Journal of Computational and Applied Mathematics 425 (2023) 115063

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Optimal sampling algorithms for block matrix multiplication☆

Chengmei Niu, Hanyu Li *

College of Mathematics and Statistics, Chongqing University, Chongqing 401331, PR China

ARTICLE INFO

Article history:
Received 11 December 2021
Received in revised form 7 October 2022

MSC:
68W20

Keywords:
Optimal sampling
Block matrix multiplication
A-optimal design criterion
Two step algorithm
Probability error bound

ABSTRACT

In this paper, we investigate the randomized algorithms for block matrix multiplication from random sampling perspective. Specifically, based on the A-optimal design criterion, we obtain the optimal sampling probabilities and sampling block sizes. To improve the practicability of the block sizes, two modified ones with less computation cost are provided. With respect to the second modified block size, we devise a two step algorithm. Moreover, the probability error bounds for the proposed algorithms are also given. Extensive numerical results show that our methods outperform the state-of-the-art ones given in the literature.

© 2023 Elsevier B.V. All rights reserved.

1. Introduction

As we know, matrix multiplication is a classical problem in numerical linear algebra. The algorithms of this problem are well-known and can be found in any book on matrix computations, see e.g., [1]. However, in the age of big data, these famous algorithms are encountered enormous challenges because of their computation cost. So, some scholars introduced the randomized ideas into matrix multiplication and proposed some randomized algorithms for this problem.

To the best of our knowledge, Cohen and Lewis [2] first applied the randomized idea to approximate matrix multiplication. In 2006, motivated by a fast sampling algorithm for low-rank approximations given in [3], Drineas et al. [4] proposed the now-famous randomized algorithm for matrix multiplication called the BasicMatrixMultiplication algorithm. It picks the outer products using the nonuniform sampling probabilities which are derived from the norms of columns and rows of the involved matrices M and N, respectively, that is, the following probabilities

$$p_i = \frac{\|M^{(i)}\|_2\|N_{(i)}\|_2}{\sum_{i=1}^{n}\|M^{(i)}\|_2\|N_{(i)}\|_2}, \ i = 1, \ldots, n, \tag{1.1}$$

where $M^{(i)}$ denotes the ith column of $M \in \mathbb{R}^{m\times n}$, $N_{(i)}$ stands for the ith row of $N \in \mathbb{R}^{n\times p}$, and $\|\cdot\|_2$ represents the Euclidean norm of a vector. The specific algorithm is given in Algorithm 1. Later, the BasicMatrixMultiplication algorithm was extended to the block version by Wu [5]. That is, a set of submatrices were sampled by using the following sampling probabilities

$$p_k = \frac{\|M^k N_k\|_F}{\sum_{k=1}^{K}\|M^k N_k\|_F}, \ k = 1, \ldots, K, \tag{1.2}$$

☆ The work was supported by the National Natural Science Foundation of China (No. 11671060) and the Natural Science Foundation of Chongqing, China (No. cstc2019jcyj-msxmX0267).

* Corresponding author.
E-mail addresses: chengmeiniu@cqu.edu.cn (C. Niu), hyli@cqu.edu.cn (H. Li).

https://doi.org/10.1016/j.cam.2023.115063
0377-0427/© 2023 Elsevier B.V. All rights reserved.

where $M^k \in \mathbb{R}^{m\times n_k}$ represents the kth block of $M = \begin{bmatrix} M^1 & M^2 & \cdots & M^K \end{bmatrix}$, $N_k \in \mathbb{R}^{n_k\times p}$ symbolizes the kth block of $N^T = \begin{bmatrix} N_1^T & N_2^T & \cdots & N_K^T \end{bmatrix}$, and $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. In 2019, Chang et al. [6] proposed another block version of the BasicMatrixMultiplication algorithm with the following sampling probabilities

$$p_{\mathbb{K}} = \frac{\|\sum_{k\in\mathbb{K}} M^k N_k\|_F}{\sum_{\mathbb{K}'} \|\sum_{k\in\mathbb{K}} M^k N_k\|_F}, \tag{1.3}$$

where $\mathbb{K} \subset \{\mathbb{K}'\}$ and $\mathbb{K}'$ denote the subsets of $\{1, 2, 3, \ldots, K\}$. Recently, the following sampling probabilities,

$$p_k = \frac{\|M^k\|_F \|N_k\|_F}{\sum_{k=1}^K \|M^k\|_F \|N_k\|_F}, \ k = 1, \ldots, K, \tag{1.4}$$

were devised for the block matrix multiplication by Charalambides et al. [7]. They are easier to compute compared with (1.2) and (1.3). In addition, there are some other generalizations of the BasicMatrixMultiplication algorithm [8,9] and some randomized algorithms for matrix multiplication based on random projection [10–12]. In particular, a block diagonal random projection method with different block sizes was developed in [12].

Algorithm 1 BasicMatrixMultiplication Algorithm [4]

Input: $M \in \mathbb{R}^{m\times n}$, $N \in \mathbb{R}^{n\times p}$, the number of sampling $c \in \mathbb{Z}^+$ such that $1 \le c \le n$, and $\{p_i\}_{i=1}^n$ given as (1.1).
Output: $C \in \mathbb{R}^{m\times c}$ and $D \in \mathbb{R}^{c\times p}$.

1. for $t = 1$ to c
   - sample $i_t \in \{1, \cdots, n\}$ with $\Pr(i_t = s) = p_s$, $s = 1, \cdots, n$, independently and with replacement.
   - set $C^{(t)} = \frac{M^{(i_t)}}{\sqrt{cp_{i_t}}}$, and $D_{(t)} = \frac{N_{(i_t)}}{\sqrt{cp_{i_t}}}$.
2. end
3. return C and D.

In this paper, we consider the randomized algorithms for block matrix multiplication based on random sampling further by using the technique of optimal subsampling proposed recently in the field of statistics; see e.g., [13–16]. Specifically, we derive the optimal sampling probabilities and sampling block sizes by the A-optimal design criterion [17], i.e., minimizing the trace of the variance of an estimator. Moreover, unlike [5–7], we do not sample the blocks directly but sample the outer products on each block with the optimal sampling probabilities and sampling block sizes.

The remainder of this paper is organized as follows. The randomized algorithm framework for block matrix multiplication, the optimal sampling probabilities, and the optimal sampling block sizes are presented in Section 2. In Section 3, we modify the block sizes to make them easier to compute and provide a two step algorithm. Furthermore, the probability error bounds of the corresponding algorithms are also given in Sections 2 and 3, respectively. Extensive numerical experiments are shown in Section 4. Finally, we make the concluding remarks of the whole paper.

2. Randomized algorithm and optimal sampling criterion

We first rewrite the product of the block matrices $M \in \mathbb{R}^{m\times n}$ and $N \in \mathbb{R}^{n\times p}$ appearing in Section 1 as follows

$$MN = \sum_{k=1}^K M^k N_k = \sum_{k=1}^K \sum_{i=1}^{n_k} M^{k(i)} N_{k(i)},$$

where $M^{k(i)}$ is viewed as the ith column of the kth block of M and $N_{k(i)}$ is the ith row of the kth block of N. Then, Algorithm 1 is applied to each block. Thus, we have K estimations for the K blocks as follows

$$C^k D_k = \sum_{t=1}^{c_k} C^{k(t)} D_{k(t)} = \sum_{t=1}^{c_k} \frac{M^{k(i_t)} N_{k(i_t)}}{c_k p_{k_{i_t}}}, \ k = 1, \ldots, K,$$

where c_k represents the number of extracted outer products from the kth block, $C^{k(t)} = \frac{M^{k(i_t)}}{\sqrt{c_k p_{k_{i_t}}}}$ and $D_{k(t)} = \frac{N_{k(i_t)}}{\sqrt{c_k p_{k_{i_t}}}}$ with $p_{k_{i_t}}$ being the sampling probability satisfying $\sum_{i=1}^{n_k} p_{k_i} = 1$. Note that these probabilities as well as the sampling block sizes $\{c_k\}_{k=1}^K$ need to be determined later in this section. Therefore, the final estimation is

$$CD = \sum_{k=1}^K C^k D_k = \sum_{k=1}^K \sum_{t=1}^{c_k} C^{k(t)} D_{k(t)} = \sum_{k=1}^K \sum_{t=1}^{c_k} \frac{M^{k(i_t)} N_{k(i_t)}}{c_k p_{k_{i_t}}}.$$

The specific algorithm is presented in Algorithm 2.

Algorithm 2 Sampling Algorithm for Block Matrix Multiplication

Input: $M \in \mathbb{R}^{m\times n}$ and $N \in \mathbb{R}^{n\times p}$ set as in Section 1, $\{n_k\}_{k=1}^K$ such that $\sum_{k=1}^K n_k = n$, $\{c_k\}_{k=1}^K$ with $c_k \in \mathbb{Z}^+$ and $1 \le c_k \le n_k$ such that $\sum_{k=1}^K c_k = c$ for $c \in \mathbb{Z}^+$, and $\{p_{k_i}\}_{i=1}^{n_k}$ with $p_{k_i} \ge 0$ such that $\sum_{i=1}^{n_k} p_{k_i} = 1$ for $k = 1, \cdots, K$.
Output: $C \in \mathbb{R}^{m\times c}$, $D \in \mathbb{R}^{c\times p}$, and CD.

1. for $k \in 1, \cdots, K$ do
   - $[C^k, D_k]$ =BasicMatrixMultiplication$(M^k, N_k, c_k, \{p_{k_i}\}_{i=1}^{n_k})$.
2. end
3. $C = \begin{bmatrix} C^1 & C^2 & \cdots & C^K \end{bmatrix}$, $D^T = \begin{bmatrix} D_1^T & D_2^T & \cdots & D_K^T \end{bmatrix}$.
4. $CD = \sum_{k=1}^K C^k D_k$.
5. return C, D, and CD.

In the following, we discuss the asymptotic properties of the estimation obtained by Algorithm 2. Based on these asymptotic properties and the A-optimal design criterion [17], we can construct the optimal sampling probabilities and sampling block sizes. One condition and two lemmas are first listed as follows, which are necessary for the proof of the main theorem, i.e., Theorem 2.1 below. More specifically, the condition is used to derive the Lyapunov's condition listed in the first lemma, while the lemma is applied to arrive the conclusion in Theorem 2.1. For the second lemma, its main aim is to simplify the proof of Theorem 2.1.

Condition 2.1.

$$\sum_{i=1}^{n_k} \frac{(M_{(h,i)}^k)^2 (N_{k(i,f)})^2}{c_k^2 p_{k_i}} < C_0, \tag{2.1}$$

$$d_1 n_k^{-(1+\alpha_1)} \le p_{k_i} \le d_2 n_k^{-(1+\alpha_1)}, \tag{2.2}$$

$$\ell_1 n_k^{\alpha_2} \le c_k \le \ell_2 n_k^{\alpha_2}, \tag{2.3}$$

where $M_{(h,i)}^k$ with $h = 1, \ldots, m$ and $i = 1, \ldots, n_k$ stand for the elements at the (h, i)-th position of the kth block of M, $N_{k(i,f)}$ with $f = 1, \ldots, p$ and $i = 1, \ldots, n_k$ denote the elements at the (i, f)-th position of the kth block of N, C_0 is a large enough positive constant, $k = 1, \ldots, K$, $0 \le \alpha_1 < 1$, and $0 \le \alpha_2 < 1$.

