Efficient Matrix Sketching over Distributed Data

Zengfeng Huang
UNSW
Sydney, Australia
z3515594@unsw.edu.au

Xuemin Lin
UNSW
Sydney, Australia
lxue@cse.unsw.edu.au

Wenjie Zhang
UNSW
Sydney, Australia
zhangw@cse.unsw.edu.au

Ying Zhang
UTS
Sydney, Australia
Ying.Zhang@uts.edu.au

ABSTRACT

A sketch or synopsis of a large dataset captures vital properties of the original data while typically occupying much less space. In this paper, we consider the problem of computing a sketch of a massive data matrix $A \in \mathbb{R}^{n\times d}$, which is distributed across a large number of s servers. Our goal is to output a matrix $B \in \mathbb{R}^{\ell\times d}$ which is significantly smaller than but still approximates A well in terms of *covariance error*, i.e., $\|A^TA - B^TB\|_2$. Here, for a matrix A, $\|A\|_2$ is the spectral norm of A, which is defined as the largest singular value of A. Following previous works, we call B a covariance sketch of A. We are mainly focused on minimizing the communication cost, which is arguably the most valuable resource in distributed computations. We show a gap between deterministic and randomized communication complexity for computing a covariance sketch. More specifically, we first prove a tight deterministic lower bound, then show how to bypass this lower bound using randomization. In *Principle Component Analysis* (PCA), the goal is to find a low-dimensional subspace that captures as much of the variance of a dataset as possible. Based on a well-known connection between covariance sketch and PCA, we give a new algorithm for distributed PCA with improved communication cost. Moreover, in our algorithms, each server only needs to make one pass over the data with limited working space.

CCS Concepts

•**Theory of computation** → **Massively parallel algorithms;** *Sketching and sampling;*

Keywords

Matrix sketching; communication complexity; distributed data

1. INTRODUCTION

Computing a compact sketch or synopsis of the data, then executing queries against the sketch rather than the entire dataset is a classical technique in big data processing. Traditionally, a sketch is computed through a streaming algorithm which makes one pass over the data using limited working space. However, modern massive data is often distributed across a shared-nothing cluster with a large number of servers. In these systems, the communication cost and the number of rounds become the most critical complexity parameters.

In many applications such as textual analysis, machine learning, and image processing, the underlying data sets are represented as large-scale matrices and stored in massively distributed databases. To analyze such large data matrices, exact computations are often infeasible and unnecessary, and thus randomized and approximate methods are widely used. Computing a sketch matrix first, and then doing expensive computations such as Singular Value Decomposition (SVD) on the sketch matrix is also a common approach (e.g. [31, 7, 29]). Therefore, the task is reduced to designing communication- and round-efficient algorithms to compute sketches with required accuracy.

Covariance sketch. Given a matrix $A \in \mathbb{R}^{n\times d}$, a *covariance sketch* of A is another much smaller matrix $B \in \mathbb{R}^{\ell\times d}$ such that the *covariance error* $\|A^TA - B^TB\|_2$ is small, where $\|A\|_2$ denotes the spectral norm of A. Equivalently, the Euclidean norm $\|Ax\|_2$ for any $x \in \mathbb{R}^d$ is approximately preserved by $\|Bx\|_2$. Covariance sketch has a wide range of applications including low-rank approximation, PCA, clustering, anomaly detection, etc. [11, 16, 8, 23, 20, 36]. Due to its importance, computing a covariance sketch has been extensively studied in various computational models e.g., [10, 27, 17, 15, 34, 9]. In this paper, we study the problem of computing a covariance sketch in the distributed model, and its application to PCA and low-rank approximation.

Distributed models. We assume that the rows of the input matrix are initially partitioned into s parts, each of which is held by a distributed server (with no replication of data). The partition can be arbitrary; we do not make any assumption on how the data is partitioned. The s servers can communicate with each other through an inter-connected communication network. The communication is point-to-point, and we call this the *message passing model* in comparison to the *broadcast model* (aka. the *blackboard model* in complexity community), where each message can be seen by all servers. All of the algorithms in this paper only needs a small constant number of rounds (mostly one or two rounds), so the focus of this paper is to characterize the communication complexity, i.e., the amount of data exchanged. While our algorithms work in the message passing model, our communication lower bound holds for the *blackboard model*, and also holds for protocols with any number of rounds. In our model, there is one special server which acts as the central *coordinator*.

PODS'17, May 14-19, 2017, Chicago, IL, USA

DOI: http://dx.doi.org/10.1145/3034786.3056119

We exhibit new algorithms that improve on the communication cost of the best previous algorithms. For instance, assume $s = d$ and we want to compute a covariance sketch with error $\|A\|_F^2/d$, where $\|A\|_F^2$ is the Frobenius norm of A (the sum of squares of the entries of A). The deterministic algorithm of [27] has cost $O(d^3)$, and the cost of using random sampling is also $O(d^3)$ [10], which is the same as the cost of the trivial algorithm. However, our new randomized algorithm can achieve this with communication cost $\tilde{O}(d^{2.5})$. On the other hand, we show that $\Omega(d^3)$ is the lower bound for any deterministic algorithms, and thus separate the randomized and deterministic communication complexity.

Moreover, in our algorithms, each server only needs to make one pass over the data with limited working space. The algorithms follows the same framework: each server first independently computes a local sketch using a streaming algorithm, then all servers run a distributed algorithm on top the local sketches without further access of the original data. Therefore, they are still efficient even when the local input does not fit into the main memory or is received in a streaming fashion. This is similar to the *distributed streaming model* of [18, 19]. In this model, each server processes a stream of items with bounded memory, and when a query is requested, a special player called *coordinator*, who can communicate with the servers, needs to output the answer for the union of the streams. We call these *distributed streaming* algorithms, in comparison to *distributed batch* algorithms if servers needs to access local data multiple times.

1.1 Matrix preliminaries and notations

We always use s for the number of machines, n for the number rows, and d for the dimension of each row. For a d-dimensional vector x, $\|x\|$ is the ℓ_2 norm of x. We use x_i to denote the ith entry of x, and $\mathsf{Diag}(x) \in \mathbb{R}^{d\times d}$ is a diagonal matrix such that the i-th diagonal entry is x_i. Let $A \in \mathbb{R}^{n\times d}$ be a matrix of dimension $n \times d$ with $d \leq n$. We use A_i to denote the i-th row of A, and $a_{i,j}$ for the (i, j)-th entry of A. $\mathsf{rows}(A)$ is the number of rows in A.

We write the (reduced) singular value decomposition of A as $(U, \Sigma, V) = \mathsf{SVD}(A)$, where $A = U\Sigma V^T$, $U \in \mathbb{R}^{n\times d}$ and $V \in \mathbb{R}^{d\times d}$ are orthonormal matrices, and $\Sigma \in \mathbb{R}^{d\times d}$ is a diagonal matrix with nonnegative diagonal values. The diagonal entries of Σ are singular values of A sorted in non-decreasing order; the columns of U and V are called left and right singular vectors respectively. The computation time of standard SVD algorithms is $O(nd^2)$. We use $\|A\|_2$ or $\|A\|$ to denote the Spectral Norm of A, which is the largest singular value of A, and $\|A\|_F$ for the *Frobenius Norm*, which is $\sqrt{\sum_{i,j} a_{i,j}^2}$. For a real symmetric matrix $X \in \mathbb{R}^{d\times d}$, let $\lambda_i(X)$ be the i-th largest eigenvalue of X. We can also characterize the spectral norm of X as

$$\|X\|_2 = \max(|\lambda_1|, |\lambda_d|) = \max_{y:\|y\|=1} |y^T X y|.$$

Hence,

$$\begin{aligned}\|A^TA - B^TB\|_2 &= \max_{x:\|x\|=1} |x^T(A^TA - B^TB)x| \\ &= \max_{x:\|x\|=1} \left|\|Ax\|_2^2 - \|Bx\|_2^2\right|.\end{aligned}$$

Let $\sigma_i(A)$ be the i-th singular values of A in non-decreasing order. We have $\sigma_i^2(A) = \lambda_i(A^TA)$. It is well-known that

$$\|A\|_F^2 = \mathsf{trace}(A^TA) = \sum_i \lambda_i(A^TA) = \sum_i \sigma_i^2(A).$$

For $k \leq \mathsf{rank}(A)$, we will use $[A]_k$ to denote the best rank k approximation of A, i.e.,

$$[A]_k = \mathsf{arg\ min}_{C:\mathsf{rank}(C)\leq k} \|A - C\|_F.$$

We define $[A]_0 = \mathbf{0}$. Given another matrix B with the same number of columns as A, we will use $\pi_B^k(A)$ to denote the right projection of A on the top-k right singular vectors of B, i.e. $\pi_B^k(A) = AVV^T$, where the columns of V are the top-k right singular vectors of B. We use $[A; B]$ to denote the matrix formed by concatenating the rows of A and B.

1.2 Definitions

Given a matrix $A \in \mathbb{R}^{n\times d}$, we want to compute a much smaller matrix $B \in \mathbb{R}^{\ell\times d}$, which approximates A well. We are interested in *covariance sketch* and its application to PCA.

DEFINITION 1. *The covariance error of B with respect to A is defined as $\|A^TA - B^TB\|_2$. For notational convenience, we will also use $\mathsf{coverr}(A, B)$ to denote this.*

A different but related error measure is so called *projection error* or *low rank approximation error* [16].

DEFINITION 2. *The k-projection error of B with respect to A is defined as $\|A - \pi_B^k(A)\|_F^2$, where $\pi_B^k(A)$ is the rank-k matrix resulting from project each row of A onto the subspace spanned by the top-k right singular vectors of B.*

These two error measures are related by the following lemma from [16]. For completeness, we provide a proof in Appendix A.

LEMMA 1 ([16] (MODIFIED)).

$$\|A - \pi_B^k(A)\|_F^2 \leq \|A - [A]_k\|_F^2 + 2k \cdot \|A^TA - B^TB\|_2.$$

It is well-known that if we randomly sample $O(1/\varepsilon^2)$ rows from A with replacement according to probability proportional to the squared norm of each row, and rescale sampled rows appropriately, the resulting matrix has covariance error at most $\varepsilon\|A\|_F^2$ with constant probability [11]. Since many matrices of interest in practice can be well approximated by a matrix with relatively lower rank, i.e., $\|A - [A]_k\|_F^2$ is much smaller than $\|A\|_F^2$, where $[A]_k$ is the best rank-k approximation of A. Hence an error bound in terms of $\|A - [A]_k\|_F^2$ can potentially be much stronger. In addition, a covariance error of $\varepsilon\|A - [A]_k\|_F^2/2k$ directly implies a relative error low rank approximation by Lemma 1.

