mathematics

MDPI

Article

A More Efficient and Practical Modified Nyström Method

Wei Zhang [1,*], Zhe Sun [2,3], Jian Liu [4] and Suisheng Chen [1]

[1] Fair Friend Institute of Intelligent Manufacturing, Hangzhou Vocational & Technical College, Hangzhou 310018, China
[2] Post Industry Technology Research and Development Center of the State Posts Bureau (Internet of Things Technology), Nanjing University of Posts and Telecommunications, Nanjing 210023, China
[3] Post Big Data Technology and Application Engineering Research Center of Jiangsu Province, Nanjing University of Posts and Telecommunications, Nanjing 210023, China
[4] College of Information Engineering, Nanjing University of Finance and Economics, Nanjing 210023, China
* Correspondence: zhw618@hzvtc.edu.cn

Abstract: In this paper, we propose an efficient Nyström method with theoretical and empirical guarantees. In parallel computing environments and for sparse input kernel matrices, our algorithm can have computation efficiency comparable to the conventional Nyström method, theoretically. Additionally, we derive an important theoretical result with a compacter sketching matrix and faster speed, at the cost of some accuracy loss compared to the existing state-of-the-art results. Faster randomized SVD and more efficient adaptive sampling methods are also proposed, which have wide application in many machine-learning and data-mining tasks.

Keywords: kernel method; Nyström method; low-rank approximation; machine learning

MSC: 68T05

Citation: Zhang, W.; Sun, Z.; Liu, J.; Chen, S. A More Efficient and Practical Modified Nyström Method. *Mathematics* **2023**, *11*, 2433. https://doi.org/10.3390/math11112433

Academic Editor: Luca Gemignani

Received: 29 March 2023
Revised: 17 May 2023
Accepted: 22 May 2023
Published: 24 May 2023

1. Introduction

The Nyström method is a widely used technique to speed up kernel machines. Its efficiency in computation has attracted much attention in the past few years [1–8]. Given a kernel matrix $\mathbf{K} \in \mathbb{R}^{n\times n}$, the Nyström method tries to approximate the kernel by random sampling to save computation cost. At the cost of computational efficiency, it suffers from a relatively large matrix approximation error in real applications [9,10]. Given the target rank k and target precision parameter $0 < \epsilon \leq 1$, Wang and Zhang [4] gave a theoretical analysis that, with the Nyström method, it is impossible to obtain a $1+\epsilon$ bound relative to $\|\mathbf{K}-\mathbf{K}_k\|_F^2$ unless the number of sampled columns $c > \Omega(\sqrt{nk/\epsilon})$. Here, $\mathbf{K}_k$ denotes the best rank-k approximation to the kernel matrix $\mathbf{K}$. Several modified Nyström methods were proposed in recent years [3,4,11,12]. In the work of [11], a modified Nyström method just needs k/ϵ columns of the kernel matrix to obtain a $1+\epsilon$ bound relative to $\|\mathbf{K}-\mathbf{K}_k\|_F^2$. To the best of our knowledge, it is the fastest algorithm, costing $O(nk^2) + T_{Multiply}(\mathrm{nnz}(\mathbf{K})\log n)$ to achieve a $1+\epsilon$ relative error of $\|\mathbf{K}-\mathbf{K}_k\|_F^2$, where $\mathrm{nnz}(\mathbf{K})$ means the number of non-zero entries of $\mathbf{K}$. Although these modified Nyström methods are superior in approximation accuracy, it needs a much higher computational burden compared to the conventional Nyström method.

In this paper, we propose a much faster modified Nyström method which runs in $\mathcal{O}(n^{\frac{1}{2}}k^3/\epsilon^{\frac{5}{2}}) + T_{Multiply}(\mathcal{O}(\mathrm{nnz}(\mathbf{K})\log n)$ time to achieve a $1+\epsilon$ bound relative to $\|\mathbf{K}-\mathbf{K}_k\|_F^2$. When $\epsilon > \sqrt{2}-1$, our algorithm will be accelerated to

$$\mathcal{O}(k^3) + T_{Multiply}(\mathcal{O}(\mathrm{nnz}(\mathbf{K})\log n),$$

which is guaranteed by Lemma 3. Our algorithm is given in Algorithm 3. It needs $T_{Multiply}(\mathcal{O}(\mathrm{nnz}(\mathbf{A})\log n))$ times to conduct matrix multiplication which is easily imple-

mented in parallel. The computation complexity of matrix multiplication in Algorithm 3 is near linear in input sparsity. In addition, for the arithmetic operations which are hard to implement in parallel, such as SVD, pseudoinverse and QR decomposition, Algorithm 3 needs $\mathcal{O}(n^{\frac{1}{2}}k^3/\epsilon^{\frac{5}{2}})$ time which is sublinear in the input size n. At the cost of sacrificing a certain accuracy, $\mathcal{O}(k^3)$ can be reached with the same computational complexity as the conventional Nyström method, needing $\mathcal{O}(k^3)$ arithmetic operations when sampling $\mathcal{O}(k)$ columns. Our empirical studies further validate the efficiency of our algorithm.

In this paper, we improve several key algorithms which constitute a faster modified Nyström method. We summarized our contributions as follow.

- First and most importantly, we propose an efficient modified Nyström method with theoretical guarantees.
- Second, a more computationally efficient adaptive sampling method is proposed in Lemma 2. Adaptive sampling is a cornerstone of column selection, CUR decomposition and the Nyström method [4,5,11,13], and it is also very popular in other matrix problems [14].
- Finally, our proposed practical Nyström method can achieve computation efficiency in real applications, as shown by our experiments.

The rest of this paper is structured as follows. In Section 2, we provide the notations used in this study. Section 3, several key algorithms that constitute the modified Nyström are improved. Section 4 gives our modified Nyström method. We conduct empirical analysis and comparison in Section 5, and conclude our work in Section 6. All detailed proofs are omitted except computation complexity analysis.

2. Notation and Preliminaries [15]

Firstly, we introduce the notation and concepts that will be utilized here and hereafter. $\mathbf{I}_m$ is used to represent the identity $m \times m$ matrix. Sometimes we just use $\mathbf{I}$ for simplicity. We also use $\mathbf{0}$ to signify a zero vector or a zero matrix with an appropriate size. The number of non-zero entries in $\mathbf{A}$ is indicated by the notation $\mathrm{nnz}(\mathbf{A})$.

Let $k \leq \rho$ and $\rho = \mathrm{rank}(\mathbf{A}) \leq \min\{m, n\}$. The singular value decomposition (SVD) of $\mathbf{A}$ may be expressed as

$$\mathbf{A} = \sum_{i=1}^{\rho} \sigma_i \mathbf{u}_i \mathbf{v}_i^T = \begin{bmatrix} \mathbf{U}_k & \mathbf{U}_{k\perp} \end{bmatrix} \begin{bmatrix} \mathbf{\Sigma}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_{k\perp} \end{bmatrix} \begin{bmatrix} \mathbf{V}_k^T \\ \mathbf{V}_{k\perp}^T \end{bmatrix},$$

where the top k singular values are represented by $\mathbf{U}_k$ $(m \times k)$, $\mathbf{V}_k$ $(n \times k)$ and $\mathbf{\Sigma}_k$ $(k \times k)$. The best (or closest) rank-k approximation to $\mathbf{A}$ is denoted by $\mathbf{A}_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T$. The i-th greatest singular value of $\mathbf{A}$ is denoted by $\sigma_i = \sigma_i(\mathbf{A})$. The SVD is the same as the eigenvalue decomposition when $\mathbf{A}$ is symmetric positive semi-definite (SPSD), in which case we obtain $\mathbf{U}_\mathbf{A} = \mathbf{V}_\mathbf{A}$.

Furthermore, let $\mathbf{A}^\dagger$ be the Moore–Penrose inverse of $\mathbf{A}$, defined as $\mathbf{A}^\dagger = \mathbf{V}_\rho \mathbf{\Sigma}_\rho^{-1} \mathbf{U}_\rho^T$. When $\mathbf{A}$ is non-singular, the matrix inverse is the same as the Moore–Penrose inverse.

The matrix norms are defined in the manner as follows. Assume that the spectral norm is $\|\mathbf{A}\|_2 = \max_{\mathbf{x} \in \mathbb{R}^n, \|\mathbf{x}\|_2 = 1}, \|\mathbf{A}\mathbf{x}\|_2 = \sigma_1$ and the Frobenius norm is $\|\mathbf{A}\|_F = (\sum_{i,j} a_{ij}^2)^{1/2} = (\sum_i \sigma_i^2)^{1/2}$.

When given the matrices, $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{C} \in \mathbb{R}^{m \times r}$ with $r > k$, we explicitly define matrix $\Pi_{C,k}^{\zeta}(\mathbf{A})$ as the closest representation of $\mathbf{A}$ in the column space of $\mathbf{C}$ with the rank of the most k. The function $\Pi_{C,k}^{\zeta}(\mathbf{A})$ minimizes the residual $\|\mathbf{A} - \hat{\mathbf{A}}\|_\zeta$ across all $\hat{\mathbf{A}}$ in the column space of C. Here, "ζ" denotes either the spectral norm or the Frobenius norm.

