Randomized block Krylov space methods for trace and log-determinant estimators

Hanyu Li · Yuanyang Zhu

Received: date / Accepted: date

Abstract We present randomized algorithms based on block Krylov space method for estimating the trace and log-determinant of Hermitian positive semi-definite matrices. Using the properties of Chebyshev polynomial and Gaussian random matrix, we provide the error analysis of the proposed estimators and obtain the expectation and concentration error bounds. These bounds improve the corresponding ones given in the literature. Numerical experiments are presented to illustrate the performance of the algorithms and to test the error bounds.

Keywords Randomized algorithm · Krylov space method · Trace estimator · Log-determinant estimator · Chebyshev polynomial

Mathematics Subject Classification (2010) 68W20 · 68W25 · 15A18 · 15A15

1 Introduction

Computing the trace and the log-determinant of Hermitian positive semi-definite matrices finds many applications in various problems such as inverse problem [11], generalized cross validation [9], spatial statistics [28], and so on. Naturally, it is a straightforward problem if the matrices are explicitly defined and we can access the individual entries. For example, a standard approach for computing the determinant of Hermitian positive definite matrices is to leverage its LU decomposition or...
Cholesky decomposition [15, Section 14.6]. However, in big data age, it is difficult or expensive to explicitly access the individual entries or we can only access the matrix through matrix vector products. We will focus on the latter case in this paper. For this case, seeking the estimators with high accuracy for trace and log-determinant will be of great interest.

For the trace of Hermitian positive semi-definite matrix $A$, the popular and simple estimator is the Monte Carlo estimator proposed by Hutchinson [18]:

$$\text{Tr}(A) \approx \frac{1}{N} \sum_{i=1}^{N} c_i A^T c_i,$$

where $N$ is the sample size, and $c_i$ are the independently and identically distributed (i.i.d.) Rademacher random vectors. Hutchinson [18] showed that this estimator is unbiased. Later, by replacing the i.i.d. random vectors $c_i$ with Gaussian random vectors, random unit vectors, or columns of the identity matrix sampled uniformly, some scholars produced some other unbiased estimators [3, 24], where Avron and Toledo [3] first considered the number of Monte Carlo samples $N$ with which a relative error $\varepsilon$ with probability $1 - \delta$ can be achieved and defined an $(\varepsilon, \delta)$ estimator:

$$P \left[ \| \text{Tr}(A) - \frac{1}{N} \sum_{i=1}^{N} c_i A^T c_i \| \leq \varepsilon \text{Tr}(A) \right] \geq 1 - \delta.$$

Some lower bounds for $N$ for different choice of the random vectors $c_i$ were provided in [3], which were improved by Roosta-Khorasani and Ascher [24]. In 2017, Lin [20] proposed two new trace estimators from the view of the randomized low-rank approximation of the matrix $A$ with order $n$:

$$\text{Tr}(A) \approx \text{Tr}(A \Omega (A^* \Omega)^\dagger (A \Omega)^*),$$

$$\text{Tr}(A) \approx \text{Tr}(A \Omega ((A \Omega)^* A \Omega)^\dagger ((A \Omega)^* A \Omega)^*)),$$

where $\Omega$ is a Gaussian random matrix of size $n \times k$ with $k \ll n$ and, for a matrix $X$, $X^\dagger$ denotes its Moore-Penrose inverse. The author mainly investigated the first estimator and found that the method can be much faster than the Monte Carlo estimator. However, there was no formal error analysis for this estimator provided in [20]. Later, Saibaba et al. [26] also presented a new trace estimator based on randomized low-rank approximation and provided detailed error analysis to validate the theoretical reliability of the estimator.

For the log-determinant of Hermitian positive definite matrix $A$, a popular approach is to combine the identity $\log \det A = \text{Tr}(\log A)$ with the Monte Carlo estimators for trace introduced above. With this idea, Barry and Pace [4] first proposed the Monte Carlo estimator of log-determinant for large sparse matrix. Later, Zhang and Leithead [29] generalized the estimator to general Hermitian positive definite matrix. However, both of the above two papers didn’t provide the rigorous error analysis of these estimators. Recently, Boutsidis et al. [5] continued the above work and investigated the error analysis in detail based on the results from [3]. In the above log-determinant estimators [4, 29, 5], the Taylor expansion was used to expand
log(A). Pace and LeSage [23] first introduced Chebyshev approximation to approximate log(A), however, they didn’t combine the approximation with Monte Carlo estimators for trace and there was no formal error analysis. These works were done by Han et al. [14] and they also generalized the method to trace estimator for matrix function [13]. In addition, some scholars also used Cauchy integral formula or spline to expand log(A) and to devise the estimators for log-determinant [21]. Recently, based on randomized low-rank approximation of matrix, Saibaba et al. [26] proposed a new log-determinant estimator without using Taylor expansion or Chebyshev approximation, and discussed the error analysis for this estimator in detail.

For the more accurate trace and log-determinant estimators given in [26], a main and attractive feature is that they took advantage of randomized subspace iteration algorithm, which has been extensively studied and found many applications [12, 8, 10]. In recent years, some scholars found that the randomized block Krylov space methods have more advantage compared randomized subspace iteration algorithm [21, 6, 27]. For example, the former has faster eigenvalue convergence rate when the target matrix has a large eigenvalue gap whose location is known. As a result, the randomized block Krylov space method receives more and more attention from the points of view of theory and applications [21, 6, 27, 7]. Inspired by the advantage of the randomized block Krylov space method and the work of Saibaba et al. [26], we consider the new estimators for trace and log-determinant of Hermitian positive definite matrices and their error analysis in the present paper. The obtained error bounds for these estimators will be tighter than the corresponding ones given in [26] in most of cases.

The rest of this paper is organized as follows. Section 2 presents some preliminaries. In Section 3, we provide the main algorithms and the error analysis of our estimators. The comparisons between our results with the ones from [26] are also discussed in this section. Section 4 is devoted to numerical experiments to illustrate our new randomized estimators and to test the error bounds. Finally, the concluding remarks of the whole paper are presented.

2 Preliminaries

In this section, we first clarify our assumptions. Then, we review some results on Chebyshev polynomials and the algorithms from [26].

2.1 Assumptions

Throughout this paper, we assume that \( A \in \mathbb{C}^{n \times n} \) is a Hermitian positive semi-definite matrix and has the following eigenvalue decomposition:

\[
A = U \Lambda U^*, \quad \Lambda = \text{diag} (\lambda_1 \cdots \lambda_n) \in \mathbb{R}^{n \times n},
\]

(2.1)

where \( U \in \mathbb{C}^{n \times n} \) is unitary, the eigenvalues satisfy \( \lambda_1 \geq \cdots \geq \lambda_n \), and we assume there is a gap in these eigenvalues: \( \lambda_k > \lambda_{k+1} \). As done in [26], we partition \( \Lambda \) and \( U \)
as follows
\[ \Lambda = \begin{pmatrix} \Lambda_1 \\ \Lambda_2 \end{pmatrix}, \quad U = \begin{pmatrix} U_1 \\ U_2 \end{pmatrix}, \]
where \( \Lambda_1 \in \mathbb{R}^{k \times k}, \Lambda_2 \in \mathbb{R}^{(n-k) \times (n-k)}, U_1 \in \mathbb{C}^{n \times k}, \) and \( U_2 \in \mathbb{C}^{n \times (n-k)} \).