Remark 2.1. Combining (2.1), (2.2), and (2.3), we can get

$$\sum_{i=1}^{n_k} (M_{(h,i)}^k)^2 (N_{k(i,f)})^2 < C_0 \ell_2^2 n_k^{2\alpha_2} d_2 n_k^{-(1+\alpha_1)} = C_0 \ell_2^2 d_2 n_k^{2\alpha_2 - 1 - \alpha_1},$$

which yields that

$$\alpha_2 > \frac{1}{2}\left(1 + \frac{\ln \sum_{i=1}^{n_k} \frac{(M_{(h,i)}^k)^2 (N_{k(i,f)})^2}{C_0 \ell_2^2 d_2}}{\ln n_k} + \alpha_1\right). \tag{2.4}$$

From the above discussion, we get that (2.1) holds if (2.4) is satisfied. Furthermore, due to $\alpha_1 < 1 + \frac{\ln \sum_{i=1}^{n_k} \frac{(M_{(h,i)}^k)^2 (N_{k(i,f)})^2}{C_0 \ell_2^2 d_2}}{\ln n_k}$, it is straightforward to have $\alpha_2 > \alpha_1$ from (2.4).

In addition, assuming

$$\tau_1 n \le n_k \le \tau_2 n \tag{2.5}$$

with $0 < \tau_1 \le \tau_2$ and $k = 1, \ldots, K$, we can transform (2.2) and (2.3) into

$$d_1 \tau_2^{-(1+\alpha_1)} n^{-(1+\alpha_1)} \le p_{k_i} \le d_2 \tau_1^{-(1+\alpha_1)} n^{-(1+\alpha_1)} \tag{2.6}$$

and

$$\ell_1 \tau_1^{\alpha_2} n^{\alpha_2} \le c_k \le \ell_2 \tau_2^{\alpha_2} n^{\alpha_2}. \tag{2.7}$$

Lemma 2.1 ([18]). *Assume that $X_1, \ldots, X_n$ are independent and identically distributed random variables, which satisfy that each expected value μ_i and variance ρ_i^2 with $i = 1, \ldots, n$ are finite. Set*

$$\rho^2 = \sum_{i=1}^{n} \rho_i^2,$$

then, when the Lyapunov's condition

$$\lim_{n\to\infty} \frac{\sum_{i=1}^{n} \mathrm{E}[|X_i - \mu_i|^3]}{\rho^3} = 0$$

is satisfied, we have

$$\frac{\sum_{i=1}^{n}(X_i - \mu_i)}{\rho} \xrightarrow{L} N(0,1), \ \text{as } n \to \infty,$$

where $\xrightarrow{L}$ denotes the convergence in distribution.

Lemma 2.2. *The matrices C and D constructed by Algorithm 2 satisfy*

$$\mathrm{E}[(CD)_{(h,f)}] = (MN)_{(h,f)}$$

and

$$\mathrm{Var}[(CD)_{(h,f)}] = \sum_{k=1}^{K}\sum_{i=1}^{n_k} \frac{(M^k_{(h,i)})^2 (N_{k(i,f)})^2}{c_k p_{k_i}} - \sum_{k=1}^{K} \frac{((M^k N_k)_{(h,f)})^2}{c_k},$$

where $(CD)_{(h,f)}$ represents the element at the (h,f)-th position of CD, $(MN)_{(h,f)}$ denotes the element at the (h,f)-th position of MN, $h = 1, \ldots, m$, and $f = 1, \ldots, p$.

Proof. The proof can be completed easily along the line of the proof of [4, Lemma 3].

Now we present the asymptotic distribution of the estimation errors of matrix elements.

Theorem 2.1. *Assume that (2.1), (2.2), (2.3), and (2.5) hold, and set*

$$\mu_1 L \le |M_{(h,i)}| \le \mu_2 L, \tag{2.8}$$

$$\mu_1 L \le |N_{(i,f)}| \le \mu_2 L, \tag{2.9}$$

$$\sum_{k=1}^{K} \frac{((M^k N_k)_{(h,f)})^2}{c_k} \le \alpha \sum_{k=1}^{K}\sum_{i=1}^{n_k} \frac{(M^k_{(h,i)})^2 (N_{k(i,f)})^2}{c_k p_{k_i}}, \tag{2.10}$$

where $0 < \mu_1 \le \mu_2$, $L \ge 0$, and $0 \le \alpha < 1$. Then the matrices C and D constructed by Algorithm 2 satisfy

$$\frac{(CD)_{(h,f)} - (MN)_{(h,f)}}{\sigma} \xrightarrow{L} N(0,1), \ \text{as } n \to \infty, \ c \to \infty, \tag{2.11}$$

where

$$\sigma^2 = \sum_{k=1}^{K}\sum_{i=1}^{n_k} \frac{(M^k_{(h,i)})^2 (N_{k(i,f)})^2}{c_k p_{k_i}} - \sum_{k=1}^{K} \frac{((M^k N_k)_{(h,f)})^2}{c_k}.$$

Proof. Note that

$$\begin{aligned}(CD)_{(h,f)} - (MN)_{(h,f)} &= \sum_{k=1}^{K}\sum_{t=1}^{c_k} \Big(\frac{M^{k(i_t)} N_{k(i_t)}}{c_k p_{k_{i_t}}}\Big)_{(h,f)} - \sum_{k=1}^{K}\sum_{i=1}^{n_k} (M^{k(i)} N_{k(i)})_{(h,f)} \\ &= \sum_{k=1}^{K}\sum_{t=1}^{c_k} \Big[\Big(\frac{M^{k(i_t)} N_{k(i_t)}}{c_k p_{k_{i_t}}}\Big)_{(h,f)} - \sum_{i=1}^{n_k} \Big(\frac{M^{k(i)} N_{k(i)}}{c_k}\Big)_{(h,f)}\Big].\end{aligned}$$

Now, let $\eta_{k(t)} = \big(\frac{M^{k(i_t)} N_{k(i_t)}}{c_k p_{k_{i_t}}}\big)_{(h,f)} - \sum_{i=1}^{n_k} \big(\frac{M^{k(i)} N_{k(i)}}{c_k}\big)_{(h,f)}$ with $k = 1, \ldots, K$ and $t = 1, \ldots, c_k$. Thus, based on Lemma 2.2, it is easy to deduce that

$$\mathrm{E}[\eta_{k(t)}] = 0 \tag{2.12}$$

and

$$\mathrm{Var}[\eta_{k(t)}] = \sum_{i=1}^{n_k} \frac{(M^k_{(h,i)})^2 (N_{k(i,f)})^2}{c_k^2 p_{k_i}} - \frac{((M^k N_k)_{(h,f)})^2}{c_k^2} < C_0, \tag{2.13}$$

where (2.13) is from (2.1). Then, considering that $\eta_{k(t)}$ are independent for the given matrices M and N, and noting (2.12), we find that, to prove (2.11), it suffices to show that

$$\lim_{c\to\infty} \frac{\sum_{k=1}^{K}\sum_{t=1}^{c_k} \mathrm{E}[|\eta_{k(t)}|^3]}{\sigma^3} = 0 \tag{2.14}$$

holds, where

$$\sigma^2 = \sum_{k=1}^{K}\sum_{t=1}^{c_k} \mathrm{Var}[\eta_{k(t)}] = \sum_{k=1}^{K}\sum_{i=1}^{n_k} \frac{(M^k_{(h,i)})^2 (N_{k(i,f)})^2}{c_k p_{k_i}} - \sum_{k=1}^{K} \frac{((M^k N_k)_{(h,f)})^2}{c_k}.$$

Now, we prove (2.14). By the basic triangle inequality, we have

$$\begin{aligned}
\sum_{k=1}^{K}\sum_{t=1}^{c_k} \mathrm{E}[|\eta_{k(t)}|^3] &= \sum_{k=1}^{K}\sum_{t=1}^{c_k} \mathrm{E}[|\frac{M^k_{(h,i_t)} N_{k(i_t,f)}}{c_k p_{k_{i_t}}} - \sum_{i=1}^{n_k} \frac{M^k_{(h,i)} N_{k(i,f)}}{c_k}|^3] \\
&\le \sum_{k=1}^{K} \frac{1}{c_k^2} [\sum_{i=1}^{n_k} \frac{|M^k_{(h,i)}|^3 |N_{k(i,f)}|^3}{p_{k_i}^2} + 4(\sum_{i=1}^{n_k} |M^k_{(h,i)}\| N_{k(i,f)}|)^3 \\
&\quad + 3\sum_{i=1}^{n_k} \frac{(M^k_{(h,i)})^2 (N_{k(i,f)})^2}{p_{k_i}} (\sum_{i=1}^{n_k} |M^k_{(h,i)}\| N_{k(i,f)}|)].
\end{aligned}$$

While, combining (2.6), (2.7), (2.8), (2.9), and (2.10), we can get

$$\begin{aligned}
&\frac{\sum_{k=1}^{K} \frac{1}{c_k^2} \sum_{i=1}^{n_k} \frac{|M^k_{(h,i)}|^3 |N_{k(i,f)}|^3}{p_{k_i}^2}}{\sigma^3} \\
&\le \frac{\mu_2^2 L^2 \sum_{k=1}^{K} \sum_{i=1}^{n_k} \frac{(M^k_{(h,i)})^2 (N_{k(i,f)})^2}{c_k^2 p_{k_i}^2}}{(1-\alpha)^{\frac{3}{2}} (\sum_{k=1}^{K} \sum_{i=1}^{n_k} \frac{(M^k_{(h,i)})^2 (N_{k(i,f)})^2}{c_k p_{k_i}})^{\frac{3}{2}}} \quad \text{by (2.8), (2.9), and (2.10)} \\
&\le \frac{\mu_2^2 L^2 (d_1\ell_1)^{-1} \tau_1^{-\alpha_2} \tau_2^{1+\alpha_1} n^{1+\alpha_1-\alpha_2} \sum_{k=1}^{K} \sum_{i=1}^{n_k} \frac{(M^k_{(h,i)})^2 (N_{k(i,f)})^2}{c_k p_{k_i}}}{(1-\alpha)^{\frac{3}{2}} (\sum_{k=1}^{K} \sum_{i=1}^{n_k} \frac{(M^k_{(h,i)})^2 (N_{k(i,f)})^2}{c_k p_{k_i}})^{1+\frac{1}{2}}} \quad \text{by (2.6) and (2.7)} \\
&= \frac{\mu_2^2 L^2 (d_1\ell_1)^{-1} \tau_1^{-\alpha_2} \tau_2^{1+\alpha_1} n^{1+\alpha_1-\alpha_2}}{(1-\alpha)^{\frac{3}{2}} (\sum_{k=1}^{K} \sum_{i=1}^{n_k} \frac{(M^k_{(h,i)})^2 (N_{k(i,f)})^2}{c_k p_{k_i}})^{\frac{1}{2}}} \\
&\le \frac{\mu_2^2}{\mu_1^2 (1-\alpha)^{\frac{3}{2}}} \frac{(d_1\ell_1)^{-1} \tau_1^{-\alpha_2} \tau_2^{1+\alpha_1} n^{1+\alpha_1-\alpha_2}}{(n d_2^{-1} \tau_1^{1+\alpha_1} n^{1+\alpha_1} \ell_2^{-1} \tau_2^{-\alpha_2} n^{-\alpha_2})^{\frac{1}{2}}} \quad \text{by (2.6), (2.7), (2.8), and (2.9)} \\
&= \frac{\mu_2^2}{\mu_1^2 (1-\alpha)^{\frac{3}{2}}} \frac{(d_1\ell_1)^{-1} \tau_1^{-\frac{1+\alpha_1}{2}-\alpha_2} \tau_2^{1+\alpha_1+\frac{\alpha_2}{2}}}{(d_2\ell_2)^{-\frac{1}{2}}} n^{\frac{\alpha_1-\alpha_2}{2}},
\end{aligned} \tag{2.15}$$

which together with $\alpha_2 > \alpha_1$ (see Remark 2.1) implies

$$\lim_{c\to\infty} \frac{\sum_{k=1}^{K} \frac{1}{c_k^2} \sum_{i=1}^{n_k} \frac{|M^k_{(h,i)}|^3 |N_{k(i,f)}|^3}{p_{k_i}^2}}{\sigma^3} = 0.$$