DEFINITION 3. *We call B an (ε, k)-sketch of A if*

$$\|A^TA - B^TB\|_2 \leq \varepsilon\|A - [A]_k\|_F^2/k.$$

By abusing notation slightly, we say B is an $(\varepsilon, 0)$-sketch if

$$\mathsf{coverr}(A, B) \leq \varepsilon\|A\|_F^2.$$

In *Principle Component Analysis* (PCA), the goal is to find a low-dimensional subspace that captures as much of the variance of a dataset as possible. The following approximate version of PCA with Frobenius norm error is widely studied (see e.g. [25, 5]).

DEFINITION 4. *In the (approximate) PCA problem, given $A \in \mathbb{R}^{n\times d}$, an integer $k \leq \mathsf{rank}(A)$, and $0 < \varepsilon < 1$, the goal is to output a $d \times k$ orthonormal matrix such that*

$$\|A - AVV^T\|_F^2 \leq (1 + \varepsilon)\|A - [A]_k\|_F^2. \tag{1}$$

The columns of V are also known as $(1 + \varepsilon)$-approximate top-k principle components (PCs).

Here V can also be viewed as a type of matrix sketch. By Lemma 1, the top k right singular vectors of any $(\varepsilon/2, k)$-sketch of A satisfy (1). However, a set of approximate PCs does not directly give good covariance error, although it has low projection error by definition. The distributed version studied in [5] requires all servers to output the same answer, while we only require one server (the coordinator) to know the answer. However, the coordinator can broadcast the answer to all servers using $O(skd)$ communication cost, which is dominated by the cost of computing the answer.

Note that, without further restrictions, an (ε, k)-sketch B may have Frobenius norm much larger than the original matrix. For technical reasons, we will always want the Frobenius norm of the sketch matrix B to be bounded: $\|B\|_F^2 \leq \|A\|_F^2 + O(\|A - [A]_k\|_F^2)$.

We study the word/bit complexity of communication costs, and assume each machine word has $O(\log(nd/\varepsilon))$ bits and each entry of the input matrix can be represented by a single machine word. W.l.o.g. we assume the entries are integers of magnitude at most $\mathrm{poly}(nd/\varepsilon)$.

1.3 Previous results

Liberty [27] adapts a well-known algorithm for finding frequent items, the MG algorithm [28], to sketching a matrix, which is called *Frequent Directions* (FD). It computes an (ε, k)-sketch containing only $O(k/\varepsilon)$ rows in one pass, which is deterministic and directly applicable to the distributed setting: each server computes a local (ε, k)-sketch independently using FD and sends it to the coordinator; the coordinator combines these local sketches and compute an (ε, k)-sketch of the combined sketch matrix. It is shown that the result is a (ε, k)-sketch of the input matrix. The communication cost is thus $O(skd/\varepsilon)$ real numbers.[1]

An alternative approach is to use random sampling. The matrix formed by a random sample of $O(1/\varepsilon^2)$ rescaled rows from A has covariance error at most $\varepsilon\|A\|_F^2$ with constant probability [10]. Random sampling can be implemented in the distributed model with communication cost $O(s + d/\varepsilon^2)$. However, it has a quadratic dependent on ε and only gives a weak error bound.

For the PCA problem, the current best algorithm is by Boutsidis et al. [5]. See section 1.4 for more details on the communication bounds.

1.4 Our contributions

In this paper, we obtain distributed streaming algorithms for computing covariance sketch with improved communication cost, and prove a tight deterministic lower bound in the blackboard model. We also improve the communication cost for the distributed PCA problem.

As there is no bound on maximum wordsize required by the original FD of Liberty [27], the communication cost is only in terms of real numbers. We show how to modify it so that the communication cost is $O(skd/\varepsilon)$ words.[2] Note this communication cost is simply s times the size of a single sketch, where s is the number of servers. Since the sketch size $O(kd/\varepsilon)$ was shown to be optimal [35] (up to a log factor) for an (ε, k)-sketch, it seems difficult to reduce this communication cost. Indeed, we show that this is optimal for deterministic algorithms by proving a deterministic communication lower bound of $O(skd/\varepsilon)$ bits.

On the other hand, we propose a new algorithms with communication cost $o(s)\times$sketch size, which is the first improvement over [27]. In particular, for (ε, k)-sketch, our communication cost is $O(sdk + \frac{\sqrt{s}dk}{\varepsilon} \cdot \sqrt{\log d})$ words, while the optimal sketch size is $\Theta(dk/\varepsilon)$ [35]. We achieve this by designing a new algorithm which further compresses the local (ε, k)-sketches computed by FD. Since this new algorithm is applied on sketches, the servers do not need to access the original data in this step. The results are summarized in Table 1.

	$\varepsilon\|A\|_F^2$	$\varepsilon\|A - [A]_k\|_F^2/k$
[27, 16]	$O(sd/\varepsilon)$	$O(skd/\varepsilon)$
Sampling [10]	$O(s + d/\varepsilon^2)$	
New	$O(\frac{\sqrt{s}d}{\varepsilon} \cdot \sqrt{\log d})$	$O(sdk + \frac{\sqrt{s}kd}{\varepsilon} \cdot \sqrt{\log d})$
Deterministic LB	$\Omega(sd/\varepsilon)$	$\Omega(skd/\varepsilon)$

Table 1: Communication costs for covariance sketch. The communication costs our algorithms are in words. The lower bounds are in terms of bits

	Communication
[5]	$O(skd + \frac{sk}{\varepsilon^2} \cdot \min\{d, k/\varepsilon^2\})$
New	$O(skd + \frac{\sqrt{s \log d} \cdot k}{\varepsilon} \cdot \min\{d, k/\varepsilon^2\})$

Table 2: Communication costs for distributed PCA. The communication costs are in words.

Directly applying our (ε, k)-sketch algorithm to the PCA problem (by Lemma 1), the cost is also $O(sdk) + \tilde{O}(\sqrt{s}dk/\varepsilon)$ words. Boutsidis et al. [5] recently gave an algorithm with communication cost $O(sdk + \frac{sk}{\varepsilon^2} \cdot \min\{d, k/\varepsilon^2\})$ words in the distributed batch model, which is the first algorithm that beats the $O(sdk/\varepsilon)$ bound of [22] when d is larger than k/ε^3. Our result is $o(sdk/\varepsilon)$ as long as $d = 2^{o(s)}$. They also showed an $\Omega(sdk)$ low bound if all servers are required to know the answer. When $s \geq \tilde{\Omega}(\frac{1}{\varepsilon^2})$, our cost is $O(skd)$ which is optimal in this setting[3]. But for smaller s, the cost is dominated by the $\tilde{O}(\sqrt{s}dk/\varepsilon)$ term. We then show how to improve this to $\tilde{O}(\sqrt{s}k/\varepsilon \cdot \min\{d, k/\varepsilon^2\})$ by utilizing the algorithm of [5].

In our algorithm, each server can independently computes a matrix $B^{(i)}$ (with little communication), such that $B = [B^{(1)}; \cdots; B^{(s)}]$ is a (ε, k)-sketch of A and the number of rows in B is $O(sk + \frac{\sqrt{s}k}{\varepsilon} \cdot \sqrt{\log d})$. We call B a *distributed covariance sketch*. To solve PCA, we do not need to send B; we could compute the top k singular vectors of B using any distributed algorithm. We show that only approximate singular vectors of B are needed. More precisely, if B is an (ε, k)-sketch of A, then any $(1 + \varepsilon)$-approximate top-k PCs of B is a $(1 + O(\varepsilon))$-approximate answer for A. Therefore, we can run the algorithm of [5] on B to compute the approximate PCs of B, which then solves the PCA problem for A. The communication cost of this combined algorithm is $O(skd) + \tilde{O}(\sqrt{s}k/\varepsilon \cdot \min\{d, k/\varepsilon^2\})$. While the algorithm of [5] needs to access the input multiple times, our combined algorithm is a distributed streaming algorithm (since the algorithm of [5] is only applied on top of a distributed sketch).

We emphasize that the PCA algorithm of [5] works for *arbitrary partition* model, where each server gets a matrix $A^{(i)} \in \mathbb{R}^{n \times d}$ and $A = \sum_{i=1}^{s} A^{(i)}$, while our algorithm only works for row-partition models.

1.5 Other related works

The problem of computing covariance matrix sketch was widely studied in the row-wise update streaming model [27, 16, 15, 34]. Optimal space lower bounds was proved in [35]. In the distributed setting, there was no improvement in communication cost since [27].

[1] Currently there is no word complexity analysis on FD.

[2] Our technique only works for communication cost, and the space usage of each server is still in real numbers.

[3] Broadcasting the answer only needs $O(skd)$ communication.

Ghashami et al [17] studied the problem in the distributed monitoring model. This model is similar to the distributed streaming model but the coordinator needs to track the answer continuously, which is a stronger requirement. It is an interesting question whether our techniques can be used to improve the communication costs of their algorithms. The approximate distributed PCA problem was studied in [14, 22, 25, 26, 4, 5]. Zhang et al. [37] studied the distributed generalized matrix rank problem. Streaming numerical linear algebra problems were studied in [6]. The communication complexity of boolean matrix multiplication in the two-party model was studied in [33]. Li et al. proved multi-party communication lower bounds for several linear algebraic problems, but only hold in the message passing model. Our distributed computation model is similar to the Massively Parallel Communication model (MPC) [2, 3], which is motivated by MapReduce.

2. DETERMINISTIC MATRIX SKETCHING

In this section, we investigate the deterministic communication complexity of computing a covariance sketch in the distributed model. Recall that each server S_i gets a local input matrix $A^{(i)} \in \mathbb{R}^{n_i \times d}$; and the entire input matrix is $A = [A^{(1)}; \cdots; A^{(s)}]$ with $n = \sum_i n_i$ rows.

Frequent Directions. We will use the *Frequent Directions* (FD) algorithm by Liberty [27], denoted as FD. Ghashami and Phillips [16] gave an improved analysis, which is summarized in the following theorem.

THEOREM 1 ([27, 16]). *Given $A \in \mathbb{R}^{n\times d}$, $\mathsf{FD}(A, \varepsilon, k)$ processes A in one pass using $O(dk/\varepsilon)$ working space. It maintains a sketch matrix $B \in \mathbb{R}^{O(k/\varepsilon)\times d}$ such that*

$$\|A^T A - B^T B\|_2 \le \varepsilon \|A - [A]_k\|_F^2/k.$$

The original FD algorithm of [27] is deterministic and has running time $O(ndk/\varepsilon)$ to process an $n \times d$ matrix. More recently, the running time was improved to $\tilde{O}(\mathsf{nnz}(A)k/\varepsilon)$ by [15] based on fast randomized (approximate) singular value decomposition (thus the algorithm of [15] is randomized), where $\mathsf{nnz}(A)$ denotes the number of non-zero entries in A. We remark that the working space of FD is in terms of real numbers rather than words.

