JMLR: Workshop and Conference Proceedings vol 35:1–18, 2014

Principal Component Analysis and Higher Correlations for Distributed Data

Ravindran Kannan KANNAN@MICROSOFT.COM
Microsoft Research India

Santosh S. Vempala VEMPALA@GATECH.EDU
Georgia Tech

David P. Woodruff DPWOODRU@US.UBM.COM
IBM Research Almaden

Abstract

We consider algorithmic problems in the setting in which the input data has been partitioned arbitrarily on many servers. The goal is to compute a function of all the data, and the bottleneck is the communication used by the algorithm. We present algorithms for two illustrative problems on massive data sets: (1) computing a low-rank approximation of a matrix $A = A^1 + A^2 + \ldots + A^s$, with matrix A^t stored on server t and (2) computing a function of a vector $a_1 + a_2 + \ldots + a_s$, where server t has the vector a_t; this includes the well-studied special case of computing frequency moments and separable functions, as well as higher-order correlations such as the number of subgraphs of a specified type occurring in a graph. For both problems we give algorithms with nearly optimal communication, and in particular the only dependence on n, the size of the data, is in the number of bits needed to represent indices and words ($O(\log n)$).

1. Introduction

In modern large-scale machine learning problems the input data is often distributed among many servers, while the communication as well as time and space resources per server are limited. We consider two well-studied problems: (1) Principal Component Analysis (PCA), and (2) Generalized Higher-order correlations. Both problems study correlations between vectors. For the first problem, the vectors correspond to the rows of a matrix and we are interested in second-order correlations, while in the second problem we are interested in higher-order correlations among the vectors.

PCA is a central tool in many learning algorithms. The goal of PCA is to find a low-dimensional subspace that captures as much of the variance of a dataset as possible. By projecting the rows of a matrix onto this lower-dimensional subspace, one preserves important properties of the input matrix, but can now run subsequent algorithms in the lower-dimensional space, resulting in significant computational and storage savings. In a distributed setting, by having each server first locally project his/her own data onto a low-dimensional subspace, this can also result in savings in communication. PCA is useful for a variety of downstream tasks, e.g., for clustering or shape-fitting problems (Feldman et al. (2013)) and latent semantic analysis.

The second problem we consider is the Generalized Higher Order Correlation Problem. For this problem we assume server t has an n-dimensional vector a_t with non-negative entries. Note that for PCA, it is useful and more general to allow the entries to be positive, negative, or zero. On the other hand, the non-negativity assumption for Generalized Higher Order Correlations is justified both by the applications we give , as well as the fact that it is impossible to achieve low communication without this assumption, as described in more detail below.

A special case of this problem is the well-studied frequency moment problem. That is, if server t holds the vector a_t, with coordinates $a_{t1}, a_{t2}, \ldots, a_{tn}$, then the k-th frequency moment of $\sum_{t=1}^{s} a_t$ is $\sum_{i=1}^{n}(\sum_{t=1}^{s} a_{ti})^k$, where, k is a positive integer. This problem has been extensively studied in the data stream literature, starting with the work of Alon et al. (1999). Known lower bounds for this problem from that literature rule out low communication algorithms when $k > 2$ in the distributed setting when the number of servers grows as a power of n (Bar-Yossef et al. (2004); Chakrabarti et al. (2003); Gronemeier (2009)), or when there are only two servers and the entries are allowed to be negative Bar-Yossef et al. (2004). Here we overcome these lower bounds for smaller s and indeed will develop algorithms and lower bounds for estimating $\sum_{i=1}^{n} f(\sum_{t=1}^{s} a_{ti})$, for a general class of functions $f : \mathbf{R}_+ \to \mathbf{R}_+$.

We then extend these results to the following more general problem: there is a collection of vectors that is partitioned into s parts - $W_1, W_2, \ldots, W_s$ - and server t holds W_t. For each t and each $i \in W_t$, there is an n-dimensional vector $v_i = (v_{i1}, v_{i2}, \ldots, v_{in})$ wholly residing on server t. Let $f : \mathbf{R}_+ \to \mathbf{R}_+$ and $g : \mathbf{R}_+^k \to \mathbf{R}_+$ be functions. For a natural number k, define the k-th generalized moment $M(f, g, k)$ as

$$M(f,g,k) = \sum_{j_1,j_2,\ldots,j_k \in [n] \text{ distinct}} f\left(\sum_i g(v_{i,j_1}, v_{i,j_2}, \ldots, v_{i,j_k})\right).$$

There are many applications of higher-order correlations, and we only mention several here. For a document collection, we seek statistics (second, third and higher moments) of the number of documents in which each trigram (triples of terms) occurs. For a bipartite graph $G(V_1, V_2, E)$ and constants (r, u), we want to estimate the number of $K_{r,u}$ (complete bipartite graph) subgraphs. For a time series of many events, we want to estimate the number of tuples $(E_1, E_2, \ldots, E_r; t_1, t_2, \ldots, t_u)$ for which each of the events $E_1, E_2, \ldots, E_r$ occurs at each of the times $t_1, t_2, \ldots, t_u$.

Conceptually, for each i, we can think of a vector a_i with $\binom{n}{k}$ components - one for each distinct tuple $(j_1, j_2, \ldots, j_k)$. Suppose $a_{i;j_1,j_2,\ldots,j_k} = g(v_{i,j_1}, v_{i,j_2}, \ldots, v_{i,j_k})$, and let $a_t = \sum_{i \in W_t} a_i$. Our first theorem describes a way of estimating $M(f, g, k)$ up to a $(1+\varepsilon)$-factor, where, each server uses polynomial time and polynomial space, but we try to optimize total communication while keeping the number of rounds constant. For this algorithm, server t explicitly constructs the vector a_t first, so it uses $O(n^k|W_t|)$ space. Thereafter the space is linear in the total size of all the a_t. Our second theorem shows how to reduce space to linear in n. This algorithm does not construct a_t explicitly, but instead performs a rejection sampling procedure.

Before stating our theorems, we need some notation. Let $c_{f,s}$ be the least positive real number such that

$$f(x_1 + x_2 + \cdots + x_s) \le c_{f,s}(f(x_1) + f(x_2) + \cdots + f(x_s)) \ \forall x_1, x_2, \ldots, x_s \in \mathbf{R}_+. \quad (1)$$

Note that for $f(x) = x^k$ (as in the k-th frequency moment), $c_{f,s} = s^{k-1}$, since for any non-negative real numbers $b_1, b_2, \ldots, b_s$, we have $(b_1 + b_2 + \cdots + b_s)^k \le s^{k-1}(b_1^k + b_2^k + \cdots + b_s^k)$, and taking $b_t = 1$, we see that the factor s^{k-1} cannot be improved.

Model. The communication and computations are not assumed to be synchronous. We arbitrarily denote one of the s servers as the Central Processor (CP). A round consists of the CP sending a message to each server and each server sending an arbitrary length message to the CP. A round is complete when the CP has received messages from all servers from which it is expecting a message in that round. All servers communicate only with the CP, which, up to a factor of two, is equivalent to the servers communicating directly with each other (provided they indicate in their message who the message is being sent to). For formal details of this model, we refer the reader to Section 3 of Braverman et al. (2013). Our algorithms take polynomial time, linear space and $O(1)$ rounds of communication.

Our Results.

Low-rank matrix approximation and approximate PCA. Our first set of results is for low-rank approximation: given an $n \times d$ matrix A, a positive integer k and $\varepsilon > 0$, find an $n \times d$ matrix B of rank at most k such that

$$||A - B||_F \leq (1 + \varepsilon) \cdot \min_{X : \text{rank}(X) \leq k} ||A - X||_F.$$

Here, for a matrix A, the Frobenius norm $||A||_F^2$ is the sum of squares of the entries of A. A basis for the rowspace of B provides an approximate k-dimensional subspace to project the rows of A onto, and so is a form of approximate PCA. We focus on the frequently occurring case when A is rectangular, that is, $n \gg d$.