When given three matrices, $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{X} \in \mathbb{R}^{m \times p}$, and $\mathbf{Y} \in \mathbb{R}^{q \times n}$, the projection of $\mathbf{A}$ onto $\mathbf{X}$'s column space is represented as $\mathbf{X}\mathbf{X}^\dagger\mathbf{A} = \mathbf{U}_\mathbf{X}\mathbf{U}_\mathbf{X}^T\mathbf{A} \in \mathbb{R}^{m \times n}$, and the one onto $\mathbf{Y}$'s row space is denoted by $\mathbf{A}\mathbf{Y}^\dagger\mathbf{Y} = \mathbf{A}\mathbf{V}_\mathbf{Y}\mathbf{V}_\mathbf{Y}^T \in \mathbb{R}^{m \times n}$.

We now give the definition of leverage score sampling and subspace embedding, which are key tools to construct our Nyström algorithm.

Definition 1 (Leverage score sampling, [13,15]). *Allow* $\mathbf{V} \in \mathbb{R}^{n\times k}$ *to be column orthonormal with* $n > k$, *and* $\mathbf{v}_{i,*}$ *to signify the i-th row of* $\mathbf{V}$. *Allow* $\ell_i = \|\mathbf{v}_{i,*}\|_F^2/k$. *Given that the* ℓ_i *are leverage scores, let r be an integer in the range* $1 \le r \le n$. *Create the sampling matrix* $\mathbf{\Omega} \in \mathbb{R}^{n\times r}$ *and the rescaling matrix* $\mathbf{D} \in \mathbb{R}^{r\times r}$ *as follows. Pick an index i from the set of* $\{1, 2 \ldots, n\}$ *with probability* ℓ_i, *for each column* $j = 1, \ldots, r$ *of* $\mathbf{\Omega}$ *and* $\mathbf{D}$, *separately and with replacements. Let* $\mathbf{\Omega}_{ij} = 1$ *and* $\mathbf{D}_{jj} = 1/\sqrt{\ell_i r}$. *The number of operations required by this procedure is* $\mathcal{O}(nk + n)$. *This procedure is designated as*

$$[\mathbf{\Omega}, \mathbf{D}] = LeverageScoreSampling(\mathbf{V}, r).$$

Definition 2 ([16]). *Assuming* $\varepsilon > 0$ *and* $\delta > 0$, *define a distribution on* $\ell \times n$ *matrix* $\mathbf{S}$ *as* Π, *where* ℓ *depends on n, d,* ε *and* δ. *Assume that, any given* $n \times d$ *matrix* $\mathbf{A}$, *with a probability of at least* $1 - \delta$, *a matrix* $\mathbf{S}$ *chosen from distribution* Π *is a* $(1 + \varepsilon)$ ℓ_2*-subspace embedding for* $\mathbf{A}$. *Meaning that, for every* $\mathbf{x} \in \mathbb{R}^d$, $\|\mathbf{SAx}\|_2^2 = (1 \pm \varepsilon)\|\mathbf{Ax}\|_2^2$ *with probability* $1 - \delta$. *After that, we designate* Π *as an* (ε, δ)*-oblivious* ℓ_2*-subspace embedding.*

The sparse subspace embedding matrix $\mathbf{S}$ and subsampled Hadamard matrix $\mathbf{H}$ are the two most popular subspace embedding matrices. For an $n \times k$ matrix $\mathbf{A}$ with k dimension subspace, we can construct a sparse subspace embedding matrix $\mathbf{S}$ for $\mathbf{A}$ with $m = O(k^2/\epsilon^2)$ rows, and the subsampled Hadamard matrix $\mathbf{H}$ with $m = O(k \log k)/\epsilon^2$ [16]. Combining $\mathbf{S}$ with $\mathbf{H}$ still has the property.

Let's discussed the computational costs about the matrix operations mentioned above. Matrix multiplication is an intrinsic parallel operation; hence, it can be easily implemented in parallel efficiently just as many mathematical software do. However, SVD decomposition and QR decomposition are much harder to implement in parallel. Hence, we denote the time complexity of such a matrix multiplication by $T_{Multiply}$. For a general $m \times n$ matrix $\mathbf{A}$ with $m \ge n$, computing the full SVD requires $\mathcal{O}(mn^2)$ flops, whereas computing the truncated SVD of rank k ($k < n$), requires $\mathcal{O}(mnk)$ flops. Additionally, computing $\mathbf{A}^\dagger$ requires $\mathcal{O}(mn^2)$ flops, too. Given a $m \times m$ Hadamard–Walsh transform matrix $\mathbf{H}$, $T_{Multiply}(\tilde{\mathcal{O}}(mn))$ is the cost for the Hadamard–Walsh transform $\mathbf{HA}$, which is substantially quicker than $T_{Multiply}(\mathcal{O}(m^2 n))$ for the typical matrix multiplication. A sparse subspace embedding matrix $\mathbf{S}$ for an $n \times d$ matrix $\mathbf{A}$, $\mathbf{SA}$ needs $T_{Multiply}(\mathcal{O}(\mathrm{nnz}(\mathbf{A})))$ arithmetic operations.

3. Main Lemmas and Theorems

In this part, we will outline our principal theorems and lemmas, which are the key tools to implement Algorithm 3. In addition, these lemmas and theorems are of independent interest and have wide application.

First, we give a fast randomized SVD method which is depicted in Algorithm 1 which is the fastest randomized SVD method as far as we know.

Lemma 1. *Given matrix* $\mathbf{A} \in \mathbb{R}^{m\times n}$, *target rank k and error parameter* $0 < \epsilon \le 1$, $\mathbf{Z}$ *is returned from Algorithm 1; then, the following formula holds with high probability.*

$$\|\mathbf{A} - \mathbf{ZZ}^T\mathbf{A}\|_F^2 \le (1 + \epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2$$

In addition, $\mathbf{Z}$ *can be computed in* $\tilde{\mathcal{O}}(k^3/\epsilon^5) + T_{Multiply}(\mathcal{O}(\mathrm{nnz}(\mathbf{A})) + \tilde{\mathcal{O}}(mk^2/\epsilon^4 + k^3/\epsilon^3))$. *We denote Algorithm 1 as*

$$\mathbf{Z} = SparseSVD(\mathbf{A}, k, \epsilon).$$

Algorithm 1 Sparse SVD

1: **Input:** a real matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, error parameter ϵ and target rank k;
2: Compute $\mathbf{AR}^T$, where $\mathbf{R} = \mathbf{\Pi S} \in \mathbb{R}^{c\times n}$ with $c = \mathcal{O}(k\log k/\epsilon)$. $\mathbf{S} \in \mathbb{R}^{s\times n}$ is a sparse subspace embedding matrix with $s = \mathcal{O}(k^2 + k/\epsilon)$ and $\mathbf{\Pi} \in \mathbb{R}^{c\times s}$ is a subsampled randomized Hadamard matrix with $c = \mathcal{O}(k\log k/\epsilon)$;
3: Compute an orthonormal basis $\mathbf{U}$ for $\mathbf{AR}^T$ by $\mathbf{U} = \mathbf{AR}^T\mathbf{C}^{-1}$, where $\mathbf{C}$ is the Cholesky decomposition of $\mathbf{RA}^T\mathbf{AR}^T$;
4: Compute $\mathbf{\Gamma} = \mathbf{U}^T\mathbf{AW}^T \in \mathbb{R}^{c\times d}$, where $\mathbf{W} = \mathbf{HF} \in \mathbb{R}^{d\times n}$ with $d = \mathcal{O}(k\log k/\epsilon^3)$. $\mathbf{F} \in \mathbb{R}^{n\times t}$ is a sparse subspace embedding matrix with $t = \mathcal{O}(k^2\log^2 k/\epsilon^3)$ and $\mathbf{H} \in \mathbb{R}^{d\times t}$ is a subsampled randomized Hadamard matrix with $d = \mathcal{O}(k\log k/\epsilon^3)$.
5: Compute the SVD of $\mathbf{\Gamma}$ and let $\mathbf{\Delta} \in \mathbb{R}^{c\times k}$ contain the top k left singular vectors of $\mathbf{\Gamma}$;
6: **Output:** $\mathbf{Z} = \mathbf{U\Delta}$.