One nice property of FD is that it is mergeable [1]. Formally, let $B = \mathsf{FD}(A, \varepsilon, k)$ and $B' = \mathsf{FD}(A', \varepsilon, k)$ for any A and A' with the same number of columns, then it was shown that $C' = \mathsf{FD}([B; B'], \varepsilon, k)$ has the same covariance error as $C = \mathsf{FD}([A; A'], \varepsilon, k)$. By induction, the number of sketches to merge can be arbitrary. This mergeable property makes FD directly applicable to the distributed model, i.e., each server sketches its local matrix independently, and sends the covariance sketch to the coordinator; the coordinator simply combines the sketches using another FD. Therefore, the algorithm is deterministic and works in the distributed streaming model. Note that the total communication cost is $O(skd/\varepsilon)$ real numbers. A real number may take a large number of bits to represent. We will show how to resolve this issue in section 3.3.

THEOREM 2. *There is a deterministic algorithm in the distributed streaming model which computes a sketch matrix with covariance error at most $\varepsilon\|A-[A]_k\|_F^2/k$. The communication cost is $O(sdk/\varepsilon)$ words ,and the space usage of each server is $O(kd/\varepsilon)$ real numbers.*

2.1 Deterministic Lower Bound

In this section we will prove communication lower bound in the s-party number-in-hand model with a shared blackboard. We first provide some preliminaries on multi-party communication complexity.

2.1.1 Rectangle property of communication complexity in the blackboard model

Let Π be any deterministic protocol in this model for some problem f, and let π be a particular transcript of Π (concatenation of all messages). Define ρ to be the subset of all possible inputs for f, which generate the same transcript π under protocol Π. It is well-known [24] that ρ is a combinatorial rectangle. Formally, ρ is a Cartesian product $\rho = B_1 \times B_2 \times \cdots \times B_s$, where B_i is a subset of all possible inputs for player i. For a deterministic protocol, each input always generates the same transcript, therefore Π partitions the set of all possible inputs into a set of combinatorial rectangles $P = \{\rho_1, \rho_2, \cdots, \rho_h\}$, each of which corresponds to a unique transcript. In particular, the protocol cannot distinguish those inputs in each ρ_j—in other words any correct protocol Π should produce a rectangle partition such that all inputs in any rectangle share a common correct output. Since the transcript corresponds to each rectangle is unique, the maximum length among all transcripts (i.e. the communication cost) is at least $\log |P|$.

2.1.2 Proof of the lower bound

In this section, we show a deterministic lower bound of $\Omega(sd/\varepsilon)$ bits for $(\varepsilon, 0)$-sketch, i.e. with covariance error $\varepsilon\|A\|_F^2$. Note that this directly implies a lower bound of $\Omega(skd/\varepsilon)$ bits for (ε, k)-sketch. Put $t = \frac{\sigma}{\varepsilon}$ for constant $\sigma \le 1$ to be determined later. In our lower bound proof, each server S_i gets a $t \times d$ matrix $A^{(i)} \in \{-1, +1\}^{t\times d}$, and thus the total number of possible inputs is 2^{std} and all input matrices have Frobenius norm exactly std. Our goal is to show that if the size of a combinatorial rectangle ρ is larger than $2^{(1-\beta)std}$ for some absolute constant β, then there must be two input matrices A, A' in ρ such that $\|A^T A - A'^T A'\|_2$ is too large, which means they cannot share the same answer. Therefore, the rectangle partition produced by any correct determnisitic protocol cannot contain a rectangle of size above $2^{(1-\beta)std}$, which implies the communication cost is at least $\Omega(\beta std) = \Omega(sd/\varepsilon)$.

LEMMA 2. *There exists a constant $\beta < 1/2$, such that, for any rectangle ρ of size larger than $2^{(1-\beta)std}$, we can find $A, A' \in \rho$ satisfying*

$$\|A^T A - A'^T A'\|_2 \ge \Omega(sd) - st.$$

PROOF. We write $\rho = B_1 \times B_2 \cdots \times B_s$, where $B_i \subseteq \{-1, +1\}^{t\times d}$ for each i, and define $U = \{i \mid |B_i| \ge 2^{(1-2\beta)td}\}$. We have the following simple property.

CLAIM 1. *If $|\rho| \ge 2^{(1-\beta)td}$, then $|U| \ge s/2$.*

PROOF. U is the same as $\{i \mid \log |B_i| \ge (1-2\beta)td\}$. By our assumption, we have

$$\sum_{i=1}^{s} \log |B_i| = \log |\rho| \ge (1-\beta)std.$$

Using the fact that $\log |B_i| \le td$ for all i and an averaging argument, we conclude that $|U| \ge s/2$. □

Note that each B_i is a subset of $\{-1, +1\}^{t\times d}$; we define $B_{i,j}$ be the projection of B_i onto the jth row, i.e.,

$$B_{i,j} = \{b \mid b \text{ is the } j\text{th row of some matrix in } B_i\}.$$

It is not hard to verify that

$$|B_i| \le \prod_{j=1}^{t} |B_{i,j}|,$$

and thus there exists j with $|B_{i,j}| \geq 2^{(1-2\beta)d}$, provided that $|B_i| \geq 2^{(1-2\beta)td}$. For each $i \in U$, we fix such a j_i, and for simplicity we write $Q_i = B_{i,j_i}$. The following lemma is proved in [21], where x is distributed uniformly in $\{-1,+1\}^d$.

LEMMA 3 ([21]). *Assume $L \subseteq \{-1,+1\}^d$ and $|L| \geq 2^{(1-\alpha)d}$ for a sufficient small constant α, then we have*

$$\mathsf{Pr}_x[\max_{y \in L} x^T y \geq 0.2d] \geq 3/4.$$

Applying the above lemma we prove the follow result.

CLAIM 2. *Let $\ell = |U|$. We have*

$$\mathsf{E}_x\left[\sum_{i \in U} \max_{y^{(i)} \in Q_i} (x^T y^{(i)})^2\right] = \Omega(\ell d^2).$$

PROOF. We have $|Q_i| \geq 2^{(1-2\beta)d}$ for each $i \in U$, therefore, by setting β small enough, we can apply Lemma 3 on each Q_i and get

$$\mathsf{Pr}_x\left[\max_{y^{(i)} \in Q_i} x^T y^{(i)} \geq 0.2d\right] \geq 3/4.$$

It follows that

$$\begin{aligned}\mathsf{E}_x[\max_{y \in Q_i}(x^T y^{(i)})^2] &\geq \mathsf{Pr}_x\left[\max_{y^{(i)} \in Q_i} x^T y^{(i)} \geq 0.2d\right] \cdot \Omega(d^2) \\ &= \Omega(d^2).\end{aligned}$$

Hence,

$$\begin{aligned}\mathsf{E}_x\left[\sum_{i \in U} \max_{y^{(i)} \in Q_i} (x^T y^{(i)})^2\right] &= \sum_{i \in U} \mathsf{E}_x\left[\max_{y^{(i)} \in Q_i} (x^T y^{(i)})^2\right] \\ &= \ell \cdot \Omega(d^2).\end{aligned}$$

□

Obviously, for any x

$$\max_{M^{(i)} \in B_i} \|M^{(i)} x\|^2 \geq \max_{y^{(i)} \in Q_i} (x^T y^{(i)})^2.$$

By Claim 2 and the fact $|U| \geq s/2$, it holds

$$\mathsf{E}_x\left[\sum_{i=1}^{s} \max_{M^{(i)} \in B_i} \|M^{(i)} x\|^2\right] = \Omega(sd^2). \tag{2}$$

Let $W^{(i)}$ be any matrix in $B^{(i)}$ for $i = 1, \cdots, s$. According to standard calculation, it can be shown that

$$\mathsf{E}_x\left[\|W^{(i)} x\|^2\right] = td.$$

Therefore, we have

$$\mathsf{E}_x\left[\sum_{i=1}^{s} \|W^{(i)} x\|^2\right] = std.$$

Combined with (2), we get

$$\mathsf{E}_x\left[\sum_{i=1}^{s} \left(\max_{M^{(i)} \in B_i} \|M^{(i)} x\|^2 - \|W^{(i)} x\|^2\right)\right] = \Omega(sd^2) - std.$$

Then there exist a vector $x^* \in \{-1,+1\}^d$ and matrices $M^{(i)} \in B_i$, for $i = 1, \cdots, s$, such that

$$\sum_{i=1}^{s} \|M^{(i)} x^*\|^2 - \sum_{i=1}^{s} \|W^{(i)} x^*\|^2 = \Omega(sd^2) - std. \tag{3}$$

We set $A = [M^{(1)}; \cdots; M^{(s)}]$ and $A' = [W^{(1)}; \cdots; W^{(i)}]$, and obviously $A, A' \in \rho$. From (3), we have

$$\begin{aligned}\|A^T A - A'^T A'\|_2 &= \max_{x \in \mathbb{R}^d} \frac{\left|\|Ax\|^2 - \|A'x\|^2\right|}{\|x\|^2} \\ &\geq \frac{\left|\sum_{i=1}^{s} \|M^{(i)} x^*\|^2 - \sum_{i=1}^{s} \|W^{(i)} x^*\|^2\right|}{\|x^*\|_2^2} \\ &= \Omega(sd) - st,\end{aligned}$$

which proves the lemma. □

Now we are ready to prove our main theorem for deterministic complexity.

THEOREM 3. *Let $A \in \{-1,+1\}^{n \times d}$ be the input matrix which is row-partitioned across s servers. If $1/\varepsilon \leq d$, then the deterministic communication complexity of computing a $(\varepsilon, 0)$-sketch matrix X is $\Omega(sd/\varepsilon)$ bits.*

PROOF. In our hard instance, each server gets a matrix $A^{(i)} \in \{-1,+1\}^{t \times d}$ where $t = \sigma/\varepsilon$ for a sufficiently small constant σ, and thus $n = st$.[4] As a result, $\varepsilon\|A\|_F^2 = \sigma sd$, which is the maximum covariance error of X allowed. Let us consider any correct deterministic protocol Π, which partitions the set of all possible inputs into combinatorial rectangles $P = \{\rho_1, \rho_2, \cdots, \rho_h\}$.

Assume, for some i, $|\rho_i| \geq 2^{(1-\beta)td}$, then by Lemma 2, there exist A and A' in ρ_i such that $\|A^T A - A'^T A'\|_2 = \Omega(sd) - st$. By our assumption, $t \leq \sigma d$, and thus $\|A^T A - A'^T A'\|_2 > 2\varepsilon\|A\|_F^2$ for sufficiently small σ. Let X be the output corresponding to ρ_i. We have

$$\begin{aligned}\|A^T A - X^T X\|_2 + \|A'^T A' - X^T X\|_2 &\geq \|A^T A - A'^T A'\|_2 \\ &> 2\varepsilon\|A\|_F^2.\end{aligned}$$

It implies that the error of X is too large for either A or A', which contradicts the correctness, so $|\rho_i| < 2^{(1-\beta)std}$ for all i. Since the number of all possible inputs is 2^{std}, we have $|P| \geq 2^{\beta std}$, thus the communication cost of Π is at least $\log|P| = \Omega(std) = \Omega(sd/\varepsilon)$ bits. □

Since the problem can be trivially solved with $O(sd^2)$ words of communication, our bound is also tight for the case when $1/\varepsilon \geq d$. The error allowed for an (ε, k)-sketch is $\varepsilon\|A - [A]_k\|_F^2/k \leq \varepsilon\|A\|_F^2/k$, so the above theorem implies a lower bound of $\Omega(skd/\varepsilon)$ bits for (ε, k)-sketch.