Given a number \( q \geq 1 \) and a Gaussian random matrix \( \Omega \in \mathbb{C}^{n \times l} \) with \( k \leq l = k + p \ll n \), set
\[ K_q = (A\Omega, A^2\Omega, \cdots, A^q\Omega). \]
Like [6], we write
\[ K_q = \text{range} (A\Omega, A^2\Omega, \cdots, A^q\Omega), \]
call it the block Krylov space in \( A \) and \( \Omega \), and assume \( \text{dim} (K_q) = ql \). That is, \( K_q \) has full column rank. It is known that the elements of the Krylov subspace \( K_q \) can be expressed in terms of matrices \( \phi (A) \Omega \in \mathbb{C}^{n \times l} \) [19, 6], where \( \phi (\cdot) \) is a polynomial of degree \( q \). Considering the eigenvalue decomposition of \( A \) in (2.1), it is easy to verify that
\[ K = \phi (A) \Omega = U \phi (A) U^* \Omega, \]
where
\[ \phi (A) = \text{diag} (\phi (\lambda_1), \phi (\lambda_2), \cdots, \phi (\lambda_n)) = \begin{pmatrix} \phi (\Lambda_1) \\ \phi (\Lambda_2) \end{pmatrix}. \]

Like [10, 26], we denote \( \widehat{\Omega} = U^* \Omega \) and hence
\[ \widehat{\Omega} = \begin{pmatrix} U_1^* \Omega \\ U_2^* \Omega \end{pmatrix} = \begin{pmatrix} \widehat{\Omega}_1 \\ \widehat{\Omega}_2 \end{pmatrix}, \]
where \( \widehat{\Omega}_1 = U_1^* \Omega \in \mathbb{C}^{k \times l} \) and \( \widehat{\Omega}_2 = U_2^* \Omega \in \mathbb{C}^{(n-k) \times l} \). We assume that \( \text{rank} (\widehat{\Omega}_1) = k \), and hence its Moore-Penrose inverse \( \widehat{\Omega}_1^\dagger \) satisfies \( \widehat{\Omega}_1 \widehat{\Omega}_1^\dagger = I_k \).

2.2 Chebyshev polynomials

The \( q \)th degree Chebyshev polynomial is recursively defined as follows
\[ T_0 (x) \equiv 1; \quad T_1 (x) \equiv x; \quad T_q (x) \equiv 2qT_{q-1} (x) - T_{q-2} (x). \]
They can also be expressed as
\[ T_q (x) = \begin{cases} \frac{(x + \sqrt{x^2 - 1})^q + (x - \sqrt{x^2 - 1})^q}{2}, & x \geq 1, \\ \cos (q \arccos (x)), & -1 \leq x \leq 1. \end{cases} \]
By Chebyshev polynomials, we construct a polynomial \( f(\cdot) \) with degree \( q - 1 \), which will play an important role in our error analysis for algorithms.

\[
f(x) = \frac{T_{q-1}\left(\frac{2x-\lambda_{k+1}}{\lambda_{k+1}}\right)}{T_{q-1}\left(\frac{2\lambda_k-\lambda_{k+1}}{\lambda_{k+1}}\right)},
\]

where \( \lambda_{k+1} \) and \( \lambda_k \) are the eigenvalues of \( A \). By the properties of Chebyshev polynomials [21, 6], we can check that the polynomial \( f(x) \) has the following properties:

1. \( f(\lambda_i) \geq 1 \), when \( 1 \leq i \leq k \);
2. \( |f(\lambda_i)| \leq T_{q-1}\left(\frac{2\lambda_k-\lambda_{k+1}}{\lambda_{k+1}}\right) \), when \( k + 1 \leq i \leq n \).

From the property 1, it follows that \( f(A_1) \) is nonsingular, and

\[
\|f^{-1}(A_1)\|_2 \leq 1.
\]  

From the property 2, we have that

\[
\|f(A_2)\|_2 \leq T_{q-1}\left(\frac{2\lambda_k-\lambda_{k+1}}{\lambda_{k+1}}\right),
\]

2.3 Randomized subspace iteration algorithm

The following algorithm was proposed by Saibaba et al. [26]. Based on the algorithm, the authors presented the estimators of trace and log-determinant: \( \text{Tr}(A) \approx \text{Tr}(T) \) and \( \log \det(I + A) \approx \log \det(I + T) \).

**Algorithm 1:** Randomized subspace iteration [26]

**Input:** \( A \in \mathbb{C}^{n \times n} \): Hermitian positive semi-definite matrix; \( k \): target rank; \( q \): number of subspace iteration; \( \Omega \in \mathbb{C}^{n \times l} \): Gaussian random matrix with \( k \leq l = k + p \ll n \).

**Output:** \( T \in \mathbb{C}^{l \times l} \).

1. Multiply \( Y = A^q \Omega \);
2. Thin QR factorization \( Y = QR \);
3. Compute \( T = Q^*A^q \).

From this algorithm, it is easy to find that the information in \( A\Omega, A^2 \Omega, \cdots, A^{q-1} \Omega \) is discarded when computing \( Q \). Collecting these information and using them for computing \( Q \) is one of the main motivations of this study, and is also an important topic in the field of randomized algorithm [21, 6, 27, 7].

3 Algorithms and error analysis

In this section, we first introduce our algorithms for estimating the trace and log-determinant, and then present the error bounds of these new estimators and their proofs.
3.1 Algorithms

Algorithm 2: Randomized block Krylov space method

**Input:** $A \in \mathbb{C}^{n \times n}$: Hermitian positive semi-definite matrix; $k$: target rank; $q$: number of block Krylov space iterations; $\Omega \in \mathbb{C}^{n \times l}$: Gaussian random matrix with $k \leq l = k + p \ll n$.

**Output:** $T \in \mathbb{C}^{ql \times ql}$.

1. Multiply and collect $K_q = (A\Omega, A^2\Omega, \cdots, A^q\Omega)$;
2. Thin QR factorization $K_q = Q_qR_q$;
3. Compute $T = Q_q^*AQ_q$.

Comparing with Algorithm 1, we can find that the randomized block Krylov space method collects the information discarded in Algorithm 1 and hence will be more accurate in theory. Numerical experiments in Section 4 also confirm this result. Moreover, the complexity of Algorithm 2 is only a little higher than that for Algorithm 1. This is because the main parts of complexity of the two algorithms, i.e., the complexity in step 1, are the same. Only in steps 2 and 3, the complexity of our algorithm increases. The factor $l$ in complexity is increased to $ql$. Since $q$ is a small number, the total complexity doesn’t change very much. In addition, similar to the discussions in [26], Algorithm 2 is only an idealized version and the idealized block Krylov space iteration can be numerically unstable. In practice, we can alternate matrix products and QR factorizations to tackle this problem [25, Algorithm 5.2].

