Superlinear Convergence of Randomized Block Lanczos Algorithm

Qiaochu Yuan  
Department of Mathematics  
UC Berkeley  
Berkeley, CA, USA  
qyuan@berkeley.edu

Ming Gu  
Department of Mathematics  
UC Berkeley  
Berkeley, CA, USA  
mgu@math.berkeley.edu

Bo Li  
Department of Mathematics  
UC Berkeley  
Berkeley, CA, USA  
bo_li@berkeley.edu

Abstract—The low rank approximation of matrices is a crucial component in many data mining applications today. A competitive algorithm for this class of problems is the randomized block Lanczos algorithm - an amalgamation of the traditional block Lanczos algorithm with a randomized starting matrix. While empirically this algorithm performs quite well, there has been scant new theoretical results on its convergence behavior and approximation accuracy, and past results have been restricted to certain parameter settings. In this paper, we present a unified singular value convergence analysis for this algorithm, for all valid choices of the block size parameter. We present novel results on the rate of singular value convergence and show that under certain spectrum regimes, the convergence is superlinear. Additionally, we provide results from numerical experiments that validate our analysis.

Index Terms—low-rank approximation, randomized block Lanczos, block size, singular values.

I. INTRODUCTION

The low rank approximation of matrices is a crucial component in many data mining applications today. In addition to functioning as a stand alone technique for dimensionality reduction [1], denoising [2], signal processing [3], data compression [4], and more, it has also been incorporated into more complex algorithms as a computational subroutine [5], [6]. As part of large scale modern data processing, low rank approximations help to reveal important structural information in the raw data and to transform the data into forms that are more efficient for computation, transmission, and storage.

The singular value decomposition (SVD) is a matrix factorization of both theoretical and practical importance, and it has a number of useful properties related to matrix nearness and rank. In particular, it is used to identify nearby matrices of lower rank, and, leaving aside the question of computational complexity, it is known that the rank-$k$ truncated SVD is the “gold standard” for approximating a matrix by another matrix of rank at most $k$ [7].

While procedures for computing the exact rank-$k$ truncated SVD have existed since the 1960s [8], the computational cost of these algorithms are prohibitive at the scale of many of today’s datasets. The recent applications of low rank matrix approximation techniques to big-data problems differ in both the computation efficiency requirement and the accuracy requirement of the algorithms. Firstly, we are increasingly leaving behind the era of moderately sized matrices and entering an age of web-scale datasets and big-data applications. The matrices arising from such are often extraordinarily large, exceeding the order of $10^6$ in one or both of the dimensions [9]–[11], and have much higher computational efficiency demands on the algorithms. Secondly, while the truncated SVD may be the final desired object for previous scientific computing questions, for big-data applications, it is usually an intermediate representation for the overall classification or regression task. Empirically, the final accuracy of the task only weakly depends on the accuracy of the matrix approximation [12]. Thus, while previous variants of truncated SVD algorithms focused on computing up to full double precision, newer iterations of these algorithms aimed at big-data applications can comfortably get by with only 2-3 digits of accuracy.

These considerations have led to the development of randomized variants of traditional SVD algorithms suited to large, sparse matrices, in particular randomized subspace iteration (RSI) and randomized block Lanczos (RBL) [13]–[16]. By applying either a randomized sketching or projecting operation on the original matrix, these algorithms balance reducing computational complexity with producing an acceptably accurate approximation. While empirically they have shown to be effective and have been widely adopted by popular software packages, e.g. [17], there has been scant new theoretical work on the convergence guarantees of the latter algorithm, the better performing but more complicated randomized block Lanczos algorithm.

In this paper, we present novel theoretical convergence results concerning the rate of singular value convergence for the RBL algorithm, along with numerical experiments supporting these results. Our analysis presents a unified singular value convergence theory for variants of the Block Lanczos algorithm, for all valid parameter choices of block size $b$. To our knowledge, all previous results in the literature are applicable only for the choice of $b \geq k$, the target rank. We present a generalized theorem, applicable to all block sizes $b$, which coincide asymptotically with previous results for the case $b \geq k$, while providing equally strong rates of convergence for the case $b < k$.

In Section II we present the randomized block Lanczos algorithm and discuss some previous convergence results for...
this algorithm. In Section [III] we dive into our main theoretical result and its derivation, followed by corollaries for special cases. In Section [IV] we investigate the behavior of this algorithm for different parameter settings and empirically verify the results of the previous section. Finally, we give concluding remarks in Section [V].

II. BACKGROUND

A. Preliminaries

Throughout this paper, our analysis assumes exact arithmetics.

We denote matrices by bold-faced uppercase letters, e.g. \( M \), entries of matrices by the plain-faced lowercase letter that the entry belongs to, e.g. \( m_{11} \), and block matrices by the bold-faced or script-faced uppercase letter that the submatrix belongs to subscripted by position, possibly with subscripts, e.g. \( M_{11}, M_{11} \) or \( M_{a \times b} \). Double numerical subscripts denote the position of the element or the submatrix, i.e. \( M_{11} \) and \( m_{11} \) are the topmost leftmost subblock or entry of \( M \) respectively. \( m \times n \) subscripts denote the dimensions of a submatrix, when such information is relevant, i.e. \( M_{a \times b} \) denote a subblock of \( M \) that has dimensions \( a \times b \). Constants are denoted by script-faced uppercase or lower-case letters, e.g. \( C \) or \( \alpha \), when it is asymptotically insignificant, i.e. constant with respect to the convergence parameter.

The SVD of a matrix \( A \) is defined as the factorization

\[
A = U \Sigma V^T
\]

where \( U = [u_1 \cdots u_n] \) and \( V = [v_1 \cdots v_n] \) are orthogonal matrices whose columns are the set of left and right singular vectors respectively, and \( \Sigma \) is a diagonal matrix whose entries \( \Sigma_{ii} = \sigma_i \) are the singular values ordered descendingly \( \sigma_1 \geq \cdots \geq \sigma_n \geq 0 \).