3. RANDOMIZED ALGORITHMS

The key step that enables us to bypass the deterministic lower bound is a better randomized algorithm which computes a sketch with covariance error $\varepsilon\|A\|_F^2/k$ for any input matrix A. For this problem, the deterministic lower bound is $\Omega(skd/\varepsilon)$, however, we give a new randomized algorithm with communication cost $O(\frac{\sqrt{s}kd}{\varepsilon} \cdot \sqrt{\log d})$.

We first present our communication-efficient algorithm in distributed model with covariance error $\alpha\|A\|_F^2$. Then show how to use this algorithm as a subroutine to get an efficient algorithm with stronger error bound.

[4]Note that, for general n, we can append 0 rows at the end of each $A^{(i)}$, which will not affect the proof.

3.1 Covariance error $\alpha\|A\|_F^2$

As we have shown, this problem can be solve deterministically with $O(sd/\alpha)$ communication. An alternative approach is to use random sampling [10, 30, 12]. Sampling can be adapted to the distributed setting, however the communication cost is $O(d/\alpha^2)$, which has an undesirable quadratic dependence on $1/\alpha$.

Our new approach has $\tilde{O}(\sqrt{s}d/\alpha)$ communication cost, which is $o(s)$ times the optimal sketch size for $(\alpha, 0)$-sketch (which is $\Theta(d/\alpha)$ [35]). In this section, we give an algorithm with cost in terms of real numbers, then will discuss how to improve this to word complexity in section 3.3. Our approach also performs random sampling, but we sample the rows of an "aggregated" form of the input matrix.

3.1.1 Our algorithm

The core procedure is the *singular-value-sampling* algorithm (SVS), which is presented in Algorithm 1. In this algorithm, given an input matrix A, we first compute its SVD $A = U\Sigma V^T$, and then sample each right singular vector v_j with probability depending on the corresponding (squared) singular value σ_j^2. We use a function $g()$ to characterize the sampling distribution, i.e., $g(\sigma_j^2)$ is the probability to sample jth singular vector. If it is sampled, we rescale v_j by $\frac{\sigma_j}{\sqrt{g(\sigma_j^2)}}$. In our distributed algorithm, each server S_i runs SVS on the input matrix $A^{(i)}$, and then sends the output to the coordinator (Algorithm 2).

Note that, in our algorithm, each server also performs row sampling. The differences between our algorithms and the row sampling algorithms from [10, 30, 12] for covariance sketch are as follows. (1) We sample the rows of the "aggregated" form of the input matrix instead of the original. Let $U\Sigma V^T$ be the singular value decomposition of A. We view $\mathsf{agg}(A) := \Sigma V^T$ as the "aggregated" form of A. (2) Our sampling scheme is different—in previous works, each row of the sketch matrix is an i.i.d. sample from the original matrix (with replacement) and rescaled, but we use independent Bernoulli sampling. Although this seems insignificant, it is actually crucial to our analysis. It is not clear whether a similar bound holds if we use i.i.d. sampling instead.

Algorithm 1: $\mathsf{SVS}(A, g)$: $A \in \mathbb{R}^{n\times d}$; g is the sampling function.

1: Set B empty
2: Compute $(U, \Sigma, V) = \mathsf{SVD}(A)$
3: **for** $j = 0$ **to** d **do**
4: Set $x_j = 1$ with probability $g(\sigma_j^2)$, and $x_j = 0$ with probability $1 - g(\sigma_j^2)$
5: Let $w_j = \sigma_j/\sqrt{g(\sigma_j^2)}$
6: Append v_j^T rescaled by $x_j w_j$ (i.e., $x_j \cdot w_j \cdot V_j^T$) to B, where v_j is the j-th right singular vector
7: Remove zero rows in B
8: Output B

Algorithm 2: Algorithm for S_i. Input: $A^{(i)} \in \mathbb{R}^{n_i\times d}$ and sampling function g.

1: $B^{(i)} = \mathsf{SVS}(A^{(i)}, g)$
2: Send $B^{(i)}$ to the coordinator

Let $B = [B^{(1)}, \cdots, B^{(s)}]$. For technical reasons, we also want the Frobenius norm of the sketch matrix B to be bounded: $\|B\|_F^2 \le O(1) \cdot \|A\|_F^2$. It is quite standard to bound the norm of B for our algorithm (see Appendix B), so we will focus on bounding the covariance error $\|A^TA - B^TB\|_2$. We first prove a theorem for a general sampling function g, then discuss how to pick a good sampling function in the next section.

THEOREM 4. *Let $A^{(i)}$ be the input of the i-th server, and $B^{(i)}$ be matrix sent by S_i. Let A and B be the concatenation of $A^{(i)}$'s and $B^{(i)}$'s respectively. We define*

$$M = \max_{i,j} \frac{\sigma_{i,j}^2}{g(\sigma_{i,j}^2)}, \text{ and } \kappa^2 = \sum_{i=1}^{s} \max_j \frac{\sigma_{i,j}^4 \cdot (1 - g(\sigma_{i,j}^2))}{g(\sigma_{i,j}^2)},$$

where $\sigma_{i,j}$ is the jth largest singular value of $A^{(i)}$, then the following inequality holds:

$$\Pr[\|B^TB - A^TA\|_2 \ge t] \le 2d \cdot \exp\left(\frac{-t^2/2}{\kappa^2 + Mt/3}\right).$$

Before giving the proof, we first briefly summarize the main idea behind the proof.

Main idea. Previous row sampling approaches sample t rows from A using i.i.d. sampling, so, in the analysis, B^TB can be treated as the sum of t i.i.d. random matrices of rank 1. To bound the covariance error, [30, 12] used a matrix concentration inequality, while [10] used a variance argument. On the other hand, in our analysis, we will view B^TB as the sum of s random matrices with potentially high rank, i.e., $\sum_{i=1}^{s} B^{(i)T}B^{(i)}$. Since we sample the rows of the "aggregated" matrix (with orthogonal rows) using Bernoulli sampling, each resulting random matrix $B^{(i)}$ has orthogonal rows, which is important to our analysis. However, the random matrices have high rank and are not i.i.d. now, so we cannot apply the same inequality used in [30, 12]. Instead, we use *Matrix Bernstein Inequality* (see [32]).

The main theorem follows from the following three claims, which are about the output matrix of the SVS sampling algorithm. Let $x = [x_1, \cdots, x_d]$ be a random vector, where x_j is defined in Algorithm 1. More precisely, x_j's are Bernoulli random variables:

$$x_j = \begin{cases} 1 & \text{the } j\text{th singular vector is sampled} \\ 0 & \text{otherwise.} \end{cases}$$

Clearly, the distribution of x_j^2 is the same as x_j, which take value 1 with probability $g(\sigma_j^2)$, where σ_j is the jth largest singular value of A. Let $w \in \mathbb{R}^d$ be a vector, with w_j defined in Algorithm 1, i.e. $w_j = \frac{\sigma_j}{\sqrt{g(\sigma_j^2)}}$.

CLAIM 3. *If A and B be the input and output of Algorithm 1 respectively, then $\mathsf{E}[B^TB] = A^TA$.*

PROOF. Let v_j be the jth column of V. We have

$$A^TA = \sum_{j=1}^{d} \sigma_j^2 v_j v_j^T.$$

By definition, $B = \mathsf{Diag}(x)\mathsf{Diag}(w)V^T$, then we have

$$\begin{aligned} B^TB &= V \cdot \mathsf{Diag}(x)^2\mathsf{Diag}(w)^2 \cdot V^T \\ &= \sum_{j=1}^{d} x_j^2 w_j^2 v_j v_j^T \end{aligned} \quad (4)$$

So we have

$$\begin{aligned}\mathsf{E}[B^TB] &= \mathsf{E}[\sum_{j=1}^{d} x_j^2 w_j^2 v_j v_j^T] \\ &= \sum_{j=1}^{d} \mathsf{E}[x_j^2] w_j^2 v_j v_j^T \\ &= \sum_{j=1}^{d} \sigma_j^2 v_j v_j^T = A^TA.\end{aligned}$$

□

CLAIM 4. *If A and B are the input and output of Algorithm 1 respectively, then we have*

$$\lambda_{\max}(B^TB - A^TA) \le \max_j \frac{\sigma_j^2}{g(\sigma_j^2)}.$$

PROOF. Using Eq (4), we have

$$B^TB - A^TA = \sum_{j=1}^{d} (x_j^2 w_j^2 - \sigma_j^2) v_j v_j^T = VDV^T, \qquad (5)$$

where D is a diagonal matrix with $D_{j,j} = x_j^2 w_j^2 - \sigma_j^2$. Since V is orthonormal, VDV^T is the eigen-decomposition of $B^TB - A^TA$, and thus

$$\begin{aligned}\lambda_{\max}(B^TB - A^TA) &= \max_j (x_j^2 w_j^2 - \sigma_j^2) \\ &\le \max w_j^2 = \max_j \frac{\sigma_j^2}{g(\sigma_j^2)}.\end{aligned}$$

□

CLAIM 5. *If A and B are the input and output of Algorithm 1 respectively, then we have*

$$\|\mathsf{E}[(B^TB - A^TA)^2]\|_2 = \max_j \frac{\sigma_j^4 \cdot (1 - g(\sigma_j^2))}{g(\sigma_j^2)}.$$

PROOF. From Eq (5), we have

$$(B^TB - A^TA)^2 = VD^2V^T = \sum_{j=1}^{d} (x_j^2 w_j^2 - \sigma_j^2)^2 \cdot v_j v_j^T.$$

By definition, we have $\mathsf{E}[x_j^2 w_j^2] = \sigma_j^2$, and thus

$$\begin{aligned}\mathsf{E}[(x_j^2 w_j^2 - \sigma_j^2)^2] &= \mathsf{E}[(x_j^2 w_j^2 - \mathsf{E}[x_j^2 w_j^2])^2] \\ &= \mathsf{Var}[x_j^2 w_j^2] = w_j^4 \cdot \mathsf{Var}[x_j^2] \\ &= \frac{\sigma_j^4}{g^2(\sigma_j^2)} \cdot g(\sigma_j^2)(1 - g(\sigma_j^2)) \\ &= \sigma_j^4 \cdot \frac{1 - g(\sigma_j^2)}{g(\sigma_j^2)}.\end{aligned}$$

Here we use the fact that the variance of a Bernoulli random variable with parameter p is $p(1-p)$. So we have

$$\begin{aligned}\mathsf{E}[(B^TB - A^TA)^2] &= \sum_{j=1}^{d} \mathsf{E}[(x_j^2 w_j^2 - \sigma_j^2)^2] \cdot v_j v_j^T \\ &= \sum_{j=1}^{d} \sigma_j^4 \cdot \frac{1 - g(\sigma_j^2)}{g(\sigma^2)} \cdot v_j v_j^T \\ &= VD'V^T,\end{aligned}$$

where D' is a diagonal matrix with $D'_{j,j} = \sigma_j^4 \cdot \frac{1-g(\sigma_j^2)}{g(\sigma_j^2)}$ for all j. Therefore, $VD'V^T$ is the eigen-decomposition of $\mathsf{E}[(B^TB - A^TA)^2]$, and the diagonals of D' are the eigenvalues. Since $g(\sigma_j^2) \le 1$, the eigenvalues are all non-negative. It follows that

$$\begin{aligned}\|\mathsf{E}[(B^TB - A^TA)^2]\|_2 &= \max_j |D'_{j,j}| \\ &= \max_j \frac{\sigma_j^4 \cdot (1 - g(\sigma_j^2))}{g(\sigma^2)}.\end{aligned}$$

□

Now we are ready to prove the main theorem.