Theorem 1 *Consider the arbitrary partition model where an $n \times d$ matrix A_t resides in server t and the data matrix $A = A^1 + A^2 + \cdots + A^s$. For any $1 \geq \varepsilon > 0$, there is an algorithm that, on termination, leaves a $n \times d$ matrix C^t in server t such that the matrix $C = C^1 + C^2 + \cdots + C^s$ with arbitrarily large constant probability achieves $\|A - C\|_F \leq (1 + \varepsilon) \min_{X : rank(X) \leq k} ||A - X||_F$, using linear space, polynomial time and with total communication complexity $O(sdk/\varepsilon + sk^2/\varepsilon^4)$ real numbers. Moreover, if the entries of each A_t are b bits each, then the total communication is $O(sdk/\varepsilon + sk^2/\varepsilon^4)$ words each consisting of $O(b + \log(nd))$ bits.*

In contrast to the guarantees in Theorem 1, in the streaming model even with multiple passes, a simple encoding argument formalized in Theorem 4.14 of Clarkson and Woodruff (2009) shows the problem requires $\Omega(n + d)$ communication. We bypass this problem by allowing the s different servers to locally output a matrix C_t so that $\sum_t C_t$ is a $(1 + \varepsilon)$-approximation to the best rank-k approximation. We are not aware of any previous algorithms with less than n communication in the arbitrary partition model.

In the row-partition model, in which each row of A is held by a unique server, there is an $O(sdk/\varepsilon)$ word upper bound due to Feldman et al. (2013). This is also achievable by the algorithms of Ghashami and Phillips (2013); Liang et al. (2013); Balcan et al. (2014). As the row-partition model is a special case of our model in which for each row of A, there is a unique server with a non-zero vector on that row, our result implies their result up to the low order $O(sk^2/\varepsilon^4)$ term, but in a stronger model. For example, consider the case in which a customer corresponds to a row of A, and a column to his/her purchases of a specific item. These purchases could be distributed across multiple servers corresponding to different vendors. Or in the case of search data, each column could correspond to a search term of a user, and the searches may be distributed across multiple servers for storage and processing considerations. These examples are captured by the arbitrary partition model but not by the row partition model.

The technique for our upper bound is based on a two-stage adaptive sketching process, and has played an important role in several followup works, including CUR Matrix Factorizations of Boutsidis and Woodruff (2014) and subspace embeddings for the polynomial kernel by Avron et al. (2014).

We also show an $\tilde{\Omega}(skd)$ communication lower bound, showing our algorithm is tight up to a $\tilde{O}(1/\varepsilon)$ factor. The argument involves an upper bound showing how a player can communication-efficiently learn a rank-k matrix given only a basis for its row space.

Theorem 2 *Suppose each of s servers has an $n \times d$ matrix A^i and the CP wants to compute a rank-k approximation of $A = \sum_{i=1}^{s} A^i$ to within relative error $\varepsilon \geq 0$. The total communication required is $\tilde{\Omega}(skd)$ bits. Note that the lower bound holds for computing a $(1+\varepsilon)$-approximation for any $\varepsilon \geq 0$.*

Frequency moments and higher-order correlations. Our next set of results are for estimating higher moments and higher-order correlations of distributed data.

Theorem 3 *Let $f : \mathbf{R}_+ \to \mathbf{R}_+$ and $c_{f,s}$ be as in (1). There are s polynomial time, linear space bounded servers, where server t holds a non-negative n-vector $a_t = (a_{t1}, a_{t2}, \ldots, a_{tn})$. We can estimate $\sum_{i=1}^{n} f\left(\sum_{t=1}^{s} a_{ti}\right)$ up to a $(1+\varepsilon)$ factor by an algorithm using $O(s^2 c_{f,s}/\varepsilon^2)$ total words of communication (from all servers) in $O(1)$ rounds. Moreover, any estimation up to a $(1+\varepsilon)$ factor needs in the worst case $\Omega(c_{f,s}/\varepsilon)$ bits of communication.*

We remark that the lower bound applies to *any* function f with parameter $c_{f,s}$, not a specific family of such functions.

Theorem 4 *Let $f : \mathbf{R}_+ \to \mathbf{R}_+$, $g : \mathbf{R}_+^k \to \mathbf{R}_+$ be monotone functions with $c_{f,s}$ as in (1). $k \in O(1)$ is a natural number and let $M(f, g, k)$ be the generalized moment. We can approximate $M(f, g, k)$ to relative error ε by an algorithm with communication at most $O(s^3 c_{f,s}/\varepsilon^2)$ words in $O(1)$ rounds. Further, we use polynomial time and linear space.*

A key feature of this algorithm, and our following ones, is worth noting: they involve *no dependence* on n or $\ln n$, so they can be used when a_t are implicitly specified and n itself is very large, possibly infinite (provided, we can communicate each index i). In the theorem below Ω is the set of coordinates of each vector. It is analogous to $[n]$. We use $\sum_{x \in \Omega}$, which when Ω is infinite and the probabilities are densities, should be replaced with an integral; our theorem is also valid for the case when we have integrals.

Theorem 5 *Let $f : \mathbf{R}_+ \to \mathbf{R}_+$, $g : \mathbf{R}_+^k \to \mathbf{R}_+$ be monotone functions with $c_{f,s}$ as in (1). Server t is able to draw (in unit time) a sample $x \in \Omega$ according to a probability distribution h_t on Ω. Also, server t can estimate $\sum_{x \in \Omega} f(h_t(x))$. Then with $O(s^3 c_{f,s}/\varepsilon^2)$ words of communication, CP can estimate $\sum_{x \in \Omega} f\left(\sum_{t=1}^{s} h_t(x)\right)$ to within relative error ε.*

As a special case we consider the well-studied case of frequency moments. The best previous upper bound for the k-th frequency moment problem in the distributed setting is by Woodruff and Zhang (2012) who gave an algorithm that achieves $s^{k-1}\left(\frac{C \log n}{\varepsilon}\right)^{O(k)}$ communication, so the complexity still depends, albeit mildly, on n. Theorem 3 implies an algorithm with $O(s^{k+1}/\varepsilon^2)$ words of communication. We further improve this:

Theorem 6 *There are s servers, with server t holding a non-negative vector $a_t = (a_{t1}, a_{t2}, \ldots, a_{tn})$.*[1] *Then, to estimate $A = \sum_{i=1}^{n} \left(\sum_{t=1}^{s} a_{ti}\right)^k$ to within relative error ε, there is an algorithm that communicates $O((s^{k-1} + s^3)(\ln s/\varepsilon)^3)$ words* [2] *in $O(1)$ rounds.*

Thus, for $k \geq 4$, the complexity is $\tilde{O}(s^{k-1}/\varepsilon^3)$. Our algorithm has no dependence on n, though it does have the restriction that $k \geq 4$. It nearly matches a known lower bound of $\Omega(s^{k-1}/\varepsilon^2)$ due to Woodruff and Zhang (2012). In Theorem 12, we extend the algorithm and its near-optimal guarantees to a broader class of functions.

2. Low-rank Approximation

For a matrix A, define $f_k(A)$ as: $f_k(A) = \min_{X:\text{rank}(X)\leq k} ||A - X||_F$. Recall that the **rank-k approximation problem** is the following: Given an $n \times d$ matrix A, and $\varepsilon > 0$, find an $n \times d$ matrix B of rank at most k such that $||A - B||_F \leq (1 + \varepsilon) \cdot f_k(A)$.

2.1. Upper bound for low rank approximation

One of the tools we need is a subspace embedding. A random $m \times n$ matrix P with $m = O(d/\varepsilon^2)$ is a *subspace embedding* if for all vectors $x \in \mathbb{R}^d$, $\|PAx\|_2 = (1 \pm \varepsilon)\|Ax\|_2$. There are many choices for P, including a matrix of i.i.d. $N(0, 1/m)$ random variables or a matrix of i.i.d. Rademacher random variables (uniform in $\{-1/\sqrt{m}, +1/\sqrt{m}\}$) with $m = O(d/\varepsilon^2)$ (combining the Johnson-Lindenstrauss transform with a standard net argument) by Arriaga and Vempala (1999); Achlioptas (2003); Arriaga and Vempala (2006). With a slightly larger value of m, one can also use Fast Johnson-Lindenstrauss transforms by Ailon and Chazelle (2009) and the many optimizations to them, or the recent fast sparse subspace embeddings by Clarkson and Woodruff (2013) and its optimizations in Nelson and Nguyen (2012); Meng and Mahoney (2013). Such mappings can also be composed with each other.