The rank-\( k \) truncated SVD of a matrix is defined as

\[
\operatorname{svd}_k(A) = U_k \Sigma_k V_k
\]

where \( U_k = [u_1 \cdots u_k] \) and \( V_k = [v_1 \cdots v_k] \) contain the first \( k \) left and right singular vectors respectively, and \( \Sigma_k = \text{diag}(\sigma_1, \cdots, \sigma_k) \).

The \( i \)th singular values of an arbitrary matrix \( M \) is denoted by \( \sigma_i(M) \), or simply \( \sigma_i \) when it is clear from context the matrix in question.

The \( p \)th degree Chebyshev polynomial is defined by the recurrence

\[
T_0(x) = 1 \\
T_1(x) = x \\
T_p(x) = 2pT_{p-1}(x) - T_{p-2}(x)
\]

Alternatively, they may be expressed as

\[
T_p(x) = \frac{1}{2} \left( (x + \sqrt{x^2 - 1})^p + (x - \sqrt{x^2 - 1})^{-p} \right)
\]

for \( |x| > 1 \), and estimated as

\[
T_p(1 + \epsilon) \approx \frac{1}{2} \left( 1 + \epsilon + \sqrt{2\epsilon} \right)^p
\]

for \( p \) large and \( \epsilon \) small.

B. The Algorithm

The randomized block Lanczos algorithm is a straightforward combination of the classical block Lanczos algorithm [13] with the added element of a randomized starting matrix \( V = A\Omega \).

The pseudocode for this algorithm is outlined in Algorithm [I] Of the parameters of the algorithm, \( k \) (target rank) is problem dependent, while \( b \) (block size), \( q \) (no. of iterations) are chosen by the user to control the quality and computational cost of the approximation. The algorithm requires the choices of \( b, q \) to satisfy \( qb \geq k \), to ensure that the Krylov subspace be at least \( k \) dimensional.

**Algorithm 1** randomized block Lanczos algorithm pseudocode

\[
\begin{align*}
A & \in \mathbb{R}^{m \times n} \\
\Omega & \in \mathbb{R}^{n \times b}, \text{ random Gaussian matrix} \\
\text{Input:} & \quad k \quad \text{, target rank} \\
& \quad b \quad \text{, block size} \\
& \quad q \quad \text{, number of Lanczos iterations} \\
\text{Output:} & \quad B_k \in \mathbb{R}^{m \times n} \quad \text{, a rank}-k \text{-approximation to } A \\
1: & \text{Form the block column Krylov subspace matrix} \\
& K = [A \Omega (AA^T) \Omega \cdots (AA^T)^{q-1} \Omega] \\
2: & \text{Compute an orthonormal basis } Q \text{ for the column span of} \\
& K, \text{ using e.g. } QR \leftarrow qr(K) \\
3: & \text{Project } A \text{ onto the Krylov subspace by computing} \\
& B = QQ^T A \\
4: & \text{Compute } k\text{-truncated SVD } B_k = \operatorname{svd}_k(B) = \operatorname{svd}_k(QQ^T A) = Q \cdot \operatorname{svd}_k(Q^T A) \\
5: & \text{Return } B_k.
\end{align*}
\]

We present the algorithm pseudocode in this form in order to highlight the mathematical ideas that are at the core of this algorithm. It is well known that a naive implementation of any Lanczos algorithm is plagued by loss of orthogonality of the Lanczos vectors due to roundoff errors [19]. A practical implementation of Algorithm [I] should involve, at the very least, a reorganization of the computation to use the three-step recurrence and bidiagonalization [20], and reorthogonalizations of the Lanczos vectors at each step using one of the numerous schemes that has been proposed [20]–[22].

C. Previous Work

Historically, the classical Lanczos algorithm was developed as an eigenvalue algorithm for symmetric matrices. Its convergence analysis focused on theorems concerning the approximation quality of the approximant’s eigenvalues as a function of \( k \), the target rank. The analysis relied heavily on the analysis of the \( k \)-dimensional Krylov subspace and the choice of the associated \( k \)-degree Chebyshev polynomial. Classical results in this line of inquiry include those by Kaniel [23], Paige [24], Underwood [25], Saad [26].

More recently, while there has been much work on the analysis of randomized algorithms, such efforts have been focused mostly on RBL’s simpler cousins, such as randomized power iteration or randomized subspace iteration [12], [15]. The exception is the results from [16]. To our knowledge, this
is one of the few works that provide convergence analysis for randomized block Lanczos and the first work that gives “gap”-independent theoretical bounds for this algorithm. The analysis found therein is restricted to the case for the block size, b, chosen at least the size of k, the desired target rank. Our theoretical analysis will give a more generally applicable convergence bound, encompassing the case for both 1 ≤ b < k and b ≥ k. In the latter case, our theoretical results will coincide with those in [16]. In the former case, we show that the rapid convergence of the algorithm for any block size b larger than the largest singular value cluster size is assured. We draw attention to this distinction in choosing the block size parameter b - in our numerical experiments, we show that generally smaller choices for b are favored.

Our current work is based partially on the analysis found in [12]. This work established aggressive multiplicative convergence bounds for the randomized subspace iteration algorithm, for both singular values and norms (Frobenius, spectral) matrix convergence. These bounds depend on both the singular value gap and the number of iterations taken by the algorithm - the former is a property of the matrix in question, and the latter is proportional to the computational complexity of the algorithm. The analysis presented in this work is linear algebraic in nature, drawing on deterministic matrix analysis, as well expectation bounds on randomized Gaussian matrices and their concentration of measure characteristics. Our current work employs similar methods, and achieves bounds of a similar form. While the details differ, core ideas, such as creating an artificial “gap” in the spectrum and choosing an opportune orthonormal basis for the analysis, are the same.

III. THEORETICAL RESULTS

A. Problem Statement

Given an arbitrary matrix \( A \in \mathbb{R}^{m \times n} \) and a target rank \( k \leq \text{rank}(A) \), the goal of a low-rank matrix approximation algorithm is to compute another matrix \( B_k \in \mathbb{R}^{m \times n} \) whose rank is at most \( k \).