3.2 Error bounds

**Theorem 3.1** (Expectation bounds) Let $T = Q_q^*AQ_q$ be computed by Algorithm 2 and furthermore, let $p \geq 2$. Then

$$0 \leq \mathbb{E}[\text{Tr}(A) - \text{Tr}(T)] \leq \left(1 + \frac{\lambda_{k+1}^{T-2} - \lambda_{k+1}^{T-1}}{\lambda_k^{T-1}} \right) \frac{C_{ge}}{\lambda_{k+1}^{T-1}} \text{Tr}(A_2)$$

and

$$0 \leq \mathbb{E} \left[ \log \det(I_n + A) - \log \det(I_q + T) \right] \leq \log \det \left( I_{n-k} + \frac{\lambda_{k+1}^{T-2} - \lambda_{k+1}^{T-1}}{\lambda_k^{T-1}} \right) C_{ge}A_2 + \log \det(I_{n-k} + A_2),$$

where $C_{ge} = \frac{\mu + \sqrt{2}}{\mu + \sqrt{2} + \frac{1}{\mu + \sqrt{2} + \frac{1}{\mu + \sqrt{2}}}}$ with $\mu = \sqrt{n-k + \sqrt{t}}$.

**Theorem 3.2** (Concentration bounds) Let $T = Q_q^*AQ_q$ be computed by Algorithm 2 and furthermore, let $p \geq 2$. If $0 \leq \delta \leq 1$, then with probability at least $1 - \delta$

$$0 \leq \text{Tr}(A) - \text{Tr}(T) \leq \left(1 + \frac{\lambda_{k+1}^{T-2} - \lambda_{k+1}^{T-1}}{\lambda_k^{T-1}} \right) \frac{C_{ge}}{\lambda_{k+1}^{T-1}} \text{Tr}(A_2)$$
and

\[ 0 \leq \log \det(I_n + A) - \log \det(I_{ql} + T) \leq \log \det \left( I_{n-k} + \frac{\lambda_{k+1}}{\lambda_k} T_q^{-2} \left( \frac{2 \lambda_k - \lambda_{k+1}}{\lambda_k} \right) C_g A_2 \right) + \log \det(I_{n-k} + A_2), \]

where \( C_g = \left( \mu + \sqrt{2 \log \frac{2}{\delta}} \right)^2 \left( \frac{2}{\delta} \right)^{p+1} \left( \sqrt{\frac{T}{p+1}} \right) \) with \( \mu = \sqrt{n-k+\sqrt{l}}. \)

In [26], Saibaba et al. presented the following error bounds for estimators of trace and log-determinant produced by Algorithm 1.

**Theorem 3.3 (Expectation bounds)** [26] Let \( T = Q^*AQ \) be computed by Algorithm 1 and furthermore, let \( p \geq 2. \) Then

\[ 0 \leq \mathbb{E} [\text{Tr}(A) - \text{Tr}(T)] \leq \left( 1 + \left( \frac{\lambda_{k+1}}{\lambda_k} \right)^{2q-1} C_g \right) \text{Tr}(A_2) \]

and

\[ 0 \leq \mathbb{E} [\log \det(I_n + A) - \log \det(I_{I} + T)] \leq \log \det \left( I_{n-k} + \left( \frac{\lambda_{k+1}}{\lambda_k} \right)^{2q-1} C_g A_2 \right) + \log \det(I_{n-k} + A_2). \]

**Theorem 3.4 (Concentration bounds)** [26] Let \( T = Q^*AQ \) be computed by Algorithm 1 and furthermore, let \( p \geq 2. \) If \( 0 \leq \delta \leq 1, \) then with probability at least \( 1 - \delta \)

\[ 0 \leq \text{Tr}(A) - \text{Tr}(T) \leq \left( 1 + \left( \frac{\lambda_{k+1}}{\lambda_k} \right)^{2q-1} C_g \right) \text{Tr}(A_2) \]

and

\[ 0 \leq \log \det(I_n + A) - \log \det(I_{I} + T) \leq \log \det \left( I_{n-k} + \left( \frac{\lambda_{k+1}}{\lambda_k} \right)^{2q-1} C_g A_2 \right) + \log \det(I_{n-k} + A_2). \]

Note that

\[ \frac{2 \lambda_k - \lambda_{k+1}}{\lambda_{k+1}} \geq \frac{\lambda_{k+1}}{\lambda_{k+1}} > 1, \]

and \( T_{q-1}(x) \) increases faster than \( x^{q-1} \) when \( q \geq 3 \) and \( x > 1. \) Thus, the term

\[ T_{q-1}^{-2} \left( (2 \lambda_k - \lambda_{k+1})/\lambda_{k+1} \right) \]

in our bounds is smaller than \( (\lambda_{k+1}/\lambda_k)^{2q-2} \) in the bounds in Theorems 3.3 and 3.4 when \( q \geq 3. \) However, it must be pointed out that the order of the matrix \( T \) in the bounds in Theorems 3.3 and 3.4 is \( l \) not \( ql. \) If we set the order to be \( ql, \) i.e., set \( p = ql - k, \) the terms \( C_{ge} \) and \( C_g \) in Theorems 3.3 and 3.4 will become small and hence the bounds will be reduced. In this case, our bounds can’t be always tighter than
the corresponding ones in Theorems 3.3 and 3.4 when $q \geq 3$. However, numerical experiments show that in most of cases, our bounds are tighter. The following are some simple examples. We set $n = 3000$, $k = 30$, $p = 10$, $\lambda_k / \lambda_{k+1} = 20$, and $\delta = 0.01$, and $q = 3, 4, 5$, respectively. Upon some computations, we have Table 3.1 where $C_{geq}$ and $C_{gq}$ are derived from $C_{ge}$ and $C_{g}$, respectively, by replacing $p$ with $ql - k$.

| $q$ | 3   | 4   | 5   |
|-----|-----|-----|-----|
| $\lambda_k T^{-1} \lambda_{k+1} \left( \frac{2k}{\lambda_k} \right) C_{ge}$ | 4.2541e-05 | 6.9946e-09 | 1.1500e-12 |
| $\left( \frac{\lambda_k}{\lambda_{k+1}} \right)^{q-1} C_{geq}$ | 1.4266e-04 | 2.4408e-07 | 4.7055e-10 |
| $\lambda_k T^{-1} \lambda_{k+1} \left( \frac{2k}{\lambda_k} \right) C_{g}$ | 1.4210e-04 | 2.3363e-08 | 3.8414e-12 |
| $\left( \frac{\lambda_k}{\lambda_{k+1}} \right)^{q-1} C_{gq}$ | 1.7747e-04 | 2.8924e-07 | 5.4284e-10 |

3.3 Proofs of Theorems 3.1 and 3.2

We only prove the structural (deterministic) error bounds for trace and log-determinant estimators, i.e., we consider $\Omega$ to be any matrix satisfying assumptions given in Section 2.1. The final results, i.e., the expectation and concentration error bounds in Theorems 3.1 and 3.2, can be derived immediately as done in [26, Sections 4.1.1 and 4.1.2] by combining the structural error bounds in Theorems 3.5 and 3.6 given below and [26, Lemmas 4 and 5]. We first do some preparation for these proofs. A useful lemma is listed as follows.