We are mainly concerned with communication, so we omit the tradeoffs of different compositions and just use a composition for which $m = O(d/\varepsilon^2)$, PA is an $m \times d$ matrix of words each consisting of $O(b + \log(nd))$ bits, and P can be specified using $O(d \log n)$ bits (using a d-wise independent hash function, as first shown in Clarkson and Woodruff (2009)), see Theorem 7 below. Since we will assume that b is at least $\log n$, the $O(d \log n)$ bits to specify P will be negligible, though we remark that the number of bits to specify P can be further reduced using results of Kane et al. (2011).

We will prove the following property about the top k right singular vectors of PA for a subspace embedding P.

Theorem 7 *Suppose A is an $n \times d$ matrix. Let P be an $m \times d$ matrix for which $(1 - \varepsilon)\|Ax\|_2 \leq \|PAx\|_2 \leq (1 + \varepsilon)\|Ax\|_2$ for all $x \in \mathbb{R}^d$, that is, P is a subspace embedding for the column space of A. Suppose VV^T is a $d \times d$ matrix which projects vectors in $\mathbb{R}^d$ onto the space of the top k singular vectors of PA. Then $\|A - AVV^T\|_F \leq (1 + O(\varepsilon)) \cdot f_k(A)$. Furthermore, if $m = O(d/\varepsilon^2)$*

[1] The vector a_t need not be written down explicitly in server t. it just has to have the ability to (i) find $\sum_{i=1}^{n} a_{ti}^k$ to relative error ε and draw a sample according to $\{a_{ti}^k / \sum_j a_{tj}^k\}$.

[2] Each communicated word is either an index i or a value a_{ti}.

and P is a random sign matrix with entries uniform in $\{-1/\sqrt{m}, 1/\sqrt{m}\}$, then with $O(d)$-wise independent entries, P satisfies the above properties with probability at least[3] $1 - \exp(-d)$.

We will combine this property with the following known property.

Theorem 8 *(combining Theorem 4.2 and the second part of Lemma 4.3 of Clarkson and Woodruff (2009)) Let $S \in \mathbb{R}^{m \times n}$ be a random sign matrix with $m = O(k \log(1/\delta)/\varepsilon)$ in which the entries are $O(k + \log(1/\delta))$-wise independent. Then with probability at least $1 - \delta$, if UU^T is the $d \times d$ projection matrix onto the row space of SA, then if $(AU)_k$ is the best rank-k approximation to matrix AU, we have*

$$\|(AU)_k U^T - A\|_F \leq (1 + O(\varepsilon))\|A - A_k\|_F.$$

We can now state the algorithm, which we call ADAPTIVECOMPRESS. In ADAPTIVECOMPRESS, the matrix P is of size $O(k/\varepsilon^3) \times n$.

ADAPTIVECOMPRESS(k,ε, δ)

1. Server 1 chooses a random seed for an $m \times n$ sketching matrix S as in Theorem 8, given parameters k, ε, and δ, where δ is a small positive constant. It communicates the seed to the other servers.
2. Server i uses the random seed to compute S, and then SA^i, and sends it to Server 1.
3. Server 1 computes $\sum_{i=1}^s SA^i = SA$. It computes an $m \times d$ orthonormal basis U^T for the row space of SA, and sends U to all the servers.
4. Each server i computes $A^i U$.
5. Server 1 chooses another random seed for a $O(k/\varepsilon^2) \times n$ matrix P which is to be $O(k)$-wise independent and communicates this seed to all servers.
6. The servers then agree on a subspace embedding matrix P of Theorem 7 for AU, where P is an $O(k/\varepsilon^3) \times n$ matrix which can be described with $O(k \log n)$ bits.
7. Server t computes PA_tU and send it to Server 1.
8. Server 1 computes $\sum_{t=1}^s PA_tU = PAU$. It computes VV^T, which is an $O(k/\varepsilon) \times O(k/\varepsilon)$ projection matrix onto the top k singular vectors of PAU, and sends V to all the servers.
9. Server t outputs $C_t = A_tUVV^TU^T$. Let $C = \sum_{t=1}^s C_t$. C is not computed explicitly.

Proof (of Theorem 7.) Suppose P is a subspace embedding for the column space of A. Form an orthonormal basis of $\mathbf{R}^d$ using the right singular vectors of PA. Let $v_1, v_2, \ldots, v_d$ be the basis.

$$||A - A\sum_{i=1}^{k} v_i v_i^T||_F^2 = \sum_{i=k+1}^{d} |Av_i|^2 \leq (1+\varepsilon)^2 \sum_{i=k+1}^{d} |PAv_i|^2 = (1+\varepsilon)^2 f_k^2(PA).$$

[3] $\exp(-d)$ denotes $2^{-\Theta(d)}$.

Also, suppose now $u_1, u_2, \ldots, u_d$ is an orthonormal basis consisting of the singular vectors of A. Then, we have

$$f_k(PA)^2 \leq ||PA - PA\sum_{i=1}^{k} u_i u_i^T||_F^2 = \sum_{i=k+1}^{d} |PAu_i|^2 \leq (1+\varepsilon)^2 \sum_{i=k+1}^{d} |Au_i|^2 = (1+\varepsilon)^2 f_k(A)^2.$$

Thus,$\|A - A\sum_{i=1}^k v_i v_i^T\|_F^2 \leq (1+\varepsilon)^4 f_k(A)^2$, as desired.

For the second part of the theorem, regarding the choice of P, fix attention on one particular $x \in \mathbf{R}^d$. We apply Theorem 2.2 of Clarkson and Woodruff (2009) with A, B of that theorem both set to Ax of the current theorem and $m = O(d/\varepsilon^2)$ in the notation of that theorem. This states that for $m = O(d/\varepsilon^2)$, if P is an $m \times n$ matrix with $O(d)$-wise independent entries uniform in $\{-1/\sqrt{m}, +1/\sqrt{m}\}$, then for any fixed vector x, $\|PAx\|_2 = (1 \pm \varepsilon)\|Ax\|_2$ with probability $1 - \exp(-d)$. We combine this with Lemma 4 in Appendix A of Arora et al. (2006), based on Feige and Ofek (2005), to conclude that for all vectors x, $\|PAx\|_2 = (1 \pm \varepsilon)\|Ax\|_2$ with probability $1 - \exp(-d)$ (for a different constant in the $\exp()$ function). ■

Proof of Theorem 1: By definition of the ADAPTIVECOMPRESS protocol, we have $\|A - C\| = \|A - AUVV^TU^T\|$, where all norms in this proof are the Frobenius norm.

Notice that UU^T and $I_d - UU^T$ are projections onto orthogonal subspaces, where I_d is the $d \times d$ identity matrix. It follows by the Pythagorean theorem applied to each row that

$$\begin{aligned} \|AUVV^TU^T - A\|^2 &= \|(AUVV^TU^T - A)(UU^T)\|^2 + \|(AUVV^TU^T - A)(I - UU^T)\|^2 \\ &= \|AUVV^TU^T - AUU^T\|^2 + \|A - AUU^T\|^2, \end{aligned} \tag{2}$$

where the second equality uses that $U^TU = I_c$, where c is the number of columns of U.