There are many ways to ask and answer the question, “how good of an approximation is \( B_k \) to the original \( A \)?” In particular, for various low-rank approximation algorithms, the answer has been provided in terms of normed approximation error [12], [15], [16], [27], singular subspace error [28], [29], and singular value error [12], [26].

In this paper, we focus on the singular value error for the randomized block Lanczos algorithm. As \( B \) is an orthogonal projection of \( A \) in Alg. 1 by the Cauchy interlacing theorem for singular values, we immediately have the upper bound

\[
\sigma_j \geq \sigma_j(B_k)
\]

for \( j = 1, \cdots, k \).

The optimal lower bound is achieved, of course, by the rank-\( k \) truncated SVD of \( A \), giving the tight inequality

\[
\sigma_j \geq \sigma_j(\text{svd}_k(A)) \geq \sigma_j
\]

for \( j = 1, \cdots, k \).

We will to show that the randomized block Lanczos algorithm provides competitive accuracy, and produces singular value estimates at least some fraction of the optimum.

\[
\sigma_j \geq \sigma_j(B_k) \geq \frac{\sigma_j}{\sqrt{1 + \{\text{some convergence factor}\}}}
\]

for \{some convergence factor\} → 0.

B. Key Results

Our convergence analysis will show that if the randomized block Lanczos algorithm converges, then the \( k \) desired singular values of the approximation \( B_k \) converges to the corresponding true singular values of \( A \) exponentially in the number of iteration \( q \). Moreover, convergence occurs as long as the block size \( b \) is chosen to be larger than the maximum cluster size for the \( k \) relevant singular values.

We present our main results here and delay their proofs to Subsection III-D. Our main theorem is as follows.

**Theorem III.1.** Let \( B_k \) be the matrix returned by Alg. 1. Assume that \( \Omega \) is chosen such that the two conditions in Remark III.1 hold. For any choices of \( r, s \geq 0 \), and any parameter choice \( b, q \) satisfying \( k + r = (q - p)b \geq k \), for \( j = 1, \cdots, k \),

\[
\sigma_j \geq \sigma_j(B_k) \geq \frac{\sigma_j}{\sqrt{1 + \text{C}^2 T_{2p+1}^{-2} \left( 1 + 2 \cdot \frac{\sigma_j - \sigma_{j+r+1}}{\sigma_j + \sigma_{j+r+1}} \right)}}
\]

where \( \text{C} \) is a constant that is independent of \( q \).

This inequality shows that for all valid choices of parameters \( b, q \), the convergence of the approximate singular values are governed by the growth of the Chebyshev polynomial term

\[
T_{2p+1} \left( 1 + 2 \cdot \frac{\sigma_j - \sigma_{j+r+1}}{\sigma_j + \sigma_{j+r+1}} \right)
\]

, with the bound holding across all choices of the analysis parameters \( s, r \).

Theorem III.1 admits the following corollaries about two special choices for the block size parameter \( b \), where the constants in each case can be expressed in an algebraically closed form.

**Corollary III.2** (Special case: \( b = 1 \)). For any choices of \( r, s \geq 0 \) satisfying \( k + r = (q - p) \geq k \), for \( j = 1, \cdots, k \),

\[
\sigma_j \geq \sigma_j(B_k) \geq \frac{\sigma_j}{\sqrt{1 + \text{C}_{b=1} T_{2p+1}^{-2} \left( 1 + 2 \cdot \frac{\sigma_j - \sigma_{j+r+1}}{\sigma_j + \sigma_{j+r+1}} \right)}}
\]

where

\[
\text{C}_{b=1} = \left( \max_{1 \leq i \leq k} \frac{\omega_i}{\omega_s} \right)^2 \cdot \left( \sum_{s=1}^{j} \sum_{r=j+1}^{n} \prod_{i=1}^{r-1} \left( \frac{\omega_i^2 - \omega_s^2}{\omega_s^2 - \omega_i^2} \right)^2 \right)
\]

is a constant independent of \( q \).
Corollary III.3 (Special case: \( b \geq k + r \)). For any choices of \( r, s \geq 0 \), for \( j = 1, \cdots, k \),

\[
\sigma_j \geq \sigma_j (B_k) \geq \sigma_{j+s} \sqrt{1 + C_{b \geq k+r}^2 T_{2q+1}^{-2} \left( 1 + 2 \cdot \frac{\sigma_j - \sigma_{j+s+r+1}}{\sigma_{j+s+r+1}} \right)}
\]  

(15)

where

\[
C_{b \geq k+r} = \left\| \Omega_{11} \right\|_2 \left\| \Omega_{11}^{-1} \right\|_2
\]  

(16)

is a constant independent of both \( q \), the iteration parameter, and \( \Sigma \), the spectrum of \( A \).

Choosing optimally the analysis parameters \( r, s \), we arrive at a result coinciding asymptotically with the conclusions reached in [16].

Theorem III.4. Let \( B_k \) be the matrix returned by running Alg. 7 with the block size \( b = k \). Assume \( \Omega \) is chosen such that \( \Omega_{11} \) is nonsingular. Then, for \( j = 1, \cdots, k \),

\[
\sigma_j \geq \sigma_j (B_k) \geq \sigma_j e^{-\frac{\log(\log(\log(k+1)))^2}{(4q+2)^2}}
\]  

(17)

where \( A = 2C_{b \geq k+r} \) is a constant independent of \( q \).

Finally, from Theorem III.1 we may derive the following result, which states that for certain matrices with singular spectrum rapidly decaying to 0, the RBL algorithm converges superlinearly.

Theorem III.5. Assume the singular value spectrum of \( A \) decays such that \( \sigma_i \to 0 \). Let \( B_k \) be the rank \( k \) approximation of \( A \) returned by Alg. 7. Assume additionally that the hypothesis and notation of Theorem III.1 hold. Then

\[
\sigma_j (B_k) \to \sigma_j
\]  

(18)

superlinearly in \( q \), the number of iterations.