Proof. Lemma A2 shows that $\|\mathbf{A} - \mathbf{UU}^T\mathbf{A}\|_F^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2$, where $\mathbf{U}$ is of $\mathcal{O}(k\log k/\epsilon)$ columns. Applying Lemma A1 and replacing $\mathbf{V}$ with $\mathbf{U}$, we can obtain the result that

$$\|\mathbf{A} - \mathbf{ZZ}^T\mathbf{A}\|_F^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2$$

For computation time analysis, computing $\mathbf{AR}^T$ takes $T_{Multiply}(\mathcal{O}(\mathrm{nnz}(\mathbf{A}) + \tilde{\mathcal{O}}(mk(k + \epsilon^{-1})))$, and then $T_{Multiply}(\tilde{O}(mk^2/\epsilon^2 + k^3/\epsilon^3))$ computes the $\mathbf{U} = \mathbf{AC}^{-1}$, where $\mathbf{C}$ is the Cholesky decomposition of $\mathbf{A}^T\mathbf{A}$. Computing $\mathbf{U}^T(\mathbf{AW}^T)$ requires $T_{Multiply}(\mathcal{O}(\mathrm{nnz}(\mathbf{A}) + mk^2/\epsilon^3 + mk^2/\epsilon^4))$. Computing the SVD of $\mathbf{\Gamma}$ requires $\tilde{\mathcal{O}}(k^3/\epsilon^5)$. In addition, computing $\mathbf{Z} = \mathbf{U\Delta}$ requires $T_{Multiply}(\tilde{\mathcal{O}}(mk^2/\epsilon^3))$. Hence, Algorithm 1 takes

$$\tilde{\mathcal{O}}(k^3/\epsilon^5) + T_{Multiply}(\mathcal{O}(\mathrm{nnz}(\mathbf{A})) + \tilde{\mathcal{O}}(mk^2/\epsilon^4 + k^3/\epsilon^3))$$

computation complexity. □

A faster adaptive sampling, Algorithm 2, is developed based on the work of [13]. Boutsidis and Woodruff [13] tried to compute norms of each column of $\mathbf{GB} = \mathbf{GA} - \mathbf{GC}_1\mathbf{C}_1^\dagger\mathbf{A}$. To further reduce the computation cost, we introduce the sketched $\mathbf{G\hat{B}} = \mathbf{GA} - \mathbf{GC}_1(\mathbf{RC}_1)^\dagger(\mathbf{RA})$ to approximate $\mathbf{GB}$. By such sketching, $\mathbf{GC}_1(\mathbf{RC}_1)^\dagger(\mathbf{RA})$ can be computed more efficiently than $\mathbf{GC}_1\mathbf{C}_1^\dagger\mathbf{A}$.

Algorithm 2 Adaptive Sampling

1: **Input:** a real matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{C}_1 \in \mathbb{R}^{m\times c_1}$ and the number of selected columns c;
2: Construct $\hat{\mathbf{B}} = \mathbf{A} - \mathbf{C}_1(\mathbf{RC}_1)^\dagger(\mathbf{RA})$, where $\mathbf{R} = \mathbf{\Pi S} \in \mathbb{R}^{t\times m}$ with $t = 2c_1\log c_1$. $\mathbf{S} \in \mathbb{R}^{s\times m}$ is a sparse subspace embedding matrix with $s = c_1^2 + 2c_1$ and $\mathbf{\Pi} \in \mathbb{R}^{t\times s}$ is a subsampled randomized Hadamard matrix;
3: Construct $\tilde{\mathbf{B}} = \mathbf{G\hat{B}}$ where $\mathbf{G} \in \mathbb{R}^{g\times m}$ is a normalized Gaussian matrix with $g = 9\log n$;
4: Compute sampling probabilities $p_j = \|\tilde{\mathbf{b}}_j\|_F^2/\|\tilde{\mathbf{B}}\|_F^2$ for $j = 1,\ldots,n$, where $\tilde{\mathbf{b}}_j$ is the $j-$th column of $\tilde{\mathbf{B}}$;
5: **Output:** Obtain $\mathbf{C}_2$ by selecting c columns from $\mathbf{A}$ in c i.i.d. trials; in each trial the index j is chosen with probability p_j.

Lemma 2. *Given* $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{C}_1 \in \mathbb{R}^{m\times c_1}$ *and* $\mathbf{V} \in \mathbb{R}^{r\times n}$ *such that* $rank(\mathbf{V}) = rank(\mathbf{AV}^\dagger\mathbf{V}) = \rho$, *with* $\rho \leq c \leq n$, *let* $\mathbf{C}_2 \in \mathbb{R}^{m\times c_2}$ *be returned from Algorithm 2 containing* c_2 *columns of* $\mathbf{A}$. *Then, the matrix* $\mathbf{C} = [\mathbf{C}_1, \mathbf{C}_2] \in \mathbb{R}^{m\times(c_1+c_2)}$ *satisfies that for any integer* $k > 0$, *and with a high probability which is at least* 0.9.

$$\|\mathbf{A} - \mathbf{CC}^\dagger\mathbf{AV}^\dagger\mathbf{V}\|_F^2 \leq \|\mathbf{A} - \mathbf{AV}^\dagger\mathbf{V}\|_F^2 + \frac{40\rho}{c_2}\|\mathbf{A} - \mathbf{C}_1\mathbf{C}_1^\dagger\mathbf{A}\|_F^2.$$

In addition, this randomized algorithm can be implemented in

$$\tilde{\mathcal{O}}(c_1^3) + T_{Multiply}(\mathcal{O}(\mathrm{nnz}(\mathbf{A})\log n + \tilde{\mathcal{O}}(nc_1^2 + nc_1\log n + c_1^3))$$

computation time. We denote this randomized algorithm as

$$\mathbf{C}_2 = AdaptiveSampling(\mathbf{A}, \mathbf{V}, \mathbf{C}_1, c_2).$$

Proof. Let $\mathbf{B} = \mathbf{A} - \mathbf{C}_1\mathbf{C}_1^{\dagger}\mathbf{A}$ be the residual matrix and $\mathbf{b}_i$ is the i-th column of $\mathbf{B}$. By Theorem A4, with high probability, it holds that

$$\|\mathbf{B}\|_F^2 \leq \|\hat{\mathbf{B}}\|_F^2 \leq (1+2\epsilon)\|\mathbf{B}\|_F^2 = 2\|\mathbf{B}\|_F^2\|\mathbf{b}_i\|_F^2 \leq \|\hat{\mathbf{b}}_i\|_F^2 \leq (1+2\epsilon)\|\mathbf{b}_i\|_F^2 = 2\|\mathbf{b}_i\|_F^2$$

Besides, by the JL property of $\mathbf{G}$, we have $\frac{1}{3}\|\hat{\mathbf{b}}_i\|_F \leq \|\tilde{\mathbf{b}}_i\|_F \leq \frac{4}{3}\|\hat{\mathbf{b}}_i\|_F$. Hence, after utilizing the below distribution for sampling,

$$p_i = \frac{\|\tilde{\mathbf{b}}_i\|_F}{\|\tilde{\mathbf{B}}\|_F} \geq \frac{2}{3}\cdot\frac{3}{4}\cdot\frac{\|\hat{\mathbf{b}}_i\|_F^2}{\|\hat{\mathbf{B}}\|_F^2} \geq \frac{1}{2}\cdot\frac{1}{2}\cdot\frac{\|\mathbf{b}_i\|_F^2}{\|\mathbf{B}\|_F^2} = \frac{1}{4}\frac{\|\mathbf{b}_i\|_F^2}{\|\mathbf{B}\|_F^2}$$

Using Lemma A3, we obtain

$$\mathbb{E}\left[\|\mathbf{A} - \mathbf{C}\mathbf{C}^{\dagger}\mathbf{A}\mathbf{V}^{\dagger}\mathbf{V}\|_F^2\right] \leq \|\mathbf{A} - \mathbf{A}\mathbf{V}^{\dagger}\mathbf{V}\|_F^2 + \frac{4k}{c_2}\|\mathbf{A} - \mathbf{C}_1\mathbf{C}_1^{\dagger}\mathbf{A}\|_F^2.$$

Using the Markov inequality, we have that

$$\|\mathbf{A} - \mathbf{C}\mathbf{C}^{\dagger}\mathbf{A}\mathbf{V}^{\dagger}\mathbf{V}\|_F^2 \leq \|\mathbf{A} - \mathbf{A}\mathbf{V}^{\dagger}\mathbf{V}\|_F^2 + \frac{40\rho}{c_2}\|\mathbf{A} - \mathbf{C}_1\mathbf{C}_1^{\dagger}\mathbf{A}\|_F^2.$$

holds with a probability of at least 0.9.

As to the running time, it needs $T_{Multiply}(\mathcal{O}(\mathrm{nnz}(\mathbf{A})) + \tilde{\mathcal{O}}(nc_1^2))$ arithmetic operations to compute $\mathbf{RA}$. To compute $\mathbf{RC}_1$ costs $T_{Multiply}(\mathcal{O}(\mathrm{nnz}(\mathbf{C}_1)) + \tilde{\mathcal{O}}(c_1^3))$. To compute $(\mathbf{RC}_1)^{\dagger}$, it requires $\tilde{O}(c_1^3)$. In addition, computing $\mathbf{GA}$ and $\mathbf{GC}_1$ require $T_{Multiply}$ $(\mathcal{O}(\mathrm{nnz}(\mathbf{A})\log n))$ and $T_{Multiply}(mc_1 \log n)$, respectively. In addition, to compute $(\mathbf{GC}_1)(\mathbf{RC}_1)^{\dagger}(\mathbf{RA})$ needs

$$T_{Multiply}(\tilde{\mathcal{O}}(nc_1 \log n + c_2 \log n))$$

computation. In addition, $\mathbf{GA} - \mathbf{GC}_1(\mathbf{RC}_1)^{\dagger}\mathbf{RA}$ needs another $T_{Multiply}(\mathcal{O}(n \log n))$ arithmetic operations. Thus, all these need $\tilde{\mathcal{O}}(c_1^3) + T_{Multiply}(\mathcal{O}(\mathrm{nnz}(\mathbf{A})\log n + \tilde{\mathcal{O}}$ $(nc_1^2 + nc_1 \log n + c_1^3))$ □