Analogously, we can get

$$\lim_{c\to\infty} \frac{\sum_{k=1}^{K} \frac{1}{c_k^2} (\sum_{i=1}^{n_k} |M^k_{(h,i)}\| N_{k(i,f)}|)^3}{\sigma^3} = 0,$$

$$\lim_{c\to\infty} \frac{\sum_{k=1}^{K} \frac{1}{c_k^2} \sum_{i=1}^{n_k} \frac{(M^k_{(h,i)})^2 (N_{k(i,f)})^2}{p_{k_i}} (\sum_{i=1}^{n_k} |M^k_{(h,i)}\| N_{k(i,f)}|)}{\sigma^3} = 0.$$

Thus, put the above discussions together, we find that the Lyapunov's condition in Lemma 2.1 is satisfied, namely, (2.14) holds. As a result, we gain (2.11).

Remark 2.2. Note that by Theorem 2.1, we can construct the confidence interval for $(CD)_{(h,f)}-(MN)_{(h,f)}$ with $h = 1, \ldots, m$ and $f = 1, \ldots, p$. Whereas, large n leads σ^2 to be prohibitive. Thus, we can use σ_*^2 established by randomized sampling, i.e.,

$$\sigma_*^2 = \sum_{k=1}^{K}\sum_{t=1}^{c_k}\frac{(M^k_{(h,i_t)})^2(N_{k(i_t,f)})^2}{c_k^2 p^2_{k_{i_t}}} - \sum_{k=1}^{K}\frac{1}{c_k^3}(\sum_{t=1}^{c_k}\frac{M^k_{(h,i_t)}N_{k(i_t,f)}}{p_{k_{i_t}}})^2,$$

to replace σ^2.

Combining the A-optimal design criterion [17] and the sum of asymptotic variances of elements, i.e., by minimizing $\sum_{h=1}^{m}\sum_{f=1}^{p}\sigma^2$, we can obtain the optimal sampling probabilities $\{p_{k_i}\}_{i=1}^{n_k}$ with $k = 1, \ldots, K$ and the optimal sampling block sizes $\{c_k\}_{k=1}^{K}$ for Algorithm 2.

Theorem 2.2. *For Algorithm 2, the sum of the asymptotic variances,*

$$\sum_{h=1}^{m}\sum_{f=1}^{p}\sigma^2$$

attains its minimum when

$$p_{k_i}^{OPL} = \frac{\|M^{k(i)}\|_2\|N_{k(i)}\|_2}{\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2}, \quad \text{for } k = 1, \ldots, K \text{ and } i = 1, \ldots, n_k, \tag{2.16}$$

and

$$c_k^{OPL} = c\frac{[(\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2)^2 - \|M^kN_k\|_F^2]^{\frac{1}{2}}}{\sum_{k=1}^{K}[(\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2)^2 - \|M^kN_k\|_F^2]^{\frac{1}{2}}}, \quad \text{for } k = 1, \ldots, K. \tag{2.17}$$

Proof. Considering

$$(\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2)^2 - \|M^kN_k\|_F^2 \geq 0$$

and by the Cauchy–Schwarz inequality, it is easy to get

$$\begin{aligned}
\sum_{h=1}^{m}\sum_{f=1}^{p}\sigma^2 &= \sum_{k=1}^{K}\sum_{i=1}^{n_k}\frac{\|M^{k(i)}\|_2^2\|N_{k(i)}\|_2^2}{c_k p_{k_i}} - \sum_{k=1}^{K}\frac{\|M^kN_k\|_F^2}{c_k}\\
&= \sum_{k=1}^{K}\sum_{i=1}^{n_k}p_{k_i}\sum_{i=1}^{n_k}\frac{\|M^{k(i)}\|_2^2\|N_{k(i)}\|_2^2}{c_k p_{k_i}} - \sum_{k=1}^{K}\frac{\|M^kN_k\|_F^2}{c_k}\\
&\geq \sum_{k=1}^{K}\frac{1}{c_k}(\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2)^2 - \sum_{k=1}^{K}\frac{\|M^kN_k\|_F^2}{c_k}\\
&= \sum_{k=1}^{K}\frac{c_k}{c}\sum_{k=1}^{K}\frac{1}{c_k}[(\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2)^2 - \|M^kN_k\|_F^2]\\
&\geq [\sum_{k=1}^{K}\frac{1}{\sqrt{c}}((\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2)^2 - \|M^kN_k\|_F^2)^{\frac{1}{2}}]^2,
\end{aligned} \tag{2.18}$$

where the equality in the inequality (2.18) holds if and only if

$$p_{k_i} = W_1\|M^{k(i)}\|_2\|N_{k(i)}\|_2$$

for some constant $W_1 \geq 0$, and the equality in the last inequality holds if and only if

$$c_k = W_2[(\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2)^2 - \|M^kN_k\|_F^2]^{\frac{1}{2}}$$

for some $W_2 \geq 0$. Thus, considering $\sum_{k=1}^{K}c_k = c$ and $\sum_{i=1}^{n_k}p_{k_i} = 1$, the desired results are derived.

Remark 2.3. It is not a complicated matter to find that

$$\sum_{h=1}^{m}\sum_{f=1}^{p}\sigma^2 = \sum_{h=1}^{m}\sum_{f=1}^{p}\mathrm{Var}[(CD)_{(h,f)}] = \mathrm{E}[\|MN - CD\|_F^2],$$

hence, the statistical criterion in Theorem 2.2 for getting the optimal sampling probabilities and sampling block sizes is equivalent to the optimization criterion used in [4].

In addition, if we only want to find the optimal sampling probabilities and sampling block sizes, it suffices to calculate the sum of variance of all elements. In this case, Condition 2.1 is not needed because it is mainly used to find the asymptotic distribution in Theorem 2.1.

Remark 2.4. Supposing that

$$v_k\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2 = \|M^kN_k\|_F, \tag{2.19}$$

where $0 \le v_k < 1$ and $\theta_1 \le 1 - v_k^2 \le \theta_2$ with $0 < \theta_1 \le \theta_2 \le 1$ and $k = 1, \ldots, K$, and considering (2.16) and (2.17), the sum of asymptotic variances of elements can be rewritten as

$$\begin{aligned}\sum_{h=1}^{m}\sum_{f=1}^{p}\sigma_{OPL}^2 &= \sum_{k=1}^{K}\sum_{i=1}^{n_k}\frac{\|M^{k(i)}\|_2^2\|N_{k(i)}\|_2^2}{c_k^{OPL}p_{k_i}^{OPL}} - \sum_{k=1}^{K}\frac{\|M^kN_k\|_F^2}{c_k^{OPL}} \\ &= \frac{1}{c}[\sum_{k=1}^{K}(1-v_k^2)^{\frac{1}{2}}\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2]^2.\end{aligned} \tag{2.20}$$

Remark 2.5. Considering (2.5), (2.8), and (2.9), we can deduce that

$$\frac{\mu_1^2}{n_k\mu_2^2} = \frac{\sqrt{mp}\mu_1^2L^2}{n_k\sqrt{mp}\mu_2^2L^2} \le p_{k_i}^{OPL} \le \frac{\sqrt{mp}\mu_2^2L^2}{n_k\sqrt{mp}\mu_1^2L^2} = \frac{\mu_2^2}{n_k\mu_1^2},$$

which indicates that, with $\alpha_1 = 0$, $d_1 = (\frac{\mu_1}{\mu_2})^2$, and $d_2 = (\frac{\mu_2}{\mu_1})^2$, the condition (2.2) holds.

On the other hand, assuming

$$c = \tau_0 n^{\alpha_2} \tag{2.21}$$

with $0 < \tau_0 < 1$, and noting (2.5), (2.8), (2.9), and (2.19), it is easy to get

$$\begin{aligned}c_k^{OPL} &= c\frac{(1-v_k^2)^{\frac{1}{2}}\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2}{\sum_{k=1}^{K}(1-v_k^2)^{\frac{1}{2}}\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2} \quad \textit{by (2.19)} \\ &\le c\frac{\theta_2^{\frac{1}{2}}\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2}{\theta_1^{\frac{1}{2}}\sum_{k=1}^{K}\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2} \\ &\le c\frac{\theta_2^{\frac{1}{2}}n_k\sqrt{mp}\mu_2^2L^2}{\theta_1^{\frac{1}{2}}n\sqrt{mp}\mu_1^2L^2} \quad \textit{by (2.8) and (2.9)} \\ &\le c\frac{\theta_2^{\frac{1}{2}}\mu_2^2\tau_2 n}{\theta_1^{\frac{1}{2}}\mu_1^2 n} = \frac{\theta_2^{\frac{1}{2}}\tau_0\tau_2\mu_2^2}{\theta_1^{\frac{1}{2}}\mu_1^2}n^{\alpha_2} \quad \textit{by (2.5) and (2.21)} \\ &\le \frac{\theta_2^{\frac{1}{2}}\tau_0\tau_2\mu_2^2}{\theta_1^{\frac{1}{2}}\tau_1^{\alpha_2}\mu_1^2}n_k^{\alpha_2}. \quad \textit{by (2.5)}\end{aligned}$$

Similarly, we have

$$c_k^{OPL} \ge \frac{\theta_1^{\frac{1}{2}}\tau_0\tau_1\mu_1^2}{\theta_2^{\frac{1}{2}}\tau_2^{\alpha_2}\mu_2^2}n_k^{\alpha_2}.$$

Thus, for c_k^{OPL}, with $\ell_1 = \frac{\theta_1^{\frac{1}{2}}\tau_0\tau_1\mu_1^2}{\theta_2^{\frac{1}{2}}\tau_2^{\alpha_2}\mu_2^2}$ and $\ell_2 = \frac{\theta_2^{\frac{1}{2}}\tau_0\tau_2\mu_2^2}{\theta_1^{\frac{1}{2}}\tau_1^{\alpha_2}\mu_1^2}$, the condition (2.3) holds.

Next, we present the error bounds of the estimation obtained by Algorithm 2. To make the analysis more general, we consider a set of sampling probabilities $\{p_{k_i}\}_{i=1}^{n_k}$ such that $p_{k_i} \geq \frac{\beta\|M^{k(i)}\|_2\|N_{k(i)}\|_2}{\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2}$ with a positive constant $\beta \leq 1$, which can be named as the nearly optimal sampling probabilities.

