An Improved Frequent Directions Algorithm for Low-Rank Approximation via Block Krylov Iteration

Chenhao Wang, Qianxin Yi, Xiuwu Liao, and Yao Wang

Abstract—Frequent directions (FDs), as a deterministic matrix sketching technique, have been proposed for tackling low-rank approximation problems. This method has a high degree of accuracy and practicality but experiences a lot of computational cost for large-scale data. Several recent works on the randomized version of FDs greatly improve the computational efficiency but unfortunately sacrifice some precision. To remedy such an issue, this article aims to find a more accurate projection subspace to further improve the efficiency and effectiveness of the existing FDs’ techniques. Specifically, by utilizing the power of the block Krylov iteration and random projection technique, this article presents a fast and accurate FDs algorithm named r-BKIFD. The rigorous theoretical analysis shows that the proposed r-BKIFD has a comparable error bound with original FDs, and the approximation error can be arbitrarily small when the number of iterations is chosen appropriately. Extensive experimental results on both synthetic and real data further demonstrate the superiority of r-BKIFD over several popular FDs algorithms both in terms of computational efficiency and accuracy.

Index Terms—Block Krylov iteration, frequent directions (FDs), low-rank approximation, randomized sketching, streaming algorithms.

I. INTRODUCTION

The objective of low-rank approximation is to approximate a given matrix by one with a low-rank structure. It is an essential tool in many applications, such as computer vision [1], signal processing [2], recommender systems [3], natural language processing [4], machine learning [5], principal component analysis (PCA) [6], and data mining [7], to name a few examples. The reason for finding a low-rank approximation is that, if we know in advance that the given matrix possesses a low-rank structure, then doing a low-rank approximation is a neat way to strip off meaningless noise and obtain a more compact representation. The distance between the original matrix and the approximate matrix is usually measured by the Frobenius norm. On this basis, the optimal low-rank approximation can be obtained by truncated singular value decomposition (SVD). However, considering a squared matrix $A$ with dimension $n$, the computational complexity of SVD is up to $O(n^3)$. This is prohibitive for large-scale datasets, especially when data are collected sequentially or parallelly, for instance, various applications receive data on the fly by varying the time period, including online advertising [8], sensor network [9], and network traffic [10].

One popular solution to remedy the computational burden of processing large data matrices is the so-called matrix sketching technique. The main idea is to construct a sketch matrix $B$, which is much smaller than the original matrix $A$, but can retain most of the information of $A$ and then use $B$ instead of $A$ to do the subsequent operations, such as SVD. More precisely, given an $n \times d$ matrix $A$, the goal is to find an $\ell \times d$ matrix $B$ with $\ell \ll n$ such that $A^T A \approx B^T B$. The efficiency of doing operations on the concisely representable sketch matrix makes this technique widely used in various applications, including kernel ridge regression [11], neural network training [12], and online learning [13], among many others.

For getting an approximate sketch matrix, many randomized techniques have drawn great attention, such as random sampling and random projection. The random sampling technique [14], [15], [16] obtains a precise representation of $A$ by sampling a small number of rows or columns and reweighting them. The most well-known random sampling technique is leverage score sampling, in which the sampling probability is proportional to the leverage score of each column. This obviously poses the difficulty that the leverage score involves the calculation of the singular vectors of $A$ and, thus, hard to process streaming data and large-scale data. Therefore, one may pay more attention to the random projection technique [17] whose key is to find a random matrix $X$ used to project the original matrix $A$ to a much smaller matrix $B$. It is requested that the construction of $X$ should guarantee that $B$ captures the principal subspace of $A$. In addition, even a single pass over data is sufficient [18]; this enables approximation for dense matrices that cannot be loaded completely in memory. These two techniques can be combined together to obtain more efficient estimation [19].

Besides the aforementioned randomized method for constructing the sketch, a deterministic method named frequent
directions (FDs) [21], [22] was proposed recently. The inspiration behind this method is from estimating item frequency in a stream [23]. Precisely, for a given matrix rows of $A$ in a stream and maintains an $\ell \times d$ sketch $B$. Then, by setting $\ell = k + \frac{k}{\varepsilon}$, it achieves $(1 + \varepsilon)$ best rank-$k$ approximation, while it costs $O(d\ell)$ space and runs in $O(nd\ell)$ time. Hence when facing massive data, such as surveillance video data, the efficiency is still limited due to the space restriction. Several attempts have been made to tackle this issue by combining the idea from random projection [20], [24], [25]. For example, Teng and Chu [20] proposed a method named SpFD that uses the CountSketch matrix to capture more information from the original matrix and, thus, can accelerate the computation significantly. However, there is still room for further improvement, considering that the projection procedure cannot be often capable of compressing the information accurately, leading to deteriorated performance. As a typical case shown in Fig. 1, when the singular values decay rapidly, SpFD can obtain a good approximation when the sketch size is over 50. However, when the decrease is slow, SpFD could not provide a satisfactory result even for a larger sketch size. This is mainly because the efficiency of random projection relies on the gap between the $k$th singular value and the $(k + 1)$th singular value [26]. Therefore, to improve the accuracy of the randomized variants of FD, there is an urgent need for finding a more accurate projection subspace.

The Krylov subspace method was first introduced in solving a system of linear equations [27] and then generalized by Golub and Underwood [28], and more recently, Musco and Musco [26] brought it into the randomized SVD and established the first gap-free theoretical analysis for the low-rank approximation. Compared with another popular technique, i.e., subspace iteration [17], [29], for improving the precision in low-rank approximation, the Krylov subspace method utilizes fewer iterations for achieving the same precision theoretically and experimentally. This motivates us to apply the Krylov subspace method to accelerate the FD algorithm, while it could maintain its precision.

To sum up, we propose in this work a fast and accurate FDs algorithm by incorporating the power of Krylov subspace and random projection techniques. Primarily, each block of matrix $A$ is compressed by a nonoblivious projection matrix constructed from the Krylov subspace and then embedded into the FD procedure to update the sketch matrix $B$. Our main contributions are summarized as follows.

1) The newly proposed FD algorithm named r-BKIFD can skillfully integrate block Krylov iteration into randomized FD so that we can obtain a more accurate subspace during the projection procedure. Taking the dense Gaussian random matrix and the sparse CountSketch matrix as examples, we demonstrate the merits of the proposed r-BKIFD in terms of the approximation error and computational speed.

2) The theoretical analysis shows that our method has comparable theoretical guarantees with the original FD. Specifically, we derive the error bounds in terms of both the covariance and projection errors. It is also shown that such error bounds would be arbitrarily small if we choose an appropriate number of iterations.

3) Extensive experiments are carried out on both synthetic and real data to show that the proposed r-BKIFD outperforms traditional FD and its variants in most cases.

**Notations:** For an $n \times d$ matrix $A$, $a_i$ denotes its $i$th row, and $a_{ij}$ denotes its $(i, j)$th element. The matrices $I_n$ represent the $n$-dimensional identity matrix, and $0_{n \times d}$ is the all-zero valued matrix with dimension $n \times d$. The Frobenius norm of $A$ is defined as $\|A\|_F = (\sum_i \sum_j a_{ij}^2)^{1/2}$, and the spectral norm of $A$ is $\|A\|_2 = \sup_{\|x\|=1} \|Ax\|$. The rank-$k$ approximation of $A$ is expressed as $A_k = U_k \Sigma_k V_k^T$, where $A = U \Sigma V^T$ represents the SVD of $A$. Let $\sigma_i$ denote the $i$th singular value of $A$. The notation $nnz(A)$ denotes the number of nonzero entries of $A$. We use $\tilde{O}(n)$ to hide the logarithmic factor on $n$.

**II. RELATED WORK**

As a basic dimension reduction method, low-rank approximation of large-scale matrices is a ubiquitous tool in scientific
computing, machine learning, and numerical analysis, among a number of other areas [30], [31], [32]. It can be mainly formulated as the problem that for a given matrix $M$ and an input parameter $k$, one would like to find a matrix $M'$ of rank at most $k$ to minimize the Frobenius norm of the discrepancy between $M$ and $M'$, i.e.,

$$\min_{\text{rank}(M') \leq k} \|M - M'\|_F.$$ 

The classic Eckart–Young–Mirsky theorem shows the best low-rank matrix approximation can be obtained from the truncated SVD. However, for a matrix $M \in \mathbb{R}^{n \times d}$, the computational complexity of calculating the truncated SVD is $O(nd^2)$, which is unacceptable for large-scale matrix data. Sketching algorithms have been proposed to alleviate the heavy computational cost by mapping the input matrix to a smaller surrogate matrix called sketch; thus, one can perform the low-rank approximation on the such sketch as an alternative.