Lemma 3 ([15,17]). *Given the matrices $\mathbf{C} \in \mathbb{R}^{m\times c}$, $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{R} \in \mathbb{R}^{n\times r}$, let's suppose that $\mathbf{S}$ is the leverage-score sketching matrix of $\mathbf{C}$ with $s = \mathcal{O}(c/\epsilon + c\log c)$ rows, and $\mathbf{T}$ is the leverage-score sketching matrix of $\mathbf{R}$ with $t = \mathcal{O}(r/\epsilon + r\log r)$ columns. Let*

$$\mathbf{U}^{\star} = \mathbf{C}^{\dagger}\mathbf{A}\mathbf{R}^{\dagger} = \underset{\mathbf{U}}{\operatorname{argmin}} \|\mathbf{A} - \mathbf{C}\mathbf{U}\mathbf{R}\|_F$$

and

$$\hat{\mathbf{U}} = (\mathbf{SC})^{\dagger}\mathbf{SAT}(\mathbf{RT})^{\dagger},$$

then we can obtain

$$\|\mathbf{A} - \mathbf{C}\hat{\mathbf{U}}\mathbf{R}\|_F \leq (1+\epsilon)\|\mathbf{A} - \mathbf{C}\mathbf{U}^{\star}\mathbf{R}\|_F.$$

The number of sampled rows in Lemma 3 is independent on the input dimension of $\mathbf{A}$ and is linear to c. By losing some accuracy, a much faster algorithm can be implemented.

4. Practical Modified Nyström Method

We use our new lemmas and theorems developed in Section 3 to implement an efficient modified Nyström algorithm.

4.1. Description of The Algorithm

A $n \times n$ real symmetric matrix $\mathbf{A}$, an error parameter $0 < \epsilon < 1$ and a target rank k are the inputs of Algorithm 3. Meanwhile, a matrix $\mathbf{C} \in \mathbb{R}^{n \times c}$ with $c = \mathcal{O}(k/\epsilon + k \log k)$ columns of $\mathbf{A}$, and a matrix $\mathbf{U} \in \mathbb{R}^{c \times c}$ are the results. There are primarily 3 steps in Algorithm 3: (i) using the definition of the leverage score sampling, it samples a number of columns of $\mathbf{A}$ to obtain $\mathbf{C}_1$; and using the adaptive sampling method to obtain $\mathbf{C}_2$ and $\mathbf{R}_2$; (ii) it calculates the leverage scores of $\mathbf{C}$ using the method in [18]; and (iii) it constructs the intersection matrix $\mathbf{U}$. Note that $\widehat{\mathbf{U}}$ in Lemma 3 is asymmetric even when $\mathbf{A}$ is positive semi-definite. Thus, when applied to kernel approximation, we need to construct a positive semi-definite $\mathbf{U}$ shown in Algorithm 3.

Algorithm 3 Practical Nyström

1: **Input:** a real symmetric matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, error parameter ϵ and target rank k;
2: $\mathbf{Z} = SparseSVD(\mathbf{A}, k, 1)$;
3: $[\mathbf{\Omega}, \Gamma] = LeverageScoreSampling(\mathbf{Z}, \mathcal{O}(k \log k))$ and construct $\mathbf{C}_1 = \mathbf{A\Omega}$;
4: $\mathbf{C}_2 = AdaptiveSampling(\mathbf{A}, \mathbf{V}_k^T, \mathbf{C}_1, \mathcal{O}(k/\epsilon))$ and $\mathbf{C}_3 = AdaptiveSampling(\mathbf{A}, \mathbf{V}_k^T, \mathbf{C}_1, \mathcal{O}(k/\epsilon))$, constructing $\mathbf{C} = [\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3] \in \mathbb{R}^{n \times \mathcal{O}(k/\epsilon + k \log k)}$;
5: Compute approximate leverage scores of $\mathbf{C}$ using the method of [18] and construct the leverage sketch matrix $\mathbf{S}_1$ and $\mathbf{S}_2$ of $n \times s$ size, where $s = \mathcal{O}(\frac{c}{\epsilon} + c \log c)$;
6: Compute $\widehat{\mathbf{U}} = (\mathbf{S}_1\mathbf{C})^\dagger \mathbf{S}_1 \mathbf{A} \mathbf{S}_2^T (\mathbf{C}^T \mathbf{S}_2^T)^\dagger$.
7: Compute $\mathbf{U} = \mathbf{\Pi}_{\mathbb{H}_+^s}(\widehat{\mathbf{U}})$ by conducting eigenvalue decomposition of $\widetilde{\mathbf{U}} = \frac{\widehat{\mathbf{U}} + \widehat{\mathbf{U}}^T}{2}$ and setting the negative eigenvalues of $\widehat{\mathbf{U}}$ to zero.
8: **Output:** $\mathbf{C}$ and $\mathbf{U}$.

4.2. Analysis of Running-Time

Here, we provide a detailed analysis of the Algorithm 3's arithmetic operations.

1. The computation complexity of Algorithm 3 is $\tilde{\mathcal{O}}(k^3) + T_{Multiply}(\mathcal{O}(\text{nnz}(\mathbf{A}) \log n + \tilde{\mathcal{O}}(nk^2 + nk \log n + k^3))$ to find $\mathcal{O}(k/\epsilon + k \log k)$ columns of $\mathbf{A}$ to construct $\mathbf{C}$.
   (a) To obtain $\mathbf{Z} \in \mathbb{R}^{n \times k}$ from Theorem 1, it takes $\tilde{\mathcal{O}}(k^3) + T_{Multiply}(\mathcal{O}(\text{nnz}(\mathbf{A})) + \tilde{\mathcal{O}}(nk^2 + k^3))$.
   (b) To obtain the leverage score and sample $\mathbf{C}_1$ and C_2, it takes $T_{Multiply}(\mathcal{O}(nk))$.
   (c) To construct $\mathbf{C}_3$ and $\mathbf{R}_2$ Lemma 2, it takes $\tilde{\mathcal{O}}(k^3) + T_{Multiply}(\mathcal{O}(\text{nnz}(\mathbf{A}) \log n + \tilde{\mathcal{O}}(nk^2 + nk \log n + k^3))$.
2. The computation complexity of Algorithm 3 is $\mathcal{O}(k^3/\epsilon^4) + T_{Multiply}(\mathcal{O}(nk^2/\epsilon^2) + \tilde{\mathcal{O}}(nk^2 + k^3/\epsilon^5))$ to construct $\mathbf{U}$ when $s = \mathcal{O}(\frac{c}{\epsilon} + c \log c)$ is the row dimension of $\mathbf{S}_1$ and $\mathbf{S}_2$ in Algorithm 3.
   (a) To obtain the leverage scores of $\mathbf{C}$, it takes $\mathcal{O}(k^3/\epsilon^3) + T_{multiply}(\mathcal{O}(n(k/\epsilon)^2 + \tilde{\mathcal{O}}(nk^2))$.
   (b) To compute $(\mathbf{S}_1^T\mathbf{C})^\dagger$ and $(\mathbf{S}_2^T\mathbf{C})^\dagger$, it takes $\tilde{\mathcal{O}}(k^3/\epsilon^4)$.
   (c) To compute matrix multiplication, it takes $T_{Multiply}(\mathcal{O}(k^3/\epsilon^5))$.
   (d) To compute the eigenvalue decomposition of $\mathbf{U}$, it takes $\tilde{\mathcal{O}}(k^3/\epsilon^3)$.

   The algorithm's overall asymptotic arithmetic operation is

$$T_{Multiply}(\mathcal{O}(\text{nnz}(\mathbf{A}) \log n + nk^2/\epsilon^2 + k^3/\epsilon^5) + \tilde{\mathcal{O}}(nk^2 + nk \log n + k^3/\epsilon^4)).$$

4.3. Error Bound

Primary approximate result regarding Algorithm 3 is shown as the following theorem.

Theorem 1. *Given an error parameter ϵ and a target rank k, run Algorithm 3, then the below inequality holds with high probability.*

$$\|\mathbf{A} - \mathbf{CUC}^T\|_F \leq (1 + \epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F$$

5. Empirical Study

In this section, we compare our Practical Nyström algorithm with the uniform+adaptive algorithm [11,19], near-optimal+adaptive algorithm [4,11,13] and conventional Nyström using just uniform sampling. All algorithms were implemented in Matlab and experiments were conducted on a workstation with 32 cores of 2G Hz and 24G RAM.

On each data set, we give the approximation error and the execution duration of each algorithm. The approximation error is

$$\text{Approximation Error} = \frac{\|\mathbf{A} - \mathbf{C}\mathbf{U}\mathbf{C}^T\|_F}{\|\mathbf{A}\|_F},$$

where $\mathbf{U}$ is the intersection matrix defined in the Nyström method.

