Journal of Machine Learning Research 22 (2021) 1-38 Submitted 7/20; Revised 1/21; Published 2/21

Communication-Efficient Distributed Covariance Sketch, with Application to Distributed PCA

Zengfeng Huang HUANGZF@FUDAN.EDU.CN
School of Data Science
Fudan University
Shanghai, China

Xuemin Lin LXUE@CSE.UNSW.EDU.AU
School of Computer Science and Engineering
The University of New South Wales
Sydney, Australia

Wenjie Zhang ZHANGW@CSE.UNSW.EDU.AU
School of Computer Science and Engineering
The University of New South Wales
Sydney, Australia

Ying Zhang YING.ZHANG@UTS.EDU.AU
School of Computer Science
University of Technology, Sydney
Sydney, Australia

Editor: Michael Mahoney

Abstract

A sketch of a large data set 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 $A \in \mathbb{R}^{n\times d}$ that is distributed across s machines. 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., $\|A^TA - B^TB\|_2$. Such a matrix B is called 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 that there is a nontrivial gap between deterministic and randomized communication complexity for computing a covariance sketch. More specifically, we first prove an almost tight deterministic communication lower bound, then provide a new randomized algorithm with communication cost smaller than the deterministic lower bound. Based on a well-known connection between covariance sketch and approximate principle component analysis, we obtain better communication bounds for the distributed PCA problem. Moreover, we also give an improved distributed PCA algorithm for sparse input matrices, which uses our distributed sketching algorithm as a key building block.

Keywords: Matrix Sketching, PCA, Distributed Streaming, Low Rank Approximation, Communication Complexity

1. Introduction

Sketching techniques have now become popular algorithmic tools for processing big data. For many applications in machine learning, the underlying data sets are represented as large-scale matrices.

To analyze such large data matrices, exact computations are often infeasible and unnecessary; thus, randomized and approximate methods are widely used. The "sketch-and-solve" framework, i.e. computing a sketch matrix first and then executing expensive computations (e.g., Singular Value Decomposition (SVD) and regression) on the sketch matrix, has been successfully used to approximately solve many important linear algebraic problems (e.g. Sarlos (2006); Clarkson and Woodruff (2013); Nelson and Nguyên (2013); Boutsidis and Woodruff (2014)).

Traditionally, a sketch is computed using 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 machines. In these systems, the communication cost and the number of computation rounds become the most critical complexity parameters. The key to extending the sketch-and-solve framework to the distributed setting is to design communication- and round-efficient algorithms for computing matrix sketches with required error guarantees. In this paper, we study communication-efficient distributed algorithms for computing *covariance sketches*, which is an important type of matrix sketch with a broad spectrum of applications.

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 all $x \in \mathbb{R}^d$ is approximately preserved by $\|Bx\|_2$. Covariance sketch has a wide range of applications including low-rank approximation, PCA, clustering, anomaly detection, online learning, etc. (Drineas et al., 2006b; Ghashami and Phillips, 2014; Cohen et al., 2017; Karnin and Liberty, 2015; Luo et al., 2016; Yoo et al., 2016; Luo et al., 2018; Sharan et al., 2018). Due to its importance, computing a covariance sketch has been extensively studied in various computational models e.g., Drineas et al. (2006a); Liberty (2013); Ghashami et al. (2014b); Wei et al. (2016); Desai et al. (2016); Huang (2019); Luo et al. (2019). In this paper, we study the problem of computing a covariance sketch in the distributed model and its applications to distributed PCA and low-rank matrix approximations.

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 machine (with no replication of data). We do not make any assumption on how the data is partitioned; the partition can be arbitrary or even adversarial. The s machines 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*. The *broadcast model* (aka. the *blackboard model* in communication complexity community) is also widely studied, in which each message can be seen by all machines. All algorithms proposed in this work only need 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. Apparently, the more powerful broadcast model may achieve lower communication costs, but for all the problem studied in this paper, broadcast doesn't seem to have an advantage. In particular, all the state-of-the-art algorithms only require point-to-point communication and their communication costs cannot be improved (up to a constant) even if broadcast is allowed. Moreover, our deterministic communication lower bound holds for the *broadcast model* (and also holds against protocols using any number of rounds), which is matched by our message passing algorithm, and thus, the two communication model are provably equivalent for deterministic algorithms. In our model, there is one special machine which acts as the central *coordinator*. For simplicity, we assume that the s machines only communicates with the coordinator. This is often called the *coordinator model* and one can simulate

arbitrary message-passing protocols within a constant factor in communication together with an additive $O(\log(mn))$-bit overhead per message (Phillips et al., 2016).

We exhibit new algorithms for distributed covariance sketch with improved communication costs. For instance, assume there are $s = d$ machines 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 Liberty (2013) has cost $O(d^3)$, and the cost of using random sampling is also $O(d^3)$ Drineas et al. (2006a), which is the same as the cost of the trivial algorithm (that sends all data to the coordinator). On the other hand, our new randomized algorithm can achieve the same covariance error with communication cost $\tilde{O}(d^{2.5})$. Furthermore, we show that $\Omega(d^3)$ is the lower bound for any deterministic algorithms, and thus separate the randomized and deterministic communication complexity.

In our algorithms (except for the distributed PCA algorithm for sparse matrices), each machine only needs to make one pass over the data with limited working space. The algorithms follows the same framework: each machine i first independently computes a local sketch B_i using a streaming algorithm, then all machines run a distributed algorithm on top the local sketches without further access to 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 essentially the same as the *distributed streaming model* (Gibbons and Tirthapura, 2001, 2002). where each machine processes a stream of items with bounded memory, and when a query is requested, the central server, who can communicate with the machines, needs to output the answer over the union of the streams. We call algorithms work in this model *distributed streaming* algorithms, in comparison to *distributed batch* algorithms if each machine needs to access local data multiple times.

1.1 Preliminaries and Notation

Before formally define our problems, we first provide some basic notation that will be repeatedly used and preliminaries on matrices. 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 $d \le n$. 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,

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

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 $[A]_0 = \mathbf{0}$, the all-zero matrix. 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 Problem 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* (Ghashami and Phillips, 2014).

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 (Ghashami and Phillips, 2014). For completeness, we provide a proof in Appendix A.

Lemma 3 (Ghashami and Phillips 2014)

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

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 (Drineas et al., 2006b). Since many matrices of interest in practice can be well approximated by a matrix with a relatively lower rank, $\|A - [A]_k\|_F^2$ could be 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 3. It was also shown that a sketch with the above error guarantee is a projection-cost-preserving sketch, which preserves the distance of the matrix's rows to any k-dimensional subspace (see e.g. Musco and Musco 2020), and can be used to approximately solve a number of low-rank optimization problems e.g., k-means clustering, constrained low rank approximation, etc. Therefore, our main focus is to obtain matrix sketches with covariance error $\varepsilon\|A - [A]_k\|_F^2/k$.

Definition 4 *We call B an (ε, k)-sketch of A if*

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

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

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

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: $\|B\|_F^2 \leq \|A\|_F^2 + O(\|A - [A]_k\|_F^2)$. However, this is typically not a restriction, since our sketch matrix will have rank at most $O(k/\varepsilon)$; and one can easily check that any (ε, k)-sketch of A with rank bounded by $O(k/\varepsilon)$ has Frobenius norm at most $\|A\|_F^2 + O(\|A - [A]_k\|_F^2)$.

Principle component analysis. In this problem, the goal is to find a low-dimensional subspace that captures as much of the variance of a data set as possible. The following approximate version of PCA with Frobenius norm error is widely studied and has applications to distributed k-means and other problems (see e.g. Liang et al. 2014; Boutsidis et al. 2016).

Definition 5 *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 V 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, the matrix V can also be viewed as a type of matrix sketch. By Lemma 3, the top k right singular vectors of any $(\varepsilon/2, k)$-sketch of A satisfy (1). On the other hand, a set of approximate PCs does not directly give good covariance error (although it has low projection error by definition), which is required for many applications (Karnin and Liberty, 2015; Luo et al., 2016; Sharan et al., 2018). The distributed PCA problem studied by Boutsidis et al. (2016) requires all machines to get the same answer, while we only require the coordinator to output the answer. Note the coordinator can broadcast the output to all machines using $O(skd)$ communication cost, which is typically dominated by the cost of computing the answer.

Real number vs word/bit complexity. Matrix sketches typically consist of real numbers, e.g. singular vectors of certain matrices, and the approximation errors are often analyzed assuming infinite precision. It could be quite nontrivial to bound the number of bits needed to encode each real number (Clarkson and Woodruff, 2009; Boutsidis et al., 2016). It may not be an serious concern for practical usage, but is still an important theoretical question, since one can always encode the entire input into a single real number if maximum wordsize is not specified. In this paper, we provide word/bit complexity for communication costs assuming that 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 in the input are integers of magnitude at most $\mathrm{poly}(nd/\varepsilon)$.