PROOF OF THEOREM 4. To prove this theorem, we will use the following *Matrix Bernstein Inequality*, which can be found in [32] (Theorem 6.1).

LEMMA 4 (MATRIX BERNSTEIN). *Consider a finite sequence $\{X_k\}$ of independent, random, self-adjoint (or Hermitian) matrices with dimension d. Assume that*

$$\mathsf{E}[X_k] = 0 \text{ and } \lambda_{\max}(X_k) \le R$$

almost surely for all k. Define $\sigma^2 := \|\sum_k \mathsf{E}[X_k^2]\|_2$. Then the following inequality holds for all $t \ge 0$.

$$\mathsf{Pr}[\lambda_{\max}(\sum_k X_k) \ge t] \le d \cdot \exp\left(\frac{-t^2/2}{\sigma^2 + Rt/3}\right).$$

To use the Matrix Bernstein Inequality, we define

$$X_i = B^{(i)T}B^{(i)} - A^{(i)^T}A^{(i)}.$$

By Claim 3, $\mathsf{E}[X_i] = 0$ for all i. By Claim 4, $\lambda_{\max}(X_i) = \max_j \frac{\sigma_{i,j}^2}{g(\sigma_{i,j}^2)}$ for all i, which will always be bounded in our case, and thus we just set $R = \max_i \lambda_{\max}(X_i)$, that is

$$R = \max_i \lambda_{\max}(X_i) = \max_{i,j} \frac{\sigma_{i,j}^2}{g(\sigma_{i,j}^2)} = M.$$

The last equality is by definition of M in the statement of Theorem 4. Using Claim 5, we can bound σ^2:

$$\begin{aligned}\sigma^2 = \|\sum_i \mathsf{E}[X_i^2]\| &\le \sum_i \|\mathsf{E}[X_i^2]\| && \text{Triangle inequality} \\ &= \sum_i \max_j \frac{\sigma_{i,j}^4 \cdot (1 - g(\sigma_{i,j}^2))}{g(\sigma_{i,j}^2)} && \text{Claim 5.} \\ &= \kappa^2 && \text{By definition of } \kappa^2\end{aligned}$$

Now we can directly use Lemma 4 and prove that

$$\mathsf{Pr}[\lambda_{\max}(B^TB - A^TA) \ge t] \le d \cdot \exp\left(\frac{-t^2/2}{\kappa^2 + Mt/3}\right).$$

To establish the theorem, we still need to show

$$\mathsf{Pr}[\lambda_{\max}(A^TA - B^TB) \ge t] \le d \cdot \exp\left(\frac{-t^2/2}{\kappa^2 + Mt/3}\right),$$

but this inequality can be proved in exactly the same way, which we omit. □

3.1.2 *Sampling functions*

Next we discuss which sampling functions to use. For our application, we need to set $t = \alpha\|A\|_F^2$ in Theorem 4. Observe that, given any g, the total communication cost is $d \cdot \sum_{i,j} g(\sigma_{i,j}^2)$ in expectation. The most natural choice is a linear function, i.e., $g(x) = ax$ for some a. We present the analysis of linear functions in Appendix C.

THEOREM 5 (LINEAR). *If we set*

$$g(x) = \min\{\frac{\sqrt{s}}{\alpha\|A\|_F^2}\log(d/\delta)\cdot x, 1\},$$

then with probability $1-\delta$

$$\|B^TB - A^TA\|_2 \le 3\alpha\|A\|_F^2, \text{ and } \|B\|_F \le 2\|A\|_F.$$

The communication cost is $O(\frac{\sqrt{s}d}{\alpha}\cdot\log\frac{d}{\delta})$.

However, due to technical reasons, the above linear function is sub-optimal. We show that a less intuitive quadratic function gives a better bound on the communication cost.

THEOREM 6 (QUADRATIC). *If we set*

$$g(x) = \begin{cases}\min\{\frac{s}{\alpha^2\|A\|_F^4}\log(d/\delta)\cdot x^2, 1\} & \text{if } x \ge \frac{\alpha\|A\|_F^2}{s}\\ 0 & \text{otherwise}\end{cases},$$

then with probability $1-\delta$,

$$\|B^TB - A^TA\| \le 4\alpha\|A\|, \text{ and } \|B\|_F \le 2\|A\|_F.$$

The communication cost is $O(\frac{\sqrt{s}d}{\alpha}\cdot\sqrt{\log\frac{d}{\delta}})$.

PROOF. We observe that $\frac{\sigma^4\cdot(1-g(\sigma^2))}{g(\sigma^2)} \le \frac{\sigma^4}{g(\sigma^2)}$. If we use a quadratic function, i.e., $g(x) = bx^2$, the above inequality is bounded by $1/b$. So we have

$$\kappa^2 \le \sum_i \frac{1}{b} = \frac{s}{b}. \tag{6}$$

Since we set $b = \tilde{O}\left(\frac{s}{\alpha^2\|A\|_F^4}\right)$, it holds that $\kappa^2 \le \tilde{O}\left(\alpha^2\|A\|_F^4\right)$. However, now

$$M = \max_{i,j}\frac{\sigma_{i,j}^2}{g(\sigma_{i,j}^2)} = \tilde{O}\left(\max_{i,j}\frac{\alpha^2\|A\|_F^4}{s\cdot\sigma_{i,j}^2}\right),$$

which could be infinitely small when $\sigma_{i,j}$ is very close to zero. Therefore, in order to make this sampling function work, we need to drop all the small singular values. This is the reason why we set $g(x) = 0$ for $x \le \frac{\alpha\|A\|_F^2}{s}$.

For the i-th server, given $A^{(i)}$, we define a new matrix $\bar{A}^{(i)}$ as follows. We write its SVD as $A^{(i)} = (U, \Sigma, V)$, and define a diagonal matrix $\bar{\Sigma}$:

$$\bar{\Sigma}_{j,j} = \begin{cases}\sigma_j & \text{if } \sigma_j \ge \frac{\sqrt{\alpha}\|A\|_F}{\sqrt{s}}\\ 0 & \text{otherwise.}\end{cases}$$

Let $\bar{A}^{(i)} = \bar{\Sigma}V^T$. It holds that

$$\|A^{(i)T}A^{(i)} - \bar{A}^{(i)T}\bar{A}^{(i)}\|_2 = \|V(\Sigma^2 - \bar{\Sigma}^2)V^T\|_2 \le \frac{\alpha\|A\|_F^2}{s}.$$

Let $\bar{A}$ be the concatenation of $\bar{A}^{(i)}$'s, we have

$$\begin{aligned}\|A^TA - \bar{A}^T\bar{A}\|_2 &\le \sum_{i=1}^s\|A^{(i)T}A^{(i)} - \bar{A}^{(i)T}\bar{A}^{(i)}\|_2\\ &\le \sum_{i=1}^s\frac{\alpha\|A\|_F^2}{s} = \alpha\|A\|_F^2.\end{aligned} \tag{7}$$

This is the error resulting from dropping all the small singular values. By triangle inequality, it is now sufficient to bound $\|\bar{A}^T\bar{A} - B^TB\|_2 \le \alpha\|A\|_F^2$. Here B is the output for A, but B essentially has the same distribution as the output of the algorithm being applied on $\bar{A}$. So, to bound $\|\bar{A}^T\bar{A} - B^TB\|_2$, we can use Theorem 4 on $\bar{A}$ which has the property that all the squared singular values are larger than $\frac{\alpha\|A\|_F^2}{s}$. We set $t = \alpha\|A\|_F^2$, and it is easy to verify that

$$\begin{aligned}Mt/3 &\le \frac{t}{3}\cdot\max_{i,j}\frac{\sigma_{i,j}^2}{g(\sigma_{i,j}^2)} = \frac{t}{3}\cdot\max_{i,j}\frac{\alpha^2\|A\|_F^4}{s\cdot\sigma_{i,j}^2\cdot\log\frac{d}{\delta}}\\ &\le \frac{t}{3}\cdot\alpha\|A\|_F^2/\log\frac{d}{\delta} = \frac{\alpha^2\|A\|_F^4}{3\log\frac{d}{\delta}}.\end{aligned} \tag{8}$$

Since $\kappa^2 \le s/b = \alpha^2\|A\|_F^4/\log\frac{d}{\delta}$ (Eq (6)). By Theorem 4 with $t = \alpha\|A\|_F^2$, we get

$$\Pr\left[\|B^TB - \bar{A}^T\bar{A}\| \ge \alpha\|A\|_F^2\right] \le \delta.$$

By triangle inequality and Eq. (7), we have

$$\begin{aligned}\|B^TB - A^TA\| &\le \|B^TB - \bar{A}^T\bar{A}\| + \|A^TA - \bar{A}^T\bar{A}\|\\ &\le 2\alpha\|A\|_F^2\end{aligned}$$

with probability at least $1-\delta$.

Since $x \le \sqrt{x}$ for all $0 \le x \le 1$, we have

$$\begin{aligned}g(\sigma_{i,j}^2) &\le \min\{\frac{s\sigma_{i,j}^4}{\alpha^2\|A\|_F^4}\cdot\log\frac{d}{\delta}, 1\}\\ &\le \min\{\frac{\sqrt{s}\sigma_{i,j}^2}{\alpha\|A\|_F^2}\cdot\sqrt{\log\frac{d}{\delta}}, 1\}\end{aligned}$$

Then the communication cost is

$$\begin{aligned}d\cdot\sum_{i,j}g(\sigma_{i,j}^2) &\le d\cdot\sum_{i,j}\frac{\sqrt{s}\sigma_{i,j}^2}{\alpha\|A\|_F^2}\cdot\sqrt{\log\frac{d}{\delta}}\\ &= \frac{\sqrt{s}d}{\alpha}\cdot\sqrt{\log\frac{d}{\delta}}.\end{aligned}$$

□

3.2 Covariance error $\varepsilon\|A-[A]_k\|_F^2/k$ via adaptive sampling

We will first present a randomized algorithm with communication cost $O(skd + \frac{\sqrt{s}kd}{\varepsilon}\cdot\sqrt{\log d})$ in terms of real numbers, then discuss how to obtain bit/word complexity.