### A. Randomized Sketching Techniques

Popular matrix sketching techniques include many randomized algorithms, such as random sampling [33] and random projection [34]. Random sampling forms a sketch by finding the small subset of rows or columns based on a predefined probability distribution. Random projection allows the original matrix to be efficiently processed in a lower dimensional space by using a random matrix, such as Gaussian [35], CountSketch [36], and substituted randomized Hadamard transform (SRHT) [37]. The Johnson–Lindenstrauss (JL) lemma [38] shows that such random matrices can preserve the pairwise distance between any two data points. It is known that the $d \times m$ Gaussian random matrix $S$ is defined in the form of $S = (1/(m)^{1/2})G$, where each entry of $G$ is sampled independent identically distributed (i.i.d.) from $N(0, 1)$. The CountSketch matrix stems from estimating the most frequent items in a data stream [39], further applied in performing low-rank approximation [36]. Mathematically, it is constructed as $X = D\Phi^T \in \mathbb{R}^{d \times m}$, where the following holds.

1. $\Phi \in \mathbb{R}^{m \times d}$ is a binary matrix with $\Phi_{h(i), j} = 1$ and $\Phi_{j, i} = 0$ for all $j \neq h(i)$. Here, $h$ is a uniformly random map from $[d] \rightarrow [m]$.
2. $D$ is a $d \times d$ diagonal matrix with each diagonal element chosen from $\{-1, 1\}$ with equal probability.

Note that the CountSketch matrix is an extremely sparse matrix due to the structure of one nonzero element per row. Thus, given the input matrix $A \in \mathbb{R}^{n \times d}$ and the CountSketch matrix $X \in \mathbb{R}^{d \times m}$, the computation cost to get $AX$ is $O(mn(A))$, which is superior to the costs of $O(ndm)$ for Gaussian and $O(nd\log(m))$ for SRHT [37], [40]. Clarkson and Woodruff [36] further illustrated that CountSketch is the fastest known procedure for low-rank approximation. Thus, it is well suitable for sparse data and has been frequently applied to various applications, including differential privacy [41] and deep learning [42], among others.

### B. Frequent Directions

Different from the above randomized techniques, the FDs algorithm is first proposed by Liberty [21] as a deterministic matrix sketching technique. Instead of projecting or sampling the whole matrix at once, it processes the matrix in a row-update approach. That is, given an input matrix $A \in \mathbb{R}^{n \times d}$, the goal is to construct a sketch matrix $B \in \mathbb{R}^{(\ell - 1)\times d}$, which is much smaller than $A$ but is still a good estimation. Precisely, the matrix $B$ is initialized to an all-zero valued matrix. We insert the rows of $A$ into $B$ until it is fulfilled. Then, a shrinkage procedure is conducted by computing the SVD and subtracting the squared singular value from all squared singular values. Considering that the last row of $B$ is always all-zero valued after the shrinkage procedure, we could insert continually until all rows are processed. The running time is dominated by the SVD, which takes $O(d\ell^2)$ time, so the total time cost is $O(nd\ell^2)$. Later, Ghashami et al. [22] modify the original FD by doubling the space of $B$ to reduce the running time to $O(nd\ell)$. See more details in Algorithm 1. It has been shown that FD has the following error bound holds for any $k < \ell$:

$$\|A^T A - B^T B\|_2 \leq \frac{\|A - A_k\|_F^2}{\ell - k},$$

(1)

Noting that setting $\ell = [k + 1/\epsilon]$ yields error of $\epsilon\|A - A_k\|_F^2$, that is, the sketch matrix $B$ is a good low-rank approximation.

Since FD is of high accuracy guarantee and well suitable for streaming settings, several studies have embedded it into online learning. Boutsidis et al. [43] proposed the online version of PCA (OPCA) and embedded FD in it to reduce both the time and space complexities. Leng et al. [44] utilized FD to learn hash functions in an online fashion with low computational complexity and storage space. Besides, Lin et al. [45] showed that FD can efficiently preserve the second-order information in neural network pruning problems. Recently, many improvements have been considered in improving the accuracy and efficiency. Luo et al. [46] proposed robust FDs (RFDs) by introducing an adaptive regularizer to improve the approximation error bound by a factor of $1/2$. Furthermore, Huang [47] considered using the random sampling technique to sketch the part removed in the singular values shrinkage procedure and proved that it is a space-optimal algorithm with a faster running time. Besides, several studies considered random projection techniques to improve efficiency. Ghashami et al. [48] considered the sparsity of the original

**Algorithm 1** Fast-FDs [22]

**Require:** $A \in \mathbb{R}^{n \times d}$, sketch size $\ell$

**Ensure:** $B \in \mathbb{R}^{(\ell - 1)\times d}$

1. $B \leftarrow 0^{\ell\times d}$
2. for $i = 1, \ldots, n$ do
3. Insert $A_i$ into a zero valued row of $B$
4. if $B$ has no zero valued rows then
5. $[U, \Sigma, V] \leftarrow \text{svd}(B)$
6. $\delta \leftarrow \sigma_k^2$
7. $B \leftarrow \sqrt{\text{max}(\Sigma^2 - \delta I_{\ell}, 0)} \cdot V^T$
8. end if
9. end for
10. return $B \leftarrow (B_1 : \ell - 1 : \cdot)$

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
matrix and combined the randomized SVD to accelerate FD. Chen et al. [24] proposed the so-called Faster FDs by utilizing SRHT on each data batch and then performing FD on the compact small matrix. Teng and Chu [20] combined FD with the CountSketch matrix to achieve comparable accuracy with low computational cost. Yi et al. [49] extended FD to deal with streaming low-rank tensor approximation problems. Considering that the integration of random projection techniques would deteriorate the accuracy, this work aims at finding a more precise projection method without losing much accuracy while maintaining the algorithm running efficiently.

C. Block Krylov Iteration

Historically, the classical Lanczos algorithm was first proposed by Lanczos [27] to compute the extremal eigenvalues and corresponding eigenvectors of symmetric matrices and then generalized by Golub and Kahan [50] to solve the singular value pairs of $m \times n$ nonsymmetric matrices. The basic idea is to construct the Krylov subspace for the initial vector $v$ and then project the original matrix onto this subspace. Thus, by using the eigenvalue pairs of the projection matrix to approximate the counterpart of the original matrix, we get the Krylov subspace described as

$$K := [v, A v, A^2 v, \ldots, A^{q-1} v].$$

Note that the involved $v$ is a single vector; the works [28], [51] modified the Lanczos algorithm from single vector $v$ to a block of $b$ vectors $V = [v_1, \ldots, v_b]$ and built the Krylov subspace as

$$K := [V, A V, A^2 V, \ldots, A^{q-1} V].$$

Compared to classical Lanczos, block Lanczos is more efficient in terms of memory and cache. Recently, with the explosive development of the randomized algorithm, the randomized block Krylov algorithm has emerged, which could be seen as an integration of classical block Lanczos algorithm with randomized starting matrix $AX$, where $X$ is a random matrix that is chosen as the Gaussian random matrix. The detailed procedure is presented in Algorithm 2.

The key idea is to take the random projection $V = AX$ as the initial matrix, instead of the arbitrary set of vectors $V$ that may result in poor convergence. The convergence analysis proposed by Musco and Musco [26] reveals that this method has a faster convergence rate with respect to the number of iterations and can capture a more accurate range space compared with the popular simultaneous iteration (also known as power iteration) technique, which is defined as

$$K := (AA^T)^q AX.$$ 

The requirement of $q$ is just $\Theta((\log d/\epsilon)^{1/2})$ for getting the $(1 + \epsilon)$ relative-error bound for the block Krylov iteration, instead of $\Theta((\log d/\epsilon^3)$ for power iteration. This shows that the block Krylov iteration can get the accuracy guarantee with fewer iterations. A more detailed theoretical analysis could be found in [52] and [53].