1.3 Previous Results

Liberty (Liberty, 2013) adapts a well-known algorithm for finding frequent items, the MG algorithm (Misra and Gries, 1982), to sketching matrices, 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 machine 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 an (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 (Drineas et al., 2006a). Random sampling can be implemented in the distributed model with communication cost $O(s + d/\varepsilon^2)$. However, it has a quadratic dependence on $1/\varepsilon$ and only gives a weaker error bound.

For the distributed PCA problem, one can apply simultaneous power iteration or its improvements (Musco and Musco, 2015), however, the communication cost is suboptimal and such methods also require super constant number of rounds. The current best algorithms are from Boutsidis et al. (2016). See section 1.4 for more details on the communication bounds.

1.4 Our Contributions

In this paper, we give 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 currently there is no bound on maximum wordsize required by the original FD algorithm of Liberty (2013), 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 (recall s is the number of machines). Since the sketch size $O(kd/\varepsilon)$ was shown to be optimal by Woodruff (2014) (up to a $\log$ factor) for an (ε, k)-sketch, it may seem 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 randomized algorithm, which is the first algorithm with communication cost $o(s)\times$sketch size. In particular, for (ε, k)-sketch, our communication cost is $O\left(sdk + \frac{\sqrt{s}dk}{\varepsilon} \cdot \sqrt{\log d}\right)$ words, while the optimal sketch size is $\Theta(dk/\varepsilon)$ (Woodruff, 2014). We achieve this by giving a new algorithm, call *singular value sampling*, which further compresses the local (ε, k)-sketches computed by FD. This new algorithm is applied on top of FD sketches and the machines do not need to access the original data in this step, and thus the combined algorithm is a distributed streaming algorithm. The results are summarized in Table 1.

Directly applying our (ε, k)-sketch algorithm to the PCA problem (by Lemma 3), the cost is also $O(sdk) + \tilde{O}(\sqrt{s}dk/\varepsilon)$ words. Boutsidis et al. (2016) gave an algorithm with communication cost $O\left(sdk + \frac{sk}{\varepsilon^2} \cdot \min\{d, k/\varepsilon^2\}\right)$ words in the distributed model, which is the first algorithm that beats the $O(sdk/\varepsilon)$ bound of Kannan et al. (2014) when d is larger than k/ε^3. They also proved an $\Omega(sdk)$ low bound if all machine are required to know the answer. Note that our result is also

1. Currently there is no word complexity analysis on FD.
2. Our technique only works for communication cost; the space usage of each machine is still in real numbers.

	$\varepsilon\|A\|_F^2$	$\varepsilon\|A-[A]_k\|_F^2/k$
Liberty (2013); Ghashami and Phillips (2014)	$O\left(\frac{sd}{\varepsilon}\right)^*$	$O\left(\frac{skd}{\varepsilon}\right)^*$
Sampling (Drineas et al., 2006a)	$O\left(s+\frac{d}{\varepsilon^2}\right)$	
New	$O\left(\frac{\sqrt{s}d}{\varepsilon}\cdot\sqrt{\log d}\right)$	$O\left(sdk+\frac{\sqrt{s}kd}{\varepsilon}\cdot\sqrt{\log d}\right)$
Deterministic LB	$\Omega\left(\frac{sd}{\varepsilon}\right)$	$\Omega\left(\frac{skd}{\varepsilon}\right)$

Table 1: Communication costs for covariance sketches. $\varepsilon\|A\|_F^2$ and $\varepsilon\|A-[A]_k\|_F^2/k$ are two target error bounds considered in the literature. $*$ indicates the communication cost is in *real numbers*; the costs of our algorithms are in words and the lower bounds are in terms of bits.

Boutsidis et al. (2016)	$O\left(skd+\frac{sk^2}{\varepsilon^4}\right)$
New	$O\left(skd+\frac{\sqrt{s\log d}\cdot k^2}{\varepsilon^3}\right)$
Boutsidis et al. (2016)	$O\left(\frac{sk\phi}{\varepsilon}+\frac{sk^2}{\varepsilon^4}\right)$
New	$O\left(\frac{sk\phi}{\varepsilon}+\frac{sk^2}{\varepsilon}+\frac{\sqrt{s\log(k/\varepsilon)}\cdot k^2}{\varepsilon^2}\right)$

Table 2: Communication costs for distributed PCA (i.e., for computing $(1+\varepsilon)$-approximate top-k principle components). The communication costs are in words. ϕ is the row sparsity of the input matrix.

$o(sdk/\varepsilon)$ (ignoring the $\sqrt{\log d}$ factor). 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 using the algorithm of Boutsidis et al. (2016).

In the row-partition model, Boutsidis et al. (2016) also show that it is possible to bypass the $\Theta(skd)$ communication bound if each row of the input matrix has low sparsity (number of nonzero entries). In particular, they provided a distributed PCA algorithm with communication cost $O\left(\frac{sk\phi}{\varepsilon}+\frac{sk^2}{\varepsilon^4}\right)$ words, where ϕ is the maximum row sparsity. When the dimension d is very large and $\phi \ll d$, the cost of this algorithm could be significantly smaller. Based on our new distributed covariance sketch algorithm, we give an improved communication bound in this setting as well. We show that there is a distributed algorithm with communication cost $O\left(\frac{sk\phi}{\varepsilon}+\frac{sk^2}{\varepsilon}+\frac{\sqrt{s\log(k/\varepsilon)}\cdot k^2}{\varepsilon^2}\right)$ words. The improvement is significant when ϕ is small. Our new algorithm uses the adaptive sampling technique from Boutsidis et al. (2016) and it is unclear whether it has a one-pass implementation. Therefore, those improved algorithms for sparse matrices are currently not distributed streaming algorithms. The communication bounds for distributed PCA are summarized in Table 2.

1.5 Other Related Work

The problem of computing covariance matrix sketch was widely studied in the row-wise update streaming model (Liberty, 2013; Ghashami and Phillips, 2014; Ghashami et al., 2016; Wei et al.,

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

2016). Optimal space lower bounds was proved in Woodruff (2014). In the distributed setting, there was no improvement in communication cost since Liberty (2013). Ghashami et al. (2014b); Zhang et al. (2017) 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 Feldman et al. (2013); Kannan et al. (2014); Liang et al. (2014, 2016); Bhojanapalli et al. (2015); Boutsidis et al. (2016). PCA is closely related to the Euclidean k-means problem; its distributed version is studied in Balcan et al. (2013); Ding et al. (2016). Zhang et al. (2015) investigated the distributed generalized matrix rank problem. Motivated by distributed Newton's method, Derezinski and Mahoney (2019); Dereziński et al. (2020) considered the problem of computing an estimate of the inverse Hessian multiplied by the gradient and the Hessian in many machine learning applications is the covariance matrix. The main difficulty is that the sum of the inverses does not equal the inverse of the sum. However, they considered the setting where each single machine has access to a subsampled version of the loss function, while our model doesn't make any assumptions on the data distribution. Wang et al. (2018) also studied approximate newton method for distributed optimization and show that if each machine locally holds a random sample of the training data and the data is incoherent, the sum of the local inverse Hessians is very close to the inverse of the global Hessian. Distributed matrix computations were recently studied for serverless systems (Gupta et al., 2019, 2020), where the main challenge is to make algorithms resilient against stragglers, i.e., a subset of much slower machines which can be a result of limited availability of shared resources, hardware failure, network latency (Dean and Barroso, 2013). Streaming numerical linear algebra problems were studied in Clarkson and Woodruff (2009). The communication complexity of boolean matrix multiplication in the two-party model was studied in Van Gucht et al. (2015). Li et al. (2014) proved multi-party communication lower bounds for several linear algebraic problems in the message passing model.

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 machine 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 (2013), denoted as FD. Ghashami and Phillips (2014) gave an improved analysis, which is summarized in the following theorem.

Theorem 6 (Liberty (2013); Ghashami and Phillips (2014)) *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*

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

The original FD algorithm of Liberty (2013) is deterministic and has running time $O(ndk/\varepsilon)$ to process an $n \times d$ matrix. We note that the working space of FD is in terms of real numbers rather than words; it is still unclear how one can obtain a space bound in terms of words/bits (Boutsidis et al., 2016; Cohen et al., 2017).

One nice property of FD is that it is mergeable (Agarwal et al., 2013). Informally speaking, 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 covariance error no larger than $C = \mathsf{FD}([A; A'], \varepsilon, k)$. By induction, the number of sketches to be merged can be arbitrary. This mergeable property makes FD directly applicable to the distributed model, i.e., each machine sketches its local matrix independently, and sends the covariance sketch to the coordinator; the coordinator simply combines the sketches running another FD algorithm. Therefore, the algorithm is deterministic and works in the distributed streaming model. Note that the total communication cost is $O(skd/\varepsilon)$ real numbers. We will show how to improve the communication cost to $O(skd/\varepsilon)$ words in section 4, and here we summarize our deterministic upper bound as follows.