This theorem validates long observed empirical behaviors of block Lanczos algorithms. In Section IV we show two examples of typical data matrices with spectra that fall under this regime, and the expected superlinear convergence behavior.

C. Intuition

Our analysis makes use of the following three ideas:

1) the growth behavior of Chebyshev polynomials, a traditional ingredient in the analysis of Lanczos iteration methods, (Fig. 1)
2) the choice of a clever orthonormal basis for the Krylov subspace, an idea adapted from [12],
3) the creation of a spectrum “gap”, by separating the spectrum of \( A \) into those singular values that are “close” to \( \sigma_k \), and those that are sufficiently smaller in magnitude, using auxiliary analysis parameters \( r, s \). (Fig. 2)

D. Analysis

We are interested in the column span of the Krylov subspace matrix \( K \). Let the singular value decomposition of \( A \) be denoted as \( A = U \Sigma V^T \). Then, we may write

\[
K = \begin{bmatrix} A \Omega & (AA^T) \Omega & \cdots & (AA^T)^s \Omega \end{bmatrix}
\]

\[
= \begin{bmatrix} U \Sigma V^T \Omega & U \Sigma^{2+1} V^T \Omega & \cdots & U \Sigma^{2q+1} V^T \Omega \end{bmatrix}
\]

\[
= \begin{bmatrix} \tilde{\Omega} & \tilde{\Sigma} \tilde{\Omega} & \cdots & \tilde{\Sigma}^q \tilde{\Omega} \end{bmatrix}
\]  

(19)

where for notational convenience we have defined the quantities \( \tilde{\Omega} \equiv V^T \Omega \) and \( \tilde{\Sigma} \equiv \Sigma^2 \).

We “factor out” the component of the Krylov subspace that drives convergence from the component that is related to the initial starting subspace but independent of \( q \). To this end, define for \( 0 \leq p \leq q \),

\[
K_p \equiv UT_{2p+1}(\Sigma) \begin{bmatrix} \tilde{\Omega} & \tilde{\Sigma} \tilde{\Omega} & \cdots & \tilde{\Sigma}^q \tilde{\Omega} \end{bmatrix}
\]  

(20)

The matrices \( K \) and \( \tilde{K} \) are related as

\[
\text{span} \{ K_p \} \subseteq \text{span} \{ K \}
\]  

(21)

In light of this, since Step 3 of Alg. 1 is a projection, we are justified in our analysis to work with \( K_p \) instead of the more complicated \( K \).

Next, we multiply \( K_p \) by a specially constructed, full rank matrix \( X \). This operation will preserve the subspace spanned by the columns of \( K_p \), but align, as much as possible, the first \( k \) columns to the direction of the leading \( k \) singular vectors.
For all $0 \leq p \leq q$, let
\[
V_p = \begin{bmatrix}
\hat{\Omega} & \hat{\Sigma} & \cdots & \hat{\Sigma}^{r-p} & \hat{\Omega}
\end{bmatrix}
\] (22)
denote the generalized Vandermonde matrix from Eqn. 20 and partition this matrix as follows:
\[
V_p = \begin{bmatrix}
V_{11} & V_{12} \\
V_{21} & V_{22} \\
V_{31} & V_{32} \\
V_{41} & V_{42}
\end{bmatrix}
\] (23)
where the blocks in the first dimension are sized $k, s, r, t = n - (k + s + r)$ and the blocks in the second dimension are sized $k, r$. Intuitively, $s$ is used to handle duplicate or clustered singular values, while $r$ is used to create the “gap” that drives convergence (Fig. 2). With this partition, we examine the convergence behavior viewed as an accentuation of the “gap” by the appropriate Chebyshev polynomial.

We show the existence of at least one special non-singular $X \in \mathbb{R}^{(k+r)\times(k+r)}$ such that
\[
K_pX = UT_{2p+1}(\Sigma)V_pX
\] (24)
\[
= U
\begin{bmatrix}
Q_{11} & \hat{V}_{12} \\
Q_{21} & \hat{V}_{22} \\
0 & \hat{V}_{32} \\
H & \hat{V}_{42}
\end{bmatrix}
\] (25)
with $[Q_{11}, Q_{21}]$ a column orthogonal matrix. Notice the “gap” in the $(3, 1)$ block of size $r$ is created by using $X$ to align the columns of $K_p$.

We explicit construct such an $X$. Partition
\[
X = \begin{bmatrix}
X_{11} & X_{12} \\
X_{21} & X_{22}
\end{bmatrix}
\] (26)
\[
\Sigma = \begin{bmatrix}
\Sigma_1 & & \\
& \Sigma_2 & \\
& & \Sigma_3 \\
& & & \Sigma_4
\end{bmatrix}
\] (27)
where each dimension of $X$ is sized $k, r$, and each dimension of $\Sigma$ is sized $k, s, r, t = n - (k + s + r)$. Then,
\[
T_{2p+1}(\Sigma)V_pX \equiv \begin{bmatrix}
\tilde{V}_{11} & \cdots \\
\tilde{V}_{21} & \cdots \\
\tilde{V}_{31} & \cdots \\
\tilde{V}_{41} & \cdots
\end{bmatrix}
\] (28)
where
\[
\begin{array}{l}
\left(\tilde{V}_{11} / \tilde{V}_{21}\right) = \left(T_{2p+1}(\Sigma_1) / T_{2p+1}(\Sigma_2)\right) \left(V_{11} / V_{21}\right)
\end{array}
\]
\[
\tilde{V}_{31} = T_{2p+1}(\Sigma_3)(V_{31}X_{11} + V_{32}X_{21})
\]
\[
\tilde{V}_{41} = T_{2p+1}(\Sigma_4)(V_{41}X_{11} + V_{42}X_{21})
\]
Setting
\[
X_{21} = -V_{32}^{-1}V_{31}X_{11}
\] (29)
ensures the $(2, 1)$ block of dimensions $r \times k$ to be $\tilde{V}_{31} = 0$, and causes the $(1, 1)$ block of dimensions $(k+s)\times k$ to become
\[
\left(\tilde{V}_{11} / \tilde{V}_{21}\right) = \left(T_{2p+1}(\Sigma_1) / T_{2p+1}(\Sigma_2)\right) \left(V_{11} - V_{12}V_{32}^{-1}V_{31} / V_{21} - V_{22}V_{32}^{-1}V_{31}\right)
\]
x
\[
\text{and set}
\]
\[
X_{11} = \tilde{R}^{-1}
\] (31)
This ensures that
\[
\left(\tilde{V}_{11} / \tilde{V}_{21}\right) = \left(Q \tilde{R}^{-1} / \tilde{V}_{21}\right) = \left(Q \tilde{R}^{-1}\right)^T = \left(Q \tilde{R}^{-1}\right)^T = I
\] (32)
Let Eqn. (31) and Eqn. (29) define $X_{11}$ and $X_{21}$ respectively as
\[
\tilde{X}_{11} = \left[-V_{32}^{-1}V_{31}\right] \tilde{R}^{-1}
\] (33)
We specify
\[
\begin{bmatrix}
X_{12} \\
X_{22}
\end{bmatrix} = \begin{bmatrix}
X_{11} \\
X_{21}
\end{bmatrix} \perp
\] (34)
to provide a complete description of $X$ which satisfies Eqn. 25