### Algorithm 2 Block Krylov Iteration [26]

**Require:** $A \in \mathbb{R}^{n \times d}$, error $\epsilon \in (0, 1)$, rank $k \leq n$

**Ensure:** $Z \in \mathbb{R}^{n \times k}$

1. $q := \Theta((\log d)/\sqrt{\epsilon})$, $X \sim \mathcal{N}(0, 1)^d \times k$
2. $K := [AX, (AA^T)AX, \ldots, (AA^T)^q AX]$
3. Orthogonalize the columns of $K$ to obtain $Q$
4. Compute $M := Q^T AA^T Q$
5. Set $U_k$ to the top $k$ singular vectors of $M$
6. return $Z = Q U_k$

### Algorithm 3 r-BKI

**Require:** $A \in \mathbb{R}^{n \times d}$, error $\epsilon \in (0, 1)$, integers $m, \ell$

**Ensure:** $Z \in \mathbb{R}^{n \times \ell}$, $P \in \mathbb{R}^{\ell \times d}$

1. $q := \Theta((\log d)/\sqrt{\epsilon})$, let $X \in \mathbb{R}^{d \times m}$ be a randomized matrix
2. $K := [AX, (AA^T)AX, \ldots, (AA^T)^q AX]$
3. Orthogonalize the columns of $K$ to obtain $Q \in \mathbb{R}^{n \times (q+1)m}$
4. Compute $M := Q^T AA^T Q \in \mathbb{R}^{(q+1)m \times (q+1)m}$
5. Set $U_\ell$ to the top $\ell$ singular vectors of $M$
6. $Z = Q U_\ell$
7. return $P = Z^T A$

III. PROPOSED ALGORITHM

For universal large-scale streaming datasets, although many randomized FD variants achieve low computational cost, the algorithmic accuracy is sacrificed to a certain extent for getting the low-rank approximation. Considering that the block Krylov iteration gives nearly optimal low-rank approximation with the fastest known theoretical runtime, we present a new algorithm named r-BKIFD, which incorporates the block Krylov iteration technique into FD to reduce the computational cost with better accuracy guarantee.

The classical block Krylov iteration is limited to the Gaussian random matrix for starting guess, and we can extend it to the CountSketch random matrix by observing that some of the real-world datasets are extremely sparse, such as hyperspectral data [54], recommendation data [55], and speech spectrograms [56]. For computing $AX$, when $X$ is the CountSketch matrix, the computation complexity is only $O(mn z(A))$ instead of $O(ndk)$ for the Gaussian matrix. We, thus, could sequentially transform the input matrix without explicitly generating the CountSketch matrix for the case that the input matrix could not fit in memory. The above excellent properties make CountSketch perform well when constructing the Krylov subspace especially when the data matrices are sparse. Therefore, our subsequent analysis is based on both Gaussian random and CountSketch matrices.

We now give a detailed description of the proposed r-BKI algorithm. For getting a more accurate approximation, we set $m \geq \ell$. First, we apply $(AA^T)^q A$ ($i = 0, 1, \ldots, q$) to form the Krylov matrix $K$, which contains all the information accumulated along the projection process, and then, we employ an orthonormal procedure to obtain the newly compressed matrix $Z$ and $P$. See Algorithm 3 for a detailed description of r-BKI.

Then, we illustrate how to integrate r-BKI into FD. Given the streaming data $A = [A_{(1)}; \ldots ; A_{(t)}] \in \mathbb{R}^{n \times d}$ with each
Algorithm 4 r-BKIFD

Require: $A \in \mathbb{R}^{n \times d}$, error $\varepsilon \in (0, 1)$, integers $m, \ell$
Ensure: $B \in \mathbb{R}^{(\ell-1) \times d}$
1: $B \leftarrow r$-BKIFD($A(1, :)$, $\varepsilon$, $m, \ell$)
2: for $i \in 1, \ldots, s - 1$
3: $P \leftarrow r$-BKIFD($A(i : n, :)$, $\varepsilon$, $m, \ell$)
4: $B \leftarrow B; P$
5: $[U, \Sigma, V] \leftarrow \text{svd}(B)$
6: $\delta \leftarrow \sigma_i^2$
7: $B \leftarrow \sqrt{\max(\Sigma^2 - \delta I, 0)} \cdot V^T$
8: $B \leftarrow B(1 : \ell - 1, :)$
9: end for
10: return $B$

$A(i) \in \mathbb{R}^{(n/s) \times d}$ ($i = 1, 2, \ldots, s$), our goal is to obtain a small sketch $B \in \mathbb{R}^{(\ell-1) \times d}$ that offers good performance in preserving crucial information of the original matrix $A$. We assume that $n/s$ is an integer; otherwise, we can change it into an integer by appending zero rows to $A$. Compared with the classical FD that directly performs singular values shrinkage procedure on each $\ell$ rows of $A$, we mainly embed the r-BKI technique for each batch $A(i)$ ($i = 1, 2, \ldots, s$) to obtain an intermediate sketch matrix with more compact representation but preserves the accuracy and then perform Fast-FD on the intermediate sketch matrix. Here, r-BKI is used to find a more accurate subspace representation with less computational complexity during the projection procedure. The detailed procedure is listed in Algorithm 4. Precisely, for each batch $A(i)$, we apply the r-BKI algorithm to compress it into a relatively small intermediate sketch matrix $P$. The sketch matrix $B$ is initialized as the first intermediate matrix $P$. Then, for the rest of the data, each time, we append the sketch $P$ into $B$. Similar to the traditional FD, we perform the singular values shrinkage procedure and maintain the first $\ell - 1$ rows of $B$, and as a result, the remaining rows of $B$ are set to be zeros and replaced by the next intermediate sketch matrix $P$. This iterative process continues until all batches are processed. The illustration is shown in Fig. 2.

IV. ERROR BOUNDS

In this section, we theoretically analyze the accuracy of the proposed algorithm r-BKIFD. To this end, we shall first introduce some useful lemmas.

Lemma 1 ([52], Th. 2.3): Given data matrix $A \in \mathbb{R}^{n \times d}$ and the random matrix $X \in \mathbb{R}^{d \times m}$, let the sketch $Z \in \mathbb{R}^{n \times l}$ be constructed by Algorithm 3. The best rank-$l$ approximation to $A$ can be written as $A_l = U_l \Sigma_l V_l^T$, and let $l < \text{rank}(A)$. If rank $(V_l^T X) = \ell$, then

$$\|A - ZZ^T A\|_2 \leq \|A - A_l\|_2 + \|\phi(\Sigma_{l+\ell})\|_2 \|V_{l+\ell} X (V_l^T X)\|_F.$$

Lemma 1 characterizes the distance between the input matrix $A$ and the projection matrix $P$ in the block Krylov iteration step. Note that the upper bound of $\|A - ZZ^T A\|_2$ is closely related to the properties of the random matrix $X$. That is to say, this lemma is helpful for choosing an appropriate random matrix to estimate the original matrix accurately.

A well-behaved random matrix $X$ could tighten the second part $\|V_{l+\ell} X (V_l^T X)\|_F$. In the following lemma, we will try to bound $\|\phi(\Sigma_{l+\ell})\|_2$.

Lemma 2 ([52], Lemma 2.4): If $(\sigma_i - \sigma_{i+1}/\sigma_{i+1}) \geq \gamma > 0$ holds, then there exists a polynomial $\phi(x)$ of degree $2q + 1$ with odd powers only such that $\phi(\sigma_i) \geq 0$ for $1 \leq i \leq \ell$, and

$$\|\phi(\sigma_i)\|_2 \leq \frac{4 \gamma^{q+1} \min(\gamma, 1)}{2(2q+1) \min(\gamma, 1)}.$$

Hence,

$$\|\phi(\Sigma_{l+\ell})\|_2 \leq \|\phi(\Sigma_{l+\ell})\|_2 \leq \frac{4 \gamma^{q+1} \min(\gamma, 1)}{2(2q+1) \min(\gamma, 1)}.$$

It is not hard to see that, as the number of iterations $q$ and singular value gap $\gamma$ increase, $\|\phi(\Sigma_{l+\ell})\|_2$ will exponentially decay. Moreover, when the sketch size $\ell$ increases, $\sigma_{i+1}$ gets smaller, which also makes the bound actually tighter.

Remark 1: As $q$ increases, the error bound is drastically reduced. The essential reason is the introduction of the Krylov subspace. Unlike the power method that aims at computing the dominant eigenspace, Krylov subspace contains the information accumulated along the way it is used. This construction of the Krylov matrix makes full use of the original matrix so that less information is lost during projection.