Theorem 7 *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 machine 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 (Kushilevitz and Nisan, 1997) 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 some constant $\sigma \leq 1$ to be determined later. In our lower bound proof, each machine 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 8 *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*

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

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 $U = \{i \mid |B_i| \geq 2^{(1-2\beta)td}\}$. We have the following simple property.

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

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

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

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

Note that each B_i is a subset of $\{-1, +1\}^{t \times d}$; we define $B_{i,j}$ to 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

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

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 by Huang and Yi (2017).

Lemma 9 (Huang and Yi (2017)) *Assume x is distributed uniformly in $\{-1, +1\}^d$, $L \subseteq \{-1, +1\}^d$ 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 9 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

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

Hence,

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

which completes the proof. ■

Obviously, for any x

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

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 exists 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

$$\begin{aligned}\|A^TA-A'^TA'\|_2&=\max_{x\in\mathbb{R}^d}\frac{\left|\|Ax\|^2-\|A'x\|^2\right|}{\|x\|^2}\\&\geq\frac{\left|\sum_{i=1}^{s}\|M^{(i)}x^*\|^2-\sum_{i=1}^{s}\|W^{(i)}x^*\|^2\right|}{\|x^*\|_2^2}\\&=\Omega(sd)-st,\end{aligned}$$

which proves the lemma. ■

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

Theorem 10 *Let $A\in\{-1,+1\}^{n\times d}$ be the input matrix which is row-partitioned across s machines. 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 machine 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 8, there exist A and A' in ρ_i such that $\|A^TA - A'^TA'\|_2 = \Omega(sd) - st$. By our assumption, $t \leq \sigma d$, and thus $\|A^TA - A'^TA'\|_2 > 2\varepsilon\|A\|_F^2$ for sufficiently small σ. Let X be the output corresponding to ρ_i. We have

$$\begin{aligned}\|A^TA - X^TX\|_2 + \|A'^TA' - X^TX\|_2 \geq &\|A^TA - A'^TA'\|_2\\ >&2\varepsilon\|A\|_F^2.\end{aligned}$$

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 $1/\varepsilon \geq d$.

Corollary 11 *If $k/\varepsilon \leq d$, the deterministic communication complexity of computing an (ε, k)-sketch matrix is lower bounded by $\Omega(skd/\varepsilon)$ bits.*

Proof The covariance error allowed for an (ε, k)-sketch is at most $\varepsilon\|A - [A]_k\|_F^2/k \leq \varepsilon\|A\|_F^2/k$, meaning an (ε, k)-sketch is also an $(\varepsilon/k, 0)$-sketch. So, Theorem 10 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.

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 (Drineas et al., 2006a; Oliveira, 2010; Drineas et al., 2011). 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 $\Theta(d/\alpha)$ Woodruff (2014)). 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 4.

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

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 machine i runs SVS on the input matrix $A^{(i)}$, and then sends the output to the coordinator (Algorithm 2). The coordinator computes the final sketch matrix using FD with working space $O(d/\alpha)$ (Algorithm 3).

Note that, in our algorithm, each machine also performs row sampling. The differences between our algorithms and the row sampling algorithms from Drineas et al. (2006a); Oliveira (2010); Drineas et al. (2011) for covariance sketches 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: Set $x_j = 1$ with probability $g(\sigma_j^2)$, and $x_j = 0$ with probability $1 - g(\sigma_j^2)$
4: Let $w_j = \sigma_j / \sqrt{g(\sigma_j^2)}$
5: 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
6: Remove zero rows in B
7: Output B

Algorithm 2 Algorithm for machine 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

Algorithm 3 Algorithm on coordinator. Input: $B^{(i)}, i = 1 \cdots s$.

1: $B' = \mathsf{FD}([B^{(1)}; \cdots B^{(s)}], \alpha, 0)$
2: **Return** B'

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 12 *Let $A^{(i)}$ be the input of the i-th machine, and $B^{(i)}$ be matrix sent by machine 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:

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

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, Oliveira (2010); Drineas et al. (2011) used a matrix concentration inequality, while Drineas et al. (2006a) 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 with high rank and are not i.i.d. now, so we cannot apply the same inequality used in Oliveira (2010); Drineas et al. (2011). Instead, we use *Matrix Bernstein Inequality* (see Tropp (2012)).

The main theorem follows from the following three claims, which are properties 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

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

Therefore,

$$\mathsf{E}[B^TB] \;=\; \mathsf{E}[\sum_{j=1}^{d} x_j^2 w_j^2 v_j v_j^T] = \sum_{j=1}^{d} \mathsf{E}[x_j^2] w_j^2 v_j v_j^T = \sum_{j=1}^{d} \sigma_j^2 v_j v_j^T = A^TA,$$

which means B^TB is an unbiased estimator of A^TA. ■

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

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

Proof By (4), it follows that

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

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

$$\lambda_{\max}(B^TB - A^TA) = \max_j (x_j^2 w_j^2 - \sigma_j^2) \leq \max w_j^2 = \max_j \frac{\sigma_j^2}{g(\sigma_j^2)},$$

which proves the claim. ■

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 (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, $\mathsf{E}[x_j^2 w_j^2] = \sigma_j^2$, 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 $g(\sigma_j^2) \leq 1$, the eigenvalues are all non-negative. It follows that

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

which completes the proof. ■

Now we are ready to prove the main theorem.

Proof (of Theorem 12) To prove this theorem, we will use the following *Matrix Bernstein Inequality*, which can be found in e.g., (Tropp, 2012) (Theorem 6.1).

Lemma 13 (Matrix Bernstein) *Consider a finite sequence $\{X_k\}$ of independent, random, self-adjoint (or Hermitian) matrices with dimension d. Assume that*

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

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

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

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 12. Using Claim 5, we can bound σ^2:

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

Now we can directly use Lemma 13 and prove that

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

To establish the theorem, we still need to show

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

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 12. 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 14 (Linear) *If we set*

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

then with probability $1 - \delta$

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

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 suboptimal. We show that a less intuitive quadratic function gives a better bound on the communication cost.

Theorem 15 (Quadratic) *If we set*

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

then with probability $1 - \delta$,

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

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

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

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

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

For the i-th machine, 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 $\bar{\Sigma}$:

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

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

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

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

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

This is the error resulting from dropping all the small singular values. By triangle inequality, it is now sufficient to bound $\|\bar{A}^T\bar{A} - B^TB\|_2 \leq \alpha\|A\|_F^2$. 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 12 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

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

Since $\kappa^2 \leq s/b = \alpha^2\|A\|_F^4/\log\frac{d}{\delta}$ (Eqn. (6)). By Theorem 12 with $t = \alpha\|A\|_F^2$, we get

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

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

$$\|B^TB - A^TA\| \leq \|B^TB - \bar{A}^T\bar{A}\| + \|A^TA - \bar{A}^T\bar{A}\| \leq 2\alpha\|A\|_F^2$$

with probability at least $1 - \delta$.

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

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

Then the communication cost is

$$d \cdot \sum_{i,j} g(\sigma_{i,j}^2) \leq d \cdot \sum_{i,j} \frac{\sqrt{s}\sigma_{i,j}^2}{\alpha \|A\|_F^2} \cdot \sqrt{\log \frac{d}{\delta}} = \frac{\sqrt{s}d}{\alpha} \cdot \sqrt{\log \frac{d}{\delta}},$$

which completes the proof. ■

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 machine i invokes FD to compute a local sketch $B^{(i)}$ in one pass, i.e. $B^{(i)} = \mathsf{FD}(A^{(i)}, \varepsilon, k)$ (Theorem 6), 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 $A^{(i)}$ Liberty (2013). From this property, it is not difficult prove the following lemma.

Lemma 16 *Assume $B = \mathsf{FD}(A, \varepsilon, k)$, then*

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

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

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

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 $\|A - [A]_k\|_F^2 = \|A\|_F^2 - \sum_{i=1}^{k} \|Au_i\|^2$. ■

The following lemma directly follows from singular value decomposition.

Lemma 17 *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,

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

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. ■

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

Algorithm 4 Algorithm on machine i. Input: $A^{(i)} \in \mathbb{R}^{n_i \times d}$ and sampling function g.

1: $B^{(i)} = \mathsf{FD}(A^{(i)}, \varepsilon, k)$
2: $(T^{(i)}, R^{(i)}) = \mathsf{Decomp}(B^{(i)}, k)$
3: $W^{(i)} = \mathsf{SVS}(R^{(i)}, g)$
4: Send $Q^{(i)} = [T^{(i)}; W^{(i)}]$ to the coordinator

Algorithm 5 Algorithm on coordinator. Input: $Q^{(i)} = [T^{(i)}; W^{(i)}], i = 1 \cdots s$.