On three data sets we test all three algorithms, and the results are listed in Table 1. We create a RBF kernel matrix $\mathbf{A}$ for each dataset, with $a_{ij} = \exp(\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\gamma^2})$, where $\mathbf{x}_i$ and $\mathbf{x}_j$ are data instances and γ is the parameter of the RBF kernel function. By the definition of $\mathbf{A}$, the size n of $\mathbf{A}$ is the number of instances of the dataset. Thus, the kernel matrices in our experiments are of large sizes. We set γ different values for each data set as Table 1 describes. However, the effectiveness of our algorithm does not depend on the setting of γ. For each data set, we set $k = 10, 30$ and 50. We sampled $c = ak$ columns from $\mathbf{A}$ and a ranges from 8 to 26. We ran each algorithm 5 times and report the average value of approximation error and running time. All results are illustrated in Figures 1–3.

Table 1. A summary of the datasets for kernel approximation.

Data Set	a9a	USPS	PenDigits
#instance	32,561	11,305	7494
γ	5	4	30
Source	UCI	TKH96a	UCI

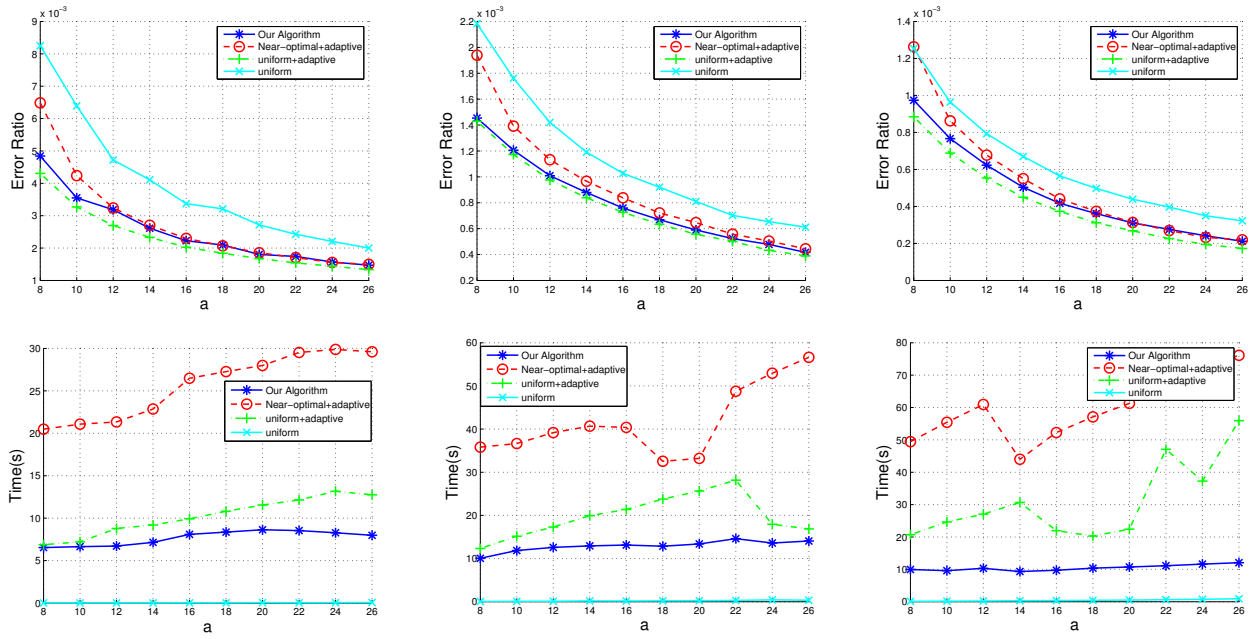

Figure 1. Results of the Nyström algorithms on the a9a dataset. In the first column, we set $k = 10$, and $c = ak$ with $a = 8, \ldots, 26$. In the middle column, we set $k = 30$, and $c = ak$. In the right column, we set $k = 50$, and $c = ak$.

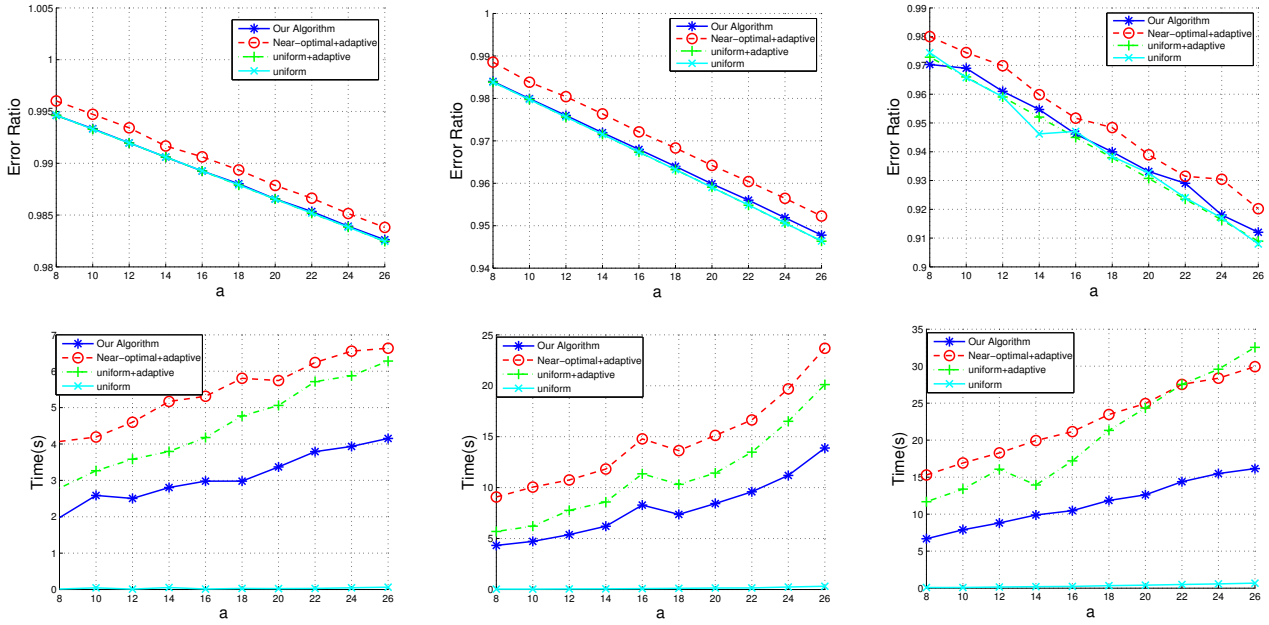

Figure 2. Results of the Nyström algorithms on the pendigit dataset. In the first column, we set $k = 10$, and $c = ak$ with $a = 8, \ldots, 26$. In the middle column, we set $k = 30$, and $c = ak$. In the right column, we set $k = 50$, and $c = ak$.

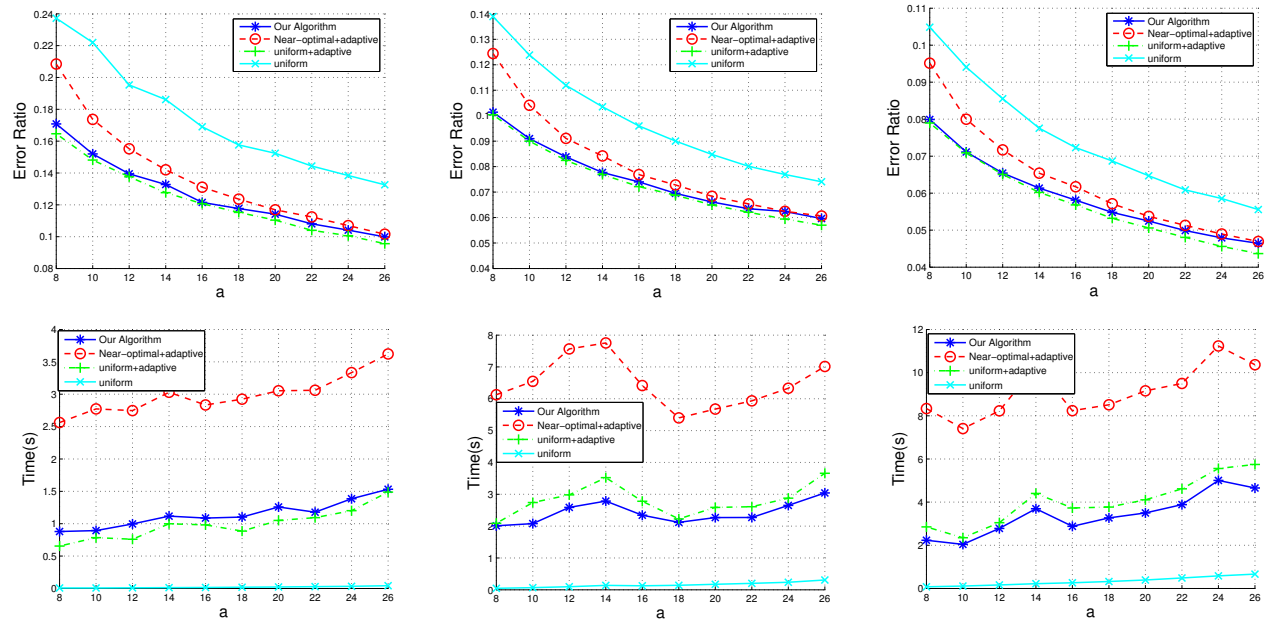

Figure 3. Results of the Nyström algorithms on the usps dataset. In the first column, we set $k = 10$, and $c = ak$ with $a = 8, \ldots, 26$. In the middle column, we set $k = 30$, and $c = ak$. In the right column, we set $k = 50$, and $c = ak$.