Lemma 3 (Matrix Bernstein Inequality [57]): Let $\{A_i\}_{i=1}^s \in \mathbb{R}^{n \times d}$ be independent random matrices with $E[A_i] = 0$ and $\|A_i\|_2 \leq R$ for all $i \in [s]$. Define a variance parameter as $\sigma^2 = \max\{\|\sum_{i=1}^s E[A_i A_i^T]\|_2, \sum_{i=1}^s E[\|A_i^T A_i\|_2]\}$. Then, for all $\epsilon \geq 0$, we have

$$\mathbb{P}\left(\left\|\sum_{i=1}^s A_i\right\|_2 \geq \epsilon\right) \leq \left(\frac{d + n}{\epsilon^2} \right) \exp\left(\frac{-\epsilon^2/2}{\sigma^2 + R\epsilon/3}\right).$$

Lemma 4 (Courant–Fischer Min–Max Theorem [58]):

Let $A$ be an $n \times n$ Hermitian matrix with eigenvalues $\lambda_1 \leq \cdots \leq \lambda_k \leq \cdots \leq \lambda_n$; then,

$$\lambda_k = \min_{x \not\in \{0\}} \max_{x \neq 0} \frac{R_A(x)}{\|x\|_2}$$

where the Rayleigh–Ritz quotient $R_A : \mathbb{C}^n \setminus \{0\} \to \mathbb{R}$ is defined by

$$R_A(x) = (Ax, x) / (x, x),$$

and $(\cdot, \cdot)$ denotes the Euclidean inner product on $\mathbb{C}^n$.

A. Error Bounds for GA-BKIFD

It is known that the Gaussian random matrix has high-quality sketch accuracy and is easy to implement [59]. In this section, we apply it to the r-BKIFD algorithm and call the algorithm as GA-BKIFD. The theoretical performance is guaranteed by the following theorem.

Theorem 1 (Covariance error of GA-BKIFD): Given data matrix $A = [A(1) ; \cdots ; A(s)] \in \mathbb{R}^{n \times d}$, where each $A(i) \in \mathbb{R}^{n \times d}$, let the small sketch $B \in \mathbb{R}^{(\ell-1) \times d}$ be constructed by Algorithm 4, where $X$ is a Gaussian random matrix. For any $\eta \in (0, 1)$ and
\( \varepsilon \geq 0, \) if \((\sigma_{\ell} - \sigma_{\ell+1}/\sigma_{\ell+1}) \geq \gamma > 0, \) then, with probability at least \( 1 - 2s \exp(-\varepsilon^2/2) - \eta, \) we have

\[
\|A^T A - B^T B\|_2 \leq \left( s(1 + \delta) + \log \left( \frac{2d}{\eta} \right) \frac{4(1 + \delta)}{3} \right. \\
\left. + \sqrt{2s(1 + \delta)^2 \log \left( \frac{2d}{\eta} \right)} \right) \times \|A - A_{\ell}\|_F^2 + \frac{\|A - A_{k}\|_F^2}{\ell - k} \quad (2)
\]

where \( 1 + \delta = (1 + (4/2^{(2q+1)\text{min}(d,\ell)/2}))((d - \ell)^{1/2}/((d)^{1/2} + (m)^{1/2} + \varepsilon)/(m)^{1/2} - (\ell^{1/2} - \varepsilon)^2), \) and \( \sigma_i \) is the singular values of \( A \) in descending order.

To explore the trend of the error bound with each variable more conveniently and clearly, we hide the logarithmic factor on \((d, \eta). \) Note that \( (\|A - A_{\ell}\|_F^2/\ell - k) \) decreases obviously as \( \ell \) increases. Thus, we focus on analyzing the effect of \( O(s(1 + \delta))\|A - A_{\ell}\|_F^2. \) First, a small increase in \( q \) and \( \gamma \) can lead to exponential decay in \( \delta; \) therefore, for the fixed singular value gap \( \gamma , \delta \) can become arbitrarily small if an appropriate \( q \) is chosen. Second, with the increase in the size of \( \ell, \) the \((\ell + 1)\text{th singular value (i.e., } \|A - A_{\ell}\|_F^2 \) becomes smaller; we stress that this advantage is even more significant when the singular value gap \( \gamma \) is large.

Note that the above analysis is based on covariance error; now, we introduce a key lemma that illustrates the relationship between covariance error and projection error.

**Lemma 5 (Covariance Error to Projection Error [47]):**

\[
\|A - \pi^k_B(A)\|_F^2 \leq \|A - A_{k}\|_F^2 + 2k \cdot \|A^T A - B^T B\|_2 \quad (3)
\]

where \( \pi^k_B(A) \) is the projection of \( A \) onto the top-k singular vectors of \( B. \)

This lemma shows that, as long as we obtain the error bound of the covariance error, we can also get the error bound of the projection error. This property is very important in the low-rank approximation. Many studies focus on the covariance error because it can reveal the difference between two matrices more substantially.

The following corollary shows the projection error of GA-BKIFD, which follows by combining (2) and (3).

**Corollary 1:** Given data \( A = [A_{(1)}; \cdots; A_{(t)}] \in \mathbb{R}^{n \times d} \), where each \( A_{(i)} \in \mathbb{R}^{(n/\gamma) \times d}, \) let the small sketch \( B \in \mathbb{R}^{(\ell - 1) \times d} \) be constructed by Algorithm 4, where \( X \) is a Gaussian random matrix. For any \( \eta \in (0, 1) \) and \( \varepsilon \geq 0, \) if \((\sigma_{\ell} - \sigma_{\ell+1}/\sigma_{\ell+1}) \geq \gamma > 0, \) then, with probability at least \( 1 - 2s \exp(-\varepsilon^2/2) - \eta, \) we have

\[
\|A - \pi^k_B(A)\|_F^2 \leq 2k \left( s(1 + \delta) + \log \left( \frac{2d}{\eta} \right) \frac{4(1 + \delta)}{3} \right. \\
\left. + \sqrt{2s(1 + \delta)^2 \log \left( \frac{2d}{\eta} \right)} \right) \times \|A - A_{\ell}\|_F^2 + \frac{\|A - A_{k}\|_F^2}{\ell - k} \quad (2)
\]

where \( 1 + \delta = (1 + (4/2^{(2q+1)\text{min}(d,\ell)/2}))((d - \ell)^{1/2}/((d)^{1/2} + (m)^{1/2} + \varepsilon)/(m)^{1/2} - (\ell^{1/2} - \varepsilon)^2), \) and \( \sigma_i \) is the singular values of \( A \) in descending order.

**B. Error Bounds for CS-BKIFD**

As mentioned earlier, the Gaussian random matrix has been well applied to the proposed r-BKIFD algorithm. However, when the dimension of the input matrix reaches a larger scale, the time complexity to perform matrix multiplication is too high. This is because it destroys the sparse nature of the original matrix if the input matrix is sparse. To address this issue, we introduce the CountSketch matrix with the sparse structure to r-BKIFD in this section, which is called CS-BKIFD. Its error bound in terms of the covariance error is listed in the following theorem.

**Theorem 2 (Covariance Error of CS-BKIFD):** Given data \( A = [A_{(1)}; \cdots; A_{(t)}] \in \mathbb{R}^{n \times d} \), where each \( A_{(i)} \in \mathbb{R}^{(n/\gamma) \times d}, \) let the small sketch \( B \in \mathbb{R}^{(\ell - 1) \times d} \) be constructed by Algorithm 4, where \( X \) is a CountSketch matrix. For any \( \varepsilon, p, \eta \in (0, 1), \)
if \((\sigma_i - \sigma_{i+1})/\sigma_{i+1} \geq \gamma > 0\) and \(m \geq (\ell^2 + (\ell/\varepsilon^2) p)\), then, with probability at least \(1 - s\rho - \eta\), we have
\[
\|A^T B \|_2 \leq \left( s(1 + \delta) + \log \left( \frac{2m}{\eta} \right) \right) \frac{4(1 + \delta)}{3} + \sqrt{2s(1 + \delta)^2 \log \left( \frac{2d}{\eta} \right)} \times \|A - A_k\|_F^2 + \frac{\|A - A_k\|_F^2}{\ell - k} 
\]
where \(1 + \delta = (1 + (4/4(2q+1)^{\min(\gamma)}1/2)^{1/2}))((d(d - \ell)/1 - \varepsilon)/\varepsilon^2)^{1/2})\), and \(\sigma_i\) is the singular values of \(A\) in descending order.

Similar to the above analysis, the projection error bound of CS-BKIFD can be obtained immediately by combining (3) and (4).

**Remark 2:** The core analysis of the error bound is consistent with the algorithm GA-BKIFD. In addition, as the sketch size \(\ell\) increases, the algorithmic accuracy is improved, which will be verified in the experimental study.