**Remark III.1.** In order for the above derivation and thus Eqn. (23) and Eqn. (22) to be valid, the following conditions must hold: $\Omega$ is chosen to allow
- $V_{32}$ to be non-singular and thus invertible,
- $V_{11} - V_{12}V_{32}^{-1}V_{31}$ to be non-singular and thus $\tilde{R}$ to be invertible. Note that this expression is the Schur complement of the $(k+r)\times(k+r)$ matrix $[V_{11} V_{12} V_{31} V_{32}]$ with respect to the $V_{32}$ block.

We present a first result on a lower bound for the singular value of $B_k$.

**Lemma III.6.** Let $B_k$ be the matrix returned by Alg. [7] let $H$ be as defined in Eqn. (25), and assume that the two conditions in Remark III.1 hold. Then,
\[
s_k(B_k) > \frac{\sigma_k}{\sqrt{1 + \|H\|^2}}
\] (35)
\[
\text{Proof. The matrix returned by Alg. [7] is the k-truncated SVD of } QQ^T A, \text{ where the columns of } Q \text{ are an orthonormal basis for the column span of } K. \text{ By construction, it follows that}
\]
\[
s_k(B_k) \geq \sigma_k(Q_\tilde{Q}_p^T A)
\] (36)
where $Q_\tilde{p}$ contains columns that form an orthonormal basis for the column span of $K_pX$.

In particular, let $Q_\tilde{p}, R_\tilde{p}$ be the QR factorization of $K_pX$, partitioned as follows:
\[
K_pX = Q_\tilde{p}R_\tilde{p} = \begin{bmatrix}
Q_1 & Q_2 \\
\tilde{R}_{11} & \tilde{R}_{12} \\
\tilde{R}_{21} & \tilde{R}_{22}
\end{bmatrix}
\] (37)
where the block dimensions are sized $k, s$, as appropriate.

We can then write

$$
\hat{Q}_p \hat{Q}_p^T A = \hat{Q}_p \begin{bmatrix} \Sigma_1 & \Sigma_2 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} U \begin{bmatrix} \Sigma_1 \\ 0 \\ 0 \end{bmatrix} U^T V^T
$$

By the Cauchy interlacing theorem for singular values, it follows that

$$
\sigma_k(\hat{Q}_p \hat{Q}_p^T A) \geq \sigma_k(\hat{Q}_1^T U \begin{bmatrix} \Sigma_1 \\ 0 \\ 0 \end{bmatrix})
$$

We can compare the first $k$ columns of Eqn. (37) with the expression in Eqn. (25) to see that

$$
\hat{Q}_1 \hat{R}_{11} = U \begin{bmatrix} Q_{11} \\ Q_{21} \\ 0 \\ H \end{bmatrix}
$$

which helps us to write

$$
\hat{Q}_1^T U \begin{bmatrix} \Sigma_1 \\ 0 \\ 0 \end{bmatrix} = \left(U \begin{bmatrix} Q_{11} \\ Q_{21} \\ 0 \\ H \end{bmatrix} \hat{R}_{11}^{-1}\right) U \begin{bmatrix} \Sigma_1 \\ 0 \\ 0 \end{bmatrix} = \hat{R}_{11}^{-T} \begin{bmatrix} Q_{11} \\ Q_{21} \\ 0 \\ H \end{bmatrix} \begin{bmatrix} \Sigma_1 \\ 0 \\ 0 \end{bmatrix} = \hat{R}_{11}^{-T} [Q_{11}^T \Sigma_1 \ Q_{21}^T \Sigma_2]
$$

On the other hand,

$$
\sigma_{k+s} = \sigma_k \left( \sigma_{k+s} [Q_{11}^T \ Q_{21}^T] \right) \\
\leq \sigma_k \left( \left[Q_{11}^T \Sigma_1 \ Q_{21}^T \Sigma_2 \right] \right) \\
= \sigma_k \left( \left[Q_{11}^T \Sigma_1 \ Q_{21}^T \Sigma_2 \right] \right) \\
\leq \|R_{11}^T\|_2 \sigma_k \left( \left[Q_{11}^T \Sigma_1 \ Q_{21}^T \Sigma_2 \right] \right)
$$

Combining Eqns. (39), (38), (40), and (41), we obtain

$$
\sigma_k(B_k) \geq \frac{\sigma_{k+s}}{\|R_{11}^T\|_2}
$$

With the help of Eqn. (39),

$$
\hat{R}_{11}^T \hat{R}_{11} = \hat{R}_{11}^T \left(U^T \hat{Q}_1 \right) \left(U^T \hat{Q}_1 \right) \hat{R}_{11} = \left[U^T \hat{Q}_1 \right]^T \left[U^T \hat{Q}_1 \right] + H^T H
$$

which completes the proof.