As evidenced by the empirical results in the figures, it is clear that our approach is efficient. In terms of accuracy, Our approach is comparable to the state-of-the-art algorithm—the near-optimal+adaptive algorithm [4,11,13]. As to the running time, our approach is much faster than near-optimal+adaptive algorithm and uniform+adaptive algorithm. Our algorithm's running time grows slower than the near-optimal+adaptive algorithm and uniform+adaptive algorithm. The advantage of the running time of our algorithm grows as the dimension of kernel matrix **A** increases. Calculating kernel matrix **A** of size 7494×7494 from the 'PenDigits' data set, our alogrithm is twice as fast as the near-

optimal+adaptive algorithm. As to the 'a9a' data set of 32,561 instances, our algorithm is four times faster than near-optimal+adaptive. In addition, as c increases, the running-time superiority of our algorithm also increases. Our algorithm also has similar a advantage over the uniform+adaptive algorithm. Hence, our algorithm is suitable to scale to kernel matrices of high dimensions.

6. Conclusions

In this paper, we proposed an efficient modified Nyström method with a theoretical and emperical guarantee. In a high-level parallel-computation environment with sparse input matrices, our Nyström method can achieve comparable computation efficiency compared to the conventional Nyström method, theoretically. Hence, our Nyström method is suitable for machine-learning algorithms in big-data setting. In addition, we give a sketching generalized matrix approximation which extends the previous work [12]. Faster randomized SVD and more efficient adaptive sampling methods are proposed which have wide application in lots of areas. In addition, our modified Nyström algorithm can be easily extended to CUR decomposition which leads to more efficient CUR decomposition.

Author Contributions: Conceptualization, W.Z. and Z.S.; methodology, W.Z. and J.L.; software, W.Z.; validation, S.C. and Z.S.; writing—original draft preparation, W.Z.; writing—review and editing, S.C.; visualization, J.L.; funding acquisition, Z.S. and S.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Hangzhou Key Scientific Research Program of China (No. 20212013B06), Zhejiang Provincial Natural Science Foundation of China (No. LGG21E050005), National Natural Science Foundation of China (No. 61972208, No. 62272239 and No. 62022044) and National Natural Science Foundation of Jiangsu Province (No. BK20201043).

Acknowledgments: The authors would like to thank Professor An Yang, from Hangzhou Vocational & Technical College, for her valuable suggestions and guidance throughout this research.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Key Theorems Used in Our Proofs

Theorem A1 ([15,20]). *There is $t = \Theta(\epsilon^{-2})$ for matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ and orthonormal $\mathbf{U} \in \mathbb{R}^{m\times k}$, thus, for a $t \times m$ leverage-score sketching matrix $\mathbf{S}$ for orthonormal $\mathbf{U}$,*

$$\mathbb{P}\left[\|\mathbf{A}^T\mathbf{S}^T\mathbf{S}\mathbf{U} - \mathbf{A}^T\mathbf{U}\|_F^2 < \epsilon^2\|\mathbf{A}\|_F^2\|\mathbf{U}\|_F^2\right] \geq 1 - \delta,$$

for any fixed $\delta > 0$.

Theorem A2 ([15,20]). *There is $t = \mathcal{O}(k\epsilon^{-2}\log k)$, for any rank k matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ with row leverage scores, such that leverage-score sketching matrix $\mathbf{S} \in \mathbb{R}^{t\times m}$ is an ϵ-embedding matrix for matrix $\mathbf{A}$, i.e.,*

$$\|\mathbf{S}\mathbf{A}\mathbf{x}\|_2^2 = (1 \pm \epsilon)\|\mathbf{A}\mathbf{x}\|_2^2$$

Theorem A3 ([15,20]). *Given that A is a matrix with m rows and C is a matrix with m rows as well as rank k. $\mathbf{S}$ is a subspace embedding for $\mathbf{C}$ with error parameter $\epsilon_0 \leq 1/\sqrt{2}$, and it is also the $t \times m$ leverage-score sketching matrix of $\mathbf{C}$ with $\mathcal{O}(k/\epsilon)$ rows. Then if $\hat{\mathbf{Y}}$ and $\mathbf{Y}^\star$ are respectively the solutions to*

$$min_{\mathbf{Y}} = \|\mathbf{S}(\mathbf{C}\mathbf{Y} - \mathbf{A})\|_F^2$$

and

$$min_{\mathbf{Y}} = \|\mathbf{S}(\mathbf{C}\mathbf{Y} - \mathbf{A})\|_F^2$$

then, the below two formulas hold with a probability of at least 0.99.

$$\|\mathbf{C}\hat{\mathbf{Y}} - \mathbf{A}\|_F \leq (1 + \epsilon)\|\mathbf{C}\mathbf{Y}^\star - \mathbf{A}\|_F$$

$$\|\mathbf{C}(\hat{\mathbf{Y}} - \mathbf{Y}^*)\|_F \leq 2\sqrt{\epsilon}\|\mathbf{C}\mathbf{Y}^* - \mathbf{A}\|_F$$

Theorem A4 ([15,20]). *Given that A is a matrix with m rows, and C is a matrix with m rows as well as rank k, where* $\mathbf{R} = \mathbf{\Pi}\mathbf{S} \in \mathbb{R}^{t\times n}$ *with* $t = 2k\log k/\epsilon$. $\mathbf{\Pi} \in \mathbb{R}^{t\times s}$ *is a subsampled randomized Hadamard matrix and* $\mathbf{S} \in \mathbb{R}^{s\times m}$ *is a sparse subspace embedding matrix with* $s = k^2 + 2k/\epsilon$. *Then if* $\hat{\mathbf{Y}}$ *and* $\mathbf{Y}^\star$ *are respectively the solutions to*

$$min_{\mathbf{Y}} = \|\mathbf{R}(\mathbf{C}\mathbf{Y} - \mathbf{A})\|_F^2$$

and

$$min_{\mathbf{Y}} = \|\mathbf{R}(\mathbf{C}\mathbf{Y} - \mathbf{A})\|_F^2$$

then, the below two formulas hold with a probability of at least 0.99.

$$\|\mathbf{C}\hat{\mathbf{Y}} - \mathbf{A}\|_F \leq (1+\epsilon)\|\mathbf{C}\mathbf{Y}^\star - \mathbf{A}\|_F$$

$$\|\mathbf{C}(\hat{\mathbf{Y}} - \mathbf{Y}^*)\|_F \leq 2\sqrt{\epsilon}\|\mathbf{C}\mathbf{Y}^* - \mathbf{A}\|_F$$

Lemma A1 ([13,15]). *Let* $\mathbf{A} \in \mathbb{R}^{m\times n}$ *and* $\mathbf{V} \in \mathbb{R}^{m\times c}$. *Assume that given a particular rank parameter k and an accuracy parameter* $0 < \epsilon < 1$,

$$\|\mathbf{A} - \Pi_{V,k}^F(\mathbf{A})\|_F^2 \leq \|\mathbf{A} - \mathbf{A}_k\|_F^2.$$

$\mathbf{V}$ *is a QR-decomposition, and let* $\mathbf{V} = \mathbf{Q}\mathbf{Y}$ *where* $\mathbf{Q} \in \mathbb{R}^{m\times c}$ *and* $\mathbf{Y} \in \mathbb{R}^{c\times c}$. *Let* $\Gamma = \mathbf{Q}^T\mathbf{A}\mathbf{W}^T \in \mathbb{R}^{c\times\ell}$, *where* $\mathbf{W}^T \in \mathbb{R}^{n\times\ell}$ *is a sparse subspace embedding matrix, and* $\ell = \mathcal{O}(c^2/\epsilon^2)$. *Let* $\mathbf{\Delta} \in \mathbb{R}^{c\times k}$ *contain the top k left singular vectors of* Γ. *Then, it holds that*

$$\|\mathbf{A} - \mathbf{Q}\mathbf{\Delta}\mathbf{\Delta}^\mathbf{T}\mathbf{Q}^\mathbf{T}\mathbf{A}\|_\mathbf{F}^\mathbf{2} \leq (\mathbf{1}+\boldsymbol{\epsilon})\|\mathbf{A} - \mathbf{A_k}\|_\mathbf{F}^\mathbf{2}.$$

with high probability.

Lemma A2 ([16]). *Given matrix* $\mathbf{R}^{m\times n}$, $\mathbf{R} = \mathbf{\Pi}\mathbf{S} \in \mathbb{R}^{c\times n}$ *is a subspace embedding matrix with* $c = \mathcal{O}(k\log k/\epsilon)$. $\mathbf{S} \in \mathbb{R}^{n\times s}$ *is a sparse subspace embedding matrix with* $s = \mathcal{O}(k^2 + k/\epsilon)$ *and* $\mathbf{\Pi} \in \mathbb{R}^{c\times s}$ *is a subsampled randomized Hadamard matrix with* $c = \mathcal{O}(k\log k/\epsilon)$. *Let* $\mathbf{U}$ *be the orthonormal basis of* $\mathbf{A}\mathbf{R}^T$. *Then, it holds that*

$$\|\mathbf{A} - \mathbf{U}\mathbf{U}^T\mathbf{A}\|_F^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2.$$

with high probability.