**Corollary 2:** Given data \(A = [A^{(1)}; \ldots; A^{(m)}] \in \mathbb{R}^{n \times d}\) where each \(A^{(i)} \in \mathbb{R}^{(n/\ell) \times d}\), let the small sketch \(B \in \mathbb{R}^{(\ell - 1) \times d}\) be constructed by Algorithm 4, where \(X\) is a CountSketch matrix. For any \(\varepsilon, \rho, \eta \in (0, 1)\), if \((\sigma_i - \sigma_{i+1})/\sigma_{i+1} \geq \gamma > 0\) and \(m \geq (\ell^2 + (\ell/\varepsilon^2) p)\), then, with probability at least \(1 - s\rho - \eta\), we have
\[
\|A - \pi^k_F(A)\|_F^2 \leq 2k \left( s(1 + \delta) + \log \left( \frac{2m}{\eta} \right) \right) \frac{4(1 + \delta)}{3} + \sqrt{2s(1 + \delta)^2 \log \left( \frac{2d}{\eta} \right)} \times \|A - A_k\|_F^2 + \frac{\|A - A_k\|_F^2}{\ell - k} 
\]
where \(1 + \delta = (1 + (4/4(2q+1)^{\min(\gamma)}1/2)^{1/2}))((d(d - \ell)/1 - \varepsilon)/\varepsilon^2)^{1/2})\), and \(\sigma_i\) is the singular values of \(A\) in descending order.

**C. Comparison of GA-BKIFD and CS-BKIFD**

We shall make a comparison of GA-BKIFD and CS-BKIFD in terms of accuracy and running time. For the algorithmic accuracy, we observe that the covariance error bound can achieve \(\overline{O}(s(1 + \delta)) \|A - A_k\|_F^2 + (\|A - A_k\|_F^2/\ell - k)\) when \(q\) is large according to Theorems 1 and 2. However, the random size \(m\) should satisfy \(m \geq \ell\) for GA-BKIFD, while \(m \geq (\ell^2 + (\ell/\varepsilon^2) p)\) for CS-BKIFD. Therefore, GA-BKIFD achieves almost the same accuracy guarantees with fewer sampling numbers. For the algorithmic running time, the matrix multiplication operation generated in the construction of the Krylov subspace takes up a lot of time. Fortunately, because of the sparse structure of the CountSketch matrix, CS-BKIFD could run faster in this step, that is to say, it has a lower computational complexity just as Section IV-D shows and, thus, works very well in some practical situations.

**D. Comparison With Existing Algorithms**

Table I shows the detailed comparison in terms of the projection and covariance errors. For easy and intuitive comparison, we rewrite the original error bounds and use \(\overline{O}\) to hide the logarithmic.

First, we can observe that the error bounds of the traditional deterministic algorithm FD are sharper than all these randomized FD variants. This is mainly because the techniques that one uses to derive the error bounds of the randomized FD variants still rely on the properties of FD. Besides, it can be emphasized here that the error bounds of the proposed r-BKIFD are superior to FFD and SpFD according to the following detailed comparative analysis.

In terms of the covariance error, our bound is tighter than FFD from two aspects. First, the first term of our bound is related to the \((\ell + 1)\)th singular value of \(A\) rather than the largest singular value in FFD, which is more advantageous when sketch size \(\ell\) is large. Second, a small increase in \(q\) and \(\gamma\) can lead to exponential decay in \(\delta\), and as a result, for the fixed singular value gap \(\gamma\), if we choose an appropriate \(q, \delta\) in our bound can be arbitrarily small, whereas \(\Delta_i\) in FFD cannot be. The reason is that \(\Delta_i = \Theta((\min(n/s), d)\log(2\min(n/s), d)/\delta)/\ell/\ell/2)^{1/2})\), and one can observe that \(\Delta_i\) cannot decay exponentially with one of the variables. We stress that such two advantages come from the incorporation of the block Krylov iteration technique.

In terms of the projection error, our bound is much tighter than SpFD in most cases. That is, it is easy to check that, when \(\delta_1 \leq (1/2)\) and \(\ell \leq (k/8s)(1 + \delta)(\log(2d/\eta)) + k\), r-BKIFD can achieve a tighter upper bound than SpFD. Such two assumptions can be satisfied most often because the failure probability \(\delta_1\) should be small. By the way, although Teng and Chu [20] aim to analyze the error bound \(\|A - [AV]kV^T\|^2_F\) of low-rank approximation, it is essentially analyzing the projection error due to the use of inequality \(\|A - [AV]kV^T\|^2_F \leq \|A - AV_kV^T\|^2_F\) in our analysis.

**E. Complexity Analysis**

The running time of r-BKIFD is dominated by step 4 of performing r-BKI, that is, the procedure for computing the sketch

**TABLE I**

**Comparison With FD and Its Randomized Variants**

| Algorithm     | Covariance error | Projection error |
|---------------|------------------|-----------------|
| FD [22]       | \(\frac{1}{4\pi^k} \|A - A_k\|^2_F\) | \(\frac{1 + \frac{k}{\ell}}{4\pi^k} \|A - A_k\|^2_F\) |
| FFD [24]      | \(\max \overline{O}(\sqrt{\gamma(1 + \Delta)}(A_{(1)})^2 + \varepsilon \|A - A_k\|_F^2)\) | N/A |
| SpFD [20]     | \(\overline{O}(\sqrt{\gamma(1 + \Delta)}(A_{(1)})^2 + \varepsilon \|A - A_k\|_F^2)\) | \(\frac{1}{4\pi^k} \|A - A_k\|^2_F\) |
| r-BKIFD (this work) | \(\max \overline{O}(\epsilon(1 + \Delta)(A_{(1)})^2 + \varepsilon \|A - A_k\|_F^2)\) | \(8k\varepsilon(1 + \delta)\log \left( \frac{2d}{\eta} \right) \|A - A_k\|_F^2 + \frac{\|A - A_k\|_F^2}{\ell - k} \|A - A_k\|_F^2\) |

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
matrix $P$. Thus, if the submatrix $A_i \in \mathbb{R}^{b \times d}$ is chosen as the Gaussian random matrix, it needs the time of $O(bdmq)$ to construct the Krylov space $K$, while, if $A_i$ is chosen as CountSketch matrix, it only needs $O(nnz(A_i)q)$ time to get $K$. Then, the QR decomposition is further needed to obtain $Q$, which requires the time of $O(bmq^2)$. The truncated SVD could cost the time of $O((mq)^2\ell)$ to get $\hat{U}_\ell$. After that, the main cost lies in step 6 of performing SVD on $B$, which costs $O(d\ell^2)$. Summarizing all these calculations, the computational cost of each iteration is about $O(bdmq + bm^2q^2 + m^2q^2\ell + d\ell^2)$. Considering that we only should proceed $(n/s)$ rows, the total cost for GA-BKIFD is $O(nmq + nm^2q^2 + sm^2q^2\ell + sd\ell^2)$.
Further noting that $\sum \text{nnz}(A_i) = \text{nnz}(A)$, it is easy to conclude that the total cost for CS-BKIFD is $O(\text{nnz}(A)q + nm^2q^2 + sm^2q^2\ell + sd\ell^2)$. Therefore, CS-BKIFD would be a better choice for large-scale sparse datasets.

V. EXPERIMENTS

In this section, the proposed r-BKIFD is compared with three popular algorithms, namely, FD [22], SFD [48], and SpFD10 [20], through a series of synthetic and real datasets.
data experiments. All such algorithms are implemented in MATLAB 2018a on a 56-core CPU (2.20 GHz) with 128 GB of RAM, and we run each algorithm 30 times and take the average result. Their detailed information is listed as follows.

1) **FD:** Algorithm 1.
2) **SFD [48]:** It is a randomized FD algorithm, which bases on the Gaussian random matrix and power iteration.
3) **SpFD10:** It is a randomized FD algorithm utilizing sparse subspace embedding method, which chooses \( q \approx 0 \) to balance the precision and computational time of the SpFDq algorithm proposed by Teng and Chu [20].
4) **GA-BKIFD:** Following Algorithm 4, here, we choose to use the standard Gaussian random matrix in the block Krylov iteration step.
5) **CS-BKIFD:** Following Algorithm 4, here, we choose to use the standard CountSketch matrix in the block Krylov iteration step.

For measuring the accuracy of these computing algorithms, we consider both the covariance and projection errors. The covariance error is defined as \( \| A^T A - B^T B \|_2 / \| A \|_F^2 \), which measures the difference in singular values. The projection error is defined by projecting \( A \) onto the top-\( k \) singular vectors of \( B \), i.e., \( \| A - \pi_B^k(A) \|_F / \| A - A_k \|_F^k \). Moreover, we also measure the computational cost by changing the sketch size \( \ell \).