**Lemma 3.1** [10] Assume that $X \in \mathbb{C}^{l \times l}$ is nonsingular, and the thin QR factorizations of $K$ and $KX$ are $K = QR$ and $KX = \tilde{Q}\tilde{R}$, respectively. Then

$$QQ^* = \tilde{Q}\tilde{Q}^*.$$ 

This simple result plays an important role in deriving error bounds because we can choose a special $X$ to achieve the useful information of range of $K$. This technique was proposed by Gu [10]. Now we introduce how to find the special $X$. Using the notation introduced in Section 2.1 we write $K$ as follows,

$$K = U \phi (\Sigma) \hat{\Omega} = U \begin{pmatrix} \phi (A_1) & \phi (A_2) \end{pmatrix} \begin{pmatrix} \hat{\Omega}_1 & \hat{\Omega}_2 \\ \hat{\Omega}_1 & \hat{\Omega}_2 \end{pmatrix} = U \begin{pmatrix} \phi (A_1) \hat{\Omega}_1 \\ \phi (A_2) \hat{\Omega}_2 \end{pmatrix}. \tag{3.1}$$

To make the last block matrix in (3.1) be simplified, we choose a matrix $X$ in the following form:

$$X = \left( \hat{\Omega}_1 \phi^{-1} (A_1), X_2 \right) \in \mathbb{C}^{l \times l},$$
where $X_2 \in \mathbb{C}^{l \times p}$ is such that $X$ is nonsingular and $\hat{\Omega}_1 X_2 = 0$. Thus,

$$KX = U \begin{pmatrix} I_k & 0 \\ H_1 & H_2 \end{pmatrix},$$

where

$$H_1 = \phi (A_2) \hat{\Omega}_2 \Omega_1^{-1} (A_1), \quad H_2 = \phi (A_2) \hat{\Omega}_2 X_2.$$  \hfill (3.2)

In accord with the above block form, we write the thin QR factorization of $KX$ in the following form:

$$KX = \tilde{Q} \tilde{R} = (\tilde{Q}_1, \tilde{Q}_2) \begin{pmatrix} \tilde{R}_{11} & \tilde{R}_{12} \\ \tilde{R}_{21} & \tilde{R}_{22} \end{pmatrix} = U \begin{pmatrix} I_k & 0 \\ H_1 & H_2 \end{pmatrix}.$$  \hfill (3.3)

As a result, we have the following thin QR factorization

$$U \begin{pmatrix} I_k \\ H_1 \end{pmatrix} = \tilde{Q}_1 \tilde{R}_{11},$$  \hfill (3.4)

which will be used in our error analysis.

Furthermore, we also need the following three results.

**Lemma 3.2** Suppose $\text{range}(N) \subseteq \text{range}(M)$. Then, for any Hermitian positive semi-definite matrix $A$, the following inequality holds

$$\text{Tr}(P_N A) \leq \text{Tr}(P_M A),$$

where $P_N$ and $P_M$ are the orthogonal projections on $\text{range}(N)$ and $\text{range}(M)$, respectively.

**Proof** From the proof of [12, Proposition 8.5], we know $P_N \preceq P_M$, where $\preceq$ denotes the L"owner partial order [17, Definition 7.7.1]. Then, by the known conjugation rule (see e.g., [17, Theorem 7.7.2]),

$$A^\dagger P_N A^\dagger \preceq A^\dagger P_M A^\dagger.$$

Further, by the properties of L"owner partial order and trace, we have

$$\text{Tr} \left( A^\dagger P_N A^\dagger \right) \preceq \text{Tr} \left( A^\dagger P_M A^\dagger \right), \quad \text{i.e., Tr}(P_N A) \leq \text{Tr}(P_M A).$$

\hfill $\square$

**Lemma 3.3** For any two Hermitian positive semi-definite matrices $A$ and $B$ with the same order, the following inequalities hold

$$\text{Tr}(AB) \leq \text{Tr}(A) \lambda_{\text{max}}(B) \leq \text{Tr}(A) \text{Tr}(B),$$

$$\text{Tr}(AB) \leq \lambda_{\text{max}}(A) \text{Tr}(B) \leq \text{Tr}(A) \text{Tr}(B),$$

where $\lambda_{\text{max}}(A)$ and $\lambda_{\text{max}}(B)$ are the largest eigenvalues of $A$ and $B$, respectively.

**Proof** These inequalities are well-known results and can be derived from, e.g., von Neumann trace theorem [17, Theorem 7.4.1.1] directly. \hfill $\square$

**Lemma 3.4** [22, Corollary 2.1] If $B \in \mathbb{C}^{m \times n}$ and $C \in \mathbb{C}^{n \times m}$, then

$$\det(I_m \pm BC) = \det(I_n \pm CB).$$
3.3.1 Structural bounds for trace estimator

**Theorem 3.5** Let $T = Q_q^*AQ_q$ be computed by Algorithm[2]. Then

\[
0 \leq \text{Tr}(A) - \text{Tr}(T) \leq \left( 1 + T_{q-1}^{-1} \left( \frac{2\lambda_k - \lambda_{k+1}}{\lambda_{k+1}} \right) \|\hat{\Omega}_2\hat{\Omega}_1\|_2 \right) \text{Tr}(A_2). 
\]  

(3.5)

When $0 < \left\|\hat{\Omega}_2\hat{\Omega}_1\right\|_2 \leq \frac{\lambda_k}{\lambda_{k+1}} T_{q-1}^{-1} \left( \frac{2\lambda_k - \lambda_{k+1}}{\lambda_{k+1}} \right)$, the following bound is tighter:

\[
0 \leq \text{Tr}(A) - \text{Tr}(T) \leq \left( 1 + \frac{\lambda_{k+1}}{\lambda_k} T_{q-1}^{-2} \left( \frac{2\lambda_k - \lambda_{k+1}}{\lambda_{k+1}} \right) \|\hat{\Omega}_2\hat{\Omega}_1\|^2 \right) \text{Tr}(A_2). 
\]

(3.6)

**Proof** The lower bound has been proven in [26, Lemma 1]. In the following, we show that the upper bounds hold.

Since $K = \phi(A) \Omega$ is an element of $\mathcal{K}_q$, we have

\[
\text{range}(K) \subset \mathcal{K}_q.
\]