1: $B = \mathsf{FD}([Q^{(1)}; \cdots Q^{(s)}], \varepsilon, k)$
2: **Return** B

In our algorithm (see Algorithm 4), each machine 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 $\|A^T A - B^T B\|_2 \le \varepsilon \|A - [A]_k\|_F^2 / k$. From Lemma 17, we have

$$\|A^T A - T^T T - R^T R\|_2 \le \varepsilon \|A - [A]_k\|_F^2 / k, \text{ and } \|R\|_F^2 = \sum_{i=1}^{s} \|R^{(i)}\|_F^2 = \sum_{i=1}^{s} \|B^{(i)} - [B^{(i)}]_k\|_F^2.$$

Then by Lemma 16, we get

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

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

$$\sum_i \|A^{(i)} - [A^{(i)}]_k\|_F^2 \le \sum_i \|A^{(i)} - [A]_k^{(i)}\|_F^2 = \|A - [A]_k\|_F^2, \tag{10}$$

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

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

Now, each machine 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 15, we have

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

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

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

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 15, we also have $\|Q\|_F^2 = \|A\|_F^2 + O(\|A - [A]_k\|_F^2)$.

Theorem 18 *There is a distributed streaming algorithm which computes an (ε, k)-sketch Q with probability $1 - \delta$. The communication cost is $O(sdk + \frac{\sqrt{s}kd}{\varepsilon} \cdot \sqrt{\log \frac{d}{\delta}})$, and space used by each machine 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 $\|Q^T Q - Q'^T Q'\|_2 \leq \varepsilon \|Q - [Q]_k\|_F^2/k$, 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 16, it can be shown that $\|Q - [Q]_k\|_F^2 = O(\|A - [A]_k\|_F^2)$. As a result, it holds that

$$\|A^T A - Q'^T Q'\|_2 \leq O(\varepsilon) \cdot \|A - [A]_k\|_F^2/k.$$

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

4. Bit Complexity for Communication Costs

So far, the communication cost of our algorithms are in real numbers. In this section, we discuss how to obtain word/bit complexity. Similar to Boutsidis et al. (2016), 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 (thus onto the row space of $A^{(i)}$). Hence, if x belongs to the row space of $A^{(i)}$, $Q^+ Q x = x$. In particular, we have $Q^+ Q A^{(i)T} = A^{(i)T}$.

Based on the above observation, each machine i runs the following algorithm. Machine 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 recover A^TA exactly. For the communication cost, Q takes at most $2kd$ words, since Q consists of 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 $k \le d$, the total communication cost is $O(skd)$ words.

Algorithm 6 One-pass algorithm on machine i for case 1. Input: $A^{(i)} \in \mathbb{R}^{n_i \times d}$.

Initialize $Q^{(i)} = \{A_1^{(i)}\}, V = \{\frac{A_1^{(i)}}{\|A_1^{(i)}\|_2}\}, Z = \|A_1^{(i)}\|_2^2$
for $t = 2$ **to** n_i **do**
  if the t-th row $A_t^{(i)}$ is not in the span of $Q^{(i)}$ **then**
    Insert $A_t^{(i)}$ into $Q^{(i)}$
    Compute an unit vector u which is orthogonal to V but in the span of $Q^{(i)}$ (Gram–Schmidt)
    $U = V[V; u]^T$
    Insert u into V ▶ V is an orthonormal basis of $Q^{(i)}$
    $Z = U^TZU$ ▶ $Z = VA_{1:t-1}^{(i)T}A_{1:t-1}^{(i)}V^T$
    $Z = Z + VA_t^{(i)T}A_t^{(i)}V^T$ ▶ $Z = VA_{1:t}^{(i)T}A_{1:t}^{(i)}V^T$ (here $A_t^{(i)}$ is treated as a row vector)
  end if
end for
Send $C^{(i)} = Q^{(i)}V^TZVQ^{(i)T}$ and $Q^{(i)}$ to the coordinator

Naively, it requires two passes on each machine: one pass for computing Q and one pass for $QA^{(i)T}A^{(i)}Q^T$. We next describe how to implement the algorithm in one pass using $O(kd)$ space. See Algorithm 6 for the details. 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. Q and V are also viewed as row matrices whose *rows* consists of the vectors in Q and V respectively. 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)[5]: when a new row u is added to V, compute $U = V[V; u]^T$ (with $O(k^2)$ space, since the number of rows in V will never exceed $2k$) and then update Z as $Z = U^TZU$ and then $Z = Z + VA_t^{(i)T}A_t^{(i)}V^T$. In the end, we compute QV^TZVQ^T (using $O(kd)$ space), which is $QA^{(i)T}A^{(i)}Q^T$. Here we have used the fact that $aV^TV = a$ when a is a row vector in the row space of V. See Algorithm 6 for more details. After receiving $C^{(i)} = QA^{(i)T}A^{(i)}Q^T$ and Q from all machines, the coordinator now can compute A^TA exactly. But, naively, it takes $O(d^2)$ working space. Therefore, we directly compute the low rank factorization of A^TA instead, i.e., compute $B \in \mathbb{R}^{2k \times d}$ such that $B^TB = A^TA$ using FD. Details is presented in Algorithm 7. For the correctness, denote $B^{(i)} = \left(C^{(i)}\right)^{1/2}\left(Q^{(i)+}\right)^T = \left(QA^{(i)T}A^{(i)}Q^T\right)^{1/2}\left(Q^{(i)+}\right)^T$; then one can easily verify that $B^{(i)T}B^{(i)} = A^{(i)T}A^{(i)}$. The coordinator computes B using FD, i.e., $B = \mathsf{FD}\left([B^{(1)}, \cdots, B^{(s)}], 1, 2k\right)$, so the space and time complexity are $O(kd)$ and $O(sk^2d)$. Since the rank of $[B^{(1)}, \cdots, B^{(s)}]$ is at most $2k$, the covariance error of B is $\|B - [B]_{2k}\|_F^2 = 0$, and thus $B^TB = \sum_{i=1}^s B^{(i)T}B^{(i)} = A^TA$.

5. Note V may contain exponentially small entries Boutsidis et al. (2016), so we cannot send Z directly.

Algorithm 7 Algorithm on coordinator for case 1. Input: $C^{(i)}, Q^{(i)}, i = 1, \cdots s$

1: Initialize $B = 0$
2: **for** $i = 1$ **to** s **do**
3: Compute $Q^{(i)+}$ and $D^{(i)} = \left(C^{(i)}\right)^{1/2}$
4: $B = \mathsf{FD}\left([B; D^{(i)}\left(Q^{(i)+}\right)^T], 1, 2k\right)$
5: **end for**
6: **Return** B

Case 2*:* $\mathsf{rank}(A) > 2k$. In this case, each machine i first computes a matrix $Q^{(i)}$ as in Section 3 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 (Boutsidis et al., 2016), 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 19 (Lemma 4.1 **of Clarkson and Woodruff 2009)** *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

$$\|A - [A]_k\|_F^2 \geq \sigma_{k+1}^2 \geq (nd/\varepsilon)^{-b} \tag{12}$$

for some constant $b > 0$. 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, if we round each entry of Q to a sufficiently small additive $\mathrm{poly}^{-1}(nd/\varepsilon)$ precision, and let $\tilde{Q}$ be the matrix after rounding, then $\|Q^TQ - \tilde{Q}^T\tilde{Q}\|_2 \leq \mathrm{poly}^{-1}(nd/\varepsilon) \leq O(\varepsilon) \cdot \|A - [A]_k\|_F^2$. Now, each entry of $\tilde{Q}$ 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.

5. Distributed PCA

In this section, we show how to use our distributed sketching algorithm to obtain improved communication bounds for distributed PCA. All algorithms in this section will be randomized with success probability at least 0.9.

5.1 Distributed PCA for Dense Matrices

To solve the distributed PCA problem, we can use Theorem 18 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 machines need to output the same answer, a lower bound of $\Omega(skd)$ bits was proved in Boutsidis et al. (2016). Since it takes $O(skd)$ communication for the coordinator to broadcast the answer to all machines, 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 Boutsidis et al. (2016).

Theorem 20 (Distributed PCA of Boutsidis et al. 2016) *Given any $A \in \mathbb{R}^{n\times d}$ which is distributed across s machines, there is a batch algorithm for PCA with communication cost*

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

In our covariance sketch algorithm, each machine 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. In Lemma 21, we show that only approximate singular vectors of B are needed. Therefore, we can run the algorithm of Boutsidis et al. (2016) 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 $O(skd) + \tilde{O}(\sqrt{s}k/\varepsilon \cdot \min\{d, k/\varepsilon^2\})$. A standard implementation of the algorithm of Boutsidis et al. (2016) needs to access the input multiple times; our combined algorithm is a distributed streaming algorithm (with $O(kd/\varepsilon)$ working space on each machine), since the algorithm of Boutsidis et al. (2016) is only applied on top of a distributed sketch. We remark that the distributed PCA algorithm of Boutsidis et al. (2016) works for *arbitrary partition* model, where each machine 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.