In the deterministic algorithm, each server S_i invokes FD to compute a local sketch $B^{(i)}$ in one pass, i.e. $B^{(i)} = \mathsf{FD}(A^{(i)}, \varepsilon, k)$ (Theorem 1), then sends $B^{(i)}$ to the coordinator. To save communication, we will further compress each $B^{(i)}$ computed by FD. It was shown that not only $B^{(i)}$ has small covariance error, the Frobenius norm of $B^{(i)}$ is also smaller than the Frobenius norm of $A^{(i)}$ [27]. From this property, it is not difficult prove the following lemma.

LEMMA 5. *Assume* $B = \mathsf{FD}(A, \varepsilon, k)$, *then*

$$\|B - [B]_k\|_F^2 \le (1+\varepsilon)\|A - [A]_k\|_F^2.$$

PROOF. Let v_i be i-th right singular vector of B. We have

$$\begin{aligned}\|B - [B]_k\|_F^2 &= \|B\|_F^2 - \sum_{i=1}^k\|Bv_i\|^2\\ &\le \|B\|_F^2 - \sum_{i=1}^k\|Av_i\|^2 + k\|A^TA - B^TB\|_2\\ &\le \|A\|_F^2 - \sum_{i=1}^k\|Av_i\|^2 + \varepsilon\|A - [A]_k\|_F^2\\ &\le \|A - [A]_k\|_F^2 + \varepsilon\|A - [A]_k\|_F^2.\end{aligned}$$

The last inequality holds because

$$\sum_{i=1}^{k} \|Av_i\|^2 \leq \sum_{i=1}^{k} \|Au_i\|^2,$$

where u_i is the i-th right singular vector of A, and $\|A-[A]_k\|_F^2 = \|A\|_F^2 - \sum_{i=1}^{k} \|Au_i\|^2$. $\square$

The following lemma directly follows from singular value decomposition.

LEMMA 6. *For any matrix $B \in \mathbb{R}^{n\times d}$, there exist two matrices $T \in \mathbb{R}^{k\times d}$ and $R \in \mathbb{R}^{(d-k)\times d}$ such that*

$$B^TB = T^TT + R^TR,$$

and $\|R\|_F^2 = \|B-[B]_k\|_F^2$.

PROOF. Let $B = U\Sigma V^T$ be the singular value decomposition of B. Clearly,

$$B^TB = V\Sigma^2V^T = \sum_{i=1}^{d} \sigma_i^2 v_i v_i^T.$$

Let T be the matrix consists of the top-k rows of the matrix ΣV^T and let R contain the rest $d-k$ rows. It is well-known that $\|B-[B]_k\|_F^2 = \sum_{i=k+1}^{d} \sigma_i^2 = \|R\|_F^2$. Hence, T and R satisfy the requirements of the lemma. $\square$

For convenience, we use $(T,R) = \mathsf{Decomp}(B,k)$ to denote this decomposition.

In our algorithm, each server S_i computes

$$(T^{(i)}, R^{(i)}) = \mathsf{Decomp}(B^{(i)}, k).$$

Let $B = [B^{(1)}; \cdots; B^{(s)}]$, and we define T and R similarly. By the mergeability of FD, it holds that $\|A^TA - B^TB\|_2 \leq \varepsilon\|A-[A]_k\|_F^2/k$. From Lemma 6, we have

$$\|A^TA - T^TT - R^TR\|_2 \leq \varepsilon\|A-[A]_k\|_F^2/k, \text{ and}$$

$$\|R\|_F^2 = \sum_{i=1}^{s} \|R^{(i)}\|_F^2 = \sum_{i=1}^{s} \|B^{(i)} - [B^{(i)}]_k\|_F^2.$$

Then by Lemma 5, we get

$$\|R\|_F^2 \leq (1+\varepsilon)\sum_{i=1}^{s} \|A^{(i)} - [A^{(i)}]_k\|_F^2. \tag{9}$$

Let $[A]_k^{(i)}$ be the ith block of $[A]_k$ corresponding to the rows in $A^{(i)}$. We observe

$$\begin{aligned}\sum_i \|A^{(i)} - [A^{(i)}]_k\|_F^2 &\leq \sum_i \|A^{(i)} - [A]_k^{(i)}\|_F^2 \\ &= \|A-[A]_k\|_F^2, \end{aligned} \tag{10}$$

since $[A]_k^{(i)}$ has rank at most k, and $[A^{(i)}]_k$ is the best rank k approximation for $A^{(i)}$. Combine (9) and (10), we get

$$\|R\|_F^2 \leq (1+\varepsilon)\|A-[A]_k\|_F^2. \tag{11}$$

Now, each server S_i applies the SVS algorithm on $R^{(i)}$, and outputs $W^{(i)} = \mathsf{SVS}(R^{(i)})$. Let $W = [W^{(1)}; \cdots; W^{(s)}]$. From Theorem 6, we have

$$\|W^TW - R^TR\|_2 \leq \varepsilon\|R\|_F^2/k \leq (\varepsilon+\varepsilon^2)\|A-[A]_k\|_F^2/k,$$

and the number of rows in W is $O(\frac{\sqrt{s}kd}{\varepsilon}\cdot\sqrt{\log d})$. Then each server S_i sends $Q^{(i)} = [T^{(i)}; W^{(i)}]$ to the coordinator. Define Q similarly, and we have

$$\begin{aligned}\|A^TA - Q^TQ\|_2 &= \|A^TA - T^TT - W^TW\| \\ &= \|A^TA - T^TT - R^TR + R^TR - W^TW\| \\ &\leq \|A^TA - B^TB\|_2 + \|W^TW - R^TR\|_2 \\ &\leq \varepsilon\|A-[A]_k\|_F^2/k + 2\varepsilon\|A-[A]_k\|_F^2/k \\ &\leq 3\varepsilon\cdot\|A-[A]_k\|_F^2/k \end{aligned}$$

which means Q is an $(3\varepsilon,k)$-sketch of A. The total communication cost of this algorithm is $O(sdk + \frac{\sqrt{s}kd}{\varepsilon}\cdot\sqrt{\log d})$. Since $\|W\|_F^2 = O(1)\cdot\|R\|_F^2 = O(1)\cdot\|A-[A]_k\|_F^2$ from Theorem 6, we also have $\|Q\|_F^2 = \|A\|_F^2 + O(\|A-[A]_k\|_F^2)$.

THEOREM 7. *There is a distributed streaming algorithm which computes an (ε,k)-sketch Q. The communication cost is $O(sdk + \frac{\sqrt{s}kd}{\varepsilon}\cdot\sqrt{\log d})$, and space used by each server is $O(kd/\varepsilon)$. Moreover, $\|Q\|_F^2 = \|A\|_F^2 + O(\|A-[A]_k\|_F^2)$.*

Note that the size of Q is not optimal, but we can apply another FD on Q. Assume $Q' = \mathsf{FD}(Q,\varepsilon,k)$, we have $\|Q^TQ - Q'^TQ'\|_2 \leq \varepsilon\|Q-[Q]_k\|_F^2/k$, and thus the covariance error of Q' (w.r.t. A) depends on $\|Q-[Q]_k\|_F^2$. However, since $\|Q\|_F^2 = \|A\|_F^2 + O(\|A-[A]_k\|_F^2)$, using the same argument as in the proof of Lemma 5, it can be shown that $\|Q-[Q]_k\|_F^2 = O(\|A-[A]_k\|_F^2)$. As a result, it holds that

$$\|A^TA - Q'^TQ'\|_2 \leq O(\varepsilon)\cdot\|A-[A]_k\|_F^2/k.$$

After adjusting ε by a constant factor in the beginning, Q' is an (ε,k)-sketch of A with optimal sketch size.

3.3 Bit complexity for communication cost

Similar to [5], our main idea is to conduct a case analysis based on the rank of A.

Case 1: $\mathsf{rank}(A) \leq 2k$. In this case, each $A^{(i)}$ also has rank at most $2k$. Then we can find at most $2k$ rows of $A^{(i)}$ which span the row space of $A^{(i)}$. Let Q be the matrix consists of such a set of rows. We use the standard notation Q^+ to denote the Moore-Penrose pseudoinverse of Q. It is known that the $d\times d$ matrix Q^+Q is the orthogonal projector which projects any d-dimensional vector x onto the row space of Q, and thus onto the row space of $A^{(i)}$. Hence, if x belongs to the row space of $A^{(i)}$, $Q^+Qx = x$. In particular, we have $Q^+QA^{(i)T} = A^{(i)T}$.

Based on the above observation, each server S_i runs the following algorithm. S_i first deterministically selects any maximal set of linearly independent rows from $A^{(i)}$, denoted as Q, then sends both Q and $QA^{(i)T}A^{(i)}Q^T$ to the coordinator. Given Q, the coordinator can compute Q^+, and then computes $Q^+QA^{(i)T}A^{(i)}Q^TQ^{T+}$, which is exactly $A^{(i)T}A^{(i)}$. In other words, the coordinator can compute A^TA exactly. For the communication cost, Q takes at most $2kd$ words, since Q consists rows chosen from A. On the other hand, it is easy to verify that each entry of $QA^{(i)T}A^{(i)}Q^T$ needs at most $O(\log(nd/\varepsilon))$ bits, and thus takes $O(k^2)$ words to represent. Since $k \leq d$, the total communication cost is $O(skd)$ words.

Naively, it requires two passes for each server: one pass for computing Q and one pass for $QA^{(i)T}A^{(i)}Q^T$. With a little more effort, the algorithm can be implemented in only one pass using $O(kd)$ space. We maintain a maximal set of linearly independent rows Q along the way, and also maintain an orthonormal basis of Q,

denoted as V, on the side. The matrix $Z = VA^{(i)T}A^{(i)}V^T$ can be maintained in the streaming model using $O(k^2)$ space (in real numbers): when a new row u is added to V, compute $U = V[V; u]^T$ (with $O(k^2)$ space) and then update Z as $Z = U^T ZU$. In the end, we compute $QV^T ZVQ^T$, which is $QA^{(i)T}A^{(i)}Q^T$. Here we have used the fact that $V^T Vx = x$ if x is in the row space of V.