Thus, by Lemma[57,1] we get

\[
\text{range}(\hat{Q}) = \text{range}(Q) \subseteq \text{range}(Q_q),
\]

(3.7)

which together with Lemma[3.2] and (3.3) implies

\[
\text{Tr}(A) - \text{Tr}(T) = \text{Tr}(A) - \text{Tr}(Q_q^*AQ_q) = \text{Tr}(A) - \text{Tr}(Q_qQ_q^*)
\]

\[
\leq \text{Tr}(A) - \text{Tr}(QQ^*) = \text{Tr}(A) - \text{Tr}(QQ^*)
\]

\[
\leq \text{Tr}(A) - \text{Tr}(Q\hat{Q}^*) = \text{Tr}(A) - \text{Tr}(Q\hat{Q}^*A).
\]

(3.8)

From (3.4), we have

\[
\hat{Q}_1 = U \begin{pmatrix} I_k \\ H_1 \end{pmatrix} \hat{R}_{11}^{-1} \text{ and } \hat{R}_{11} - \hat{R}_{11} = I_k + H_1^*H_1.
\]

(3.9)

As a result,

\[
\hat{Q}^*_1A\hat{Q}_1 = (\hat{R}_{11}^{-1})(I_k, H_1^*)U^*U \begin{pmatrix} A_1 \\ A_2 \end{pmatrix} U^*U \begin{pmatrix} I_k \\ H_1 \end{pmatrix} \hat{R}_{11}^{-1}
\]

\[
= (\hat{R}_{11}^{-1})(A_1 + H_1^*A_2H_1)\hat{R}_{11}^{-1},
\]

(3.10)

and hence

\[
\text{Tr}(\hat{Q}^*_1A\hat{Q}_1) = \text{Tr}((A_1 + H_1^*A_2H_1)(\hat{R}_{11}^{-1}\hat{R}_{11}^{-1})) = \text{Tr}((A_1 + H_1^*A_2H_1)(I_k + H_1^*H_1)^{-1})
\]

\[
= \text{Tr}(A_1(I_k + H_1^*H_1)^{-1}) + \text{Tr}(A_2H_1(I_k + H_1^*H_1)^{-1}H_1^*H_1).
\]

(3.11)

Substituting (3.11) into (3.8) and noting $\text{Tr}(A) = \text{Tr}(A_1) + \text{Tr}(A_2)$ and Lemma[3.3] gives

\[
\text{Tr}(A) - \text{Tr}(T) \leq \text{Tr}(A_1(I_k - (I_k + H_1^*H_1)^{-1})) + \text{Tr}(A_2(I_k - H_1(I_k + H_1^*H_1)^{-1}H_1^*))
\]

\[
\leq \text{Tr}(A_1(I_k - (I_k + H_1^*H_1)^{-1})) + \text{Tr}(A_2) \lambda_{\text{max}}(I_k - H_1(I_k + H_1^*H_1)^{-1}H_1^*)
\]

\[
\leq \text{Tr}(A_1(I_k - (I_k + H_1^*H_1)^{-1})) + \text{Tr}(A_2).
\]

(3.12)
Note that
\[(I_k + H_1^*H_1)^{-1} = I_k - H_1^*(I_{n-k} + H_1H_1^*)^{-1}H_1.\]

Then
\[\text{Tr}(A_1(I_k - (I_k + H_1^*H_1)^{-1})) = \text{Tr}(A_1H_1^*(I_{n-k} + H_1H_1^*)^{-1}H_1).\]

Further, setting \(\phi(x) = xf(x),\) where \(f(x)\) is defined in (2.22), and considering \(H_1\) in (3.2), von Neumann trace theorem (7.4.1.1), and singular value inequalities (16), we have
\[\text{Tr}(A_1(I_k - (I_k + H_1^*H_1)^{-1})) = \sum_{j=1}^{k-1} \sigma_j\left(f^{-1}(A_1)(\tilde{V}_2\tilde{\Omega}_1^*)^*A_2f(A_2)(I_{n-k} + H_1H_1^*)^{-1}H_1\right),\]

which combined with (2.3) and (2.4) leads to
\[\text{Tr}(A_1(I_k - (I_k + H_1^*H_1)^{-1})) \leq T_q^{-1}\left(2\lambda_k - \lambda_{k+1}\right) \left\|\tilde{V}_2\tilde{\Omega}_1^*\right\|_2 \left\|(I_{n-k} + H_1H_1^*)^{-1}H_1\right\|_2 \left\|\text{Tr}(A_2)\right\|. \quad (3.13)\]

Furthermore, from (2.6), we have
\[\left\|(I_{n-k} + H_1H_1^*)^{-1}H_1\right\|_2 \leq 1, \quad (3.14)\]

or
\[\left\|(I_{n-k} + H_1H_1^*)^{-1}H_1\right\|_2 \leq \left\|H_1\right\|_2 \leq \left\|A_2f(A_2)\right\|_2 \left\|A_1^{-1}f^{-1}(A_1)\right\|_2 \left\|\tilde{V}_2\tilde{\Omega}_1^*\right\|_2 \leq \frac{\lambda_{k+1}T_q^{-1}}{\lambda_k} \left(2\lambda_k - \lambda_{k+1}\right) \left\|\tilde{V}_2\tilde{\Omega}_1^*\right\|_2. \quad (3.15)\]

Thus, combining (3.12), (3.13), (3.14), and (3.15), we derive the desired upper bounds (3.5) and (3.6).

### 3.3.2 Structural bounds for log-determinant estimator

**Theorem 3.6** Let \(Q^*AQ\) be computed by Algorithm (2). Then
\[0 \leq \log\det(I_n + A) - \log\det(I_n + T) \leq \log\det(I_{n-k} + \eta A_2) + \log\det(I_{n-k} + A_2), \quad (3.16)\]

where \(\eta = \frac{\lambda_{k+1}T_q^{-1}}{\lambda_k}\left(2\lambda_k - \lambda_{k+1}\right) \left\|\tilde{V}_2\tilde{\Omega}_1^*\right\|_2^2.\)
Proof. The lower bound has been derived in [26, Lemma 2]. It is sufficient to show that the upper bound holds.

From (3.7), it follows that there is an orthonormal matrix \( Y \in \mathbb{C}^{l \times l} \) such that \( Q = Q_q Y \), and hence \( Q^*A Q = Y^* Q^*_q A Q_q Y = Y^* T Y \).

Thus, by the proved lower bound, i.e.,
\[
\log \det(I_n + A) - \log \det(I_q + T) \geq 0,
\]
we have
\[
\log \det(I_q + T) - \log \det(I_q + Y^* T Y) \geq 0.
\]
That is, \( \log \det(I_q + Q^*_q A Q_q) - \log \det(I_q + Q^* A Q) \geq 0 \).