The key is the following lemma, the proof of which can be found in section 5.1.1.[6]

Lemma 21 *For any $\xi \geq 0$, and let B be a matrix such that $\|B\|_F^2 \leq \|A\|_F^2 + O(\|A - [A]_k\|_F^2) + \xi$ and $\|A^TA - B^TB\|_2 \leq \frac{\varepsilon}{2k}\|A - [A]_k\|_F^2 + \frac{\xi}{k}$. Let $V \in \mathbb{R}^{d\times k}$ be any orthonormal matrix satisfying $\|B - BVV^T\|_F^2 \leq (1+\varepsilon)\|B - [B]_k\|_F^2$, then*

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

In this section, we only need a special case of the above lemma with $\xi = 0$; the more general version will be used in the analysis for sparse matrices.

Corollary 22 *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 corollary can be viewed as a robust version of Lemma 3. With this result, we can apply the standard "sketch-and-solve" framework to solve the distributed PCA problem.

1. In the "sketch" step, all machines compute a distributed (ε, k)-sketch, i.e., each machine i output a matrix $Q^{(i)}$, such that $Q = [Q^{(1)}; \cdots ; Q^{(s)}]$ is an (ε, k)-sketch of A.
2. In the "solve" step, we can apply any communication-efficient distributed PCA algorithm on the input Q.

6. We remark that a similar result has been proved by Cohen et al. (2017) for a relevant problem. Their proof is quite technical, so we provide a direct proof for our application here.

Note that, in our algorithm for (ε, k)-sketch, if we do not require machines to send their local sketches to the coordinator, the communication cost is negligible[7], and the number of rows in Q is $\tilde{O}(\sqrt{s}kd/\varepsilon)$. In the "solve" step, as long as the distributed PCA algorithm approximately solves the PCA problem for Q, the output is also a valid solution for the original input A due to Corollary 22 (where we also use the property that $\|Q\|_F^2 = \|A\|_F^2 + O(\|A - [A]_k\|_F^2)$ from Theorem 18). 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 machine 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 from Boutsidis et al. (2016) 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 20 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 4, entries in Q can be rounded so that each only takes $O(\log(nd/\varepsilon))$ bits to represent, 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 Boutsidis et al. (2016).

Theorem 23 *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 machine is $O(dk/\varepsilon)$ real numbers.*

5.1.1 PROOF OF LEMMA 21

Proof By assumption, we have $\|A^TA - B^TB\|_2 \le \frac{\varepsilon}{2k}\|A - [A]_k\|_F^2 + \frac{\xi}{k}$, which is equivalent to

$$\max_{x:\|x\|=1} |\|Ax\|^2 - \|Bx\|^2| \le \frac{\varepsilon}{2k}\|A - [A]_k\|_F^2 + \frac{\xi}{k}. \tag{13}$$

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

$$\begin{aligned}
\|B - [B]_k\|_F^2 &= \|B\|_F^2 - \sum_{i=1}^{k} \|Bu_i\|^2 \\
&\le \|A\|_F^2 + O(\|A - [A]_k\|_F^2) + \xi - \sum_{i=1}^{k} \|Bu_i\|^2 \\
&\le \|A\|_F^2 + O(\|A - [A]_k\|_F^2) + \xi - \sum_{i=1}^{k} \|Bw_i\|^2 \\
&\le \|A\|_F^2 + O(\|A - [A]_k\|_F^2) + \xi - \sum_{i=1}^{k} \|Aw_i\|^2 + k\frac{\varepsilon}{2k}\|A - [A]_k\|_F^2 + k\frac{\xi}{k} \quad \text{by (13)} \\
&\le O(\|A - [A]_k\|_F^2) + 2\xi. \qquad (14)
\end{aligned}$$

7. In fact, all the computations are local and parallel; the only communication needed is to synchronize the same sampling function g.

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

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

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

$$\sum_{i=1}^{k} \|Bv_i\|^2 \geq \sum_{i=1}^{k} \|Bu_i\|^2 - \varepsilon\|B - [B]_k\|_F^2 \geq \sum_{i=1}^{k} \|Bu_i\|^2 - O(\varepsilon)\|A - [A]_k\|_F^2 - 2\varepsilon\xi. \quad (15)$$

The last inequality is from (14). Then,

$$\begin{aligned}
\|A - AVV^T\|_F^2 =& \|A\|_F^2 - \|AVV^T\|_F^2 = \|A\|_F^2 - \sum_{i=1}^{k} \|Av_i\|^2 \\
\leq& \|A\|_F^2 - \sum_{i=1}^{k} \|Bv_i\|^2 + k \cdot \frac{\varepsilon}{2k}\|A - [A]_k\|_F^2 + k\frac{\xi}{k} \\
\leq& \|A\|_F^2 - \sum_{i=1}^{k} \|Bu_i\|^2 + O(\varepsilon)\|A - [A]_k\|_F^2 + (1+2\varepsilon)\xi \quad \text{(by (15))} \\
\leq& \|A\|_F^2 - \sum_{i=1}^{k} \|Bw_i\|^2 + O(\varepsilon)\|A - [A]_k\|_F^2 + (1+2\varepsilon)\xi \\
& \qquad\qquad\qquad\qquad\qquad\qquad \text{(as } \sum_{i=1}^{k} \|Bu_i\|^2 \geq \sum_{i=1}^{k} \|Bw_i\|^2\text{)} \\
\leq& \|A\|_F^2 - \sum_{i=1}^{k} \|Aw_i\|^2 + O(\varepsilon)\|A - [A]_k\|_F^2 + (2+2\varepsilon)\xi \quad \text{by (13)} \\
=& \|A - [A]_k\|_F^2 + O(\varepsilon)\|A - [A]_k\|_F^2 + O(1)\cdot\xi,
\end{aligned}$$

where the second inequality is by (15) ■

5.2 Distributed PCA for Sparse Matrices

In this subsection, we assume each row of the input matrix is ϕ-sparse, i.e., has at most ϕ nonzero entries. The key building block to bypass the skd bound in the algorithm from Boutsidis et al. (2016) is the follow result.

Lemma 24 (Boutsidis et al. 2016) *There is a randomized distributed algorithm that outputs a subset of rows from A, denoted as $C \in \mathbb{R}^{r\times d}$, such that, with probability at least 0.98, it satisfies*

$$\min_{X:\mathsf{rank}(X)\le k} \|A - XC\|_F^2 \le (1+\varepsilon)\|A-[A]_k\|_F^2,$$

The communication cost is $O\left(sk\phi + \frac{k\phi}{\varepsilon}\right)$ words and the number of rows in C is $r = O\left(\frac{k}{\varepsilon}\right)$.

Our algorithm works as follows.

1. Apply the algorithm from Lemma 24 so that the coordinator obtains the matrix C with $r = O\left(\frac{k}{\varepsilon}\right)$ rows and $O\left(\frac{k\phi}{\varepsilon}\right)$ nonzero entries.
2. The coordinator sends C to all machines.
3. Each machine i computes an orthonormal basis of the row space of C (as column vectors), denoted as $U \in \mathbb{R}^{d\times r}$, and then computes $A_i' = A_i U$.
4. All machines computes a distributed (ε, k)-sketch for $A' = [A_1', \cdots, A_s']$ (but do not send the sketches to the coordinator), denoted as $B = [B_1, \cdots, B_s]$.
5. Each machine i rounds the entries in B_i down (in absolute value) to the nearest multiple of $(nd/\varepsilon)^{-c}$ for a sufficient large constant c, and let $\tilde{B}_i$ be the rounded version; then sends $\tilde{B}_i$ to the coordinator.
6. The coordinator computes top-k right singular vectors of $\tilde{B} = [\tilde{B}_1, \cdots, \tilde{B}_s]$, and let $Q \in \mathbb{R}^{r\times k}$ be the matrix whose columns are the top-k right singular vectors of $\tilde{B}$. The coordinator outputs $V = UQ \in \mathbb{R}^{d\times k}$.

Correctness. From the analysis in Section 4, when $\mathsf{rank}(A) \le 2k$, there is an exact algorithm with communication cost $O(sk^2)$ words. Therefore, we only need to consider the case when $\mathsf{rank}(A) > 2k$.

Claim 6 *Assume $\mathsf{rank}(A) > 2k$, then, for a large enough constant c in step 5, the covariance error of $\tilde{B}$ with respect to A' is*

$$\|A'^T A' - \tilde{B}^T\tilde{B}\|_2 \le \frac{\varepsilon}{2k}\|A' - [A']_k\|_F^2 + \frac{\varepsilon}{2k}\|A-[A]_k\|_F^2.$$

Moreover, $\|\tilde{B}\|_F^2 \le \|A'\|_F^2 + O\left(\|A'-[A']_k\|_F^2\right)$.