Theorem 2.3. *Assume that (2.19) holds, and let* $\varphi = \frac{(\theta_2-\theta_1\beta+\theta_2\theta_1\beta)^{\frac{1}{2}}}{(\theta_2\theta_1)^{1/4}}$ *with* $\beta \leq 1$. *Then, for Algorithm 2 with* $p_{k_i} \geq \frac{\beta\|M^{k(i)}\|_2\|N_{k(i)}\|_2}{\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2}$ *and* $c_k = c_k^{OPL}$, *the sum of the asymptotic variances satisfies*

$$\sum_{h=1}^{m}\sum_{f=1}^{p}\sigma_{OPL}^2 \leq \frac{\varphi^2}{\beta c}\|M\|_F^2\|N\|_F^2.$$

Furthermore, setting $\delta \in (0,1)$ *and* $\eta = \varphi + (\frac{\theta_2}{\theta_1})^{\frac{1}{2}}\sqrt{(8/\beta)\log(1/\delta)}$,

$$\|MN - CD\|_F^2 \leq \frac{\eta^2}{\beta c}\|M\|_F^2\|N\|_F^2 \tag{2.22}$$

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

Proof. Similar to the proof of [4, Theorem 1], we can derive the desired results. The specific proof is presented in Appendix.

3. Modification of the optimal criterion

Note that calculating (2.17) requires to figure out the matrix multiplication M^kN_k. This cost may be prohibitive for massive data. In this section, we develop two low-cost alternatives, $\hat{c}_k$ and $\widetilde{c}_k$, to replace the optimal sampling block size c_k^{OPL} in (2.17). Besides, a two step algorithm is also provided with respect to $\widetilde{c}_k$.

3.1. Modification with adjusting variance

The size $\hat{c}_k$ is derived from a small modification of the proof of Theorem 2.2. That is, we first let

$$\begin{aligned}\sum_{h=1}^{m}\sum_{f=1}^{p}\sigma^2 &= \sum_{k=1}^{K}\sum_{i=1}^{n_k}\frac{\|M^{k(i)}\|_2^2\|N_{k(i)}\|_2^2}{\hat{c}_k p_{k_i}} - \sum_{k=1}^{K}\frac{\|M^kN_k\|_F^2}{\hat{c}_k}\\ &\leq \sum_{k=1}^{K}\sum_{i=1}^{n_k}\frac{\|M^{k(i)}\|_2^2\|N_{k(i)}\|_2^2}{\hat{c}_k p_{k_i}},\end{aligned}$$

and then find two sets $\{\hat{c}_k\}_{k=1}^K$ and $\{p_{k_i}\}_{i=1}^{n_k}$ to make the above upper bound achieve minimum. Similar to the proof of Theorem 2.2, we have

$$\hat{c}_k = c\frac{\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2}{\sum_{k=1}^{K}\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2} \tag{3.1}$$

and $p_{k_i}^{OPL}$ as in (2.16). Obviously, $\hat{c}_k$ is much easier to compute compared with (2.17).

Remark 3.1. It is easy to find that when $\theta_2 = \theta_1$, $\hat{c}_k = c_k^{OPL}$. Moreover, noting (2.5), (2.8), (2.9), and (2.21), and considering the results in Remark 2.5, we gain

$$\frac{\tau_0\tau_1\mu_1^2}{\tau_2^{\alpha_2}\mu_2^2}n_k^{\alpha_2} \leq \hat{c}_k \leq \frac{\tau_0\tau_2\mu_2^2}{\tau_1^{\alpha_2}\mu_1^2}n_k^{\alpha_2},$$

which implies that, for $\hat{c}_k$, with $\ell_1 = \frac{\tau_0\tau_1\mu_1^2}{\tau_2^{\alpha_2}\mu_2^2}$ and $\ell_2 = \frac{\tau_0\tau_2\mu_2^2}{\tau_1^{\alpha_2}\mu_1^2}$, the condition (2.3) holds.

Below we provide the asymptotic distribution of the estimation errors of matrix elements and probability error bound of $\hat{C}\hat{D}$ constructed by putting (2.16) and (3.1) into Algorithm 2. The asymptotic distribution is first given as follows.

Theorem 3.1. *Assume that (2.5), (2.8), (2.9), (2.10), and (2.21) hold. Then, the matrices* $\hat{C}$ *and* $\hat{D}$ *constructed by Algorithm 2 with* $p_{k_i} = p_{k_i}^{OPL}$ *and* $c_k = \hat{c}_k$ *satisfy*

$$\frac{(\hat{C}\hat{D})_{(h,f)} - (MN)_{(h,f)}}{\hat{\sigma}} \xrightarrow{L} N(0,1), \ \text{as } n\to\infty, \ c\to\infty,$$

where

$$\hat{\sigma}^2 = \sum_{k=1}^{K}\sum_{i=1}^{n_k}\frac{(M^k_{(h,i)})^2(N_{k(i,f)})^2}{\hat{c}_k p^{OPL}_{k_i}} - \sum_{k=1}^{K}\frac{((M^kN_k)_{(h,f)})^2}{\hat{c}_k}. \tag{3.2}$$

Proof. From Remark 3.1, we have that the conditions (2.2) and (2.3) hold when $p_{k_i} = p^{OPL}_{k_i}$ and $c_k = \hat{c}_k$. In addition, following the conclusions in Remarks 2.1 and 2.5, we get that, when

$$\alpha_2 > \frac{1}{2}(1 + \frac{\ln\sum_{i=1}^{n_k}\frac{(M^k_{(h,i)})^2(N_{k(i,f)})^2}{c_0\ell_2^2 d_2}}{\ln n_k}), \tag{3.3}$$

the condition (2.1) holds. Thus, the proof can be completed along the line of the proof of Theorem 2.1.

Remark 3.2. From (2.20) and (3.2), it is easy to see that the difference between σ^2_{OPL} and $\hat{\sigma}^2$ lies in the sampling block sizes, c_k^{OPL} and $\hat{c}_k$.

Now, we present the probability error bound of $\hat{C}\hat{D}$.

Theorem 3.2. *Assume that (2.19) holds, and let* $\hat{\varphi} = (1 - \beta(1-\theta_2))^{\frac{1}{2}}$ *with* $\beta \le 1$. *Then, for Algorithm 2 with* $p_{k_i} \ge \frac{\beta\|M^{k(i)}\|_2\|N_{k(i)}\|_2}{\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2}$ *and* $c_k = \hat{c}_k$, *the sum of the asymptotic variances satisfies*

$$\sum_{h=1}^{m}\sum_{f=1}^{p}\hat{\sigma}^2 \le \frac{\hat{\varphi}^2}{\beta c}\|M\|_F^2\|N\|_F^2.$$

Furthermore, setting $\delta \in (0,1)$ *and* $\eta = \hat{\varphi} + \sqrt{(8/\beta)\log(1/\delta)}$,

$$\|MN - \hat{C}\hat{D}\|_F^2 \le \frac{\eta^2}{\beta c}\|M\|_F^2\|N\|_F^2 \tag{3.4}$$

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

Proof. The proof can be completed along the line of the proof of Theorem 2.3.

Remark 3.3. Letting $\theta_2 = \theta_1 < 1$ and $\beta = 1$ in Theorem 3.2, we have

$$\eta = (\theta_2)^{1/2} + \sqrt{(8/\beta)\log(1/\delta)}.$$

In this case, the probability error bound (3.4) is the same as the one in Theorem 2.3.

3.2. Modification with the BasicMatrixMultiplication algorithm

The size $\widetilde{c}_k$ is derived by the BasicMatrixMultiplication algorithm. Specifically, we use $C^{0k}D_{0k}$ constructed by Algorithm 1 with the same sampling size $[c0/K]$ and a set of sampling probabilities $\{p_{0k_i}\}_{i=1}^{n_k}$ to approximate M^kN_k, where $c0$ denotes the total sample size and p_{0k_i} with $i = 1,\dots,n_k$ are allowed to be uniform probabilities or nonuniform probabilities. Considering that $(\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2)^2 - \|C^{0k}D_{0k}\|_F^2 \ge 0$ may not hold,[1] we propose $\widetilde{c}_k$ as follows

$$\widetilde{c}_k = c\frac{|(\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2)^2 - \|C^{0k}D_{0k}\|_F^2|^{\frac{1}{2}}}{\sum_{k=1}^{K}|(\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2)^2 - \|C^{0k}D_{0k}\|_F^2|^{\frac{1}{2}}}.$$

Based on the above idea, we devise a two step algorithm summarized in Algorithm 3.

Remark 3.4. Similar to (2.19), we suppose

$$\hat{v}_k\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2 = \|C^{0k}D_{0k}\|_F, \tag{3.5}$$

[1] The main reason is that $C^{0k}D_{0k}$ may not be a good approximation of M^kN_k if $c0/K$ is not large enough. In this case, the difference may be smaller than zero.

where $0 < \hat{\theta}_1 \le |1 - \hat{v}_k^2| \le \hat{\theta}_2$ with $k = 1, \ldots, K$. Thus, we can get

$$\widetilde{c}_k = c\frac{|1-\hat{v}_k^2|^{\frac{1}{2}}\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2}{\sum_{k=1}^{K}|1-\hat{v}_k^2|^{\frac{1}{2}}\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2} \ge c\frac{\hat{\theta}_1^{\frac{1}{2}}\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2}{\hat{\theta}_2^{\frac{1}{2}}\sum_{k=1}^{K}\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2} = (\frac{\hat{\theta}_1}{\hat{\theta}_2})^{\frac{1}{2}}\hat{c}_k,$$

$$\widetilde{c}_k = c\frac{|1-\hat{v}_k^2|^{\frac{1}{2}}\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2}{\sum_{k=1}^{K}|1-\hat{v}_k^2|^{\frac{1}{2}}\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2} \le c\frac{\hat{\theta}_2^{\frac{1}{2}}\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2}{\hat{\theta}_1^{\frac{1}{2}}\sum_{k=1}^{K}\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2} = (\frac{\hat{\theta}_2}{\hat{\theta}_1})^{\frac{1}{2}}\hat{c}_k.$$

Following the results in Remark 3.1, we find that, when (2.5), (2.8), (2.9), and (2.21) are satisfied, for $\widetilde{c}_k$, with $\ell_1 = \frac{\hat{\theta}_1^{\frac{1}{2}}\tau_0\tau_1\mu_1^2}{\hat{\theta}_2^{\frac{1}{2}}\tau_2^{\alpha_2}\mu_2^2}$ and $\ell_2 = \frac{\hat{\theta}_2^{\frac{1}{2}}\tau_0\tau_2\mu_2^2}{\hat{\theta}_1^{\frac{1}{2}}\tau_1^{\alpha_2}\mu_1^2}$, the condition (2.3) is satisfied.