Lemma A3 ([4,15,16]). *Given* $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{R}_1 \in \mathbb{R}^{r_1\times n}$ *and* $\mathbf{C} \in \mathbb{R}^{m\times c}$ *such that*

$$rank(\mathbf{C}) = rank(\mathbf{C}\mathbf{C}^\dagger\mathbf{A}) = \rho,$$

with $\rho \leq c \leq n$, *given* $\mathbf{R}_1 \in \mathbb{R}^{r_1\times n}$ *and the defined residual*

$$\mathbf{B} = \mathbf{A} - \mathbf{A}\mathbf{R}_1^\dagger\mathbf{R}_1 \in \mathbb{R}^{m\times n}.$$

For $i = 1, \ldots, m$, *let* p_i *be the probability distribution such that for each i:*

$$p_i \geq \alpha\|\mathbf{b}_i\|_F^2/\|\mathbf{B}\|_F^2,$$

where $\mathbf{b}_i$ *is the i-th row of* $\mathbf{B}$. *Sample* r_2 *rows from* $\mathbf{A}$ *in* c_2 *i.i.d. trials, where in each trial the i-th column is chosen with probability* p_i. *Let* $\mathbf{R}_2 \in \mathbb{R}^{r_2\times n}$ *contain the* r_2 *sampled rows and let* $\mathbf{R} = [\mathbf{R}_1^T, \mathbf{R}_2^T]^T$. *Then*

$$\mathbb{E}\|\mathbf{A} - \mathbf{C}\mathbf{C}^\dagger\mathbf{A}\mathbf{R}^\dagger\mathbf{R}\|_F^2 \leq \|\mathbf{A} - \mathbf{C}\mathbf{C}^\dagger\mathbf{A}\|_F^2 + \frac{\rho}{\alpha r_2}\|\mathbf{A} - \mathbf{A}\mathbf{R}^\dagger\mathbf{R}\|_F^2.$$

Theorem A5 ([13,15]). *Given three matrices* $\mathbf{C} \in \mathbb{R}^{m\times c}$, $\mathbf{A} \in \mathbb{R}^{m\times n}$ *and* $\mathbf{R} \in \mathbb{R}^{r\times n}$, *we have*

$$\mathbf{C}^{\dagger}\mathbf{A}\mathbf{R}^{\dagger} = \underset{\mathbf{U}}{\operatorname{argmin}} \|\mathbf{A} - \mathbf{C}\mathbf{U}\mathbf{R}\|_F$$

Theorem A6 ([15,21]). *Given a matrix* $\mathbf{A} = \mathbf{A}\mathbf{Z}\mathbf{Z}^T + \mathbf{E} \in \mathbb{R}^{m\times n}$, *where* $\mathbf{Z}^T\mathbf{Z} = \mathbf{I}_k$ *and* $\mathbf{Z} \in \mathbb{R}^{n\times k}$, *let* $\mathbf{S} \in \mathbb{R}^{n\times t}$ *be any matrix such that rank* $k = (\mathbf{Z}^T\mathbf{S})$. *Let* $\mathbf{C} = \mathbf{A}\mathbf{S} \in \mathbb{R}^{m\times r}$. *Then*

$$\begin{aligned}\|\mathbf{A} - \mathbf{C}\mathbf{C}^{\dagger}\mathbf{A}\|_\zeta^2 \le \|\mathbf{A} - \Pi_{\mathbf{C},k}^{\zeta}(\mathbf{A})\|_\zeta^2 \\ \le \|\mathbf{A} - \mathbf{C}(\mathbf{Z}^T\mathbf{S})^{\dagger}\mathbf{Z}^T\|_\zeta^2 \le \|\mathbf{E}\|_\zeta^2 + \|\mathbf{E}\mathbf{S}(\mathbf{Z}^T\mathbf{S})^{\dagger}\|_\zeta^2.\end{aligned}$$

Appendix B. Theorem 1 Proof

We first provide an essential lemma before proving the theorem.

Lemma A4 ([15]). *Given any* $\mathbf{Z} \in \mathbb{R}^{m\times p}$, $\mathbf{C} \in \mathbb{R}^{m\times q}$ *and* $\mathbf{A} \in \mathbb{R}^{m\times n}$, *assume* $\mathcal{R}(\mathbf{Z}) \subseteq \mathcal{R}(\mathbf{C}) \subseteq \mathcal{R}(\mathbf{A})$. *Let* $\mathbf{X} \in \mathbb{R}^{n\times n}$ *be a projection matrix. Then*

$$\|\mathbf{A} - \mathbf{C}\mathbf{C}^{\dagger}\mathbf{A}\mathbf{X}\|_F \le \|\mathbf{A} - \mathbf{Z}\mathbf{Z}^{\dagger}\mathbf{A}\mathbf{X}\|_F.$$

Now we start to prove Theorem 1.

Proof. According to Theorem A6, we have

$$\|\mathbf{A} - \mathbf{C}_1\mathbf{C}_1^{\dagger}\mathbf{A}\|_F^2 \le \|\mathbf{A} - \Pi_{\mathbf{C}_1,k}^{F}(\mathbf{A})\|_F^2 \le \|\mathbf{E}\|_F^2 + \|\mathbf{E}\mathbf{S}(\mathbf{Z}^T\mathbf{S})^{\dagger}\|_F^2.$$

Let $\mathbf{S} = \mathbf{\Omega}\Gamma$ and $\mathbf{E} = \mathbf{A} - \mathbf{A}\mathbf{Z}\mathbf{Z}^T$, then we have

$$\|\mathbf{E}\|_F^2 \le 2\|\mathbf{A} - \mathbf{A}_k\|_F^2 \tag{A1}$$

because of Lemma A1 with error parameter $\epsilon = 1$. In addition, $\mathbf{S}^T$ is a row leverage score sketching matrix of $\mathbf{Z}$, where Γ, $\mathbf{\Omega}$ and $\mathbf{Z}$ are calculated in Algorithm 3. Additionally, $\mathbf{S}^T$ is also a subspace embedding matrix of $\mathbf{Z}$ with error parameter $\epsilon_0 = 1/2$. Inferring from the fact that $(\mathbf{Z}^T\mathbf{S})^{\dagger} = (\mathbf{Z}^T\mathbf{S})^T(\mathbf{Z}^T\mathbf{S}\mathbf{S}^T\mathbf{Z})^{-1}$, we obtain

$$\begin{aligned}&\|\mathbf{E}\mathbf{S}(\mathbf{Z}^T\mathbf{S})^{\dagger}\|_F^2 = \|\mathbf{E}\mathbf{S}\mathbf{S}^T\mathbf{Z}(\mathbf{Z}^T\mathbf{S}\mathbf{S}^T\mathbf{Z})^{-1}\|_F^2 \\ &\le \|\mathbf{E}\mathbf{S}\mathbf{S}^T\mathbf{Z}\|_F^2\|(\mathbf{Z}^T\mathbf{S}\mathbf{S}^T\mathbf{Z})^{-1}\|_2^2 &\text{(A2)}\\ &\le \frac{1}{4k\log k}\|\mathbf{E}\|_F^2\|\mathbf{Z}\|_F^2\|(\mathbf{Z}^T\mathbf{S}\mathbf{S}^T\mathbf{Z})^{-1}\|_2^2 &\text{(A3)}\\ &\le \frac{1}{\log k}\|\mathbf{E}\|_F^2, &\text{(A4)}\end{aligned}$$

where Equation (A2) follows from the fact that $\|\mathbf{A}\mathbf{B}\|_F \le \|\mathbf{A}\|_2\|\mathbf{B}\|_F$, and Equation (A3) follows from Theorem A1 with error parameter $\varepsilon = 4k\log k$ and $\mathbf{E}\mathbf{Z} = \mathbf{A}(\mathbf{I} - \mathbf{Z}\mathbf{Z}^T)\mathbf{Z} = \mathbf{0}$. Due to Theorem A2 with error parameter $\epsilon_0 = 1/2$, Equation (A4) can be obtained. Because we have

$$\|\mathbf{S}^T\mathbf{Z}\|_2^2 = (1 \pm \epsilon_0)\|\mathbf{Z}\|_2^2 = (1 \pm \epsilon_0).$$

therefore,

$$\|(\mathbf{Z}^T\mathbf{S}\mathbf{S}^T\mathbf{Z})^{-1}\|_2^2 \le (1 - \epsilon_0)^{-2} = 4.$$

Due to Theorem A2, $\mathbf{S}$ needs $t = 4k\log k$ columns as a subspace embedding matrix of $\mathbf{Z}$ with error parameter $\epsilon_0 = 1/2$. Theorem A2 also leads to $\varepsilon = 4k\log k$ in the proof of Equation (A3). Now we have

$$\|\mathbf{A} - \mathbf{C}_1\mathbf{C}_1^{\dagger}\mathbf{A}\|_F^2 \le \|\mathbf{E}\|_F^2 + \frac{1}{\log k}\|\mathbf{E}\|_F^2 \le 4\|\mathbf{A} - \mathbf{A}_k\|_F^2, \tag{A5}$$

where the last inequality follows from Equation (A1) and $1/\log k \leq 1$.