Observe that the row spaces of $AUVV^TU^T$ and AUU^T are both in the row space of U^T, and therefore in the column space of U. It follows that since U has orthonormal columns, $\|AUVV^TU^T - AUU^T\| = \|(AUVV^TU^T - AUU^T)U\|$, and therefore

$$\begin{aligned} \|AUVV^TU^T - AUU^T\|^2 + \|A - AUU^T\|^2 &= \|(AUVV^TU^T - AUU^T)U\|^2 + \|A - AUU^T\|^2 \\ &= \|AUVV^T - AU\|^2 + \|A - AUU^T\|^2, \end{aligned} \tag{3}$$

where the second equality uses that $U^TU = I_c$. Let $(AU)_k$ be the best rank-k approximation to the matrix AU. By Theorem 7, with probability $1 - o(1)$, $\|AUVV^T - AU\|^2 \leq (1 + O(\varepsilon))\|(AU)_k - AU\|^2$, and so

$$\begin{aligned} \|AUVV^T - AU\|^2 + \|A - AUU^T\|^2 &\leq (1 + O(\varepsilon))\|(AU)_k - AU\|_2^2 + \|A - AUU^T\|^2 \\ &\leq (1 + O(\varepsilon))(\|(AU)_k - AU\|_2^2 + \|A - AUU^T\|^2) \end{aligned} \tag{4}$$

Notice that the row space of $(AU)_k$ is spanned by the top k right singular vectors of AU, which are in the row space of U. Let us write $(AU)_k = B \cdot U$, where B is a rank-k matrix.

For any vector $v \in \mathbb{R}^d$, vUU^T is in the rowspace of U^T, and since the columns of U are orthonormal, $\|vUU^T\|^2 = \|vUU^TU\|^2 = \|vU\|^2$, and so

$$\begin{aligned} \|(AU)_k - AU\|^2 + \|A - AUU^T\|^2 &= \|(B - A)U\|^2 + \|A(I - UU^T)\|^2 \\ &= \|BUU^T - AUU^T\|^2 + \|AUU^T - A\|^2. \end{aligned} \tag{5}$$

We apply the Pythagorean theorem to each row in the expression in (5), noting that the vectors $(B_i - A_i)UU^T$ and $A_iUU^T - A_i$ are orthogonal, where B_i and A_i are the i-th rows of B and A, respectively. Hence,

$$\|BUU^T - AUU^T\|^2 + \|AUU^T - A\|^2 \quad = \quad \|BUU^T - A\|^2 = \|(AU)_kU^T - A\|^2, \qquad (6)$$

where the first equality uses that

$$\|BUU^T - A\|^2 = \|(BUU^T - A)UU^T\|^2 + \|(BUU^T - A)(I - UU^T)\|^2 = \|BUU^T - AUU^T\|^2 + \|AUU^T - A\|^2,$$

and the last equality uses the definition of B. By Theorem 8, with constant probability arbitrarily close to 1, we have

$$\|(AU)_kU^T - A\|^2 \quad \leq \quad (1 + O(\varepsilon))\|A_k - A\|^2. \qquad (7)$$

It follows by combining (2), (3), (4), (5), (6), (7), that $\|AUVV^TU^T - A\|^2 \leq (1 + O(\varepsilon))\|A_k - A\|^2$, which shows the correctness property of ADAPTIVECOMPRESS.

We now bound the communication. In the first step, by Theorem 8, m can be set to $O(k/\varepsilon)$ and the matrix S can be described using a random seed that is $O(k)$-wise independent. The communication of steps 1-3 is thus $O(sdk/\varepsilon)$ words. By Theorem 7, the remaining steps take $O(s(k/\varepsilon)^2/\varepsilon^2) = O(sk^2/\varepsilon^4)$ words of communication.

To obtain communication with $O(b + \log(nd))$-bit words if the entries of the matrices A_t are specified by b bits, Server 1 can instead send SA to each of the servers. The t-th server then computes $PA_t(SA)^T$ and sends this to Server 1. Let $SA = RU^T$, where U^T is an orthonormal basis for the row space of SA, and R is an $O(k/\varepsilon) \times O(k/\varepsilon)$ change of basis matrix. Server 1 computes $\sum_t PA_t(SA)^T = PA(SA)^T$ and sends this to each of the servers. Then, since each of the servers knows R, it can compute $PA(SA)^T(R^T)^{-1} = PAU$. It can then compute the SVD of this matrix, from which it obtains VV^T, the projection onto its top k right singular vectors. Then, since Server t knows A_t and U, it can compute $A_tU(VV^T)U^T$, as desired. Notice that in this variant of the algorithm what is sent is SA_t and $PA_t(SA)^T$, which each can be specified with $O(b + \log(nd))$-bit words if the entries of the A_t are specified by b bits. ■

3. Frequency Moments and Higher Order Correlations

In this section, we prove Theorems (3), (4) and (6).

We begin with some common notation. For $t \in [s], i \in [n]$:

$$C_t = \sum_{i=1}^{n} f(a_{ti}) \; ; \; B_i = \sum_{t=1}^{s} f(a_{ti}) \; ; \; A_i = f\left(\sum_{t=1}^{s} a_{ti}\right).$$

Let $B = \sum_i B_i = \sum_t C_t$; $A = \sum_i A_i$. The task is to estimate A. We analyze the following algorithm. Let $l = 100\frac{s \cdot c_{f,s}}{\varepsilon^2}$. The parameters in the algorithm will be specified presently.

Proof (of Theorem (3)):

To analyze the algorithm, we think of it differently: suppose CP picks t for the first of its l trials and asks that t to pick i according to its $\{f(a_{ti}/C_t\}$. Let X be the random variable BA_i/B_i for that i. Clearly the estimate made by the algorithm can be viewed as the average of l i.i.d. copies of X. So it will suffice to show that (i) X is unbiased : I.e., $E(X) = A$ and (ii) Var$(X) \leq c_{f,s}sA^2$

```
DISTRIBUTEDSUM($\varepsilon$)
  1. For $t \in [s]$, server $t$ computes $C_t$ and all servers send their $C_t$ to CP. This is round 1.
  2. CP does $l$ i.i.d. trials, in each picking a $t$, with probabilities $\{C_t/B\}$. Let $d_t$ be the number of times it picks $t$. CP sends $d_t$ to server $t$.
  3. Server $t$ picks $d_t$ samples $i_1, i_2, \ldots$ in i.i.d. trials, each according to probabilities $\{f(a_{ti})/C_t\}$ and sends the $d_t$ indices to CP. Round 2 is complete when CP receives all these indices.
  4. CP collects all the samples. Let $S$ be the set of sampled $i$ (so, $|S| = l$). CP sends all of $S$ to all servers.
  5. Server $t$ sends $a_{ti}$ for all $i \in S$ to CP.
  6. CP computes $A_i, B_i$ for all $i \in S$ and outputs $\frac{B}{l}\sum_{i\in S}\frac{A_i}{B_i}$ as its estimate of $A$.
```

(whence, the variance of the average of l i.i.d. copies of X would have variance at most $\varepsilon^2 A^2$ giving us the relative error bound.)

The first part is easy: Let p_i be the probability that we pick i by this process. Clearly, $p_i = \sum_{t=1}^{s} \text{Prob(CP picked } t\text{)Prob(}t\text{ picks } i) = \sum_t \frac{C_t}{B}\frac{f(a_{ti})}{C_t} = \frac{B_i}{B}$. So, $E(X) = \sum_{i=1}^{n} B\frac{A_i}{B_i}\frac{B_i}{B} = A$, proving (i). For (ii), we have $E(X^2) = B^2\sum_i p_i \frac{A_i^2}{B_i^2} = B\sum_i \frac{A_i^2}{B_i} \le ABc_{f,s} \le c_{f,s}sA^2$, since, $A_i = f(\sum_t a_{ti}) \le c_{f,s}\sum_{t=1}^{s} f(a_{ti})$ by the definition of $c_{f,s}$ and by monotonicity of f, we have $B_i = \sum_t f(a_{ti}) \le sf(\sum_t a_{ti})$.

To prove the claimed resource bounds, note that polynomial time and linear space bounds are obvious, since, all that each server has to do is to compute all $f(a_{ti})$, sum them up and sample at most l times. The communication is dominated by each of s servers sending $\{a_{ti}, i \in S\}$ to CP which is $sc_{f,s}/\varepsilon^2$ words per server giving us a total of $O(s^2 c_{f,s}/\varepsilon^2)$.