Algorithm 3 Two Step Algorithm for Block Matrix Multiplication

Input: $M \in \mathbb{R}^{m\times n}$ and $N \in \mathbb{R}^{n\times p}$ set as in Section 1, $\{n_k\}_{k=1}^K$such that $\sum_{k=1}^K n_k = n$, $c \in \mathbb{Z}^+$, $c0 \in \mathbb{Z}^+$ with $1 \le c0 \le c \le n$, and $\{p_{0k_i}\}_{i=1}^{n_k}$ with $p_{0k_i} \ge 0$ such that $\sum_{i=1}^{n_k} p_{0k_i} = 1$ for $k = 1, \cdots, K$.
Output: $\widetilde{C} \in \mathbb{R}^{m\times c}$, $\widetilde{D} \in \mathbb{R}^{c\times p}$, and $\widetilde{C}\widetilde{D}$.
Step 1:

1. for $k \in 1, \cdots, K$ do
   - update $[C^{0k}, D_{0k}]$ =BasicMatrixMultiplication$(M^k, N_k, [c0/K], \{p_{0k_i}\}_{i=1}^{n_k})$.
   - update $p_{k_i} = \frac{\|M^{k(i)}\|_2\|N_{k(i)}\|_2}{\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2}$, $i = 1, \cdots, n_k$.
2. end
3. replace M^kN_k in (2.17) by $C^{0k}D_{0k}$, i.e., $\widetilde{c}_k = c\frac{|(\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2)^2 - \|C^{0k}D_{0k}\|_F^2|^{\frac{1}{2}}}{\sum_{k=1}^{K}|(\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2)^2 - \|C^{0k}D_{0k}\|_F^2|^{\frac{1}{2}}}$.
4. return $\widetilde{c}_k$ and p_{k_i}, for $k = 1, \cdots, K$ and $i = 1, \cdots, n_k$.

Step 2:

1. for $k \in 1, \cdots, K$ do
   - $[\widetilde{C}^k, \widetilde{D}_k]$ =BasicMatrixMultiplication$(M^k, N_k, \widetilde{c}_k, \{p_{k_i}\}_{i=1}^{n_k})$.
2. end
3. $\widetilde{C} = \begin{bmatrix}\widetilde{C}^1 & \widetilde{C}^2 & \cdots & \widetilde{C}^K\end{bmatrix}$, $\widetilde{D}^T = \begin{bmatrix}\widetilde{D}_1^T & \widetilde{D}_2^T & \cdots & \widetilde{D}_K^T\end{bmatrix}$.
4. $\widetilde{C}\widetilde{D} = \sum_{k=1}^K \widetilde{C}^k\widetilde{D}_k$.
5. return $\widetilde{C}$, $\widetilde{D}$, and $\widetilde{C}\widetilde{D}$.

Now, we provide the asymptotic distribution of the estimation errors of matrix elements of $\widetilde{C}\widetilde{D}$ obtained by Algorithm 3.

Theorem 3.3. *To the assumption of Theorem 3.1, add that (3.5) holds. Then, the matrices $\widetilde{C}$ and $\widetilde{D}$ constructed by Algorithm 3 with $p_{k_i} = p_{k_i}^{OPL}$ and $c_k = \widetilde{c}_k$, satisfy*

$$\frac{(\widetilde{C}\widetilde{D})_{(h,f)} - (MN)_{(h,f)}}{\widetilde{\sigma}} \xrightarrow{L} N(0,1), \ \text{as } n \to \infty, \ c \to \infty,$$

where

$$\widetilde{\sigma}^2 = \sum_{k=1}^{K}\sum_{i=1}^{n_k}\frac{(M_{(h,i)}^k)^2(N_{k(i,f)})^2}{\widetilde{c}_k p_{k_i}^{OPL}} - \sum_{k=1}^{K}\frac{((M^kN_k)_{(h,f)})^2}{\widetilde{c}_k}. \tag{3.6}$$

Proof. Following the discussions in Remarks 2.5 and 3.4, we obtain that, when $p_{k_i} = p_{k_i}^{OPL}$ and $c_k = \widetilde{c}_k$, the conditions (2.2) and (2.3) hold. Besides, when α_2 is as in (3.3), the condition (2.1) also holds. Thus, the proof can be completed along the line of the proof of Theorem 2.1.

Table 1
Description of five experiments.

Number	Comparison	c	K	$c0$ (Algorithm 3)	Results
1	Algorithm 2, UNSSM, and SSM	5×10^4 to 5×10^5	10	Null	Fig. 1
2	Algorithm 2, UNSSM, and SSM	5×10^4	10 to 500	Null	Fig. 2
3	Algorithms 2 and 3	1000 to 5×10^4	10	1000	Fig. 3
4	Algorithms 2 and 3	1×10^4	10 to 500	5000	Fig. 4
5	Algorithms 2 and 3	5000	10	100 to 5×10^4	Fig. 5

Remark 3.5. Analogously, based on (2.20) and (3.6), we also observe that the difference between σ_{OPL}^2 and $\widetilde{\sigma}^2$ also lies in the sampling block sizes, c_k^{OPL} and $\widetilde{c}_k$.

In the following, the probability error bound of $\widetilde{C}\widetilde{D}$ is shown.

Theorem 3.4. *Assume that* (2.19) and (3.5) *hold, and let* $\widetilde{\varphi} = \frac{(\hat{\theta}_2 - \hat{\theta}_1\beta + \theta_2\hat{\theta}_1\beta)^{\frac{1}{2}}}{(\hat{\theta}_2\hat{\theta}_1)^{1/4}}$ *with* $\beta \leq 1$. *Then, for Algorithm 3 with* $p_{k_i} \geq \frac{\beta\|M^{k(i)}\|_2\|N_{k(i)}\|_2}{\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2}$ *and* $c_k = \widetilde{c}_k$, *the sum of the asymptotic variances satisfies*

$$\sum_{h=1}^{m}\sum_{f=1}^{p}\widetilde{\sigma}^2 \leq \frac{\widetilde{\varphi}^2}{\beta c}\|M\|_F^2\|N\|_F^2.$$

Furthermore, setting $\delta \in (0, 1)$ *and* $\eta = \widetilde{\varphi} + (\frac{\hat{\theta}_2}{\hat{\theta}_1})^{\frac{1}{2}}\sqrt{(8/\beta)\log(1/\delta)}$,

$$\|MN - \widetilde{C}\widetilde{D}\|_F^2 \leq \frac{\eta^2}{\beta c}\|M\|_F^2\|N\|_F^2$$

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

Proof. The proof can be completed along the line of the proof of Theorem 2.3.

Remark 3.6. When $\hat{\theta}_2 > 1$, $\hat{\theta}_1 \leq 1$, and $1 - \theta_2 = o(1)$, the bound in Theorem 3.4 is a little weaker than the one in Theorem 3.2. This is because the η in Theorem 3.4 is larger than $1+\sqrt{(8/\beta)\log(1/\delta)}$, while η in Theorem 3.2 is extremely close to $1 + \sqrt{(8/\beta)\log(1/\delta)}$.

Remark 3.7. The bounds in Theorems 2.3, 3.2, and 3.4 are close to the one in [4], which implies that the block sampling with the sampling probabilities and sampling block sizes proposed in this paper can achieve the similar estimation accuracy compared with the direct sampling.

Remark 3.8. For the two alternatives of c_k^{OPL}, i.e., $\hat{c}_k$ and $\widetilde{c}_k$, it is difficult to compare them in theory. Numerical results also show that the algorithms with one alternative cannot be consistently superior to the algorithms with another alternative.

4. Numerical experiments

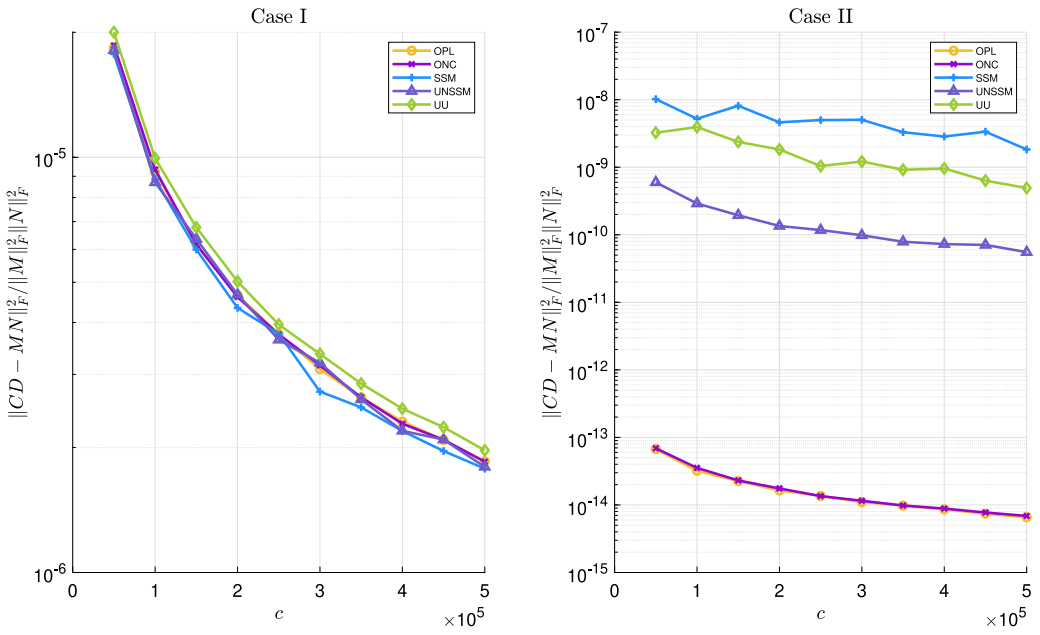
In this section, two kinds of block matrices are used to test our methods. For the first one, the matrix entries are uniform, while the entries of the second kind of matrices are nonuniform. The specific setting is given in the following. Without loss of generality, we set the sizes of the blocks of the involved block matrices M and N to be the same, namely $n_k = n/K$ for $k = 1, \ldots, K$. To construct the following matrices M and N, we let $m = 30$, $p = 50$, $n = 5 \times 10^5$, $\Sigma_1 = (1 \times 0.7^{|i-j|})$ with $1 \leq i, j \leq m$, and $\Sigma_2 = (2 \times 0.7^{|i-j|})$ with $1 \leq i, j \leq p$. As for m, n, and p, their values can be set almost arbitrarily if the basic conditions, i.e., $n \gg m$ and $n \gg p$, hold. The constraints on Σ_1 and Σ_2 are also quite loose. Our specific setting is taken from [16].

Case I: The ith column of M with $1 \leq i \leq m$, $M^{(i)}$, is generated from a multivariate normal distribution, that is, $M^{(i)} \sim N(0, \Sigma_1)$. Similarly, set $N_{(j)} \sim N(0, \Sigma_2)$.