Proof Since U is an orthonormal matrix and the entries in A are upper bounded by $\mathrm{poly}(nd/\varepsilon)$, the entries in A' are also bounded by $\mathrm{poly}(nd/\varepsilon)$. From step 4, we have $\|A'^T A' - B^T B\|_2 \le \frac{\varepsilon}{2k}\|A'-[A']_k\|_F^2 \le \mathrm{poly}(nd/\varepsilon)$. Therefore, the entries in B must also be bounded by $\mathrm{poly}(nd/\varepsilon)$, since otherwise its covariance error w.r.t. A' must be too large. It follows that $\|B^T B - \tilde{B}^T\tilde{B}\|_2 \le \mathrm{poly}(nd/\varepsilon)\cdot(nd/\varepsilon)^{-c}$. By (12), $\|A-[A]_k\|_F^2 \ge (nd/\varepsilon)^{-b}$ for some constant b, so if c is large enough, it holds that $\|B^T B - \tilde{B}^T\tilde{B}\|_2 \le \frac{\varepsilon}{2k}\|A-[A]_k\|_F^2$. Then, the first part of the lemma follows from the triangle inequality for spectral norm. For the second part, we know from Theorem 18 that $\|B\|_F^2 \le \|A'\|_F^2 + O\left(\|A'-[A']_k\|_F^2\right)$; and since we always round down the absolute values of entries in B, $\|\tilde{B}\|_F^2 \le \|B\|_F^2$. Then, the second part follows. ■

Claim 7 *Let A', Q be the matrices computed in step 3 and 6. We have*

$$\|A' - A'QQ^T\|_F^2 \leq (1+\varepsilon)\|A' - [A']_k\|_F^2 + O(\varepsilon) \cdot \|A - [A]_k\|_F^2.$$

Proof As Q contains the top-k right singular vectors of $\tilde{B}$, it holds that $\|\tilde{B} - \tilde{B}QQ^T\|_F^2 = \|\tilde{B} - [\tilde{B}]_k\|_F^2$. By Claim 6 and applying Lemma 21 with $\xi = \varepsilon\|A - [A]_k\|_F^2$, we have

$$\|A' - A'QQ^T\|_F^2 \leq (1+\varepsilon)\|A' - [A']_k\|_F^2 + O(\varepsilon) \cdot \|A - [A]_k\|_F^2,$$

which proves the claim. ■

The following lemma shows the correctness of the algorithm.

Lemma 25 *Let V be the matrix computed in step 3 and 6. We have*

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

Proof Since UU^T is an orthogonal projection, the row space of $A - AUU^T$ and $AUU^T - AUQQ^TU^T$ are orthogonal. Then,

$$\begin{aligned}
\|A - AVV^T\|_F^2 =& \|A - AUQQ^TU^T\| = \|A - AUU^T + AUU^T - AUQQ^TU^T\|_F^2 \\
=& \|A - AUU^T\|_F^2 + \|AUU^T - AUQQ^TU^T\|_F^2. && \text{(by Pythagorean theorem)} \\
=& \|A - AUU^T\|_F^2 + \|AU - AUQQ^T\|_F^2 && (U^T \text{ has orthonormal rows}) \\
=& \|A - AUU^T\|_F^2 + \|A' - A'QQ^T\|_F^2 && \text{(note that } A' = AU) \\
\leq& \|A - AUU^T\|_F^2 + (1+\varepsilon)\|A' - [A']_k\|_F^2 + O(\varepsilon) \cdot \|A - [A]_k\|_F^2 && \text{(by Claim 7)} \\
=& \|A - AUU^T\|_F^2 + \|AU - [AU]_k\| + O(\varepsilon) \cdot \|A - [A]_k\|_F^2 && (16)
\end{aligned}$$

Let $Z = \arg\min_{X:\mathsf{rank}(X)\leq k}\|A - XU^T\|_F^2$. Because U^T has the same row space as C, then

$$\|A - ZU^T\|_F^2 = \min_{X:\mathsf{rank}(X)\leq k}\|A - XC\|_F^2 \leq (1+\varepsilon)\|A - [A]_k\|_F^2, \tag{17}$$

where the inequality is from Lemma 24. Since $\mathsf{rank}(Z) \leq k$, we also have

$$\|AU - [AU]_k\| \leq \|AU - Z\|_F^2 = \|AUU^T - ZU^T\|_F^2 \quad (U^T \text{ has orthonormal rows}).$$

Then continuing from (16), we have

$$\begin{aligned}
\|A - AVV^T\|_F^2 &\leq \|A - AUU^T\|_F^2 + \|AUU^T - ZU^T\|_F^2 + O(\varepsilon) \cdot \|A - [A]_k\|_F^2 \\
&= \|A - ZU^T\|_F^2 + O(\varepsilon) \cdot \|A - [A]_k\|_F^2 && \text{(by Pythagorean theorem)} \\
&\leq (1+\varepsilon)\|A - [A]_k\|_F^2 + O(\varepsilon) \cdot \|A - [A]_k\|_F^2 && \text{(by (17))} \\
&= (1 + O(\varepsilon))\|A - [A]_k\|_F^2,
\end{aligned}$$

which completes the proof. ■

Communication cost. The communication cost of step 1 is $O\left(sk\phi + \frac{k\phi}{\varepsilon}\right)$ words by Lemma 24 and the cost of step 2 is $O\left(\frac{sk\phi}{\varepsilon}\right)$ words. In step 4, since the sketches are not sent to the coordinator, the communication cost is negligible. Note that the matrix A' has $O\left(\frac{k}{\varepsilon}\right)$ columns, so the distributed sketch matrix B in step 4 is an $O\left(sk + \frac{\sqrt{s\log(k/\varepsilon)}\cdot k}{\varepsilon}\right) \times O\left(\frac{k}{\varepsilon}\right)$ matrix (Theorem 18). Thus, the rounded version $\tilde{B}$ can be encoded using $O\left(\frac{sk^2}{\varepsilon} + \frac{\sqrt{s\log(k/\varepsilon)}\cdot k^2}{\varepsilon^2}\right)$ words (note that each entry in $\tilde{B}$ is upper bounded by $\mathrm{poly}(dn/\varepsilon)$ from the proof of Claim 6) and the communication cost in step 5 is $O\left(\frac{sk^2}{\varepsilon} + \frac{\sqrt{s\log(k/\varepsilon)}\cdot k^2}{\varepsilon^2}\right)$ words. There is no communication in step 3 and 6.

We summarize the main result on distributed PCA for sparse matrices in the following theorem.

Theorem 26 *Given $A \in \mathbb{R}^{n\times d}$ with row sparsity ϕ, the above distributed algorithm correctly solves PCA for A and the communication cost is $O\left(\frac{sk\phi}{\varepsilon} + \frac{sk^2}{\varepsilon} + \frac{\sqrt{s\log(k/\varepsilon)}\cdot k^2}{\varepsilon^2}\right)$ words.*

We remark that the $\frac{sk\phi}{\varepsilon}$ term in the communication is the cost needed to broadcast C to all s machines. In the blackboard model, where each message is seen by all machines, the cost of this step is $O(k\phi/\varepsilon)$ words. Note that the matrix U may contain exponentially small entries, and thus it is possible that the entries in A' are not representable by $\mathrm{polylog}(nd/\varepsilon)$-bit words. So we cannot use Theorem 18 directly. Instead, we have used a more careful argument to obtain the word complexity.

6. Numerical Simulations on Synthetic Data

In this section, we evaluate of the trade-off between covariance error and communication cost on synthetic data generated from a natural statistical model. The communication cost is measured in terms of the total number of rows sent. All real numbers are stored as double-precision floats. Given an input matrix, the accuracy of an approximation matrix B is measure by $\|A^TA - B^TB\|$. For randomized algorithms, the reported errors and costs are the average values of 10 independent executions. For deterministic algorithms, the actual errors they incurs could be much smaller than their worst-case guarantees; in our evaluations, we always report their actual errors in each experimental setting. All algorithms are implemented in MATLAB.

6.1 Competing Algorithms

Deterministic Frequency Directions. Since we mostly focus on the trade-off between communication cost and accuracy, we use the Exact Frequent Directions (eFD) algorithm, i.e., compute the SVD of local matrices and take their top-k right singular vectors and singular values. It can be shown that eFD is always more accurate than the more efficient Frequent Directions algorithm of Liberty (2013).

Random row sampling. In the random row sampling (RS) algorithm, each row of B is a rescaled row of A picked independently with replacement, with probability proportional to the squares of their Euclidean norms.

Our algorithms. We implement our SVS algorithm with both linear and quadratic sampling functions, named L-SVS and Q-SVS respectively. From the analysis of linear sampling function, we do not require to discard all small singular values. However we find in the experiments that this truncation

operation slightly improves the performance and makes the results more stable. Therefore, given a target message size ℓ, we will only keep the top 4ℓ singular values; by a similar argument as in the analysis for the quadratic function (Theorem 15), one can show that this truncation operation affects the theoretical bound by at most a constant factor.