Now for the lower bound, we use (rather unsurprisingly) the set-disjointness problem. It is known (Alon et al. (1999); Bar-Yossef et al. (2004); Chakrabarti et al. (2003); Gronemeier (2009); Jayram (2009); Woodruff and Zhang (2012)) that the following problem needs $\Omega(n)$ bits of communication even for a randomized algorithm: we distinguish between two situations: (a) Each of s servers holds a subset of $[n]$ and the subsets are pairwise disjoint and (b) There is exactly one element common to all s sets. We reduce this problem to ours. Let S_t be the subset held by server t. By definition of $c_{f,s}$, there exist $x_1, x_2, \ldots, x_s \in \mathbf{R}_+$ such that $f(x_1 + x_2 + \cdots + x_s) = c_{f,s}(f(x_1) + f(x_2) + \cdots + f(x_s))$. Let $n = \frac{c_{f,s}-1}{\varepsilon}$. Let a_{ti} be defined by: $a_{ti} = x_t$ if $i \in S_t$ and $a_{ti} = 0$ otherwise. If the sets are disjoint, then $\sum_{i=1}^{n} f\left(\sum_{t=1}^{s} a_{ti}\right) = \sum_{t=1}^{s} |S_t| f(x_t)$. In the case (b) when the sets all share one element in common, $\sum_{i=1}^{n} f\left(\sum_{t=1}^{s} a_{ti}\right) = \sum_{t=1}^{s}(|S_t| - 1)f(x_t) + f(x_1 + x_2 + \cdots + x_s) = \sum_{t=1}^{s} |S_t| f(x_t) + (c_{f,s} - 1)\sum_t f(x_t) = \sum_{t=1}^{s} |S_t| f(x_t) + \varepsilon n \sum_t f(x_t)$. Since $|S_t| \le n$, it follows that if we can estimate $\sum_i f(\sum_t a_{ti})$ to relative error ε, then we can distinguish the two cases. But it is known that this requires $\Omega(n)$ bits of communication which is $\Omega(c_{f,s}/\varepsilon)$ proving the lower bound.

■

Proof (of Theorem (4): The only change is in the sampling algorithm:

- Order the $j = (j_1, j_2, \ldots, j_k)$ lexicographically. Start with the first j as the sample and compute a_{tj} by making a pass through the entire data: For each $i \in W_t$, after $v_{i,j_1}, v_{i,j_2}, \ldots, v_{i,j_k}$ are read, compute $g(v_{i,j_1}, v_{i,j_2}, \ldots, v_{i,j_k})$ and sum over all $i \in W_t$.
- Process the next j similarly. After processing a j, say, $j = j_0$, compute $f(a_{tj_0})$ and keep a running total of $f(a_{tj})$ for all j seen so far. Reject the old sample and replace it by the current j_0 with probability $\frac{f(a_{tj_0})}{\text{Total of all } f(a_{tj}) \text{ including } j_0}$.
- If the old sampled j is not rejected, just keep it as the sample and go to next j.

The proof of correctness and linear space bound follow straightforwardly by plugging in this sampling algorithm into Theorem (3). ■

We next turn to a more refined algorithm for estimating frequency moments with near-optimal communication, using the specific function $f(x) = x^k$. Here is the algorithm.

Proof (of Theorem (6): Let $B_i = \sum_{t=1}^s a_{ti}^k$ and $A_i = \left(\sum_{t=1}^s a_{ti}\right)^k$ and $\rho_i = A_i/B_i$. Note: $1 \leq \rho_i \leq s^{k-1}$. CP can arrange to pick $i \in [n]$ with probabilities $\{B_i/B\}$, where, $B = \sum_i B_i$ as we already saw. First pick $m = s^{k-2}/\varepsilon^3$ sample $i \in [n]$ according to $\{B_i/B\}$. Then, CP tells all servers all these i and collects all a_{ti} and thence all A_i, B_i. Total communication is at most $ms \leq s^{k-1}/\varepsilon^3$.

The estimator of A from one i is $X = \frac{BA_i}{B_i}$. It is easy to see that $E(X) = A$, and Var$(X) \leq E(X^2) = B\sum_{i=1}^n \frac{A_i^2}{B_i} \leq ABs^{k-1}$, since each $A_i/B_i \leq s^{k-1}$. So if we estimate A by $\tilde{A} =$ average of m i.i.d. copies of X, then we would get Var$(\tilde{A}) \leq \varepsilon^3 sAB$.

Claim 9 *With a suitable choice of constants, if* $A \geq sB$), *then,* $\mathsf{Prob}\left(|\tilde{A} - A| \leq \varepsilon A \text{ and } \tilde{A} \in (1-\varepsilon)sB\right) \geq 1 - c$. *Further, since* B *is known and* $\tilde{A}$ *is computed the condition* $\tilde{A} \leq sB$ *can be checked. Conversely, if* $A \leq sB$, *then,* $\mathsf{Prob}\left(\tilde{A} \in sB/(1-\varepsilon))\right) \geq 1 - \varepsilon$.

Proof In the first case, we have $\mathsf{Var}(\tilde{A}) \leq \varepsilon^3 A^2$, from which the first assertion follows using Chebychev inequality. The checkability is clear. For the second part, $\mathsf{Prob}(\tilde{A} \geq sB/(1-\varepsilon)) \leq \mathsf{Prob}(\tilde{A} > A + \varepsilon sB)) \leq \mathsf{Prob}\left(\tilde{A} \geq A + \varepsilon\sqrt{sAB}\right) \leq \frac{\text{Var}(\tilde{A})}{\varepsilon^2 sAB} \leq \varepsilon$. ■

Given the claim, after this step, the algorithm either has found that $A > sB$ and $\tilde{A}$ is a good estimate of A and terminated or it knows that $A \leq sB$. So assume now $A \leq sB$. CP now collects a set S of $s^{k-1}(\ln s)^2/\varepsilon^3$ sampled i 's, each i.i.d. sampled according to $\{B_i/B\}$. [It cannot now afford to inform all servers of all $i \in S$.]

Let $\rho_i = A_i/B_i$. Let β range over $\{s^{k-1}, e^{-\varepsilon}s^{k-1}, e^{-2\varepsilon}s^{k-1}, \ldots 1\}$, a total of $O(\ln s)$ values and let $S_\beta = \{i \in S : \rho_i \in [\beta e^{-\varepsilon}, \beta)\}$. Then, $\sum_{i\in S}\rho_i \approx \sum_\beta |S_\beta|\beta$. Since each $\rho_i \geq 1$, we have $\sum_{i\in S}\rho_i \geq |S|$. So we need only accurate estimates of those $|S_\beta|$ with $|S_\beta| \geq \varepsilon|S|/\beta\ln s$. For each

```
FREQUENCYMOMENTS($k$, $\varepsilon$)

  1. Pick an i.i.d. sample $S_0$ of $m = s^{k-2}/\varepsilon^3$ indices $i$, where each $i$ is picked according to
     $\{B_i/B\}$.

  2. Find average $\rho_i$ among the picked $i$ by getting all the $a_{ti}$ and estimate $A$ by $\tilde{A} = B$ times this
     average.

  3. If $\tilde{A} \geq sB$, then declare it to be the final estimate of $A$ and stop.

  4. CP now gets an i.i.d. sample $S$ of $O(s^{k-1}(\ln s)^2/\varepsilon^3)$ $i$ 's, each according to $\{B_i/B\}$.

  5. For each $\beta \in \{s^{k-1}, e^{-\varepsilon}s^{k-1}, e^{-2\varepsilon}s^{k-1}, \dots, 1\}$, CP does the following:

     (a) Pick a subset $T$ of $S$ of cardinality $\Omega(\beta(\ln s)^2/\varepsilon^3)$ u.a.r.
     (b) For each $i \in T$, pick a set $L$ of $l = \frac{s^{k-1}}{\beta}$ $t \in [s]$ u.a.r. Find all the $a_{ti}, t \in L$ and find
         $\frac{s^k}{l^k}\left(\sum_{t\in L} a_{ti}\right)^k$. Repeat this $\Omega(k \ln s + \ln(1/\varepsilon))$ times and take the median of all values
         found to be the estimate $\tilde{A}_i$ of $A_i$.
     (c) For each $i \in T$, take $\tilde{B}_i = a_{t(i),i}$, where, $t(i)$ is defined in (8).
     (d) For every $i \in T$ with $\tilde{A}_i/\tilde{B}_i \in [\beta e^{-\varepsilon}, \beta)$, do an exact computation of $A_i, B_i$ by asking
         every server for all the $a_{ti}$ values.
     (e) From the above estimate $|S_\beta \cap T$ and compute $\tilde{s}_\beta = |S_\beta \cap T||T|/|S|$ as the estimate of
         $|S_\beta|$.

  6. Return $B\sum_\beta \tilde{s}_\beta \beta$ as the estimate of $A$.
```

S_β, if we pick u.a.r. a subset T of S of cardinality $\Omega(\beta(\ln s)^2/\varepsilon^3)$, then $\frac{|S|}{|T|}|T \cap S_\beta|$ estimates $|S_\beta|$ to within $(1 \pm \varepsilon)$ for every β satisfying $|S_\beta| \geq \varepsilon|S|/\beta \ln s$.