### A. Synthetic Data Experiments

We consider both dense and sparse synthetic data. The generation of dense data follows the setting in [22], that is, we generate \( A = S D U + N / \zeta \in \mathbb{R}^{n \times d} \), where \( S \in \mathbb{R}^{n \times k} \) is the coefficients matrix with \( S_{ij} \sim \mathcal{N}(0, 1) \), \( D \in \mathbb{R}^{k \times k} \) is a diagonal matrix with \( D_{ii} = 1 - (i - 1)/k \) that gives linearly diminishing singular values, \( U \in \mathbb{R}^{k \times d} \) is the row space matrix with \( U U^T = I_k \), and \( N \in \mathbb{R}^{n \times d} \) is a noise matrix with \( N_{ij} \sim \mathcal{N}(0, 1) \). The parameter \( \zeta \) determines whether the noise can dominate the signal. The generation of the sparse data is by random sampling; each row contains roughly 0.1% nonzeros chosen uniformly from \([0, 1]\), with the remaining entries as 0.

In our method, we fix the batch size to the dimension \( d \) for dense data and \( 2d \) for sparse data, and the iteration number \( q = 2 \). Empirically, setting \( m = \ell + p \) is sufficient to estimate an accurate subspace, where \( p \) is a small nonnegative integer. For other computing methods, we follow the parameter settings in their original papers. In our experiments, we consider \( n = 60,000 \), \( d = 5000 \), \( k \in \{50, 200, 500\} \), and \( \zeta = 10 \), and vary the sketch size \( \ell \) to measure the performance. We present their average results on 30 independent runs in Fig. 3.

Several observations can be easily obtained from Fig. 3. First, all compared methods have a tendency to reduce error as the growth of sketch size \( \ell \) in both projection and covariance errors. Second, the proposed GA-BKIFD and CS-BKIFD algorithms obtain much lower error bounds compared with two randomized FDs, as well as FD in most cases. SFD has a close performance to ours but has a much higher computational cost for the dense matrix cases. For FD and SpFD10 methods, they both obtain a worse estimation, especially for the sparse matrix. Third, all randomized methods generally spend less running time than FD. For the dense matrix cases, though the SpFD10 method is the fastest among all such methods, it fails to find an accurate estimation. In addition, once the input matrix is extremely sparse, by utilizing the structure of CountSketch matrix, CS-BKIFD only takes at most one-sixth of the time compared with GA-BKIFD and has the least running time among all methods. All these verify the effectiveness and the efficiency of the proposed Algorithm 4 and, thus, further support the performance guarantees provided by Theorems 1 and 2.

### B. Real Data Experiments

In this section, we evaluate the performance by considering ten real-world datasets: “w8a,” “CIFAR-10,” “sido0,” “MovieLens-10M,” “MovieLens-20M,” “Protein,” “MNIST,” “rcv1-small,” “Newsgroups,” and “amazon7-small.” The sparsity in the datasets varies from 0.017% to 99.76%, and the detailed information is listed in Table II.

For the three datasets with small feature dimensions, i.e., w8a, Protein, and MNIST, we utilize a larger batch size of 1000 in our method, and for the other datasets, we set the batch size as in the synthetic data experiments. For other competing methods, we still keep the settings of the original papers. According to the results shown in Figs. 3–5, we observe that our algorithm outperforms other ones in terms of accuracy for nearly all the datasets. For the last three extremely sparse datasets, our algorithm still achieves the lowest covariance error among nearly all these methods and a comparable projection error with the best one. The performance of SpFD10 is unstable; even with a larger sketch size, the covariance error may be higher. This may be because the sparse subspace embedding in SpFD10 fails to capture the important information underlying the input matrix. In addition, SFD attains comparable accuracy to ours, while it has a higher computational cost. All these results, together with the previous results on synthetic data, demonstrate that the
proposed r-BKIFD provides great improvement over other randomized and traditional FD algorithms, both in terms of computational efficiency and accuracy.

VI. CONCLUSION

In this article, we proposed a novel algorithm named r-BKIFD to alleviate the inaccuracy issue in the randomized FD variants for low-rank approximation. Different from the existing ones embedding random projection technique directly, which may lead to the loss of some important information of the original matrix during the projection process, the basic idea of r-BKIFD is to incorporate the block Krylov iteration technique that could capture a more accurate projection subspace into the randomized FD. In the new algorithmic framework, we consider two types of random matrices, i.e., Gaussian and CountSketch matrices. Our rigorous theoretical analysis reveals that the proposed r-BKIFD gives a comparable error bound with traditional FD. The extensive experiments on both synthetic and real data further demonstrate that r-BKIFD outperforms traditional FD and its randomized variants in most cases in terms of computational efficiency and accuracy.

Noting that some real-world data are mostly in the form of multi-dimensional arrays (or say, tensors), such as videos and hyperspectral images, and the random projection technique has been extended to deal with large-scale low-rank tensor problems, including low-rank tensor completion [66] and low multilinear rank approximation of big tensors [67]. Thus, how to combine the proposed procedure with the existing studies to get a more efficient low-rank approximation of big tensor data is the focus of our future study.

APPENDIX

Proofs

To prove Theorem 1, we shall list the following auxiliary property for Gaussian random matrix, regarding the error bound of extreme singular values.

**Lemma 6 ([68], Corollary 5.35):** Let $A$ be an $N \times n$ matrix whose entries are independent standard normal random variables. Then, for every $\epsilon \geq 0$, with probability at least $1 - 2 \exp(-\epsilon^2/2)$, one has
\[
\sqrt{N} - \sqrt{n} - \epsilon \leq s_{\min}(A) \leq s_{\max}(A) \leq \sqrt{N} + \sqrt{n} + \epsilon.
\]

**Proof of Theorem 1:** By the triangle inequality, we have
\[
\|A^T A - B^T B\|_2 \leq \|A^T A - P^T P\|_2 + \|P^T P - B^T B\|_2.
\]

We first bound $\|A^T A - P^T P\|_2$. As mentioned earlier, the input matrix $A$ exists as a streaming fashion with each batch arriving in order, that is, $A = [A(i); \cdots; A(i_j)] \in \mathbb{R}^{n \times d}$, where $A(i) \in \mathbb{R}^{n \times d}$. Then, we can rewrite $\|A^T A - P^T P\|_2$ as
\[
\|A^T A - P^T P\|_2 = \sqrt{\sum_{i=1}^{T} \|A(i)A(i) - P(i)P(i)\|_2^2}
\]

where $P = [P(1); \cdots; P(j)]$ with each $P(j)$ obtained by the BKI compression on $A(i)$.

Let $M(i) = A(i) - P(i)P(i)$. By noting that $\{M(i)\}_{i=1}^{T}$ are independent random matrices, and $\mathbb{E}[M(i) - \mathbb{E}[M(i)]] = 0$; we, thus, can apply the matrix Bernstein inequality previously given in Lemma 3 to bound $\mathbb{E}[M(i)]$. To this end, the necessary step is to calculate the covariance error bound $R$ and the variance parameter $\sigma^2$. As for $R$, combining the triangle inequality with Jensen’s inequality, we have
\[
\mathbb{E}[\|M(i) - \mathbb{E}[M(i)]\|_2^2] \leq \|\mathbb{E}[M(i)]\|_2^2 + \mathbb{E}[\|M(i)\|_2^2] 
\]

Due to $M(i) = A(i) - P(i)P(i)$ and $P(i) = Z(i)^T A(i)$, we can reformulate $M(i)$ as $A(i)(I - Z(i)Z(i)^T)A(i)$. By the fact that $I - Z(i)Z(i)^T$ is a projection, i.e.,
\[
I - Z(i)Z(i)^T = (I - Z(i)Z(i)^T)(I - Z(i)Z(i)^T)
\]

we have
\[
\|M(i)\|_2 = \|A(i)(I - Z(i)Z(i)^T)(I - Z(i)Z(i)^T)A(i)\|_2
\]

\[
= \|A(i) - Z(i)Z(i)^T A(i)\|_2 
\]

\[
\leq \left\{ \|\phi(I_{\ell \perp})\|_2 \|V(i)\|_{\ell \perp} \left(\|V(i)\|_{\ell \perp} X(V(i))^\dagger\right)_F \
\right. 
\]

\[
+ \|A(i) - [A(i)]_\ell\|_2^2 \right\}^2 
\]

where the inequality (6) follows from Lemma 1. Since the rows of $[V(i)]_\ell X$ are orthonormal, the entries of $[V(i)]_\ell X \in \mathbb{R}^{\ell \times m}$ are independent Gaussians. We use the same probability of success for each $[V(i)]_\ell X$, then, according to Lemma 6, with probability at least $1 - 2 \exp(-\epsilon^2/2)$, we have
\[
\sigma_{\ell}(\|V(i)\|_F X) \geq \sqrt{m} - \sqrt{\ell} - \epsilon, \text{ and } \|X\|_2 \leq \sqrt{d} + \sqrt{m} + \epsilon.
\]