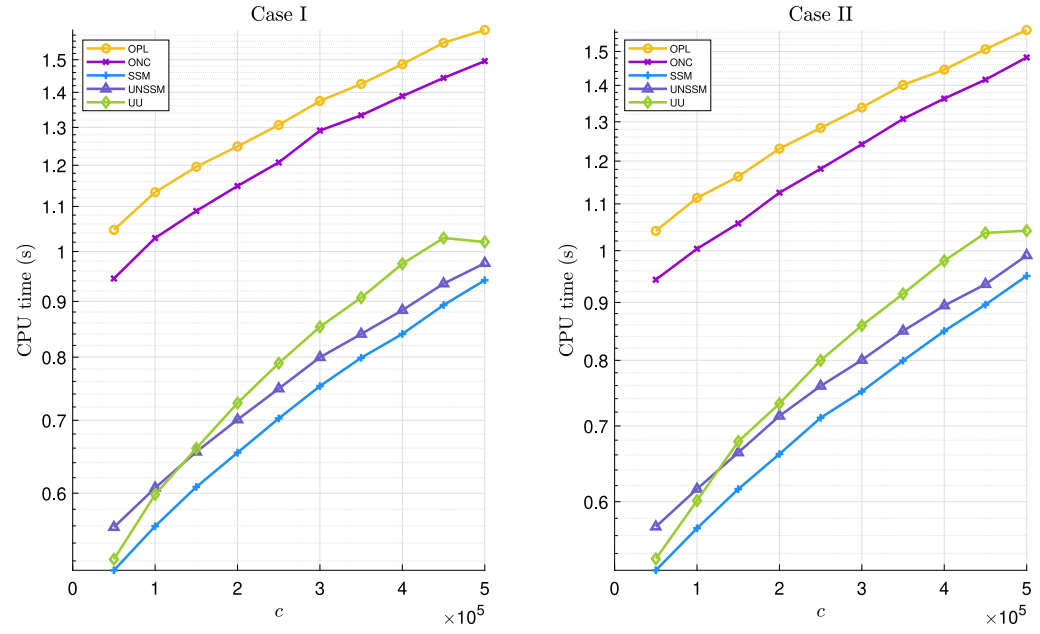
Case II: The ith column of M with $1 \leq i \leq m$, $M^{(i)}$, is generated from a multivariate t distribution with 1 degree of freedom, that is, $M^{(i)} \sim t_1(1, \Sigma_1)$. Similarly, set $N_{(j)} \sim t_1(1, \Sigma_2)$.

For the above matrices, by setting suitable values of K, $c0$, and c, we do five specific experiments summarized in Table 1[2] and report the numerical results in log scale on accuracy, i.e., $\frac{\|CD-MN\|_F^2}{\|M\|_F^2\|N\|_F^2}$, and CPU time in Figs. 1–5. Note that all

[2] To make the cases be diverse, we set the values of K, $c0$, and c to be quite different in different experiments. For the specific values of the parameters, the constraints are quite loose, and they can even be set arbitrarily. Of course, some extreme cases, e.g., the case on c and $c0$ being very small, the case on K being very large, etc., should be avoided.



(a) Relative error

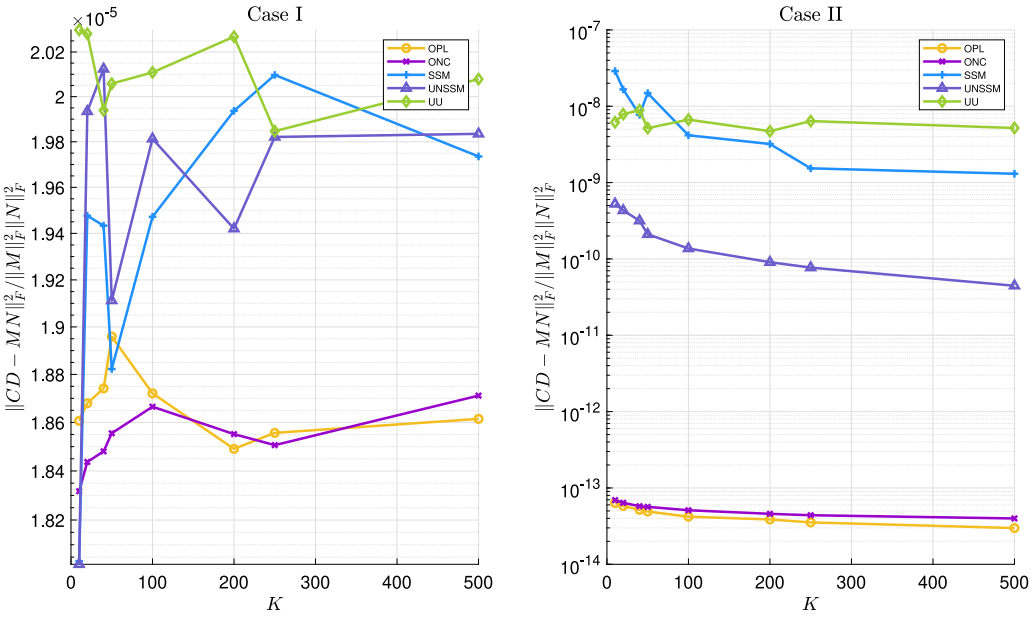


(b) CPU time

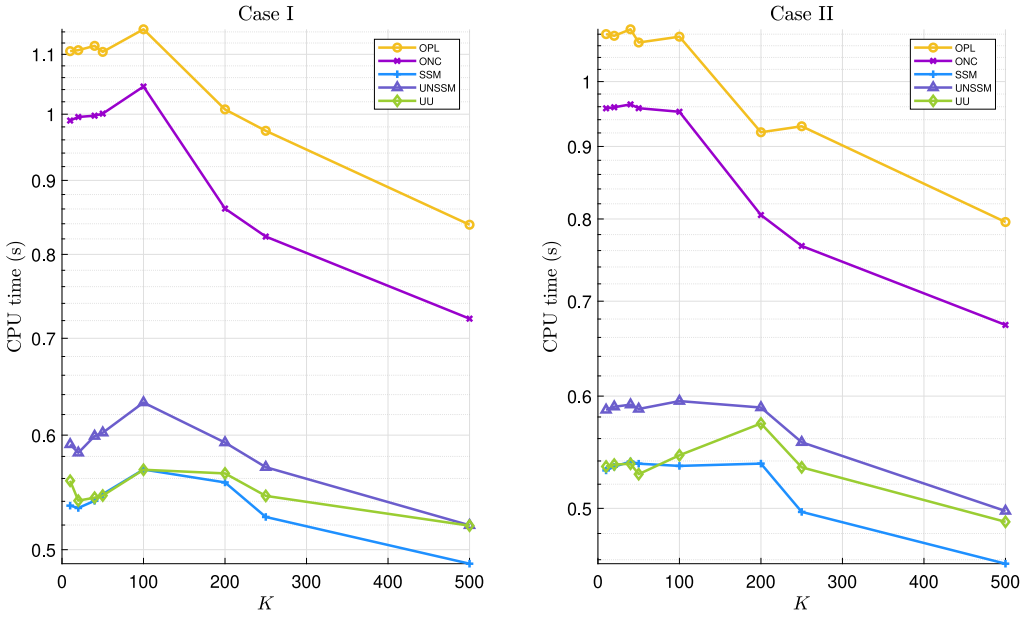
Fig. 1. Comparison of Algorithm 2, UNSSM, and SSM varying with c.

the experiments are implemented on a laptop running MATLAB software with 16 GB random-access memory and Intel Core i5-10210U processor, all the numerical results are based on 100 replications,[3] and in these figures, UNSSM represents

[3] For the 100 replications, we have used parallel computing by MATLAB's parfor and 4 threads on a single laptop. In addition, all the algorithms are run in the same setting and we do not apply any compiler optimizations.



(a) Relative error



(b) CPU time

Fig. 2. Comparison of Algorithm 2, UNSSM, and SSM varying with K.

the method from [5], whose sampling probabilities are as in (1.2), SSM denotes the method from [7], whose sampling probabilities are given in (1.4), and other notations, i.e., ONU, ONMCNR, OPL, ONC, and UU, describe Algorithms 2 or 3 with different p_{k_i}, c_k, and p_{0k_i}, respectively; see Table 2 for more details.

In the first two experiments, we compare Algorithm 2 with UNSSM and SSM for different c and K, respectively. The corresponding numerical results are shown in Figs. 1–2. From these figures, we can find that for Case II, OPL and ONC outperform UNSSM and SSM in accuracy for different c or different K, however, they need more computing time. While, the improvement in accuracy is more than the increasement in computing time. For Case I, the four methods have similar

Table 2
Explanation of sampling methods with different sampling probabilities and sampling block sizes.

Method	p_{k_i}	c_k	p_{0k_i}
ONU (from Algorithm 3)	(2.16)	$\widetilde{c}_k$	$\frac{1}{n_k}$
ONMCNR (from Algorithm 3)	(2.16)	$\widetilde{c}_k$	$\frac{\|M^{k(i)}\|_2\|N_{k(i)}\|_2}{\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2}$
OPL (from Algorithm 2)	(2.16)	(2.17)	Null
ONC (from Algorithm 2)	(2.16)	$\hat{c}_k$	Null
UU (from Algorithm 2)	$\frac{1}{n_k}$	$\frac{c}{K}$	Null

performance in accuracy, and OPL and ONC are a little expensive. These findings are consistent with the theoretical results of these methods. Furthermore, it is interesting to find that UU may be superior to SSM in accuracy for Case II.

The third and fourth experiments are utilized to compare Algorithms 2 and 3 for different c and different K, respectively. Based on the numerical results presented in Figs. 3–4, we get that for Case II, OPL always performs best in accuracy but needs the most CPU time in most of cases. For different c, ONMCNR has the similar performance in accuracy to OPL, however, for large K, i.e., small $c0/K$, it has the worst accuracy. This is because when $c0/K$ is very small, $C^{0k}D_{0k}$ may not be a good approximation of M^kN_k and hence $\widetilde{c}_k$ will not be a good alternative of the optimal sampling block size c_k^{OPL}. In this case, the performance of ONMCNR will be unsatisfactory. In addition, ONC always performs quite well. It needs the least CPU time but has the similar accuracy to OPL. For Case I, the four methods perform similarly in accuracy.

In the last experiment, we compare Algorithms 2 and 3 for different $c0$. The corresponding numerical results are shown in Fig. 5. From this figure, it is easy to see that, for Case II, OPL and ONMCNR have the almost identical performance in accuracy for large $c0$, i.e., large $c0/K$, but the latter consumes less CPU time. In addition, as before, ONC always performs quite well. For Case I, the four methods show the similar accuracy for different $c0$.

In a word, for matrices whose row or column norms are nonuniform, OPL performs best in accuracy in all cases but worst in CPU time in most cases. When $c0/K$ is large, OPL and ONMCNR have almost the same performance in accuracy. Furthermore, ONC always performs quite well.

5. Concluding remarks

In this paper, we present the optimal sampling probabilities and sampling block sizes in the randomized sampling algorithm for block matrix multiplication. Modified sampling block sizes and a two step algorithm for reducing the computation cost are also provided. Numerical experiments show that our new methods outperform the UNSSM method in [5] and the SSM method in [7] in accuracy with a little extra computation cost.

It is easy to see that the blocks of the matrices can be regarded as the single matrices scattered at multiple locations. So, the proposed methods are applicable to distributed data and distributed computations and hence should have many potential applications in the age of big data.

Data availability

Data will be made available on request.