For each β, pick such a random subset T from S. We have to recognize for each $i \in T$, whether it is in S_β. First, for each $i \in T$, we estimate A_i as follows: We pick $l = s^{k-1}/\beta$ servers $t_1, t_2, \dots, t_l$ u.a.r. and take $Z_i = \frac{s}{l}(a_{t_1,i} + a_{t_2,i} + \cdots a_{t_l,i})$ as our estimate of $A_i^{1/k} = \sum_{t=1}^s a_{ti}$. If Y is the r.v. based on just one random server (namely $Y = sa_{ti}$ for a u.a.r t), then $\mathsf{E}Y = A_i^{1/k}$ and

$$
\begin{aligned}
\frac{\mathsf{E}(Y^2)}{(\mathsf{E}Y)^2} &= s^2 \frac{\frac{1}{s}\sum_t a_{ti}^2}{(\sum_t a_{ti})^2} \\
&\leq s^2 \frac{\left(\frac{1}{s}\sum_t a_{ti}^k\right)^{2/k}}{A_i^{2/k}} \\
&\leq \frac{s^{2-(2/k)}}{\rho_i^{2/k}} \leq \frac{e^{2\varepsilon/k}s^{2-(2/k)}}{\beta^{2/k}}.
\end{aligned}
$$

From this it follows by averaging over l samples that

$$\frac{\mathsf{E}(Z^2)}{(\mathsf{E}Z)^2} \leq \frac{e^{2\varepsilon/k}s^{2-(2/k)}}{l\beta^{2/k}} = \frac{e^{2\varepsilon/k}s^{2-(2/k)}\beta}{s^{k-1}\beta^{2/k}} = e^{2\varepsilon/k}\left(\frac{\beta}{s^{k-1}}\right)^{1-(2/k)} \leq 1 + O(\frac{\varepsilon}{k}).$$

We do $\Omega(k \ln s + \ln(1/\varepsilon))$ such experiments and take the median of all of these to drive down the failure probability for a single i to less than ε^2/s^k, whence, it is small by union bound for the failure of any of the at most s^{k-1}/ε^2 indices i 's. Thus all the $A_i^{1/k}, i \in T$ are estimated to a factor of $(1+\epsilon)$ by this process whp. Since k is a fixed constant, this also means $(1+O(\varepsilon))$ relative error in the estimate of A_i.

Next we estimate the B_i to within a factor of $\tilde{O}(s)$, i.e., $\frac{\tilde{B}_i}{B_i} \in \left[\frac{1}{10s\ln s}, 1\right]$ whp. We will see shortly that such an estimate suffices. For each $i \in S$, define

$$t(i) = \text{ the index of the server which picked } i \text{ ; } \tilde{B}_i = a^k_{t(i),i}. \tag{8}$$

Then

$$\mathsf{E}(\tilde{B}_i) = \frac{\sum_t a_{t,i}^{2k}}{\sum_t a_{t,i}^k} \geq \frac{1}{s}\sum_t a_{t,i}^k = \frac{1}{s}B_i.$$

We observe that

$$\mathsf{Prob}(a^k_{t(i),i} \leq \delta\frac{B_i}{s}) \leq \delta.$$

Let $I_\delta(i)$ be an indicator random variable of whether $\tilde{B}_i \leq \delta B_i$. Then $I_\delta(i)$ are independent: the distribution of $S, t(i)$ is not changed if we imagine picking $S, t(i)$ as follows: we pick $|S|$ indices i in i.i.d. trials, according to B_i/B. Then for each i picked, independently pick a $t(i)$, where $\mathsf{Prob}(t(i) = t) = a^k_{ti}/B_i$. From this, the independence of I_δ is clear. Therefore, by Chernoff the the number of B_i which are much underestimated is small. Fixing $\delta = 1/(10\ln s)$, for each β, the number of i' for which B_i is underestimated by by less than $\delta B_i/s$ is at most a δ fraction.

We now have estimates $\tilde{\rho}_i$ of each $\rho_i, i \in T$. We need to determine from this $|S_\beta|$. From the bounds on estimation errors, we have $\tilde{\rho}_i \in [e^{-2\varepsilon/k}\rho_i, 10(s\ln s)\rho_i]$. Therefore, we see that only $i \in T$ with $\rho_i \geq \beta/(10s\ln s)$ may be mistaken for an $i \in S_\beta$. We have

$$|\{i \in S : \rho_i \geq \frac{\beta}{10s\ln s}\}| \leq \frac{\sum_S \rho_i}{\beta/(10s\ln s)}.$$

Moreover,

$$\mathsf{E}(\sum_{i\in S}\rho_i) = |S|\mathsf{E}(\rho_i) = |S|\sum_{i=1}^n \frac{A_i}{B_i}\frac{B_i}{B} = |S|\frac{A}{B} \leq s|S|.$$

Therefore,

$$\mathsf{E}(|\{i \in S : \rho_i \geq \frac{\beta}{s}\}|) \leq \frac{10s^2\ln s|S|}{\beta}.$$

The subset that intersects T is then $\{i \in T : \rho_i \geq \beta/s\}| \leq 20\frac{|T|}{|S|}\frac{s^2\ln s|S|}{\beta} = O(s^2\ln^3 s/\varepsilon^3)$. Now for these i 's in T, we collect all a_{ti} and find A_i, B_i exactly. This costs us $O(s^3(\ln s)^3/\varepsilon^2)$ communication. Thus the overall communication is bounded by

$$\frac{s^{k-1}}{\varepsilon^3} + \frac{s^{k-1}\ln^3 s}{\varepsilon^3} + \frac{s^3\ln^3 s}{\varepsilon^3} = O((s^{k-1}+s^3)\ln^3 s/\varepsilon^3).$$

This is $\tilde{O}(s^{k-1}/\varepsilon^3)$ for $k \geq 4$.

We have given the proof already of all assertions except the number of rounds. For the number of rounds, the most crucial point is that though the algorithm as stated requires $O((\ln s)^c)$ rounds, we can instead deal with all β simultaneously. CP just picks the T for all of them at once and sends them accross. Also, we just make sure that CP communicates all choices of t for each i all in one round. Also, note that the s^{k-2} sampling and checking if the first $\tilde{A} > \Omega(sB)$ can all be done in $O(1)$ rounds, so also the s^{k-1} sampling. Then the crude estimation of $\tilde{\rho}_i$ can be done in one $O(1)$ rounds followed by the finer sampling in $O(1)$ rounds. ■

Acknowledgements. We are grateful to Dick Karp and Jelani Nelson for helpful discussions, as well as the Simons Institute at Berkeley. Santosh Vempala was supported in part by NSF award CCF-1217793 and David Woodruff by the XDATA program of the Defense Advanced Research Projects Agency (DARPA), administered through Air Force Research Laboratory contract FA8750-12-C0323.

References

Dimitris Achlioptas. Database-friendly random projections: Johnson-lindenstrauss with binary coins. *J. Comput. Syst. Sci.*, 66(4):671–687, 2003.