Thus,
\[
\log \det(I_n + A) - \log \det(I_q + T) \leq \log \det(I_n + A) - \log \det(I_q + Q^* A Q).
\]

By Lemmas 3.3 and 3.1, it is seen that
\[
\log \det(I_n + A) - \log \det(I_q + T) \leq \log \det(I_n + A) - \log \det(I_q + Q Q^* A)
= \log \det(I_n + A) - \log \det(I_q + Q Q^* A)
= \log \det(I_n + A) - \log \det(I_q + Q Q^* A).
\]

From (3.3), we can check that \( \tilde{Q}_1 = \tilde{Q} \begin{pmatrix} I_k & 0 \\ 0 & 0 \end{pmatrix} \) and \( \begin{pmatrix} I_k \\ 0 \end{pmatrix} \) is orthonormal. Thus, by (3.17) again, we have
\[
\log \det(I_q + Q Q^* A) \geq \log \det(I_q + \tilde{Q}^*_1 A \tilde{Q}_1),
\]
which together with (3.19) gives
\[
\log \det(I_n + A) - \log \det(I_q + T) \leq \log \det(I_k + \tilde{Q}^*_1 A \tilde{Q}_1).
\]

Setting \( M = \tilde{Q}^*_1 A \tilde{Q}_1 \) and considering (3.10), we get
\[
M = (\tilde{R}_1^* \tilde{R}_1)^{-1} (\Lambda_1 + H_1^* A H_1) \tilde{R}_1^{-1}.
\]

Thus, by Lemma 3.4 and noting (3.9), we have
\[
\log \det(I_k + M) = \log \det(I_k + M_1) = \log \det(I_k + M_2),
\]
where
\[
M_1 = (\Lambda_1 + H_1^* A H_1) (\tilde{R}_1^* \tilde{R}_1)^{-1} = (\Lambda_1 + H_1^* A H_1) (I + H_1^* H_1)^{-1},
M_2 = (I + H_1^* H_1)^{-\frac{1}{2}} (\Lambda_1 + H_1^* A H_1) (I + H_1^* H_1)^{-\frac{1}{2}}
\geq (I + H_1^* H_1)^{-\frac{1}{2}} \Lambda_1 (I + H_1^* H_1)^{-\frac{1}{2}} = M_3.
\]
Combining the properties of Löwner partial order [17, Corollary 7.7.4] with (3.20) implies

\[ \log \det(I_k + M) \geq \log \det(I_k + M_3). \]

As done in [26], we can show that \( \log \det(I_k + M_3) = \log \det(I_k + M_4) \), where

\[ M_4 = \Lambda_1^\frac{1}{2} (I + H_1^* H_1)^{-1} \Lambda_1^\frac{1}{2}, \]

and

\[ \log \det(I_k + A_1) - \log \det(I_k + M_4) = \log \det(M_5), \]

where

\[ M_5 = (I_k + M_4)^{-\frac{1}{2}} (I_k + A_1) (I_k + M_4)^{-\frac{1}{2}} = (I_k + M_4)^{-1} + (I_k + M_4)^{-\frac{1}{2}} A_1 (I_k + M_4)^{-\frac{1}{2}}. \]

Further, as done in [26], we have

\[ M_5 = I_k + (I_k + M_4)^{-\frac{1}{2}} (A_1 - M_4) (I_k + M_4)^{-\frac{1}{2}} = I_k + M_6, \]

where

\[ M_6 = (I_k + M_4)^{-\frac{1}{2}} (A_1 - M_4) (I_k + M_4)^{-\frac{1}{2}} \leq (I_k + M_4)^{-\frac{1}{2}} \Lambda_1^\frac{1}{2} H_1^* H_1 A_1^\frac{1}{2} (I_k + M_4)^{-\frac{1}{2}}, \]

and hence

\[ M_5 \leq I_k + (I_k + M_4)^{-\frac{1}{2}} \Lambda_1^\frac{1}{2} H_1^* H_1 A_1^\frac{1}{2} (I_k + M_4)^{-\frac{1}{2}} \leq I_k + \Lambda_1^\frac{1}{2} H_1^* H_1 A_1^\frac{1}{2}. \]

Thus, considering (3.19) and (3.20) and using Lemma 3.4, we have

\[ \log \det(I_n + A) - \log \det(I_q + T) \leq \log \det(I_k + A_1) - \log \det(I_k + M) + \log \det(I_{n-k} + A_2) \leq \log \det(I_k + H_1^* A_1 H_1^*) + \log \det(I_{n-k} + A_2) = \log \det(I_{n-k} + H_1 A_1 H_1^*) + \log \det(I_{n-k} + A_2). \]  

(3.21)

Further, noting \( H_1 \) in (3.2), \( \phi(x) = xf(x) \), and Lemma 3.4, we have

\[ \log \det(I_{n-k} + H_1 A_1 H_1^*) = \log \det \left( I_{n-k} + \phi(A_2) \hat{\Omega}_2 \hat{\Omega}_1 \phi^{-1}(A_1) A_1 \phi^{-1}(A_1) (\hat{\Omega}_2 \hat{\Omega}_1)^* \phi(A_2) \right) \]

\[ = \log \det(I_{n-k} + A_2 f(A_2) \hat{\Omega}_2 \hat{\Omega}_1 f^{-1}(A_1) A_1^{-1} f^{-1}(A_1) (\hat{\Omega}_2 \hat{\Omega}_1)^* A_2 f(A_2)) \]

\[ = \log \det(I_{n-k} + \Lambda_1^\frac{1}{2} f^{-1}(A_1) (\hat{\Omega}_2 \hat{\Omega}_1)^* f(A_2) A_2^\frac{1}{2} A_2 A_2^\frac{1}{2} f(A_2) \hat{\Omega}_2 \hat{\Omega}_1 f^{-1}(A_1) A_1^{-\frac{1}{2}}) \]

\[ = \log \det(I_k + G^* A_2 G), \]  

(3.22)

where

\[ G = \Lambda_2^\frac{1}{2} f(A_2) \hat{\Omega}_2 \hat{\Omega}_1^* f^{-1}(A_1) A_1^{-\frac{1}{2}}. \]
As done in [26], from [16] Theorem 3.3.16, we can derive
\[
\det(I_k + G^* A_2 G) \leq \prod_{j=1}^{n-k} (I_{n-k} + \sigma_j(GG^*) A_2) \leq \prod_{j=1}^{n-k} (I_{n-k} + \|GG^*\|_2 \sigma_j(A_2)) \\
\leq \det(I_{n-k} + \|G\|_2^2 A_2).
\]
Substituting the about result into (3.22) and then into (3.21) gives
\[
\log \det(I_n + A) - \log \det(I_{qI} + T) \\
\leq \log \det(I_{n-k} + \|G\|_2^2 A_2) + \log \det(I_{n-k} + A_2). \quad (3.23)
\]
Further, noting (2.3) and (2.4), we obtain
\[
\|G\|_2^2 = \left\| A_2^{\frac{1}{2}} f(A_2) \tilde{\Omega}_2 \tilde{\Omega}_1 f^{-1}(A_1) A_1^{-\frac{1}{2}} \right\|_2^2 \\
\leq \frac{\lambda_{k+1}}{\lambda_k} \left\| f^{-1}(A_1) \right\|_2^2 \left\| f(A_2) \right\|_2 \left\| \tilde{\Omega}_2 \tilde{\Omega}_1 \right\|_2^2 \\
\leq \frac{\lambda_{k+1}}{\lambda_k} q^{-\frac{1}{2}} \left( \frac{2\lambda_k - \lambda_{k+1}}{\lambda_{k+1}} \right) \left\| \tilde{\Omega}_2 \tilde{\Omega}_1 \right\|_2^2 \\
= \eta,
\]
which together with (3.23) implies the desired upper bound. □