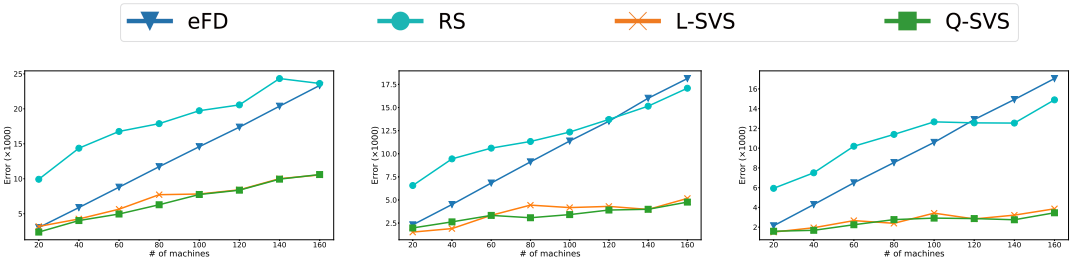
6.2 Synthetic Data

We use the same synthetic data sets as in Liberty (2013); Ghashami et al. (2014a). We generate our data sets using the following distributions. The input matrix is $A = SDU + N/\zeta$. The signal row space matrix $U \in \mathbb{R}^{t\times d}$ ($t \ll d$) contains a random t-dimensional subspace in $\mathbb{R}^d$ with $UU^T = I_t$. More precisely, we first generate a matrix $G \in \mathbb{R}^{d\times d}$ such that $g_{i,j} \sim \mathcal{N}(0,1)$ are i.i.d. standard Gaussians. Let $(Q, R) = \mathrm{QR}(G)$ be the QR decomposition, and U is a first t columns of Q. D is a diagonal matrix with $d_{i,i} = 1 - (i-1)/t$, which gives linearly diminishing signal singular values, and $S \in \mathbb{R}^{n\times t}$ is the signal coefficients matrix, where each $s_{i,j} \sim \mathcal{N}(0,1)$. The matrix SDU has rank t, which is the signal we wish to recover. The matrix $N \in \mathbb{R}^{n\times d}$ is a d-dimensional Gaussian noise added to the signal with $n_{i,j} \sim \mathcal{N}(0,1)$, and the parameter ζ controls the level of noise. From Vershynin (2011), we know the spectral norm of SDU dominate the spectral norm of N when ζ is greater than some constant c_1 ($c_1 \approx 1$ experimentally). In this case the signal is recoverable. Moreover, when $\zeta \le c_2\sqrt{d/t}$ for another constant close to 1, the Frobenius norm of A is dominated by the noise. Hence, in the experiments, we typically choose $c_1 \le \zeta \le c_2\sqrt{d/t}$, so that the signal is still recoverable even though the vast majority of the energy of each row is due to noise. The matrix A is randomly partitioned to s machines. In all of our experiments, the input on each machine will be a matrix of size 1000×500.

6.3 Performance Evaluation

In the first set of experiments, we evaluate the accuracy of the four algorithms on six sythetic data sets where noise ratio ζ varies from 4 to 12 and signal dimension t is set to 30 and 40. For the ease of comparison, the communication cost for each algorithm is tuned to be roughly 20 per machine. Figure 1 shows that the error of each algorithm increases when the number of machines grows from 20 to 160, this is because the input size on each machine keeps fixed and as the number of machines increases, the Frobenius norm of the entire input A *scales linearly*. It is observed that our two algorithms consistently outperform eFD and RS. Although there is slight difference between our two algorithms, for most of the time they demonstrate similar performance. Even though we restrict the message size to be smaller than the signal dimension, the error of our algorithm is quite small and increases very slowly as s getting larger, especially when the noise is relatively small. We also observe that the error of eFD grows almost linearly as the norm $\|A\|_F^2$ increases, and the performance of eFD is surpassed by random sampling when s is relatively large

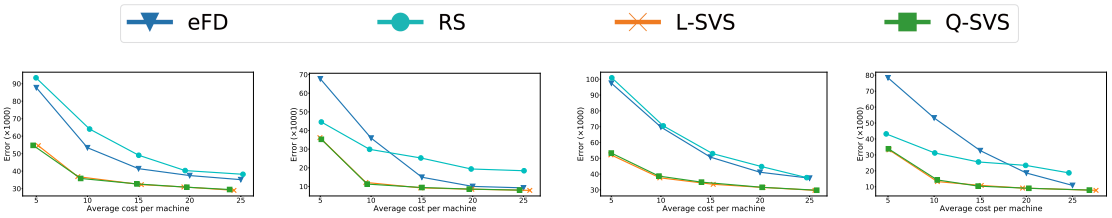
In Figure 2, we evaluate the trade-off between covariance error and communication costs. In these set of experiments, we fix the number of machines to $s = 128$. The performance of all algorithms increase with higher communication cost; it is clear that our two algorithms are the most cost-effective ones. We observe that, even though our signals have $20 - 30$ dimensions, the errors of our algorithms are already very close to optimal when each machine is only allowed to send 10 rows. On the contrary, the error of eFD becomes small only when the message size of each machine is close to the signal dimension. For random sampling, the errors decay slowly, which confirms its quadratic dependence on $1/\varepsilon$. To achieve high accuracy, the sample size needs to be very large. So



(a) $t = 30, \zeta = 4$ (b) $t = 30, \zeta = 8$ (c) $t = 30, \zeta = 12$

(d) $t = 40, \zeta = 4$ (e) $t = 40, \zeta = 8$ (f) $t = 40, \zeta = 12$

Figure 1: The number of machines s varies from 20 to 160. We tune the communication cost for each algorithm to be roughly 20 rows per machine.



(a) $t = 20, \zeta = 2$ (b) $t = 20, \zeta = 4$ (c) $t = 30, \zeta = 2$ (d) $t = 30, \zeta = 4$

Figure 2: Error vs cost. The number of machines s is 128.

our SVS-based algorithms have a property that, to recover a signal, the number of rows sent by each machine can be much smaller than the signal dimension. Both eFD and RS do not share this nice property.

7. Conclusion

In this paper, we study covariance sketch and its application to PCA in the distributed model. For covariance sketch, we give efficient one pass algorithms with improved communication costs, and prove a tight deterministic lower bound. We also provide analyses on the bit complexity for communication costs. We also show how to apply our distributed sketching algorithm to improve the communication costs of distributed PCA algorithms for dense and sparse matrices.

There are still lots of questions left unanswered. The most interesting one is whether our randomized algorithm for covariance sketch can be significantly improved. In particular, it is

still unknown whether the dependence on s can be improved further. The $\sqrt{\log d}$ factor in the communication costs might be an artifact of the matrix concentration inequality used; with a more suitable inequality or a more refined analysis, it might be removed. For PCA, it is unclear whether the $\Omega(skd)$ lower bound of Boutsidis et al. (2016) still holds in the setting where only one machine 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 to determine the communication complexity of covariance sketch in the arbitrary partition model.

Acknowledgments

This work is supported by National Natural Science Foundation of China Grant No. 61802069, Shanghai Sailing Program Grant No. 18YF1401200, Open Project of Shanghai Institute of Optics and Fine Mechanics, Shanghai Science and Technology Commission Grant No. 17JC1420200, and Science and Technology Commission of Shanghai Municipality Project Grant No. 19511120700.

Appendix A. Proof of Lemma 1

Lemma 27 (restatement of Lemma 3)

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

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

$$\left| \|Ax\|^2 - \|Bx\|^2 \right| = \left| x^T (A^T A - B^T B) x \right| \leq \|A^T A - B^T B\|_2$$

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

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

■

Appendix B. Bounding $\|B\|_F$

Theorem 28 *Assume the same setting as in Theorem 12, 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:

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

The proof of this theorem is actually a special case of the proof of Theorem 12; here we only need to bound the sum of squared singular values, so we apply the usual *Bernstein Inequality* (see e.g. Dubhashi and Panconesi (2009)) for scalar random variables.

Appendix C. Linear function (Proof of Theorem 5)

From Theorem 12 and 28, 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

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

For any σ, we have

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

So it follows that

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

We also have

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

To achieve the desired 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 12. 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 12 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 28, 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 28 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 $O(\frac{\sqrt{s}d}{\alpha} \cdot \log\frac{d}{\delta})$.

References

Pankaj K Agarwal, Graham Cormode, Zengfeng Huang, Jeff M Phillips, Zhewei Wei, and Ke Yi. Mergeable summaries. *ACM Transactions on Database Systems (TODS)*, 38(4):26, 2013.

Maria-Florina F Balcan, Steven Ehrlich, and Yingyu Liang. Distributed k-means and k-median clustering on general topologies. In *Advances in Neural Information Processing Systems*, 2013.

Srinadh Bhojanapalli, Prateek Jain, and Sujay Sanghavi. Tighter low-rank approximation via sampling the leveraged element. In *Proceedings of SODA*. SIAM, 2015.