We are now in a position to provide the proof for Theorem [III.1]

**Proof.** With an eye toward Lemma III.6 we proceed by providing a bound for $\|H\|_2^2$.

$$
\|H\|_2^2 = \sigma_1^2 (HH^T) = \sigma_1^2 \left(T_{2p+1}(\Sigma_4)(V_{41} - V_{42} V_{32}^{-1} V_{31}) \left(R_{11}^T R_{11}\right)^{-1} (V_{41} - V_{42} V_{32}^{-1} V_{31})^T T_{2p+1}(\Sigma_4) \right)
$$

$$
= \sigma_1^2 \left(T_{2p+1}(\Sigma_4)(V_{41} - V_{42} V_{32}^{-1} V_{31}) \left(V_{11} - V_{12} V_{32}^{-1} V_{31} V_{21} - V_{22} V_{32}^{-1} V_{31}) \left(T_{2p+1}(\Sigma_4) \right) \right)
$$

$$
\leq \sigma_1^2 \left(T_{2p+1}(\Sigma_4)(V_{41} - V_{42} V_{32}^{-1} V_{31}) \left((V_{11} - V_{12} V_{32}^{-1} V_{31} V_{21} - V_{22} V_{32}^{-1} V_{31})^T T_{2p+1}(\Sigma_4) \right) \right)
$$

$$
\leq \sigma_1 \left(T_{2p+1}(\Sigma_4)\right)
$$

$$
\leq \sigma_1 \left(T_{2p+1}(\Sigma_4)\right)
$$

$$
\leq \sigma_1 \left(T_{2p+1}(\Sigma_4)\right)
$$

The $1 + 2 \cdot \frac{\sigma_{k+s} - \sigma_{k+s+r+1}}{\sigma_{k+s+r+1}}$ factor is interpreted as shifting the Chebyshev polynomial $T_{2p+1}$ onto the interval $[0, \sigma_{k+s+r+1}]$, so that the tail of the singular spectrum is bounded by 1 and convergence is driven by the growth of the Chebyshev polynomial on the $[\sigma_k, \cdots, \sigma_1]$ part of the spectrum that we are interested in.

Repeating the previous argument for $1 \leq j \leq k$ completes the proof for the bound on $\sigma_j(B_k)$.
Due to space constraint, we omit the proofs for the corollaries of Theorem III.1. They are similar in flavor to the proof above and involve constructions of specifically chosen X matrices in each case.

We close by providing the proof for Theorem III.5

**Proof.** The statement of the theorem is equivalent to the statement that

$$CT^{-1}_{2p+1} \left( 1 + 2 \cdot \frac{\sigma_j - \sigma_{j+r+1}}{\sigma_{j+r+1}} \right) \rightarrow 0 \quad (46)$$

superlinearly. For notational convenience we assume $\sigma_j$ is not a multiple singular value and we have chosen $s = 0$; otherwise, the following argument can be made for the largest choice of $s$ such that $\sigma_{j+s} = \sigma_j$.

Recall that a sequence $\{a_n\}$ converges superlinearly to $a$ if

$$\lim_{n \to \infty} \frac{|a_{n+1} - a|}{|a_n - a|} = 0 \quad (47)$$

For any fixed $j = 1, \cdots, k$, define

$$a_q \equiv C(r)T^{-1}_{2(q+1 - \frac{k+1}{b})+1} \left( 1 + 2 \cdot \frac{\sigma_j - \sigma_{j+r+1}}{\sigma_{j+r+1}} \right)$$

where we have explicitly specified the dependence of the constant $C$ on the analysis parameter $r$, and expressed $p$ in terms of $q$. We approximate

$$a_q \approx C(r) \cdot \frac{1}{2} \left( 1 + g + \sqrt{2g} \right)^{-2\left(q+1 - \frac{k+1}{b}\right)+1}$$

$$\approx \frac{1}{2} \cdot C(r) \cdot \left( 1 + g + \sqrt{2g} \right)^{-2(1 - \frac{k+1}{b})+1} \cdot \left( 1 + g + \sqrt{2g} \right)$$

where $g = 2 \cdot \frac{\sigma_j - \sigma_{j+r+1}}{\sigma_{j+r+1}} = 2 \cdot \left( \frac{\sigma_j}{\sigma_{j+r+1}} - 1 \right)$

Then we argue that $a_{q+1}/a_q \to 0$ as follows.

$$\frac{a_{q+1}}{a_q} = \frac{1}{1 + g + \sqrt{2g}} \leq \frac{1}{(1 + g)^2} \quad (48)$$

Since we assume a spectrum such that $\sigma_i \to 0$ eventually, it is possible to choose $r$ sufficiently large such that $1/(1 + g)^2$ is arbitrarily small.

Rigorously, the above argument applies only to infinite dimensional operators, as in the finite dimensional case, $r \leq n$ cannot be chosen to be arbitrarily large. However, numerous previous works have noted that in practice, the convergence does tend to exhibit superlinear behavior for certain types of spectrums [30].

**IV. Numerical Experiments**

**A. Computational Complexity**

We will give an arithmetic complexity accounting of the randomized block Lanczos algorithm. The initialization of the random starting matrix $\Omega$ takes $O(nb)$ floating-point operations (flops). In step 1, the formation of the Krylov matrix $K$ consists of 1 matrix multiplications of $A\Omega$ along with $2(q-1)$ accumulated applications of either $A$ or $A^T$ for a total of $O(mnbq)$ flops. The orthonormal basis $Q$ of $K$ can be computed using a QR factorization using the standard Householder implementation, which has complexity $O(m(bq)^2)$. Finally, steps 3 and 4 consists of first forming $Q^T A$ for $O(mnbq)$ flops, then computing its truncated SVD factorization. Because the size of this matrix is $qb \times n$ and we expect $qb \approx k$ to be small, we assume its SVD computation is performed with a non-specialized dense matrix algorithm, using $O(n(bq)^2)$ flops. The final step of projecting the right $k$ singular vectors onto $Q$ is an additional $O(m(bq)^2)$ flops.