Nir Ailon and Bernard Chazelle. The fast johnson–lindenstrauss transform and approximate nearest neighbors. *SIAM J. Comput.*, 39(1):302–322, 2009.

Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. *J. Comput. Syst. Sci.*, 58(1):137–147, 1999.

Sanjeev Arora, Elad Hazan, and Satyen Kale. A fast random sampling algorithm for sparsifying matrices. In *APPROX-RANDOM*, pages 272–279, 2006.

Rosa I. Arriaga and Santosh Vempala. An algorithmic theory of learning: Robust concepts and random projection. In *FOCS*, pages 616–623, 1999.

Rosa I. Arriaga and Santosh Vempala. An algorithmic theory of learning: Robust concepts and random projection. *Machine Learning*, 63(2):161–182, 2006.

Haim Avron, Huy L. Nguyen, and David P. Woodruff. Subspace embeddings for the polynomial kernel, 2014.

Maria-Florina Balcan, Vandana Kanchanapally, Yingyu Liang, and David P. Woodruff. Fast and communication efficient algorithms for distributed pca. *Manuscript*, 2014.

Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. *J. Comput. Syst. Sci.*, 68(4):702–732, 2004.

Christos Boutsidis and David P. Woodruff. Optimal cur matrix factorizations. In *STOC*, 2014.

Mark Braverman, Faith Ellen, Rotem Oshman, Toniann Pitassi, and Vinod Vaikuntanathan. Tight bounds for set disjointness in the message passing model. *CoRR*, abs/1305.4696, 2013.

Amit Chakrabarti, Subhash Khot, and Xiaodong Sun. Near-optimal lower bounds on the multi-party communication complexity of set disjointness. In *IEEE Conference on Computational Complexity*, pages 107–117, 2003.

Kenneth L. Clarkson and David P. Woodruff. Numerical linear algebra in the streaming model. In *STOC*, pages 205–214, 2009.

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

Uriel Feige and Eran Ofek. Spectral techniques applied to sparse random graphs. *Random Struct. Algorithms*, 27(2):251–275, 2005.

Dan Feldman, Melanie Schmidt, and Christian Sohler. Turning big data into tiny data: Constant-size coresets for *k*-means, pca and projective clustering. In *SODA*, pages 1434–1453, 2013.

Mina Ghashami and Jeff M. Phillips. Relative errors for deterministic low-rank matrix approximations. *CoRR*, abs/1307.7454, 2013.

Andre Gronemeier. Asymptotically optimal lower bounds on the nih-multi-party information complexity of the and-function and disjointness. In *STACS*, pages 505–516, 2009.

T. S. Jayram. Hellinger strikes back: A note on the multi-party information complexity of and. In *APPROX-RANDOM*, pages 562–573, 2009.

Daniel M. Kane, Raghu Meka, and Jelani Nelson. Almost optimal explicit johnson-lindenstrauss families. In *APPROX-RANDOM*, pages 628–639, 2011.

Yingyu Liang, Maria-Florina Balcan, and Vandana Kanchanapally. Distributed pca and k-means clustering. In *The Big Learning Workshop at NIPS*, 2013.

Xiangrui Meng and Michael W. Mahoney. Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression. In *STOC*, pages 91–100, 2013.

Jelani Nelson and Huy L. Nguyen. Osnap: Faster numerical linear algebra algorithms via sparser subspace embeddings. *CoRR*, abs/1211.1002, 2012.

Jeff M. Phillips, Elad Verbin, and Qin Zhang. Lower bounds for number-in-hand multiparty communication complexity, made easy. In *SODA*, pages 486–501, 2012.

David P. Woodruff and Qin Zhang. Tight bounds for distributed functional monitoring. In *STOC*, pages 941–960, 2012.

4. Appendix

4.1. Lower bound for low-rank approximation

Our reduction is from the multiplayer SUM problem.

Theorem 10 *(Phillips et al. (2012)) Suppose each of s players has a binary vector a^i with n bits and the first player wants to compute $\sum_{i=1}^{s} a^i \bmod 2$ with constant probability. Then the total communication needed is $\Omega(sn)$ bits.*

Proof (of Theorem 2.) We reduce from the $s-2$ player SUM problem, in which each player has a $k \times d$ binary matrix A^i and the first player wants to learn their sum. By Theorem 10, this problem needs $\Omega(skd)$ communication, since even the mod 2 version of the problem requires this amount of communication. Now consider the s-player problem s-RESTRICT-SUM in which the first player has I_d, the second player has $-I_d$, the remaining $s-2$ players have a $k \times d$ binary matrix A^i and the first player wants to learn the sum of all inputs. This also requires $\Omega(skd)$ communication. This follows since if this problem could be solved with $o(skd)$ communication, then SUM with $s-2$ players would have $o(skd)$ communication by a simulation in which the first player of the $(s-2)$-SUM problem simulates the first three players of the s-RESTRICT-SUM problem.

In our s-player low-rank approximation problem, we give the first player I_d, the second player $-I_d$, and the remaining $s-2$ players each has a random $k \times d$ binary matrix A^i. Note that there is a unique rank-k approximation to the sum of the k player inputs, namely, it is the matrix $\sum_{i=3}^{s} A^i$. It follows that any algorithm which outputs a projection matrix VV^T for which $\|A - AVV^T\|_F^2 \leq (1+\varepsilon)\min_{X:\, \text{rank}(X) \leq k}$, for any $\varepsilon \geq 0$, must be such that VV^T is a projection onto the row space of $\sum_{i=3}^{s} A^i$. This follows because $(1+\varepsilon)\min_{X:\, \text{rank}(X)\leq k} \|A - X\|_F = (1+\varepsilon)\cdot 0 = 0$.

Now, since the first player has I_d, his output is $I_d VV^T$, where the row space of V^T equals the row space of $\sum_{i=3}^{s} A^i$. Suppose the total communication of our problem is C.

We use this to build a protocol for s-RESTRICT-SUM, which has the same inputs as in our s-player low rank approximation problem. Notice that $A = \sum_{i=3}^{s} A^i$ is a $k \times d$ matrix with rows in $\{0, 1, 2, ..., s-2\}^d$.

Claim. The span of the rows of A can intersect $\{0, 1, \ldots, s-2\}^d$ in at most $(2s)^k$ distinct points.

Proof Let r$owspace(A)$ denote the row space of A. We will bound the size of r$owspace(A) \cap GF(p)^d$ for prime p with $s-2 < p < 2(s-2)$, where $GF(p)$ is the finite field with elements $\{0, 1, 2, \ldots, p-1\}$, and $GF(p)^d$ is the vector space over $GF(p)$. This will be an upper bound on the size of r$owspace(A) \cap \{0, 1, \ldots, s-2\}^d$. Since r$owspace(A)$ is k-dimensional, so is r$owspace(A) \cap GF(p)^d$. Hence the intersection has at most k linearly independent points. These k linearly independent points can be used to generate the remaining points in r$owspace(A) \cap GF(p)^d$. The number of distinct combinations of these points is at most $p^k < (2s)^k$, bounding the intersection size. ■

Next, players $3, \ldots, s$ agree on random $\{+1, -1\}^d$ vectors $u^1, \ldots u^{k'}$ where $k' = k \log 4s$ via a public coin. The entries of $u^1, ..., u^{k'}$ need only be $O(k \log s)$-wise independent, and as such can be agreed upon by all the players using only $O(sk \log s)$ bits of communication. Each player i then computes the inner products $A^i_j \cdot u^1, \ldots, A^i_j \cdot u^{k'}$ for each $j \in \{1, \ldots, k\}$. Here A^i_j denotes the j'th row of a the i'th player's matrix A^i.