4 Numerical experiments

In this section, we take two examples from [26] to demonstrate the performance of our algorithms and compare them with randomized subspace iteration algorithms given in [26]. We also test the structural upper bounds given Theorems 3.5 and 3.6 by these two examples. In these experiments, the relative errors in trace and log-determinant estimators are defined as
\[
\Delta_t \equiv \frac{\text{Tr}(A) - \text{Tr}(T)}{\text{Tr}(A)}, \quad \Delta_l \equiv \frac{\log \det(I + A) - \log \det(I + T)}{\log \det(I + A)},
\]
and all computations are carried out in MATLAB 2016b.

4.1 Small matrices

The eigenvalues of the test matrix \( A \) satisfy \( \lambda_{j+1} = \tau \lambda_j \) for \( j = 1, 2, \ldots, n-1 \). In contrast to [26], we set the order of the matrix \( A \) to be 1280 × 1280. By setting suitable values of \( \tau \) and \( \lambda_1 \), we do the following four specific numerical experiments.

1. Test the performance of Algorithm 2 when \( p = 20, q = 3, \lambda_1 = 100, \) and \( \tau \) varies from 0.98 to 0.86.
2. Test the performance of Algorithm 2 when \( p = 20, \lambda_1 = 100, \tau = 0.90, \) and \( q \) varies from 1 to 5.
3. Compare Algorithms 1 and 2 when \( p = 20, \lambda_1 = 100, \tau = 0.90, \) and \( q = 3, \).
4. Test the structural error bounds when \( p = 20, q = 3, \lambda_1 = 100, \) and \( \tau = 0.90. \)
The first experiment is used to test the effect of gap on the algorithms. Numerical results are displayed in Fig. 4.1. It is easy to see that both the trace and log-determinant estimators are increasingly accurate as the eigenvalue gap increases. Note that hereafter the relative error is plotted against the sample size $l = k + p$. Since $p$ is fixed, increasing the sample size means to increase in the target rank $k$. As a result, the location of the gap is changing.

Fig. 4.1 Accuracy of (left) trace and (right) log-determinant estimators produced by Algorithm 2 for small matrix with $\tau$ varying from 0.98 to 0.86. The relative error is plotted against the sample size $l = k + p$.

The second experiment is used to test the effect of block Krylov space iteration parameter $q$ on the algorithms. Numerical results are displayed in Fig. 4.2, which shows that the accuracy of both the trace and log-determinant estimators increases as the parameter $q$ increases for a fixed target rank $k$. However, the growth is slowing as $q$ is increasing. As pointed out in [26], this is because the overall error is dominated by $\text{Tr}(A_2)$ and $\log\det(I_{n-k} + A_2)$.

Fig. 4.2 Accuracy of (left) trace and (right) log-determinant estimators produced by Algorithm 2 for small matrix with $q$ varying from 1 to 5. The relative error is plotted against the sample size $l = k + p$. 

In the third experiment, we compare Algorithms 1 and 2 for a special setting on sampling parameter $p$, block Krylov space iteration parameter $q$, and eigenvalue gap $\tau$. Numerical results are displayed in Fig. 4.3 from which we can find that the estimators produced by Algorithm 2 is always more accurate than the corresponding ones produced by Algorithm 1 and the differences increase as the sample size increases. In this experiment, we set $q = 3$. We also do the experiments for $q \geq 2$. The results are similar and the differences for fixed target rank $k$ will increase when $q$ increases. Of course, the two algorithms behave the same when $q = 1$.

![Fig. 4.3 Comparisons of Algorithms 1 and 2 for (left) trace and (right) log-determinant estimators for small matrix. The relative error is plotted against the sample size](image)

In the fourth experiment, we test the accuracy of our structural error bounds. Specifically, we compare the error bounds (3.5), (3.6), and (3.16) with the corresponding best error bounds $\text{Tr}(\Lambda_2)$ and $\log \det(I_{n-k} + \Lambda_2)$. It should be clarified that the results displayed in Fig. 4.4 are all relative error bounds. That is, they are divided by $\text{Tr}(A)$ and $\log \det(I_n + A)$, respectively. These results suggest that our bounds are effective. Especially, the bounds for trace estimator are qualitatively similar to the best error bound and the bound (3.5) is also quantitatively within a factor of 10 of the best error bound. For log-determinant estimator, the differences between the error bound (3.16) and the best bound become a little larger when sample size increases.
4.2 Medium sized matrices

The test matrix $A$ is defined as

$$A \equiv \sum_{j=1}^{40} h_j x_j x_j^T + \sum_{j=41}^{300} l_j x_j x_j^T.$$  \hspace{1cm} (4.1)

In contrast to [26], we set the dimension of the sparse vectors $x_j$ with random non-negative entries to be 20000. So the matrix $A$ is of size $20000 \times 20000$. Note that $x_j$ are not orthonormal. They are generated by the Matlab command $x_j = sprand(20000, 1, 0.025)$. By setting suitable values of $h$ and $l$, we do the following four specific numerical experiments.

1. Test the performance of Algorithm 2 when $p = 20$, $h = 10$, $l = 1$, and $q$ varies from 1 to 5.
2. Test the performance of Algorithm 2 when $p = 20$, $h = 1000$, $l = 1$, and $q$ varies from 1 to 5.
3. Compare Algorithms 1 and 2 when $p = 20$, $q = 3$, $h = 10$, and $l = 1$.
4. Compare Algorithms 1 and 2 when $p = 20$, $q = 3$, $h = 1000$, and $l = 1$.

Numerical results of these experiments are displayed in Figs. 4.5–4.8, respectively, which show the similar results found in the experiments in Section 4.1. That is, the accuracy of both estimators increases as the parameter $q$ increases for a fixed target rank $k$, the growth is slowing as $q$ is increasing, and the estimators produced by Algorithm 2 is always more accurate than the corresponding ones produced by Algorithm 1. As pointed out in [26], the accuracy of Algorithm 1 improves considerably around the location of eigenvalue jump, and the larger the jump, the greater the improvement. In contrast, the accuracy of our algorithm is quite high in all the locations. This is mainly because we use the information discarded by Algorithm 1. These information improves the accuracy of estimators greatly.
Fig. 4.5 Accuracy of (left) trace and (right) log-determinant estimators for medium matrix with $h = 10$ and $q$ varying from 1 to 5. The relative error is plotted against the sample size.