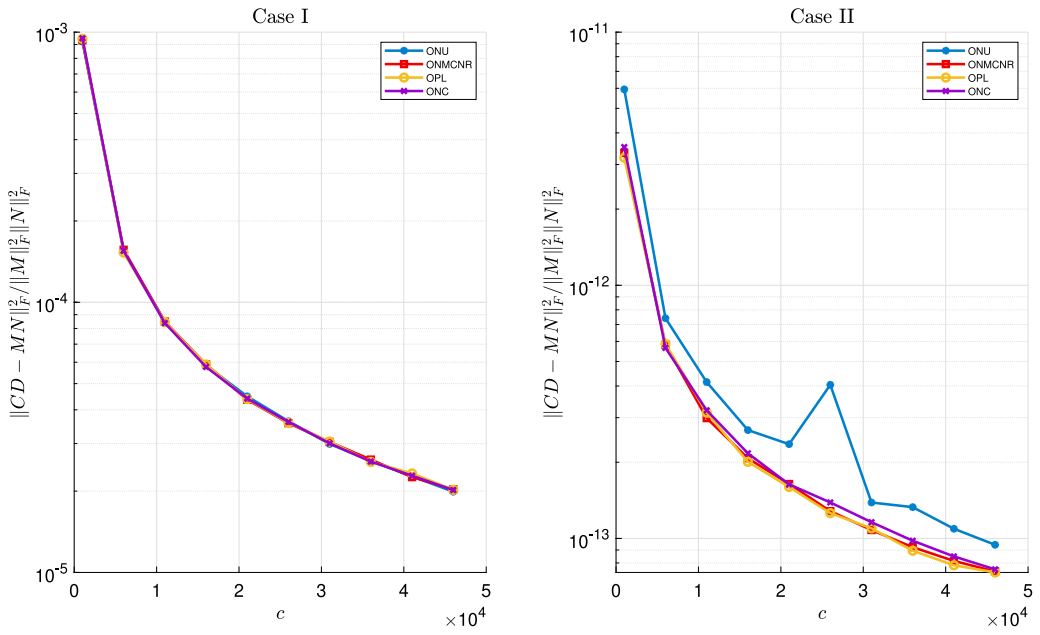
Acknowledgments

The authors would like to thank the editor and the anonymous reviewers for their detailed comments and helpful suggestions which helped considerably to improve the quality of the paper.

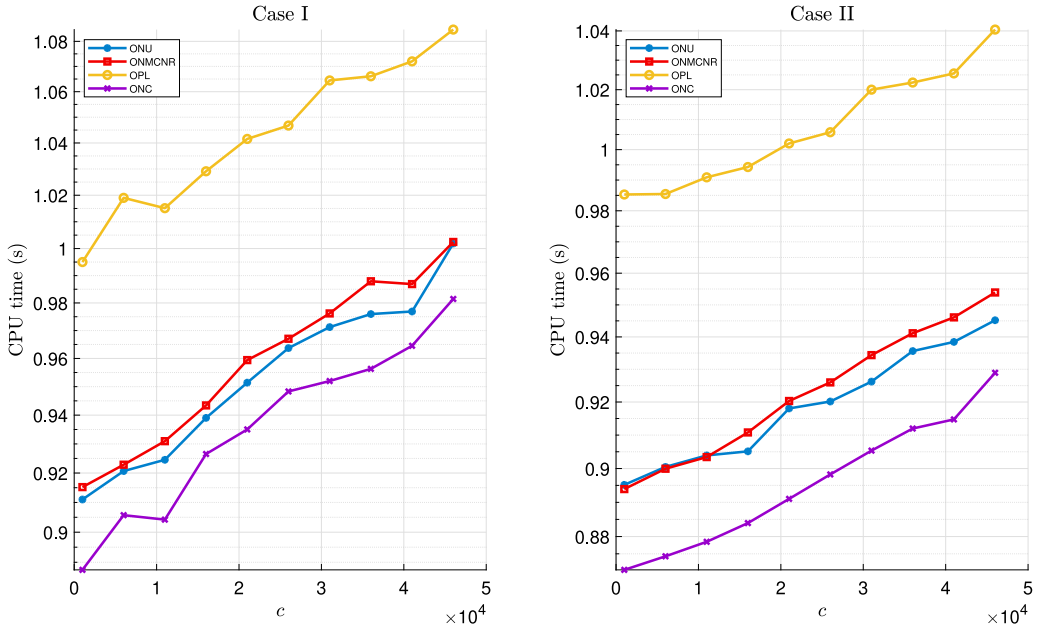
Appendix. Proof of Theorem 2.3

We first deduce that

$$\begin{aligned}\sum_{h=1}^{m}\sum_{f=1}^{p}\sigma_{OPL}^2 &= \sum_{k=1}^{K}\sum_{i=1}^{n_k}\frac{\|M^{k(i)}\|_2^2\|N_{k(i)}\|_2^2}{c_k^{OPL}p_{k_i}} - \sum_{k=1}^{K}\frac{\|M^kN_k\|_F^2}{c_k^{OPL}}\\ &\le \frac{1}{\beta c}\Big(\frac{\theta_2}{\theta_1}\Big)^{\frac{1}{2}}\Big(\sum_{k=1}^{K}\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2\Big)^2 - \frac{(1-\theta_2)}{c}\Big(\frac{\theta_1}{\theta_2}\Big)^{\frac{1}{2}}\Big(\sum_{k=1}^{K}\sum_{i=1}^{n_k}\|M^{k(i)}\|_2\|N_{k(i)}\|_2\Big)^2\\ &\le \frac{\theta_2-\theta_1\beta+\theta_2\theta_1\beta}{\beta c(\theta_2\theta_1)^{1/2}}\|M\|_F^2\|N\|_F^2\\ &= \frac{\varphi^2}{\beta c}\|M\|_F^2\|N\|_F^2,\end{aligned}$$



(a) Relative error



(b) CPU time

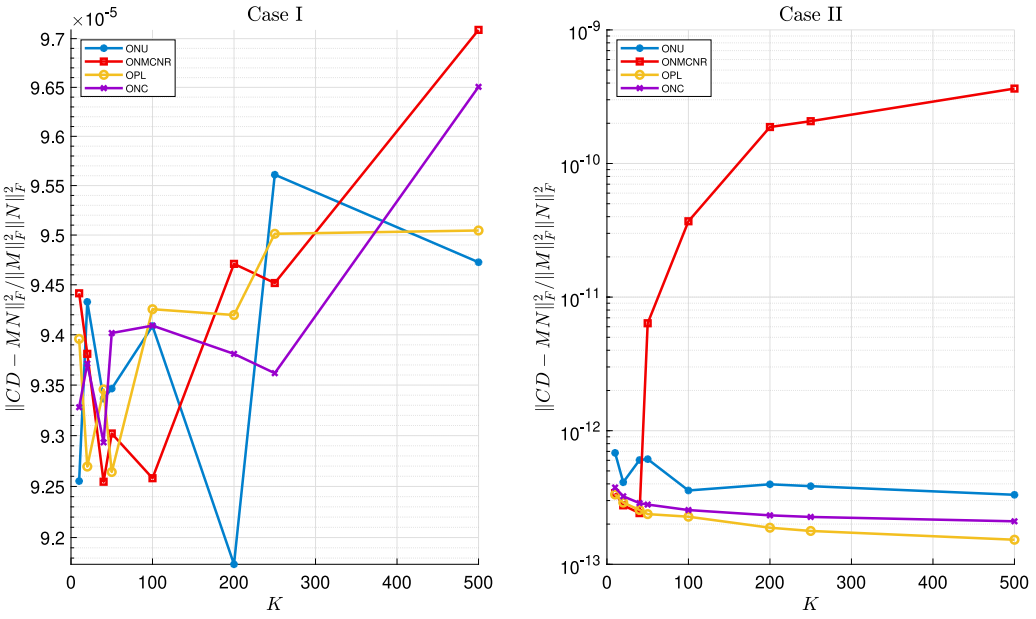
Fig. 3. Comparison of Algorithms 2 and 3 varying with c.

where the first inequality follows from (2.19), and the second inequality is derived from the Cauchy–Schwarz inequality. To prove (2.22), we define a event θ as

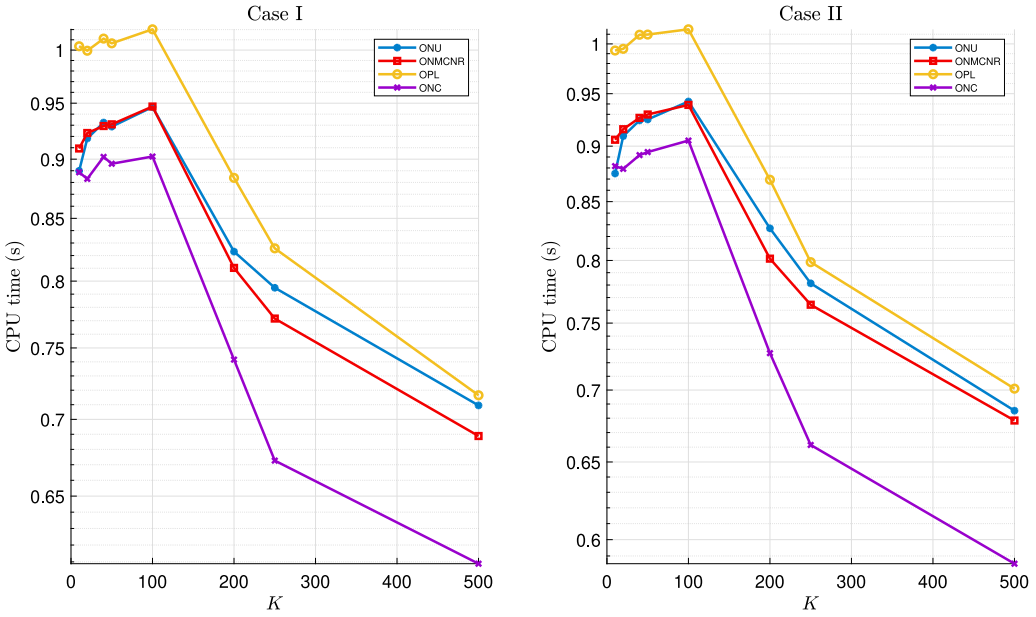
$$\|MN - CD\|_F \leq \frac{\eta}{\sqrt{\beta c}}\|M\|_F\|N\|_F.$$

Thus, as long as getting $\Pr[\theta] \geq 1 - \delta$, (2.22) is proved. To explain easily, we define a function