Overall, the computational complexity of Algorithm II is $O(mnbq + (m + n)(bq)^2)$. The first term dominates the computations and is the result of performing the matrix multiplications for the computation of the Lanczos block vectors. Fortunately, matrix multiplication is a highly optimized and highly tuned part of many matrix computation libraries, especially for suitably chosen block sizes.

We draw attention to the fact that the parameters $b$ and $q$ only appear together as the quantity $bq$ in our computational complexity count. This suggests that we may freely vary $b$, $q$ - as long as they vary inversely and the quantity $bq$ remains constant, the cost for running Algorithm II remains comparable. (In practice, this will only hold true for $b > 1$, due to the efficiency of BLAS2 and BLAS3 operations compared with BLAS1 operations.) Given the comparable computational complexity, and assuming the conditions for the convergence of Algorithm II is met, we need not privilege the block size choice $b = k$. In fact, we show empirically that in many cases, it is advantageous to choose block sizes $b$ strictly smaller than $k$.

**B. Activities and Sports Dataset**

The Activities and Sports Dataset is a dataset consisting of motion sensor data for 8 subjects performing 19 daily/sports activities, for 5 minutes, sampled at 25Hz frequency. This dataset can be found at [31].

The matrix associated with this dataset is dense and of dimensions $A \in \mathbb{R}^{5120 \times 5625}$, where each row is a sample and each entry is a double precision float. Figure 5 shows a plot of the first 500 singular values of $A$. As is typically for data matrices, this matrix exhibit spectrum decay on the order of $\sigma_j = \frac{1}{j}$, and our theory suggests that in this case, we should observe superlinear convergence for the RBL algorithm.

In this set of experiments, we investigate the convergence of a single singular value with respect to the number of iterations, in addition to the affect of the block size on convergence. We run the RSI and RBL algorithms on the Activities and Sports Dataset matrix with a target rank of $k = 200$, and examine the convergence of $\sigma_1$, $\sigma_{100}$, and $\sigma_{200}$. The results of these experiments are in Figures 5 and 6.

Each of these plots represent the convergence of a particular singular value. In each plot, each line represents a single parameter setting for the block size $b$, for either the RSI or the RBL algorithm. The $$axis is in log scale, and denote

$$\text{rel. err.} = \frac{\sigma_j - \sigma_j(B_k)}{\sigma_j} \quad (49)$$

B. Activities and Sports Dataset
Fig. 3. First 500 singular values of the Daily Activities and Sports Matrix.

Fig. 4. $k = 200$ approximation of the Daily Activities Dataset, convergence of $\sigma_1$.

Fig. 5. $k = 200$ approximation of the Daily Activities Dataset, convergence of $\sigma_{100}$.

Fig. 6. $k = 200$ approximation of the Daily Activities Dataset, convergence of $\sigma_{200}$.

, the relative error of the particular singular value we are examining. The $x$-axis is in linear scale, and denote the number of matrix-vector multiplications (MATVECs), a proxy measure for computational complexity. Markers on each line represent successive iterations of the algorithm. In these plots, down and to the left is good - we seek parameter settings that give good convergence for less computational complexity. We observe that, as expected, RSI converges linearly and RBL converges superlinearly. These trends are most clearly seen in Figure 6 and is also present in Figure 5. The convergence of $\sigma_1$ is extremely rapid in Figure 4 and reaches double precision in 2-5 iterations for all block sizes. In all cases, for both RBL and RSI, it appears that at the same computational complexity, choosing a smaller block size $b$, leads to more rapid convergence. For example, in Figure 6 we observe that in order for $\sigma_j$ to converge to a relative error of $\sim 10^{-5}$, taking $b = 1$ uses $1/2$ the number of MATVECs as taking $b = k = 200$.

C. Eigenfaces Dataset

The Eigenfaces dataset is available from the AT&T Laboratories’ Database of Faces [32], and consists of 10 different face images of 40 different subjects at $92 \times 112$ pixels resolution, varying in light, facial expressions, and other details. The widely cited technique for processing this data is via principal component analysis (PCA), where it was observed that each face can be composed in large part from a few prominent “Eigenfaces” [33].

The associated matrix is a dense matrix, which is formed by vectorizing each different face image as a column vector. It has dimensions $A \in \mathbb{R}^{10304 \times 400}$ and is of full numerical rank. The spectrum of this matrix spans 5 orders of magnitude but decays extremely rapidly, typical of data matrices. In fact, as
for all block sizes, the singular value approximation accuracy for this algorithm converges geometrically in the number of iterations, with a rate that is asymptotically superior to that achieved by the randomized subspace iteration algorithm. We have also shown for a matrix with spectrum decaying to zero, the RBL algorithm converges superlinearly. Additionally, we have provided numerical results in support of our analysis.

The current work is largely theoretical in nature, and there continues to be need for quality implementations of the Randomized Block Lanczos algorithm to aid its wider adoptability. To this end, continuations of the current work might include such an (possibly parallelized) implementation, along with further investigations of practical choices for the block size parameter \( b \) which balances the evident preference for a smaller \( b \) for convergence with the advantages of a larger \( b \) for computational efficiency and numerical stability.

**V. CONCLUSIONS**

In this paper, we have derived a novel convergence result for the randomized block Lanczos algorithm. We have shown that

**REFERENCES**

[1] M. B. Cohen, S. Elder, C. Musco, C. Musco, and M. Persu, “Dimensionality reduction for k-means clustering and low rank approximation,” in Proceedings of the forty-seventh annual ACM symposium on Theory of computing, ACM, 2015, pp. 163–172.

[2] H. M. Nguyen, X. Peng, M. N. Do, and Z.-P. Liang, “Denoising mr spectroscopic imaging data with low-rank approximations,” IEEE Transactions on Biomedical Engineering, vol. 60, no. 1, pp. 78–89, 2013.

