[7] Tamas Sarlos. Improved approximation algorithms for large matrices via random projections. In *2006 47th annual IEEE symposium on foundations of computer science (FOCS'06)*, pages 143–152. IEEE, 2006.

[8] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

[9] Yiming Sun, Yang Guo, Charlene Luo, Joel Tropp, and Madeleine Udell. Low-rank tucker approximation of a tensor from streaming data. *SIAM Journal on Mathematics of Data Science*, 2(4):1123–1150, 2020.

[10] Farhad Pourkamali Anaraki and Shannon M Hughes. Compressive k-svd. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5469–5473. IEEE, 2013.

[11] Oleg Balabanov and Anthony Nouy. Randomized linear algebra for model reduction. Part I: Galerkin methods and error estimation. *Advances in Computational Mathematics*, 45(5-6):2969–3019, December 2019.

[12] Oleg Balabanov and Laura Grigori. Randomized gram-schmidt process with application to gmres. *arXiv preprint arXiv:2011.05090*, 2020.

[13] Ahmed Alaoui and Michael W Mahoney. Fast randomized kernel ridge regression with statistical guarantees. *Advances in neural information processing systems*, 28, 2015.

[14] Francis Bach. Sharp analysis of low-rank kernel matrix approximations. In *Conference on Learning Theory*, pages 185–209. PMLR, 2013.

[15] Michal Derezinski, Rajiv Khanna, and Michael W Mahoney. Improved guarantees and a multiple-descent curve for column subset selection and the nystrom method. *Advances in Neural Information Processing Systems*, 33:4953–4964, 2020.

[16] Giacomo Meanti, Luigi Carratino, Lorenzo Rosasco, and Alessandro Rudi. Kernel methods through the roof: handling billions of points efficiently. *Advances in Neural Information Processing Systems*, 33:14410–14422, 2020.

[17] Rong Yin, Weiping Wang, and Dan Meng. Distributed nyström kernel learning with communications. In *International Conference on Machine Learning*, pages 12019–12028. PMLR, 2021.

[18] Alessandro Rudi, Luigi Carratino, and Lorenzo Rosasco. Falkon: An optimal large scale kernel method. *Advances in neural information processing systems*, 30, 2017.

[19] Yuchen Zhang, John Duchi, and Martin Wainwright. Divide and conquer kernel ridge regression. In *Conference on learning theory*, pages 592–617. PMLR, 2013.

[20] Daniele Calandriello, Alessandro Lazaric, and Michal Valko. Analysis of nyström method with sequential ridge leverage score sampling. 2016.

[21] Joel A Tropp, Alp Yurtsever, Madeleine Udell, and Volkan Cevher. Fixed-rank approximation of a positive-semidefinite matrix from streaming data. *Advances in Neural Information Processing Systems*, 30, 2017.

[22] Alex Gittens and Michael Mahoney. Revisiting the nystrom method for improved large-scale machine learning. In *International Conference on Machine Learning*, pages 567–575. PMLR, 2013.

[23] Jalaj Upadhyay. Fast and space-optimal low-rank factorization in the streaming model with application in differential privacy. *arXiv preprint arXiv:1604.01429*, 2016.

[24] Joel A Tropp, Alp Yurtsever, Madeleine Udell, and Volkan Cevher. Streaming low-rank matrix approximation with an application to scientific simulation. *SIAM Journal on Scientific Computing*, 41(4):A2430–A2463, 2019.

[25] Ravi Kannan, Santosh Vempala, and David Woodruff. Principal component analysis and higher correlations for distributed data. In *Conference on Learning Theory*, pages 1040–1057. PMLR, 2014.

[26] Joel A. Tropp. Improved analysis of the subsampled randomized Hadamard transform. *arXiv:1011.1595 [cs, math]*, July 2011. arXiv: 1011.1595.

[27] Christos Boutsidis and Alex Gittens. Improved matrix algorithms via the subsampled randomized hadamard transform. *SIAM Journal on Matrix Analysis and Applications*, 34(3):1301–1340, 2013.

[28] Jiyan Yang, Xiangrui Meng, and Michael W Mahoney. Implementing randomized matrix algorithms in parallel and distributed environments. *Proceedings of the IEEE*, 104(1):58–92, 2015.

[29] James Demmel, Laura Grigori, Mark Hoemmen, and Julien Langou. Communication-optimal parallel and sequential qr and lu factorizations. *SIAM Journal on Scientific Computing*, 34(1):A206–A239, 2012.

[30] Jiawei Chiu and Laurent Demanet. Sublinear randomized algorithms for skeleton decompositions. *SIAM Journal on Matrix Analysis and Applications*, 34(3):1361–1383, 2013.

[31] Petros Drineas, Michael W Mahoney, and Nello Cristianini. On the nyström method for approximating a gram matrix for improved kernel-based learning. *journal of machine learning research*, 6(12), 2005.

[32] Farhad Pourkamali-Anaraki and Stephen Becker. Improved fixed-rank nyström approximation via qr decomposition: Practical and theoretical aspects. *Neurocomputing*, 363:261–272, 2019.

[33] Farhad Pourkamali-Anaraki, Stephen Becker, and Michael Wakin. Randomized clustered nystrom for large-scale kernel machines. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[34] Huamin Li, George C Linderman, Arthur Szlam, Kelly P Stanton, Yuval Kluger, and Mark Tygert. Algorithm 971: An implementation of a randomized algorithm for principal component analysis. *ACM Transactions on Mathematical Software (TOMS)*, 43(3):1–14, 2017.

[35] Alex Gittens. The spectral norm error of the naive nystrom extension. *arXiv preprint arXiv:1110.5305*, 2011.

[36] Joel A Tropp, Alp Yurtsever, Madeleine Udell, and Volkan Cevher. Practical sketching algorithms for low-rank matrix approximation. *SIAM Journal on Matrix Analysis and Applications*, 38(4):1454–1485, 2017.

[37] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[38] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011.