$$G(x) = \|MN - CD\|_F^2$$



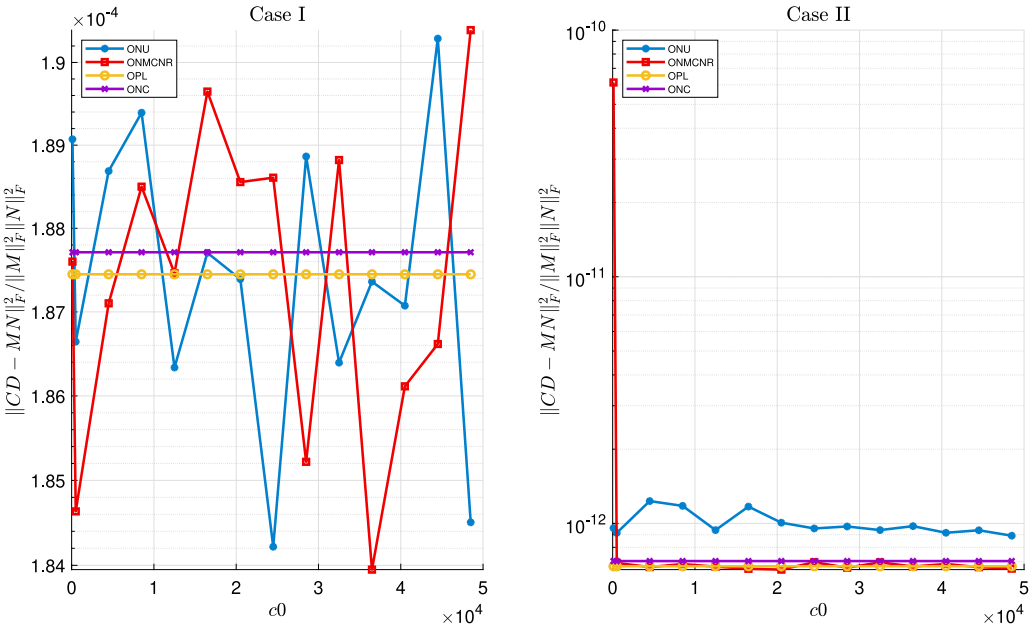
(a) Relative error



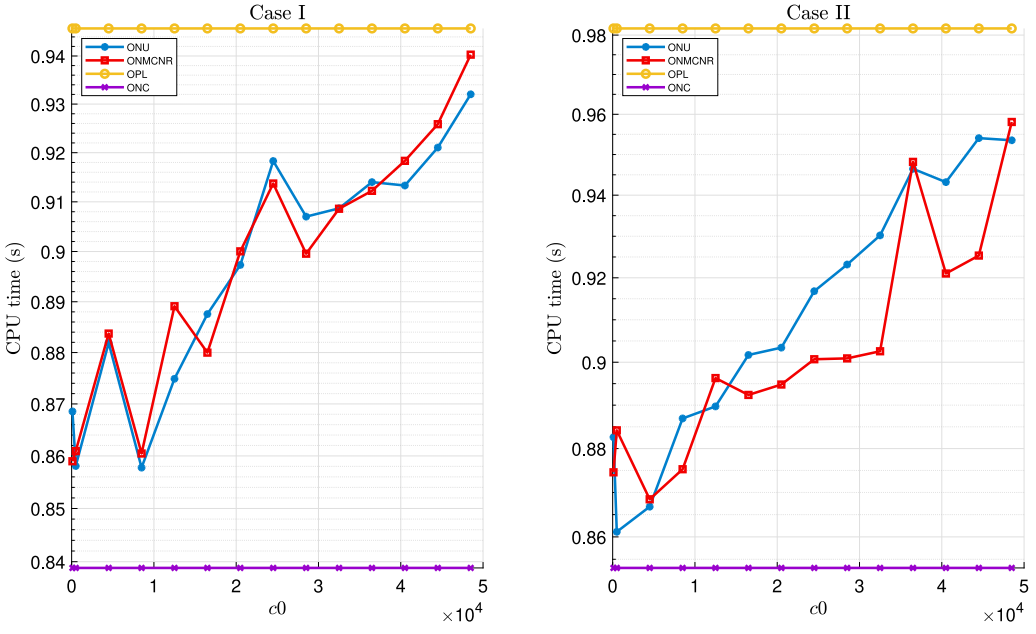
(b) CPU time

Fig. 4. Comparison of Algorithms 2 and 3 varying with K.

with random variable $x = (1(i_1), \ldots, 1(i_{c_1}), 2(i_1), \ldots, 2(i_{c_2}), \ldots, K(i_1), \ldots, K(i_{c_K}))$ standing for the positions of sampled results, where $k(i_t)$ denotes the picked i_t-th column (row) from the kth block of M (N), for $k = 1, \ldots, K$ and $t = 1, \ldots, c_k$. It will be shown that changing one coordinate $k(i_t)$ at a time does not change the value of G too much. Considering x and x' differing only in the $k(i_t)$-th coordinate, we can construct corresponding $\|MN - CD\|_F^2$ and $\|MN - C'D'\|_F^2$, respectively. Note that C' (D') differs from C (D) in only a single column (row). So, based on (2.19) and the Cauchy–Schwarz inequality,



(a) Relative error



(b) CPU time

Fig. 5. Comparison of Algorithms 2 and 3 varying with $c0$.

we have

$$\begin{aligned}\|CD - C'D'\|_F &= \|\frac{M^{k(i_t)}N_{k(i_t)}}{c_k^{OPL}p_{k_{i_t}}} - \frac{M^{k(i_{t'})}N_{k(i_{t'})}}{c_k^{OPL}p_{k_{i_{t'}}}}\|_F \\ &\leq \frac{1}{c_k^{OPL}p_{k_{i_t}}}\|M^{k(i_t)}N_{k(i_t)}\|_F + \frac{1}{c_k^{OPL}p_{k_{i_{t'}}}}\|M^{k(i_{t'})}N_{k(i_{t'})}\|_F\end{aligned}$$

$$\leq \frac{2}{c_k^{OPL}} \frac{\|M^{k(r)} N_{k(r)}\|_F}{p_{k_r}}$$
$$\leq \frac{2}{\beta c}(\frac{\theta_2}{\theta_1})^{\frac{1}{2}} \sum_{k=1}^{K} \sum_{i=1}^{n_k} \|M^{k(i)}\|_2 \|N_{k(i)}\|_2 \quad \text{by (2.19)}$$
$$\leq \frac{2}{\beta c}(\frac{\theta_2}{\theta_1})^{\frac{1}{2}} \|M\|_F \|N\|_F, \quad \text{by the Cauchy–Schwarz inequality}$$

where $\frac{\|M^{k(r)} N_{k(r)}\|_F}{p_{k_r}} = \max_{i_t=1,\ldots,n_k} \frac{\|M^{k(i_t)} N_{k(i_t)}\|_F}{p_{k_{i_t}}}$. Furthermore, since

$$\|MN - CD\|_F \leq \|MN - C'D'\|_F + \|CD - C'D'\|_F$$
$$\leq \|MN - C'D'\|_F + \frac{2}{\beta c}(\frac{\theta_2}{\theta_1})^{\frac{1}{2}} \|M\|_F \|N\|_F$$

and

$$\|MN - C'D'\|_F \leq \|MN - CD\|_F + \|CD - C'D'\|_F$$
$$\leq \|MN - CD\|_F + \frac{2}{\beta c}(\frac{\theta_2}{\theta_1})^{\frac{1}{2}} \|M\|_F \|N\|_F,$$

we have $|G(x) - G(x')| \leq \|CD - C'D'\|_F$. For convenience, let Δ denote $\frac{2}{\beta c}(\frac{\theta_2}{\theta_1})^{\frac{1}{2}} \|M\|_F \|N\|_F$ and $\gamma = \sqrt{2c \log(1/\delta)}\Delta$. Noting the associated Doob martingale, and by Hoeffding-Azuma inequality [19], the probability inequality

$$\Pr[\|MN - CD\|_F \geq \frac{\varphi}{\sqrt{\beta c}} \|M\|_F \|N\|_F + \gamma] \leq \exp(-\frac{\gamma^2}{2c\Delta^2}) = \delta$$

is attained and the theorem follows.

Remark A.1. Letting $\theta_2 = \theta_1 < 1$ and $\beta = 1$, we have $\eta = (\theta_2)^{\frac{1}{2}} + \sqrt{8 \log(1/\delta)}$ in Theorem 2.3. It is smaller than the one in [4, Theorem 1], i.e., $\eta = 1 + \sqrt{8 \log(1/\delta)}$. This is because when computing the upper bound of $\sum_{h=1}^{m} \sum_{f=1}^{p} \sigma^2$, we do not throw away the second item $-\sum_{k=1}^{K} \frac{\|M^k N_k\|_F^2}{c_k}$.

References

[1] G.H. Golub, C.F. Van Loan, Matrix Computations, fourth ed., The Johns Hopkins University Press, Baltimore, MD, 2013.
[2] E. Cohen, D.D. Lewis, Approximating matrix multiplication for pattern recognition tasks, J. Algorithms 30 (2) (1999) 211–252, http://dx.doi.org/10.1006/jagm.1998.0989.
[3] A. Frieze, R. Kannan, S. Vempala, Fast Monte-Carlo algorithms for finding low-rank approximations, J. ACM 51 (6) (2004) 1025–1041, http://dx.doi.org/10.1145/1039488.1039494.
[4] P. Drineas, R. Kannan, M.W. Mahoney, Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication, SIAM J. Comput. 36 (1) (2006) 132–157, http://dx.doi.org/10.1137/S0097539704442684.
[5] Y. Wu, A note on random sampling for matrix multiplication, 2018, arXiv preprint arXiv:1811.11237.
[6] W.-T. Chang, R. Tandon, Random sampling for distributed coded matrix multiplication, in: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2019, pp. 8187–8191, http://dx.doi.org/10.1109/ICASSP.2019.8682895.
[7] N. Charalambides, M. Pilanci, A.O. Hero, Approximate weighted *CR* coded matrix multiplication, in: ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, 2021, pp. 5095–5099, http://dx.doi.org/10.1109/ICASSP39728.2021.9413800.
[8] S. Eriksson-Bique, M. Solbrig, M. Stefanelli, S. Warkentin, R. Abbey, I.C.F. Ipsen, Importance sampling for a Monte Carlo matrix multiplication algorithm, with application to information retrieval, SIAM J. Sci. Comput. 33 (4) (2011) 1689–1706, http://dx.doi.org/10.1137/10080659X.
[9] Y. Wu, N. Polydorides, A multilevel Monte Carlo estimator for matrix multiplication, SIAM J. Sci. Comput. 42 (5) (2020) A2731–A2749, http://dx.doi.org/10.1137/19M125604X.
[10] M.B. Cohen, J. Nelson, D.P. Woodruff, Optimal approximate matrix product in terms of stable rank, in: Proceedings of the 43rd International Colloquium on Automata, Languages, and Programming, Vol. 55, ICALP 2016, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2016, pp. 11:1–11:14, http://dx.doi.org/10.4230/LIPIcs.ICALP.2016.11.
[11] A. Eftekhari, H.L. Yap, C.J. Rozell, M.B. Wakin, The restricted isometry property for random block diagonal matrices, Appl. Comput. Harmon. Anal. 38 (1) (2015) 1–31, http://dx.doi.org/10.1016/j.acha.2014.02.001.
[12] R.S. Srinivasa, M.A. Davenport, J. Romberg, Localized sketching for matrix multiplication and ridge regression, 2020, arXiv preprint arXiv:2003.09097.
[13] H. Wang, R. Zhu, P. Ma, Optimal subsampling for large sample logistic regression, J. Amer. Statist. Assoc. 113 (522) (2018) 829–844, http://dx.doi.org/10.1080/01621459.2017.1292914.
[14] P. Ma, X. Zhang, X. Xing, J. Ma, M. Mahoney, Asymptotic analysis of sampling estimators for randomized numerical linear algebra algorithms, in: Proceedings of the 23nd International Conference on Artificial Intelligence and Statistics, vol. 108, PMLR, 2020, pp. 1026–1035.
[15] H. Wang, Y. Ma, Optimal subsampling for quantile regression in big data, Biometrika 108 (1) (2021) 99–112, http://dx.doi.org/10.1093/biomet/asaa043.
[16] H. Zhang, H. Wang, Distributed subdata selection for big data via sampling-based approach, Comput. Statist. Data Anal. 153 (2021) 107072, http://dx.doi.org/10.1016/j.csda.2020.107072.
[17] F. Pukelsheim, Optimal Design of Experiments, Wiley, New York, 1993.
[18] I. Tyurin, A refinement of the remainder in the Lyapunov theorem, Theory Probab. Appl. 56 (4) (2012) 693–696, http://dx.doi.org/10.1137/S0040585X9798572X.
[19] C. McDiarmid, On the method of bounded differences, Surv. Comb. 141 (1) (1989) 148–188.