Using Lemma 2, we need to sample $\mathcal{O}(\frac{k}{\epsilon})$ columns from $\mathbf{A}$ such that $\hat{\mathbf{C}} = [\mathbf{C}_1, \mathbf{C}_2]$ has the property

$$\|\mathbf{A} - \hat{\mathbf{C}}\hat{\mathbf{C}}^{\dagger}\mathbf{A}\|_F^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2.$$

Lemma A1 shows that there exists an othonormal matrix $\mathbf{Q}_k$ with rank k in the range of $\hat{\mathbf{C}}$ such that

$$\|\mathbf{A} - \mathbf{Q}_k\mathbf{Q}_k^T\mathbf{A}\|_F^2 \leq (1+\epsilon)\|\mathbf{A} - \hat{\mathbf{C}}\hat{\mathbf{C}}^{\dagger}\mathbf{A}\hat{\|}_F^2. \tag{A6}$$

$[\mathbf{C}_3] = AdptiveSampling(\mathbf{A}, \mathbf{Q}_k, \mathbf{C}_1, k/\epsilon)$, and we define $\tilde{\mathbf{C}} = [\mathbf{C}_1, \mathbf{C}_3]$, then by Lemma A3, it holds that

$$\begin{aligned}
&\|\mathbf{A} - \tilde{\mathbf{C}}\tilde{\mathbf{C}}^{\dagger}\mathbf{A}\mathbf{Q}_k^T\mathbf{Q}_k\|_F^2 \\
\leq &\|\mathbf{A} - \mathbf{Q}_k\mathbf{Q}_k^T\mathbf{A}\|_F^2 + \epsilon\|\|\mathbf{A} - \mathbf{C}_1\mathbf{C}_1^{\dagger}\mathbf{A}\|_F^2 \\
\leq &(1+\epsilon)\|\mathbf{A} - \hat{\mathbf{C}}\hat{\mathbf{C}}^{\dagger}\mathbf{A}\|_F^2 + 4\epsilon\|\mathbf{A} - \mathbf{A}_k\|_F^2 \\
\leq &(1+\epsilon)^2\|\mathbf{A} - \mathbf{A}_k\|_F^2 + 4\epsilon\|\mathbf{A} - \mathbf{A}_k\|_F^2 \\
= &(1+6\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2.
\end{aligned}$$

By rescaling the ϵ, we can obtain a $(1+\epsilon)$ relative error bound. Since $\mathcal{R}(\mathbf{Q}_k) \subseteq \mathcal{R}(\hat{\mathbf{C}}) \subseteq \mathcal{R}(\mathbf{A})$, Lemma A4 leads to

$$\|\mathbf{A} - \tilde{\mathbf{C}}\tilde{\mathbf{C}}^{\dagger}\mathbf{A}(\hat{\mathbf{C}}^{\dagger})^T\hat{\mathbf{C}}^T\|_F^2 \leq \|\mathbf{A} - \tilde{\mathbf{C}}\tilde{\mathbf{C}}^{\dagger}\mathbf{A}\mathbf{Q}_k^T\mathbf{Q}_k\|_F^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2.$$

Inferring from the fact that $\mathcal{R}(\hat{\mathbf{C}}) \subseteq \mathcal{R}(\mathbf{C}) \subseteq \mathcal{R}(\mathbf{A})$ and $\mathcal{R}(\tilde{\mathbf{C}}) \subseteq \mathcal{R}(\mathbf{C}) \subseteq \mathcal{R}(\mathbf{A})$, utilizing Lemma A4 twice, we reach the result that

$$\|\mathbf{A} - \mathbf{C}\mathbf{C}^{\dagger}\mathbf{A}(\mathbf{C}^{\dagger})^T\mathbf{C}^T\|_F^2 \leq \|\mathbf{A} - \tilde{\mathbf{C}}\tilde{\mathbf{C}}^{\dagger}\mathbf{A}(\hat{\mathbf{C}}^{\dagger})^T\hat{\mathbf{C}}^T\|_F^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2.$$

$\mathbf{S}$ is a leverage-score sketching matrix of $\mathbf{C}$, when $s = \mathcal{O}(\frac{c}{\epsilon} + c\log c)$ is the row dimension of $\mathbf{S}$; by Theorem 3 of [12], we have,

$$\|\mathbf{A} - \mathbf{C}\mathbf{U}\mathbf{C}^T\|_F^2 \leq \|\mathbf{A} - \mathbf{C}\hat{\mathbf{U}}\mathbf{C}^{\top}\|_F^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{C}\mathbf{C}^{\dagger}\mathbf{A}(\mathbf{C}^{\dagger})^T\mathbf{C}^T\|_F^2.$$

By rescaling ϵ, we achieve the final result that

$$\|\mathbf{A} - \mathbf{C}\mathbf{U}\mathbf{C}^T\|_F \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F.$$

□

References

1. Kumar, S.; Mohri, M.; Talwalkar, A. Sampling methods for the Nyström method. *J. Mach. Learn. Res.* **2012**, *13*, 981–1006.
2. Williams, C.; Seeger, M. Using the Nyström method to speed up kernel machines. In Proceedings of the 14th Annual Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 3–8 December, 2001; Number EPFL-CONF-161322, pp. 682–688.
3. Gittens, A.; Mahoney, M.W. Revisiting the Nyström Method for Improved Large-Scale Machine Learning. *arXiv* **2013**, arXiv:1303.1849.
4. Wang, S.; Zhang, Z. Improving CUR Matrix Decomposition and the Nyström Approximation via Adaptive Sampling. *J. Mach. Learn. Res.* **2013**, *14*, 2729–2769.
5. Anderson, D.G.; Du, S.S.; Mahoney, M.W.; Melgaard, C.; Wu, K.; Gu, M. Spectral Gap Error Bounds for Improving CUR Matrix Decomposition and the Nyström Method. In Proceedings of the AISTATS, San Diego, CA, USA, 9–12 May 2015.
6. Wang, S.; Gittens, A.; Mahoney, M.W. Scalable kernel K-means clustering with Nyström approximation: Relative-error bounds. *J. Mach. Learn. Res.* **2019**, *20*, 431–479.
7. Gao, S.; Dou, S.; Zhang, Q.; Huang, X. Kernel-Whitening: Overcome Dataset Bias with Isotropic Sentence Embedding. *arXiv* **2022**, arXiv:2210.07547.
8. Hamm, K.; Lu, Z.; Ouyang, W.; Zhang, H.H. Boosting Nyström Method. *arXiv* **2023**, arXiv:2302.11032.

9. Hsieh, C.J.; Si, S.; Dhillon, I.S. Fast Prediction for Large-Scale Kernel Machines. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 3689–3697.
10. Si, S.; Hsieh, C.J.; Dhillon, I. Memory efficient kernel approximation. In Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 21–26 June 2014; pp. 701–709.
11. Wang, S.; Luo, L.; Zhang, Z. SPSD Matrix Approximation vis Column Selection: Theories, Algorithms, and Extensions. *J. Mach. Learn. Res.* **2016**, *17*, 1697–1745.
12. Wang, S.; Zhang, Z.; Zhang, T. Towards More Efficient SPSD Matrix Approximation and CUR Matrix Decomposition. *J. Mach. Learn. Res.* **2016**, *17*, 1–49.
13. Boutsidis, C.; Woodruff, D.P. Optimal CUR Matrix Decompositions. *SIAM J. Comput.* **2017**, *46*, 543–589. [CrossRef]
14. Deshpande, A.; Vempala, S. Adaptive sampling and fast low-rank matrix approximation. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 292–303.
15. Ye, H.; Li, Y.; Zhang, Z. A simple approach to optimal CUR decomposition. *arXiv* **2015**, arXiv:1511.01598.
16. Woodruff, D.P. Sketching as a tool for numerical linear algebra. *arXiv* **2014**, arXiv:1411.4357.
17. Ye, H.; Wang, S.; Zhang, Z.; Zhang, T. Fast Generalized Matrix Regression with Applications in Machine Learning. *arXiv* **2019**, arXiv:1912.12008.
18. Drineas, P.; Magdon-Ismail, M.; Mahoney, M.W.; Woodruff, D.P. Fast approximation of matrix coherence and statistical leverage. *J. Mach. Learn. Res.* **2012**, *13*, 3475–3506.
19. Wang, S.; Zhang, Z. Efficient Algorithms and Error Analysis for the Modified Nystrom Method. In Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, Reykjavik, Iceland, 22–25 April 2014; pp. 996–1004.
20. Clarkson, K.L.; Woodruff, D.P. Low rank approximation and regression in input sparsity time. In Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing, Palo Alto, CA, USA, 2–4 June 2013; pp. 81–90.
21. Boutsidis, C.; Drineas, P.; Magdon-Ismail, M. Near-optimal column-based matrix reconstruction. *SIAM J. Comput.* **2014**, *43*, 687–717. [CrossRef]