Case 2: $\mathsf{rank}(A) > 2k$. In this case, each server S_i first computes a matrix $Q^{(i)}$ as in the above section such that $Q = [Q^{(1)}; \cdots; Q^{(s)}]$ is an (ε, k)-sketch of A. Note that Q may contain entries exponentially small in k/ε [5], which leads to an extra k/ε factor in communication cost. We use the following result which gives a lower bound on the singular values of a matrix with integer entries of bounded magnitude.

LEMMA 7 (LEMMA 4.1 OF [6]). *If an $n \times d$ matrix A has integer entries bounded in magnitude by γ, and has rank ρ, then the k-th largest singular value of A satisfies*

$$\sigma_k^2 \geq (nd\gamma^2)^{-k/(\rho-k)}.$$

Since we assume each entry of the input matrix A is an integer with magnitude bounded by $\mathrm{poly}(nd/\varepsilon)$ and $\mathsf{rank}(A) > 2k$, we get from the above lemma

$$\|A - [A]_k\|_F^2 \geq \sigma_{k+1}^2 \geq \mathrm{poly}^{-1}(nd/\varepsilon).$$

Observe that each entry of Q is upper bounded by $\mathrm{poly}(nd/\varepsilon)$, since otherwise the covariance error of Q must be too large. Therefore, it suffice to round each entry of Q to an additive $\mathrm{poly}^{-1}(nd/\varepsilon)$ precision. In other words, after rounding, each entry of Q is representable with $O(\log(nd/\varepsilon))$ bits, and thus the communication cost is $O(skd) + \tilde{O}(\sqrt{s}kd/\varepsilon)$ words. The deterministic case is the same; just replace each $Q^{(i)}$ in the above argument with an (ε, k)-sketch computed by the deterministic FD.

4. DISTRIBUTED PCA

To solve the distributed PCA problem, we can use Theorem 7 to obtain an (ε, k)-sketch Q, and then the coordinator computes the top k right singular vectors of Q. The communication cost is thus $O(sdk + \frac{\sqrt{s}dk}{\varepsilon} \cdot \sqrt{\log d})$ words. When $s \geq \frac{\log d}{\varepsilon^2}$, this cost is $O(skd)$. In the model where all servers need to output the same answer, a lower bound of $\Omega(skd)$ bits was proved in [5]. Since it takes $O(skd)$ communication for the coordinator to broadcast the answer to all servers, our algorithm is optimal in this setting (up to a $\log$ factor).

On the other hand, when s is small, the term $O(\frac{\sqrt{s}dk}{\varepsilon} \cdot \sqrt{\log d})$ dominates the cost. In this case, we can further improve the communication cost when d is large using the distributed algorithm of [5].

THEOREM 8 (DISTRIBUTED PCA OF [5]). *Given any $A \in \mathbb{R}^{n\times d}$ which is distributed across s servers, there is a batch algorithm for PCA with communication cost*

$$O(sdk + \min\{d, k\varepsilon^{-2}\} \cdot \min\{n, sk\varepsilon^{-2}\}).$$

The key is the following lemma, the proof of which can be found in section 4.1.[5]

LEMMA 8. *Let Q be a strong $(\varepsilon/2, k)$-sketch of A, and we assume $\|Q\|_F^2 = \|A\|_F^2 + O(\|A - [A]_k\|_F^2)$. Let $V \in \mathbb{R}^{d\times k}$ be any orthonormal matrix satisfying $\|Q - QVV^T\|_F^2 \leq (1+\varepsilon)\|Q - [Q]_k\|_F^2$, then*

$$\|A - AVV^T\|_F^2 \leq (1 + O(\varepsilon)) \cdot \|A - [A]_k\|_F^2.$$

This lemma can be viewed as a robust version of Lemma 1. With this result, we can apply the standard "sketch-and-solve" approach to solve the distributed PCA problem. More formally, in the "sketch" step, all servers compute a distributed (ε, k)-sketch, i.e., each server S_i output matrix $Q^{(i)}$, such that $Q = [Q^{(1)}; \cdots; Q(s)]$ is an (ε, k)-sketch of A. Note that, in our algorithm for (ε, k)-sketch, if we do not require servers to send their local sketches to the coordinator, the communication cost is negligible[6], and the number of rows in Q is $\tilde{O}(\sqrt{s}kd/\varepsilon)$. In the "solve" step, we can apply any communication-efficient distributed PCA algorithm with input Q, which then solve the PCA problem for A due to Lemma 8 (where we also use the property that $\|Q\|_F^2 = \|A\|_F^2 + O(\|A - [A]_k\|_F^2)$). The communication cost of the combined algorithm is dominated by the "solve" step, while local computation cost is dominated by the "sketch" step. Since each server only makes one pass over its local data with small working space for computing an (ε, k)-sketch, the above approach can convert any batch distributed PCA algorithm to a distributed streaming algorithm.

If we use the distributed PCA algorithm of [5] to compute the approximate PCs for Q, we solve the PCA problem for A with communication cost $O(skd + \frac{\sqrt{s}k\sqrt{\log d}}{\varepsilon} \cdot \min\{d, k/\varepsilon^2\})$. The cost in Theorem 8 is in terms of words as long as the entries of the input matrix are representable by $O(\log(nd/\varepsilon))$ bits. As discussed in section 3.3, each entry of Q takes $O(\log(nd/\varepsilon))$ bits, and thus the cost of the combined algorithm is also in words. Using (ε, k)-sketch as a sketch for solving distributed PCA, our algorithm is faster and more space-efficient than the algorithm of [5].

THEOREM 9. *Given $A \in \mathbb{R}^{n\times d}$, there is a distributed streaming algorithm which solves PCA for A. The communication cost is $O(sdk + \frac{\sqrt{s}k\sqrt{\log d}}{\varepsilon} \cdot \min\{d, k/\varepsilon^2\})$ words, and space used by each server is $O(dk/\varepsilon)$ real numbers.*

4.1 Proof of Lemma 8

PROOF. By definition of $(\varepsilon/2, k)$-sketch, we have

$$\|A^T A - B^T B\|_2 \leq \frac{\varepsilon}{2k}\|A - [A]_k\|_F^2,$$

which is equivalent to

$$\max_{x:\|x\|=1} |\|Ax\|^2 - \|Bx\|^2| \leq \frac{\varepsilon}{2k}\|A - [A]_k\|_F^2 \qquad (12)$$

[5] We remark that a similar result has been proved in [8] under a slightly different setting. However, their proof is quite complicated, so we present a direct proof for our application here.

[6] In fact, all the computations are local and parallel; the only communication needed is to set the same sampling function g.

Let u_i and w_i be the ith right singular vector of B and A respectively. We have

$$\begin{aligned}
\|B-[B]_k\|_F^2 &= \|B\|_F^2 - \sum_{i=1}^{k}\|Bu_i\|^2 \\
&\le \|A\|_F^2 + O(\|A-[A]_k\|_F^2) - \sum_{i=1}^{k}\|Bu_i\|^2 \\
&\le \|A\|_F^2 + O(\|A-[A]_k\|_F^2) - \sum_{i=1}^{k}\|Bw_i\|^2 \\
&\le \|A\|_F^2 + O(\|A-[A]_k\|_F^2) - \sum_{i=1}^{k}\|Aw_i\|^2 \\
&\quad + k\frac{\varepsilon}{2k}\|A-[A]_k\|_F^2 \quad \text{by (12)} \\
&\le O(\|A-[A]_k\|_F^2). \qquad (13)
\end{aligned}$$

The first equality is from Pythagorean theorem. Let v_i be the ith column of V. Again by Pythagorean theorem, we have

$$\begin{aligned}
\|B-BVV^T\|_F^2 &= \|B\|_F^2 - \|BVV^T\|_F^2 \\
&= \|B\|_F^2 - \sum_{i=1}^{k}\|Bv_i\|^2,
\end{aligned}$$

and

$$(1+\varepsilon)\|B-[B]_k\|_F^2 = \|B\|_F^2 - \sum_{i=1}^{k}\|Bu_i\|^2 + \varepsilon\|B-[B]_k\|_F^2.$$

Since $\|B-BVV^T\|_F^2 \le (1+\varepsilon)\|B-[B]_k\|_F^2$ from our assumption, we have

$$\begin{aligned}
\sum_{i=1}^{k}\|Bv_i\|^2 &\ge \sum_{i=1}^{k}\|Bu_i\|^2 - \varepsilon\|B-[B]_k\|_F^2 \\
&\ge \sum_{i=1}^{k}\|Bu_i\|^2 - O(\varepsilon)\|A-[A]_k\|_F^2. \qquad (14)
\end{aligned}$$

The last inequality is from (13). We have

$$\begin{aligned}
\|A-AVV^T\|_F^2 =& \|A\|_F^2 - \|AVV^T\|_F^2 \\
=& \|A\|_F^2 - \sum_{i=1}^{k}\|Av_i\|^2 \\
\le& \|A\|_F^2 - \sum_{i=1}^{k}\|Bv_i\|^2 + k\cdot\frac{\varepsilon}{2k}\|A-[A]_k\|_F^2 \\
\le& \|A\|_F^2 - \sum_{i=1}^{k}\|Bu_i\|^2 + O(\varepsilon)\|A-[A]_k\|_F^2 \\
\le& \|A\|_F^2 - \sum_{i=1}^{k}\|Bw_i\|^2 + O(\varepsilon)\|A-[A]_k\|_F^2 \\
\le& \|A\|_F^2 - \sum_{i=1}^{k}\|Aw_i\|^2 + k\frac{\varepsilon}{2k}\|A-[A]_k\|_F^2 \\
&+ O(\varepsilon)\|A-[A]_k\|_F^2 \\
=& \|A-[A]_k\|_F^2 + O(\varepsilon)\|A-[A]_k\|_F^2,
\end{aligned}$$

where the second inequality is by (14) □

5. CONCLUSION

In this paper, we study the covariance sketch and its application to PCA in the distributed model. We provide efficient one pass algorithms with improved communication costs, and also prove a tight deterministic lower bound in the blackboard model. There are lots of interesting open questions left. For instance, is our randomized algorithm for covariance sketch optimal? For PCA, whether the $\Omega(skd)$ lower bound of [5] still holds in the case when only one server needs to know the answer; it is also interesting to determine the right order of the $\mathrm{poly}(s,k,1/\varepsilon)$ term in the cost. Another question is what the communication complexity of covariance sketch is in the arbitrary partition model

6. REFERENCES

[1] P. K. Agarwal, G. Cormode, Z. Huang, J. M. Phillips, Z. Wei, and K. Yi. Mergeable summaries. *ACM Transactions on Database Systems (TODS)*, 38(4):26, 2013.

[2] P. Beame, P. Koutris, and D. Suciu. Communication steps for parallel query processing. In *PODS*. ACM, 2013.

[3] P. Beame, P. Koutris, and D. Suciu. Skew in parallel query processing. In *PODS*. ACM, 2014.

[4] S. Bhojanapalli, P. Jain, and S. Sanghavi. Tighter low-rank approximation via sampling the leveraged element. In *SODA*. SIAM, 2015.