By the submultiplicativity property of the Frobenius norm, we then further get that
\[
\|V(i)_\ell X(V(i))^\dagger\|_F \leq \left\{ \|V(i)_\ell X\|_2 \right\} \leq \left( \frac{\sqrt{d} - \ell}{\sqrt{m} - \sqrt{\ell} - \epsilon} \right)^2
\]

Combining Lemma 2 with inequalities (6) and (7) can get that
\[
\|M(i)\|_2 \leq \|A(i) - Z(i)Z(i)^T A(i)\|_2^2
\]

\[
\leq \left( 1 + \frac{4}{2\delta \ell + 1} \min(\sqrt{d}, 1) \right) \frac{\sqrt{d} - \ell}{\sqrt{m} - \sqrt{\ell} - \epsilon}
\]

\[
\times \|A(i) - [A(i)]_\ell\|_2^2
\]

\[
= (1 + \delta) \|A(i) - [A(i)]_\ell\|_2^2
\]

\[
\leq \left( 1 + \delta \right) \|A(i) - [A(i)]_\ell\|_2^2
\]

\[
\leq \left( 1 + \delta \right) \|A(i) - [A(i)]_\ell\|_2^2
\]

\[
\|A(i) - [A(i)]_\ell\|_2^2 \leq \sigma_{\ell+1}(\|A(i)\|) + \lambda_{\ell+1}(A(i)^T A(i))
\]

\[
\|A(i) - [A(i)]_\ell\|_2^2 \leq \sigma_{\ell+1}(\|A(i)\|) + \lambda_{\ell+1}(A(i)^T A(i))
\]

\[
(9)
\]
According to Lemma 4 and the fact that
\[ R_{\sum_{i=1}^{s} A_{i}^{*} A_{i}}(x) \geq \max_{\ell \in [s]} R_{A_{i}^{*} A_{i}}(x) \]
we can further upper bound (9) as
\[ \lambda_{t+1}(A_{(g)}^{T} A_{(g)}) \leq \lambda_{t+1}(\sum_{i=1}^{s} A_{i}^{T} A_{i}) = \lambda_{t+1}(A^{T} A) = \sigma^{2}_{t+1}(A) = \|A - A_{\ell}\|^{2}_{2}. \]
Hence, it then follows the inequality (5) that
\[ \|M_{(i)} - E[M_{(i)}]\|_{2} \leq 2(1 + \delta)\|A - A_{\ell}\|^{2}_{2}. \]
This completes the calculation of $R$.

Next, we shall focus on calculating $\sigma^{2}$. Due to the fact that $M_{(i)} - E[M_{(i)}]$ is a symmetric matrix, we have $\sigma^{2} = E[(M_{(i)} - E[M_{(i)}])^{2}]\|_{2}$. Thus, by the triangle inequality, we further have $\sigma^{2} \leq \sum_{i=1}^{s} \|E[(M_{(i)} - E[M_{(i)}])^{2}]\|_{2}$. Expanding the square gives
\[ E[(M_{(i)} - E[M_{(i)}])^{2}] = E[M_{(i)}^{2}] - E[M_{(i)}]^{2} \leq E[M_{(i)}^{2}]. \]

Consequently, $\|E[M_{(i)}^{2}]\|_{2} \leq \|E[M_{(i)}^{2}]\|_{2}$.

We then bound $E[M_{(i)}^{2}]$ as
\[ E[M_{(i)}^{2}] = E\left[(A_{(i)}^{T} A_{(i)} - P_{(i)}^{T} P_{(i)})^{2}\right] = E\left[\left(\sum_{i=1}^{d} (A_{(i)}^{T} - Z_{(i)} Z_{(i)}^{T} A_{(i)})\right)^{2}\right] \leq \sum_{i=1}^{d} E[(A_{(i)}^{T} - Z_{(i)} Z_{(i)}^{T} A_{(i)})^{2}] \leq (1 + \delta)\|A - A_{\ell}\|^{2}_{2}. \]

where the inequality (10) follows the fact that, for any $y \in \mathbb{R}^{d}$
\[ y^{T}\left((A_{(i)} - Z_{(i)} Z_{(i)}^{T} A_{(i)})^{T} (A_{(i)} - Z_{(i)} Z_{(i)}^{T} A_{(i)})\right)y \leq \|y^{T} (A_{(i)} - Z_{(i)} Z_{(i)}^{T} A_{(i)})^{T} (A_{(i)} - Z_{(i)} Z_{(i)}^{T} A_{(i)})\|_{2} \leq \|y^{T} (A_{(i)} - Z_{(i)} Z_{(i)}^{T} A_{(i)})\|_{2} \leq \|y^{T} (A_{(i)} - Z_{(i)} Z_{(i)}^{T} A_{(i)})\|_{2}. \]

and the inequality (11) holds by (8). Therefore,
\[ \sigma^{2} \leq \sum_{i=1}^{s} \|E[M_{(i)}^{2}]\|_{2} \leq \sum_{i=1}^{s} (1 + \delta)\|A - A_{\ell}\|^{2}_{2} \|E[M_{(i)}^{2}]\|_{2}. \]

By Jensen’s inequality, we further have
\[ \sigma^{2} \leq \sum_{i=1}^{s} (1 + \delta)\|A - A_{\ell}\|^{2}_{2} \|E[M_{(i)}^{2}]\|_{2} \leq \sum_{i=1}^{s} (1 + \delta)\|A - A_{\ell}\|^{2}_{2} = (1 + \delta)^{2} \|A - A_{\ell}\|^{2}_{2}. \]

It then follows Lemma 3 that
\[ P \left( \sum_{i=1}^{s} \left| M_{(i)} - E[M_{(i)}] \right| \geq t \right) \leq 2d \exp \left( -\frac{t^{2}}{2 \sigma^{2} + Rt/3} \right) \]
for all $t \geq 0$. We denote the right-hand side of (13) by $\eta$; then,
\[ t = \log \left( \frac{2d}{\eta} \right) \left( \frac{R}{3} + \sqrt{\frac{R}{3} + \frac{2\sigma^{2}}{\log(2d/\eta)}} \right) \leq \log \left( \frac{2d}{\eta} \right) \frac{2R}{3} + \sqrt{2\sigma^{2} \log \left( \frac{2d}{\eta} \right)}. \]

Plugging $R$ and $\sigma^{2}$ into (14) can get that
\[ t \leq \left( \log \left( \frac{2d}{\eta} \right) \frac{(1 + \delta)^{2}}{3} + \sqrt{2(1 + \delta)^{2} \log \left( \frac{2d}{\eta} \right)} \right) \|A - A_{\ell}\|^{2}_{2}. \]

That is to say, with probability at least $1 - \eta$, we have
\[ \sum_{i=1}^{s} \left| M_{(i)} - E[M_{(i)}] \right| \leq \log \left( \frac{2d}{\eta} \right) \frac{(1 + \delta)^{2}}{3} + \sqrt{2(1 + \delta)^{2} \log \left( \frac{2d}{\eta} \right)} \|A - A_{\ell}\|^{2}_{2}. \]

By triangle and Jensen’s inequalities, we have
\[ \sum_{i=1}^{s} \left| M_{(i)} \right| \leq \sum_{i=1}^{s} \left| E[M_{(i)}] \right| + \sum_{i=1}^{s} \left| M_{(i)} - E[M_{(i)}] \right| \leq \sum_{i=1}^{s} \left| E[M_{(i)}] \right| + \sum_{i=1}^{s} \left| M_{(i)} - E[M_{(i)}] \right|. \]

Therefore, with probability at least $1 - \eta - 2s \exp(-\epsilon^{2}/2)$, we have
\[ \|A^{T} A - P^{T} P\|_{2} = \sum_{i=1}^{s} \left| M_{(i)} \right| \leq \sum_{i=1}^{s} \left| E[M_{(i)}] \right| + \sum_{i=1}^{s} \left| M_{(i)} - E[M_{(i)}] \right| \leq \left( s(1 + \delta) + \log \left( \frac{2d}{\eta} \right) \frac{(1 + \delta)^{2}}{3} \right) \|A - A_{\ell}\|^{2}_{2}. \]