Fig. 4.6 Accuracy of (left) trace and (right) log-determinant estimators for medium matrix with $h = 1000$ and $q$ varying from 1 to 5. The relative error is plotted against the sample size.
Randomized block Krylov space methods for trace and log-determinant estimators

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig4_7}
\caption{Comparisons of Algorithms 1 and 2 for (left) trace and (right) log-determinant estimators for medium matrix with $h = 10$. The relative error is plotted against the sample size.}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig4_8}
\caption{Comparisons of Algorithms 1 and 2 for (left) trace and (right) log-determinant estimators for medium matrix with $h = 1000$. The relative error is plotted against the sample size.}
\end{figure}

5 Concluding remarks

In this paper, we present new randomized algorithms for estimating trace and log-determinant of Hermitian positive semi-definite matrices defined implicitly. Numerical experiments show that the performance of the new algorithms is better than that for algorithms given in [26]. We also provide rigorous error bounds for our trace and log-determinant estimators. They are tighter than the corresponding ones from [26] in most of cases.

To achieve the information from $\mathcal{H}_q$, we adopt a popular method [5], that is, we consider an element of $\mathcal{H}_q$: $\phi(A) \Omega$. The method has a drawback that it requires $l \geq k$. The requirement can be relaxed by considering a method from [27]. However, it will be difficult to investigate the expectation and concentration error bounds of estimators for this method. In addition, for concise and comparing with the error bounds in [26]...
directly, we partition $\Lambda$ in (2.1) into two blocks. As done in [10][27], we can partition $\Lambda$ into three blocks or four blocks. That is, we make an artificial gap and consider a cluster of some eigenvalues. The error bounds for this setting will be more flexible compared with the results obtained in this paper. We will consider these problems in the future work.

References

1. Anitescu, M., Chen, J., Wang, L.: A matrix-free approach for solving the parametric Gaussian process maximum likelihood problem. SIAM J. Sci. Comput. 34(1), A240-A262 (2012)
2. Aune, E., Simpson, D.P., Eidsvik, J.: Parameter estimation in high dimensional Gaussian distributions. Stat. Comput. 24(2), 247-263 (2014)
3. Avron, H., Toledo, S.: Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. J. ACM 58(2), Art. 8, (2011)
4. Barry, R.P., Pace, R.K.: Monte Carlo estimates of the log determinant of large sparse matrices. Linear Algebra Appl. 289(1-3), 41-54 (1999)
5. Boutsidis, C., Drineas, P., Kambadur, P., Kontopoulou, E.M., Zouzias, A.: A randomized algorithm for approximating the log determinant of a symmetric positive definite matrix. Linear Algebra Appl. 533, 95-117 (2017)
6. Drineas, P., Ipsen, I.C.F., Kontopoulou E.M., Magdon-Ismail, M.: Structural convergence results for approximation of dominant subspaces from block Krylov spaces. SIAM J. Matrix Anal. Appl. 39(2), 567-586 (2018)
7. Feng, X., Yu, W., Li, Y.: Faster matrix completion using randomized SVD. arXiv: 1810.06860 (2018)
8. Gittens, A., Kambadur, P., Boutsidis, C.: Approximate spectral clustering via randomized sketching. arXiv: 1311.2854 (2013)
9. Golub, G.H., Von Matt, U.: Generalized cross-validation for large-scale problems. J. Comput. Graph. Stat. 6(1), 1-34 (1997)
10. Gu, M.: Subspace iteration randomization and singular value problems. SIAM J. Sci. Comput. 37(3), A1139-A1173 (2015)
11. Haber, E., Horesh, L., Tenorio, L.: Numerical methods for experimental design of large-scale linear ill-posed inverse problems. Inverse Probl. 24(5), 055012 (2008)
12. Halko, N., Martinsson, P.G., Tropp, J.A.: Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. SIAM Rev. 53(2), 217-288 (2011)
13. Han, I., Malioutov, D., Avron, H., Shin, J.: Approximating spectral sums of large-scale matrices using stochastic Chebyshev approximations. SIAM J. Sci. Comput. 39(4), A1558-A1585 (2017)
14. Han, I., Malioutov, D., Shin, J.: Large-scale log-determinant computation through stochastic Chebyshev expansions. In: Proceedings of the 32nd International Conference on Machine Learning, pp. 908-917 (2015)
15. Higham, N.J.: Accuracy and Stability of Numerical Algorithms, 2nd edn. Society for Industrial and Applied Mathematics (SIAM), Philadelphia (2002)
16. Horn, R.A., Johnson, C.R.: Topics in Matrix Analysis. Cambridge University Press, Cambridge (1991)
17. Horn, R.A., Johnson, C.R.: Matrix Analysis, 2nd edn. Cambridge University Press, Cambridge (2013)
18. Hutchinson, M.F.: A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. Commun. Stat.-Simul. Comput. 19(2), 433-450 (1990)
19. Li, R.C., Zhang, L.H.: Convergence of the block Lanczos method for eigenvalue clusters. Numer. Math. 131(1), 83-113 (2015)
20. Lin, L.: Randomized estimation of spectral densities of large matrices made accurate. Numer. Math. 136(1), 183-213 (2017)
21. Musco, C., Musco, C.: Randomized block krylov methods for stronger and faster approximate singular value decomposition. In: Advances in Neural Information Processing Systems 28, pp. 1396-1404 (2015)
22. Ouellette, D.V.: Schur complements and statistics. Linear Algebra Appl. 36, 187295 (1981)
23. Pace, R.K., LeSage, J.P.: Chebyshev approximation of log-determinants of spatial weight matrices. Comput. Stat. Data Anal. 45(2), 179-196 (2004)
24. Roosta-Khorasani, F., Ascher, U.: Improved bounds on sample size for implicit matrix trace estimators. Found. Comput. Math. 15(5), 1187-1212 (2015)
25. Saad, Y.: Numerical Methods for Large Eigenvalue Problems, revised edn. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (2011).
26. Saibaba, A.K., Alexanderian, A., Ipsen, I.C.F.: Randomized matrix-free trace and log-determinant estimators. Numer. Math. 137(2), 353-395 (2017)
27. Yuan, Q., Gu, M., Li, B.: Superlinear convergence of randomized block lanczos algorithm. arXiv:1808.06287 (2018)
28. Zhang, Y., Leith, W.E., Leith, D.J, Walsh, L.: Log-det approximation based on uniformly distributed seeds and its application to Gaussian process regression. J. Comput. Appl. Math. 220(1-2), 198-214 (2008)
29. Zhang, Y., Leith, W.E.: Approximate implementation of the logarithm of the matrix determinant in Gaussian process regression. J. Stat. Comput. Simul. 77(4), 329-348 (2007)