[5] C. Boutsidis, D. Woodruff, and P. Zhong. Optimal principal component analysis in distributed and streaming models. *STOC*, 2016.

[6] K. L. Clarkson and D. P. Woodruff. Numerical linear algebra in the streaming model. In *STOC*. ACM, 2009.

[7] K. L. Clarkson and D. P. Woodruff. Low rank approximation and regression in input sparsity time. In *STOC*, 2013.

[8] M. B. Cohen, C. Musco, and C. Musco. Input sparsity time low-rank approximation via ridge leverage score sampling. *arXiv preprint arXiv:1511.07263*, 2015.

[9] A. Desai, M. Ghashami, and J. M. Phillips. Improved practical matrix sketching with guarantees. *IEEE Transactions on Knowledge and Data Engineering*, 28(7):1678–1690, 2016.

[10] P. Drineas, R. Kannan, and M. W. Mahoney. Fast monte carlo algorithms for matrices i: Approximating matrix multiplication. *SIAM Journal on Computing*, 36(1):132–157, 2006.

[11] P. Drineas, R. Kannan, and M. W. Mahoney. Fast monte carlo algorithms for matrices ii: Computing a low-rank approximation to a matrix. *SIAM Journal on Computing*, 36(1):158–183, 2006.

[12] P. Drineas, M. W. Mahoney, S. Muthukrishnan, and T. Sarlós. Faster least squares approximation. *Numerische Mathematik*, 117(2):219–249, 2011.

[13] D. P. Dubhashi and A. Panconesi. *Concentration of measure for the analysis of randomized algorithms*. Cambridge University Press, 2009.

[14] D. Feldman, M. Schmidt, and C. Sohler. Turning big data into tiny data: Constant-size coresets for k-means, pca and projective clustering. In *SODA*. SIAM, 2013.

[15] M. Ghashami, E. Liberty, and J. M. Phillips. Efficient frequent directions algorithm for sparse matrices. *KDD*, 2016.

[16] M. Ghashami and J. M. Phillips. Relative errors for deterministic low-rank matrix approximations. In *SODA*, pages 707–717. SIAM, 2014.

[17] M. Ghashami, J. M. Phillips, and F. Li. Continuous matrix approximation on distributed data. *Proceedings of the VLDB Endowment*, 7(10):809–820, 2014.
[18] P. B. Gibbons and S. Tirthapura. Estimating simple functions on the union of data streams. In *SPAA*. ACM, 2001.
[19] P. B. Gibbons and S. Tirthapura. Distributed streams algorithms for sliding windows. In *SPAA*. ACM, 2002.
[20] H. Huang and S. P. Kasiviswanathan. Streaming anomaly detection using randomized matrix sketching. *Proceedings of the VLDB Endowment*, 2015.
[21] Z. Huang and K. Yi. The communication complexity of distributed epsilon-approximations. In *FOCS*, 2014.
[22] R. Kannan, S. Vempala, and D. P. Woodruff. Principal component analysis and higher correlations for distributed data. In *COLT*, 2014.
[23] Z. Karnin and E. Liberty. Online pca with spectral bounds. In *Proceedings of the 28th Annual Conference on Computational Learning Theory (COLT)*, 2015.
[24] E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge University Press, 1997.
[25] Y. Liang, M.-F. F. Balcan, V. Kanchanapally, and D. Woodruff. Improved distributed principal component analysis. In *NIPS*, 2014.
[26] Y. Liang, B. Xie, D. Woodruff, L. Song, and M.-F. Balcan. Communication efficient distributed kernel principal component analysis.
[27] E. Liberty. Simple and deterministic matrix sketching. In *KDD*, pages 581–588. ACM, 2013.
[28] J. Misra and D. Gries. Finding repeated elements. *Science of computer programming*, 2(2):143–152, 1982.
[29] J. Nelson and H. L. Nguyên. Osnap: Faster numerical linear algebra algorithms via sparser subspace embeddings. In *FOCS*. IEEE, 2013.
[30] R. I. Oliveira. Sums of random hermitian matrices and an inequality by rudelson. *Electron. Commun. Probab*, 15(203-212):26, 2010.
[31] T. Sarlos. Improved approximation algorithms for large matrices via random projections. In *FOCS*, 2006.
[32] J. A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012.
[33] D. Van Gucht, R. Williams, D. P. Woodruff, and Q. Zhang. The communication complexity of distributed set-joins with applications to matrix multiplication. In *PODS*. ACM, 2015.
[34] Z. Wei, X. Liu, F. Li, S. Shang, X. Du, and J.-R. Wen. Matrix sketching over sliding windows. *SIGMOD*, 2016.
[35] D. Woodruff. Low rank approximation lower bounds in row-update streams. In *NIPS*, 2014.
[36] S. Yoo, H. Huang, and S. P. Kasiviswanathan. Streaming spectral clustering. ICDE, 2016.
[37] Y. Zhang, M. Wainwright, and M. Jordan. Distributed estimation of generalized matrix rank: Efficient algorithms and lower bounds. In *ICML*, 2015.

APPENDIX

A. PROOF OF LEMMA 1

LEMMA 9 (RESTATEMENT OF LEMMA 1).

$$\|A-\pi_B^k(A)\|_F^2 \le \|A-[A]_k\|_F^2 + 2k\cdot\|A^TA-B^TB\|_2.$$

PROOF. For any x with $\|x\|=1$, we have

$$\left|\, \|Ax\|^2 - \|Bx\|^2 \,\right| = \left| x^T(A^TA-B^TB)x \right| \le \|A^TA-B^TB\|_2 \quad (15)$$

Let u_i and w_i be the ith right singular vector of B and A respectively

$$\begin{aligned}
&\|A-\pi_B^k(A)\|_F^2 = \|A\|_F^2 - \|\pi_B^k(A)\|_F^2 \\
&= \|A\|_F^2 - \sum_{i=1}^k \|Au_i\|^2 \\
&\le \|A\|_F^2 - \sum_{i=1}^k \|Bu_i\|^2 + k\cdot\|A^TA-B^TB\|_2 \\
&\le \|A\|_F^2 - \sum_{i=1}^k \|Bw_i\|^2 + k\cdot\|A^TA-B^TB\|_2 \\
&\le \|A\|_F^2 - \sum_{i=1}^k \|Aw_i\|^2 + 2k\cdot\|A^TA-B^TB\|_2 \\
&= \|A-[A]_k\|_F^2 + 2k\cdot\|A^TA-B^TB\|_2.
\end{aligned}$$

□

B. BOUNDING $\|B\|_F$

We next provide a result on bounding $\|B\|_F^2$.

THEOREM 10. *Assume the same setting as in Theorem 4, and define*

$$M = \max_{i,j}\frac{\sigma_{i,j}^2}{g(\sigma_{i,j}^2)} \text{ and } \tau^2 = \sum_{i,j}\frac{\sigma_{i,j}^4\cdot(1-g(\sigma_{i,j}^2))}{g(\sigma_{i,j}^2)},$$

then the following inequality holds:

$$\Pr[\|B\|_F^2 \ge \|A\|_F^2 + t] \le \exp\left(\frac{-t^2/2}{\tau^2 + Mt/3}\right).$$

The proof of this theorem is actually a special case of the proof of Theorem 4; here we only need to bound the sum of squared singular values, so we apply the usual *Bernstein Inequality* [13] for scalar random variables. We omit the proof here.

C. LINEAR FUNCTION (PROOF OF THEOREM 5)

From Theorem 4 and 10, we want to bound M, κ^2 and τ^2. It is easy to bound M if we pick a linear function, i.e., $g(x)=\beta x$ for some β. Since $g(x)$ is a probability, it also must be bounded by 1, and thus we will set $g(x)=\min\{\beta x, 1\}$. Then the communication cost is

$$d\sum_{i,j} g(\sigma_{i,j}^2) \le \beta d\sum_{i,j}\sigma_{i,j}^2 = \beta d\sum_{i,j}\|A^{(i)}\|_F^2 = \beta d\|A\|_F^2.$$

For any σ, we have

$$\begin{aligned}
\frac{\sigma^4\cdot(1-g(\sigma^2))}{g(\sigma^2)} &= \frac{\sigma^2}{\beta} - \sigma^4 \le \frac{\sigma^2}{\beta} - \sigma^4 - \frac{1}{4\beta^2} + \frac{1}{4\beta^2} \\
&= -\left(\frac{1}{2\beta} - \sigma^2\right)^2 + \frac{1}{4\beta^2} \\
&\le \frac{1}{4\beta^2}.
\end{aligned}$$

So it follows that

$$\kappa^2 \leq \sum_i 1/4\beta^2 = s/4\beta^2.$$

We also have

$$\tau^2 \quad = \quad \sum_{i,j} \frac{\sigma_{i,j}^4 \cdot (1 - g(\sigma_{i,j}^2))}{g(\sigma_{i,j}^2)} \leq \sum_{i,j} \frac{\sigma_{i,j}^2}{\beta} = \frac{\|A\|_F^2}{\beta}.$$

To achieve our error bound, we set $\beta = \frac{\sqrt{s}}{\alpha\|A\|_F^2} \cdot \log \frac{d}{\delta}$, and set $t = \alpha\|A\|_F^2$ in Theorem 4. We have $M \leq 1/\beta$, and thus

$$\kappa^2 + Mt/3 \leq \alpha^2\|A\|_F^4/(4 \cdot \log \frac{d}{\delta}) + \alpha^2\|A\|_F^4/(3\sqrt{s} \cdot \log \frac{d}{\delta}),$$

which is at most $\alpha^2\|A\|_F^4/(2\log(d/\delta))$. From Theorem 4 with $t = \alpha\|A\|_F^2$, the probability $\mathsf{Pr}[\|B^TB - A^TA\| \geq \alpha\|A\|_F^2]$ is smaller than

$$\begin{aligned} d \cdot \exp\left(\frac{-\alpha^2\|A\|_F^4/2}{\alpha^2\|A\|_F^4/(2\log(d/\delta))}\right) &\leq d \cdot \exp\left(-\log \frac{d}{\delta}\right) \\ &= \delta. \end{aligned}$$

To bound $\|B\|_F$, we set $t = \|A\|_F^2$ in Theorem 10, and have

$$\begin{aligned} \tau^2 + Mt/3 &\leq \frac{\alpha\|A\|_F^4}{(\sqrt{s} \cdot \log \frac{d}{\delta})} + \frac{\alpha\|A\|_F^4}{(3\sqrt{s} \cdot \log \frac{d}{\delta})} \\ &\leq \|A\|_F^4/\log \frac{d}{\delta}. \end{aligned}$$

By Theorem 10 with $t = \|A\|_F^2$, we have

$$\mathsf{Pr}[\|B\|_F^2 \geq 2\|A\|_F^2] \leq \delta/d.$$

The communication cost is at most $O(\frac{\sqrt{s}d}{\alpha} \cdot \log \frac{d}{\delta})$.