Now, we bound $\|P^{T} P - B^{T} B\|_{2}$. Based on the property of FD stated in [22], that is, $\|P^{T} P - B^{T} B\|_{2} \leq (\|P - P_{k}\|_{F}/\ell-k)$, where $P_{k}$ is the rank-$k$ approximation of $P$, let $M$ be the projection matrix onto the subspace spanned by the top-$k$ right singular vectors of $A$. Noting that $M$ is of rank $k$, and then,
\( \| P - P_k \|_F^2 / \ell - k \) \leq (\| P - PM \|_F^2 / \ell - k). \) Thus,

\[
\| P^T P - B^T B \|_2 \leq \frac{\| P - PM \|_F^2}{\ell - k} = \sum_{i=1}^s \frac{\| P(i) - P(i)M \|_F^2}{\ell - k} = \sum_{i=1}^s \frac{\| Z_{(i)}^T A(i) - Z_{(i)}^T A(i)M \|_F^2}{\ell - k} \leq \sum_{i=1}^s \frac{\| Z_{(i)}^T A(i) - A(i)M \|_F^2}{\ell - k} = \frac{\| A - AM \|_F^2}{\ell - k} = \frac{\| A - A_k \|_F^2}{\ell - k} \tag{15}
\]

where (15) holds because of \( Z_{(i)} \) is orthonormal, and (16) holds by the definition of \( M \).

From the aforementioned analysis, we, finally, have

\[
\| A^T A - B^T B \|_2 \leq \| A^T A - P^T P \|_2 + \| P^T P - B^T B \|_2 \leq \left( s(1 + \delta) + \log \frac{2d}{\eta} \right) \frac{4(1 + \delta)}{3} + \sqrt{2s(1 + \delta)^2 \log \frac{2d}{\eta}} \times \| A - A_k \|_F^2 + \frac{\| A - A_k \|_F^2}{\ell - k}.
\]

This arrives at the conclusion of Theorem 1. \( \square \)

Before proving Theorem 2, we need to list the following auxiliary property for the CountSketch matrix.

**Lemma 7 ([20], Lemma 1):** Given a matrix \( U \in \mathbb{R}^{n \times k} \) with orthonormal columns \( (k \leq n) \), for any \( \varepsilon, \rho \in (0, 1) \), let the CountSketch matrix \( S \in \mathbb{R}^{m \times n} \) be defined as with

\[
m \geq \frac{k^2 + k}{\varepsilon^2 \rho^2}.
\]

Let \( P \) be an \( n \times n \) random permutation matrix (i.e., for each \( e_j \) with \( j \in [n] \), \( P[Pe_j = e_i] = 1/n \) for \( i \in [n] \)), and then, with probability at least \( 1 - \rho \)

\[
\| U^T (SP)^T SP - I \|_2 \leq \varepsilon.
\]

This lemma reveals that CountSketch is a \( 1 \pm \varepsilon \) \( \ell_2 \)-subspace embedding for \( U \). It is noted that the permutation matrix \( P \) can be a unit matrix in special cases.

**Proof of Theorem 2:** The proof of Theorem 2 is similar to the procedure of the proof of Theorem 1 with a little bit of differences. This is mainly because the various constructions of \( X \) can result in the different error bounds of \( \| A^T A - P^T P \|_2 \). Let the CountSketch matrix be \( X = D\Phi T \in \mathbb{R}^{d \times m} \). Noting that \( \sigma_k((V(i)_T X) X) = \sigma_k((V(i)_T D\Phi T) ) = \sigma_k(\Phi D[V(i)_T] ) \), and according to Lemma 7, we can get that, for any \( \varepsilon, \rho \in (0, 1) \), if \( m \geq (\ell^2 + \ell/\varepsilon^2 \rho) \), the inequation \( \| (V(i)_T XX^T V(i)_T) - I \|_2 \leq \varepsilon \) holds with probability at least \( 1 - \rho \), that is,

\[
\sigma_k((V(i)_T X) \geq \sqrt{1 - \varepsilon}. \tag{17}
\]

By the submultiplicativity property of the Frobenius norm, we have

\[
\| (V(i)_T X) F \|_2 \leq \| (V(i)_T X) P \|_2 \| \Phi T \|_F = \sqrt{d(d - \ell)}.
\]

It then follows by inequalities (17) and (18) that

\[
\| (V(i)_T X) F \|_2 \leq \| (V(i)_T X) F \|_2 \| \Phi T \|_F \leq \sqrt{d(d - \ell)} \leq \frac{1}{1 - \varepsilon}.
\]

Next, we first derive the upper bound of \( \| M_{(i)} \|_2 \) by combining inequalities (6) and (19) and then calculate \( R \) and \( \sigma^2 \) using the same procedure as Theorem 1. Thus, with probability at least \( 1 - sp - \eta \), we can get that

\[
\| A^T A - P^T P \|_2 \leq \left( s(1 + \delta) + \log \frac{2d}{\eta} \right) \frac{4(1 + \delta)}{3} + \sqrt{2s(1 + \delta)^2 \log \frac{2d}{\eta}} \times \| A - A_k \|_F^2 + \frac{\| A - A_k \|_F^2}{\ell - k}.
\]

This completes the proof of Theorem 2. \( \square \)

**References**

[1] M. Turk and A. Pentland, “Eigenfaces for recognition,” J. Cognit. Neurosci., vol. 3, no. 1, pp. 71–86, 1991.

[2] P. Parker, P. J. Wolfe, and V. Tarokh, “A signal processing application of randomized low-rank approximations,” in Proc. IEEE ESP 13th Workshop Stat. Signal Process., Jul. 2005, pp. 345–350.

[3] P. Drineas, I. Kerenidis, and P. Raghavan, “Competitive recommendation systems,” in Proc. 34th Annu. ACM Symp. Theory Comput. (STOC), 2002, pp. 82–90.

[4] T. Ren, X. Jia, W. Li, and S. Zhao, “Label distribution learning with label correlations via low-rank approximation,” in Proc. 28th Int. Joint Conf. Artif. Intell., Aug. 2019, pp. 3325–3331.

[5] K. P. Murphy, Machine Learning: A Probabilistic Perspective. Cambridge, MA, USA: MIT Press, 2012.

[6] S. Wold, K. Esbensen, and P. Geladi, “Principal component analysis,” Chemometrics Intell. Lab. Syst., vol. 2, no. 4, pp. 37–52, 1987.

[7] D. Skillicorn, Understanding Complex Datasets: Data Mining With Matrix Decompositions. Boca Raton, FL, USA: CRC Press, 2007.

[8] L. Zhang and Y. Guan, “Detecting click fraud in pay-per-click streams of online advertising networks,” in Proc. 28th Int. Conf. Distrib. Comput. Syst., Jun. 2008, pp. 77–84.

[9] P. Bonnet, J. Gehrke, and P. Seshadri, “Towards sensor database systems,” in Proc. Int. Conf. Mobile Data Manag. Berlin, Germany: Springer, 2001, pp. 3–14.

[10] A. C. Gilbert, Y. Kotidis, S. Muthukrishnan, and M. Strauss, “Quicksand: Quick summary and analysis of network data,” DIMACS, Tech. Rep. 2001-43, 2001.
[64] J.-Y. Wang, “Application of support vector machines in bioinformatics,” M.S. thesis, Dept. Comput. Sci. Inf. Eng., Nat. Taiwan Univ., Taipei, Taiwan, 2002.

[65] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[66] Y. Feng and G. Zhou, “Orthogonal random projection for tensor completion,” IET Comput. Vis., vol. 14, no. 5, pp. 233–240, Aug. 2020.

[67] Y. Qiu, G. Zhou, Y. Zhang, and A. Cichocki, “Canonical polyadic decomposition (CPD) of big tensors with low multilinear rank,” Multimedia Tools Appl., vol. 80, no. 15, pp. 22987–23007, Jun. 2021.

[68] R. Vershynin, “Introduction to the non-asymptotic analysis of random matrices,” 2010, arXiv:1011.3027.

Chenhao Wang received the B.Sc. and M.Sc. degrees from Xi’an Jiaotong University, Xi’an, China, in 2019 and 2022, respectively. He is currently pursuing the Ph.D. degree with the School of Data Science, The Chinese University of Hong Kong, Shenzhen, China.

His current research interests include machine learning, revenue management, and online learning.

Qianxin Yi received the M.S. degree majoring in statistics from Xi’an Jiaotong University, Xi’an, China, in 2021, where she is currently pursuing the Ph.D. degree with the School of Management.

Her current research interests include tensor optimization algorithms and reinforcement learning.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.