Christos Boutsidis and David P Woodruff. Optimal cur matrix decompositions. In *Proceedings of STOC*. ACM, 2014.

Christos Boutsidis, D Woodruff, and Peilin Zhong. Optimal principal component analysis in distributed and streaming models. *Proceedings of STOC*, 2016.

Kenneth L Clarkson and David P Woodruff. Numerical linear algebra in the streaming model. In *Proceedings of STOC*. ACM, 2009.

Kenneth L Clarkson and David P Woodruff. Low rank approximation and regression in input sparsity time. In *Proceedings of STOC*, 2013.

Michael B Cohen, Cameron Musco, and Christopher Musco. Input sparsity time low-rank approximation via ridge leverage score sampling. In *Proceedings of SODA*. SIAM, 2017.

Jeffrey Dean and Luiz André Barroso. The tail at scale. *Communications of the ACM*, 56(2):74–80, 2013.

Michal Derezinski and Michael W Mahoney. Distributed estimation of the inverse hessian by determinantal averaging. In *Advances in Neural Information Processing Systems*, pages 11405–11415, 2019.

Michał Dereziński, Burak Bartan, Mert Pilanci, and Michael W Mahoney. Debiasing distributed second order optimization with surrogate sketching and scaled regularization. *arXiv preprint arXiv:2007.01327*, 2020.

Amey Desai, Mina Ghashami, and Jeff M Phillips. Improved practical matrix sketching with guarantees. *IEEE Transactions on Knowledge and Data Engineering*, 28(7):1678–1690, 2016.

Hu Ding, Yu Liu, Lingxiao Huang, and Jian Li. K-means clustering with distributed dimensions. In *International Conference on Machine Learning (ICML)*, 2016.

Petros Drineas, Ravi Kannan, and Michael W Mahoney. Fast monte carlo algorithms for matrices i: Approximating matrix multiplication. *SIAM Journal on Computing*, 36(1):132–157, 2006a.

Petros Drineas, Ravi Kannan, and Michael 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, 2006b.

Petros Drineas, Michael W Mahoney, S Muthukrishnan, and Tamás Sarlós. Faster least squares approximation. *Numerische Mathematik*, 117(2):219–249, 2011.

Devdatt P Dubhashi and Alessandro Panconesi. *Concentration of measure for the analysis of randomized algorithms*. Cambridge University Press, 2009.

Dan Feldman, Melanie Schmidt, and Christian Sohler. Turning big data into tiny data: Constant-size coresets for k-means, pca and projective clustering. In *Proceedings of SODA*. SIAM, 2013.

Mina Ghashami and Jeff M Phillips. Relative errors for deterministic low-rank matrix approximations. In *SODA*. SIAM, 2014.

Mina Ghashami, Amey Desai, and Jeff M Phillips. Improved practical matrix sketching with guarantees. In *European Symposium on Algorithms*. Springer, 2014a.

Mina Ghashami, Jeff M Phillips, and Feifei Li. Continuous matrix approximation on distributed data. *Proceedings of the VLDB Endowment*, 7(10):809–820, 2014b.

Mina Ghashami, Edo Liberty, and Jeff M Phillips. Efficient frequent directions algorithm for sparse matrices. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

Phillip B Gibbons and Srikanta Tirthapura. Estimating simple functions on the union of data streams. In *Proceedings of SPAA*. ACM, 2001.

Phillip B Gibbons and Srikanta Tirthapura. Distributed streams algorithms for sliding windows. In *Proceedings of SPAA*. ACM, 2002.

Vipul Gupta, Swanand Kadhe, Thomas Courtade, Michael W Mahoney, and Kannan Ramchandran. Oversketched newton: Fast convex optimization for serverless systems. *arXiv preprint arXiv:1903.08857*, 2019.

Vipul Gupta, Dominic Carrano, Yaoqing Yang, Vaishaal Shankar, Thomas Courtade, and Kannan Ramchandran. Serverless straggler mitigation using local error-correcting codes. *arXiv preprint arXiv:2001.07490*, 2020.

Zengfeng Huang. Near optimal frequent directions for sketching dense and sparse matrices. *Journal of Machine Learning Research*, 20(56):1–23, 2019.

Zengfeng Huang and Ke Yi. The communication complexity of distributed epsilon-approximations. *SIAM Journal on Computing*, 46(4):1370–1394, 2017.

Ravi Kannan, Santosh Vempala, and David P Woodruff. Principal component analysis and higher correlations for distributed data. In *Proceedings of the Annual Conference on Computational Learning Theory (COLT)*, 2014.

Zohar Karnin and Edo Liberty. Online pca with spectral bounds. In *Proceedings of the 28th Annual Conference on Computational Learning Theory (COLT)*, 2015.

Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, 1997.

Yi Li, Xiaoming Sun, Chengu Wang, and David P Woodruff. On the communication complexity of linear algebraic problems in the message passing model. In *International Symposium on Distributed Computing*. Springer, 2014.

Yingyu Liang, Maria-Florina F Balcan, Vandana Kanchanapally, and David Woodruff. Improved distributed principal component analysis. In *Advances in neural information processing systems*, 2014.

Yingyu Liang, Bo Xie, David Woodruff, Le Song, and Maria-Florina Balcan. Communication efficient distributed kernel principal component analysis. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2016.

Edo Liberty. Simple and deterministic matrix sketching. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2013.

Haipeng Luo, Alekh Agarwal, Nicolo Cesa-Bianchi, and John Langford. Efficient second order online learning by sketching. In *Advances in Neural Information Processing Systems*, 2016.

Luo Luo, Wenpeng Zhang, Zhihua Zhang, Wenwu Zhu, Tong Zhang, and Jian Pei. Sketched follow-the-regularized-leader for online factorization machine. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018.

Luo Luo, Cheng Chen, Zhihua Zhang, Wu-Jun Li, and Tong Zhang. Robust frequent directions with application in online learning. *Journal of Machine Learning Research*, 20(45):1–41, 2019.

Jayadev Misra and David Gries. Finding repeated elements. *Science of computer programming*, 2(2): 143–152, 1982.

Cameron Musco and Christopher Musco. Randomized block krylov methods for stronger and faster approximate singular value decomposition. In *Advances in neural information processing systems*, 2015.

Cameron Musco and Christopher Musco. Projection-cost-preserving sketches: Proof strategies and constructions. *arXiv preprint arXiv:2004.08434*, 2020.

John Nelson and Huy L Nguyên. Osnap: Faster numerical linear algebra algorithms via sparser subspace embeddings. In *annual symposium on foundations of computer science (FOCS)*. IEEE, 2013.

Roberto Imbuzeiro Oliveira. Sums of random hermitian matrices and an inequality by rudelson. *Electron. Commun. Probab*, 15(203-212):26, 2010.

Jeff M Phillips, Elad Verbin, and Qin Zhang. Lower bounds for number-in-hand multiparty communication complexity, made easy. *SIAM Journal on Computing*, 45(1):174–196, 2016.

Tamas Sarlos. Improved approximation algorithms for large matrices via random projections. In *annual symposium on foundations of computer science (FOCS)*, 2006.

Vatsal Sharan, Parikshit Gopalan, and Udi Wieder. Efficient anomaly detection via matrix sketching. In *Advances in neural information processing systems*, 2018.

Joel A Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012.

Dirk Van Gucht, Ryan Williams, David P Woodruff, and Qin Zhang. The communication complexity of distributed set-joins with applications to matrix multiplication. In *Proceedings of PODS*. ACM, 2015.

Roman Vershynin. Spectral norm of products of random and deterministic matrices. *Probability theory and related fields*, 150(3-4):471–509, 2011.

Shusen Wang, Fred Roosta, Peng Xu, and Michael W Mahoney. Giant: Globally improved approximate newton method for distributed optimization. In *Advances in Neural Information Processing Systems*, pages 2332–2342, 2018.

Zhewei Wei, Xuancheng Liu, Feifei Li, Shuo Shang, Xiaoyong Du, and Ji-Rong Wen. Matrix sketching over sliding windows. *Proceedings of SIGMOD*, 2016.

David Woodruff. Low rank approximation lower bounds in row-update streams. In *Advances in neural information processing systems*, 2014.

Shinjae Yoo, Hao Huang, and Shiva Prasad Kasiviswanathan. Streaming spectral clustering. IEEE 32nd International Conference on Data Engineering (ICDE), 2016.

Haida Zhang, Zengfeng Huang, Zhewei Wei, Wenjie Zhang, and Xuemin Lin. Tracking matrix approximation over distributed sliding windows. In *IEEE 33rd International Conference on Data Engineering (ICDE)*, 2017.

Yuchen Zhang, Martin Wainwright, and Michael Jordan. Distributed estimation of generalized matrix rank: Efficient algorithms and lower bounds. In *International Conference on Machine Learning (ICML)*, 2015.