The players $P_3, ..., P_s$ send all of these inner products to P_1. The latter, for all rows $j \in \{1, ..., k\}$, computes the inner products $A_j \cdot u^1, \ldots, A_j \cdot u^{k \log s}$, where $A = \sum_{i=3}^{s} A^i$. This can be done using $\tilde{O}(sk^2)$ communication. Since P_1 now has V^T, he can compute the $O(s)^k$ points in $\{0, 1, ..., s-2\}^d$ that each row of A could possibly be. Let p be one such possible point. For each row j, P_1 checks if $A_j \cdot u^l = p \cdot u^l$ for every $l \in \{1, 2, \ldots, k'\}$. He decides $p = A_j$ iff all $k \log s$

equalities hold for A_j. In this way, he can reconstruct $A = \sum_{i=3}^{s} A_i$. The number $k \log s$ of the different u vectors is chosen so that by a union bound, the procedure succeeds with high probability.

We can thus solve the s-RESTRICT-SUM problem using our s-player low rank problem with communication $C + \tilde{O}(sk^2)$, where C was the communication of our low-rank problem. Therefore, $C + \tilde{O}(sk^2) = \Omega(skd)$, which implies $C = \tilde{\Omega}(skd)$ since $k < d$. ■

4.2. Generalized moments

Here we extend the above theorem and proof works for a wide class of functions satisfying a weak Lipschitz condition (and generalizing the case of moments).

For a monotone function $f : \Re_+ \to \Re_+$, define

$$L_f = \min r : \forall x > y > 0, \frac{f(x)}{f(y)} \leq \left(\frac{x}{y}\right)^r.$$

Alternatively, L_f is the Lipschitz constant of f wrt the "distance" $d(x,y) = \log(x) - \log(y)$, i.e.,

$$L_f = \sup \frac{d(f(x), f(y))}{d(x,y)}.$$

For the function $f(x) = x^k$, we see that $L_f = k$.

Lemma 11 *For any function $f : \Re_+ \to R_+$ with $L = L_f$,*

$$\frac{f(\sum_{t=1}^{s} x_t)}{\sum_{t=1}^{s} f(x_t)} \leq \frac{(\sum_{t=1}^{s} x_t)^L}{\sum_{t=1}^{s} x_t^L}$$

Proof

$$\begin{aligned}
\frac{f(\sum_{t=1}^{s} x_t)}{\sum_{t=1}^{s} f(x_t)} &= \frac{f(\sum_{t=1}^{s} x_t)}{\sum_{t=1}^{s} x_t^L} \frac{\sum_{t=1}^{s} x_t^L}{\sum_{t=1}^{s} f(x_t)} \\
&\leq \frac{f(\sum_{t=1}^{s} x_t)}{\sum_{t=1}^{s} x_t^L} \min_t \frac{x_t^L}{f(x_t)} \\
&= \min_t \frac{f(\sum_{t=1}^{s} x_t)}{f(x_t)} \frac{x_t^L}{\sum_{t=1}^{s} x_t^L} \\
&\leq \min_t \frac{(\sum_{t=1}^{s} x_t)^L}{x_t^L} \frac{x_t^L}{\sum_{t=1}^{s} x_t^L} \\
&= \frac{(\sum_{t=1}^{s} x_t)^L}{\sum_{t=1}^{s} x_t^L}.
\end{aligned}$$

■

Theorem 12 *Let f be any nonnegative, superlinear real function with $L = L_f \geq 4$. Suppose there are s servers, with server t holding a non-negative vector $a_t = (a_{t1}, a_{t2}, \ldots, a_{tn})$ Then, to estimate $A = \sum_{i=1}^{n} f\left(\sum_{t=1}^{s} a_{ti}\right)$ to relative error ε, there is an algorithm that communicates $O(s^{L-1}(\ln s)^3/\varepsilon^3)$ words in $O(1)$ rounds.*

The algorithm is the following, essentially the same as in the moments case, with parameters defined in terms of L_f for a general function $f(.)$ in place of x^k.

1. Pick an i.i.d. sample S_0 of $m = s^{L-2}/\varepsilon^3$ indices i, where each i is picked according to $\{B_i/B\}$.
2. Find average ρ_i among the picked i by getting all the a_{ti} and estimate A by $\tilde{A} = B$ times this average.
3. If $\tilde{A} \geq sB$, then declare it to be the final estimate of A and stop.
4. CP now gets an i.i.d. sample S of $O(s^{L-1} \ln^2 s/\varepsilon^3)$ i 's, each according to $\{B_i/B\}$.
5. For each $\beta \in \{s^{L-1}, e^{-\varepsilon} s^{L-1}, e^{-2\varepsilon} s^{L-1}, \ldots, 1\}$, CP does the following:
   (a) Pick a subset T of S of cardinality $\Omega(\beta(\ln s)^2/\varepsilon^3)$ u.a.r.
   (b) For each $i \in T$, pick a set T' of $l = \frac{s^{L-1}}{\beta}$ $t \in [s]$ u.a.r. Find all the $a_{ti}, t \in T'$ and find $f\left(\frac{s}{l}\sum_{t \in T'} a_{ti}\right)$. Repeat this $\Omega(L \ln s + \ln(1/\varepsilon))$ times and take the median of all values found to be the estimate $\tilde{A}_i$ of A_i.
   (c) For each $i \in T$, take $\tilde{B}_i = a_{t(i),i}$, where, $t(i)$ is defined in (8).
   (d) For every $i \in T$ with $\tilde{A}_i/\tilde{B}_i \in [\beta e^{-\varepsilon}, \beta)$, do an exact computation of A_i, B_i by asking every server for all the a_{ti} values.
   (e) From the above estimate $|S_\beta \cap T$ and compute $\tilde{s}_\beta = |S_\beta \cap T||T|/|S|$ as the estimate of $|S_\beta|$.
6. Return $B \sum_\beta \tilde{s}_\beta \beta$ as the estimate of A.

Proof (of Thm. 12.) We point out the changes in the analysis from the special case of moments. Now we have $A_i = f(\sum_{t=1}^s a_{ti}), B_i = \sum_{t=1}^s f(a_{ti})$ and $A = \sum_{i=1}^n A_i, B = \sum_{i=1}^n B_i$. Also $\rho_i = A_i/B_i$. The reader will have noticed that k has been replaced by L in the above algorithm. The first phase remains the same, and at the end we either have a good approximation for A or we know that $A \leq sB$.

In the next phase we estimate $f(\sum_t a_{ti})^{1/L}$. To do this, we first estimate $\sum_{t=1}^s a_{ti}$, then apply f to this estimate. We need to analyze the error of both parts. For the first part, let $Y = sa_{ti}$ as before. Then $\mathsf{E}(Y) = \sum_{t=1}^s a_{ti}$ and since the server used to define Y is chosen uniformly at random, we have

$$\begin{aligned}
\frac{\mathsf{E}(Y^2)}{\mathsf{E}(Y)^2} &\leq s^2 \frac{\frac{1}{s}\sum_{t=1}^s a_{ti}^2}{(\sum_{i=1}^s a_{ti})^2} \\
&\leq s^2 \left(\frac{\frac{1}{s}\sum_{t=1}^s a_{ti}^L}{(\sum_{i=1}^s a_{ti})^L}\right)^{2/L} \\
&\leq s^{2-(2/L)} \left(\frac{f(\sum_{t=1}^s a_{ti})}{\sum_{t=1}^s f(a_{ti})}\right)^{2/L} \qquad \text{(using Lemma 11)} \\
&= \frac{s^{2-(2/L)}}{\rho_i^{2/L}}.
\end{aligned}$$

This is then bounded by $e^{2\varepsilon/L}$ just as before. So we get an estimate of $a_i = \sum_{t=1}^{s} a_{ti}$ to within multiplicative error $e^{\varepsilon/L}$. Let $\tilde{a}_i$ be this approximation. It remains to bound $f(\tilde{a}_i)$ in terms of $f(a_i)$. For this we observe that using the definition of L, if $a_i \le \tilde{a}_i$, then

$$1 \le \frac{f(\tilde{a}_i)}{f(a_i)} \le \left(\frac{\tilde{a}_i}{a_i}\right)^L \le e^{\varepsilon}.$$

We get a similar approximation if $a_i > \tilde{a}_i$.

The last phase for estimating B_i and putting together the estimates for all the ρ_i is again the same as in the case of moments. ■