[3] M. Fazel, E. Candes, B. Recht, and P. Parrilo, “Compressed sensing and robust recovery of low rank matrices,” in Signals, Systems and Computers, 2008 42nd Asilomar Conference on. IEEE, 2008, pp. 1043–1047.

[4] D. Anderson and M. Gu, “An efficient, sparsity-preserving, online algorithm for low-rank approximation,” in Proceedings of the 34th International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. International Convention Centre, Sydney, Australia: PMLR, 06–11 Aug 2017, pp. 156–165.

[5] J. Liu, P. Musialski, P. Wonka, and J. Ye, “Tensor completion for estimating missing values in visual data,” IEEE transactions on pattern analysis and machine intelligence, vol. 35, no. 1, pp. 208–220, 2013.

[6] N. Parikh, S. Boyd et al., “Proximal algorithms,” Foundations and Trends® in Optimization, vol. 1, no. 3, pp. 127–239, 2014.

[7] C. Eckart and G. Young, “The approximation of one matrix by another of lower rank,” Psychometrika, vol. 1, no. 3, pp. 211–218, 1936.

[8] G. Golub and W. Kahan, “Calculating the singular values and pseudo-inverse of a matrix,” Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis, vol. 2, no. 2, pp. 205–224, 1965.

[9] A. Talwalkar, S. Kumar, M. Mohri, and H. Rowley, “Large-scale svd and manifold learning,” The Journal of Machine Learning Research, vol. 14, no. 1, pp. 3129–3152, 2013.

[10] R. Mazumder, T. Hastie, and R. Tibshirani, “Spectral regularization algorithms for learning large incomplete matrices,” Journal of machine learning research, vol. 11, no. Aug, pp. 2287–2322, 2010.

[11] S. Cohen, B. Kcinelfeld, and G. Koutrika, “A survey on proximity measures for social networks,” in Search computing. Springer, 2012, pp. 191–206.

[12] M. Gu, “Subspace iteration randomization and singular value problems,” SIAM Journal on Scientific Computing, vol. 37, no. 3, pp. A1139–A1173, 2015.

[13] P. Drineas, R. Kannan, and M. W. Mahoney, “Fast monte carlo algorithms for matrices ii: Computing a low-rank approximation to a matrix,” SIAM Journal on computing, vol. 36, no. 1, pp. 158–183, 2006.

[14] V. Rokhlin, A. Szlam, and M. Tygert, “A randomized algorithm for principal component analysis,” SIAM Journal on Matrix Analysis and Applications, vol. 31, no. 3, pp. 1100–1124, 2009.

[15] N. Halko, P.-G. Martinsson, and J. A. Tropp, “Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions,” SIAM review, vol. 53, no. 2, pp. 217–288, 2011.
[16] C. Musco and C. Musco, “Randomized block krylov methods for stronger and faster approximate singular value decomposition,” in Advances in Neural Information Processing Systems, 2015, pp. 1396–1404.

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg et al., “Scikit-learn: Machine learning in python,” Journal of machine learning research, vol. 12, no. Oct, pp. 2825–2830, 2011.

[18] G. H. Golub and R. Underwood, “The block lanczos method for computing eigenvalues,” in Mathematical software. Elsevier, 1977, pp. 361–377.

[19] C. C. Paige, “Computational variants of the lanczos method for the eigenproblem,” IMA Journal of Applied Mathematics, vol. 10, no. 3, pp. 373–381, 1972.

[20] G. H. Golub, R. R. Underwood, and J. H. Wilkinson, “The lanczos algorithm for the symmetric ax= λ bx problem.” 1972.

[21] B. N. Parlett and D. S. Scott, “The lanczos algorithm with selective orthogonalization,” Mathematics of computation, vol. 33, no. 145, pp. 217–238, 1979.

[22] H. D. Simon, “The lanczos algorithm with partial reorthogonalization,” Mathematics of Computation, vol. 42, no. 165, pp. 115–142, 1984.

[23] S. Kaniel, “Estimates for some computational techniques in linear algebra,” Mathematics of Computation, vol. 20, no. 95, pp. 369–378, 1966.

[24] C. C. Paige, “The computation of eigenvalues and eigenvectors of very large sparse matrices.” Ph.D. dissertation, University of London, 1971.

[25] R. Underwood, “An iterative block lanczos method for the solution of large sparse symmetric eigenproblems,” Tech. Rep., 1975.

[26] Y. Saad, “On the rates of convergence of the lanczos and the blocklanczos methods,” SIAM Journal on Numerical Analysis, vol. 17, no. 5, pp. 687–706, 1980.

[27] J. Xiao and M. Gu, “Spectrum-revealing cholesky factorization for kernel methods,” in Data Mining (ICDM), 2016 IEEE 16th International Conference on. IEEE, 2016, pp. 1293–1298.

[28] J. Chen and Y. Saad, “Lanczos vectors versus singular vectors for effective dimension reduction,” IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 8, pp. 1091–1103, 2009.

[29] H.-C. Li and L.-H. Zhang, “Convergence of the block lanczos method for eigenvalue clusters,” Numerische Mathematik, vol. 131, no. 1, pp. 83–113, 2015.

[30] Y. Saad, “Theoretical error bounds and general analysis of a few lanczos-type algorithms,” in Proceedings of the Cornelius Lanczos International Centenary Conference (JD Brown, MT Chu, DC Ellison and RJ Plemmons, eds), SIAM, Philadelphia, PA, 1994, pp. 123–134.

[31] K. Altun, B. Barshan, and O. Tunçel, “Comparative study on classifying human activities with miniature inertial and magnetic sensors,” Pattern Recognition, vol. 43, no. 10, pp. 3605–3620, 2010.

[32] F. S. Samaria and A. C. Harter, “Parameterisation of a stochastic model for human face identification,” in Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on. IEEE, 1994, pp. 138–142.

[33] M. A. Turk and A. P. Pentland, “Face recognition using eigenfaces,” in Computer Vision and Pattern Recognition, 1991. Proceedings CVPR ’91., IEEE Computer Society Conference on. IEEE, 1991, pp. 586–591.