

# Distributed Statistical Estimation of Matrix Products with Applications

David P. Woodruff  
Carnegie Mellon University  
Pittsburgh, PA 15213  
United States  
dwoodruf@cs.cmu.edu

Qin Zhang  
Indiana University Bloomington  
Bloomington, IN 47401  
United States  
qzhangcs@indiana.edu

## ABSTRACT

We consider statistical estimations of a matrix product over the integers in a distributed setting, where we have two parties Alice and Bob; Alice holds a matrix  $A$  and Bob holds a matrix  $B$ , and they want to estimate statistics of  $A \cdot B$ . We focus on the well-studied  $\ell_p$ -norm, distinct elements ( $p = 0$ ),  $\ell_0$ -sampling, and heavy hitter problems. The goal is to minimize both the communication cost and the number of rounds of communication.

This problem is closely related to the fundamental set-intersection join problem in databases: when  $p = 0$  the problem corresponds to the size of the set-intersection join. When  $p = \infty$  the output is simply the pair of sets with the maximum intersection size. When  $p = 1$  the problem corresponds to the size of the corresponding natural join. We also consider the heavy hitters problem which corresponds to finding the pairs of sets with intersection size above a certain threshold, and the problem of sampling an intersecting pair of sets uniformly at random.

## 1. INTRODUCTION

We study the problem of statistical estimations of a matrix product in the distributed setting. Consider two parties Alice and Bob; Alice holds a matrix  $A \in \{0, 1\}^{n \times n}$  and Bob holds a matrix  $B \in \{0, 1\}^{n \times n}$ , and they want to jointly compute a function  $f$  defined on  $A$  and  $B$  by exchanging messages. The goal is to minimize both the total communication cost and number of rounds of interaction.

One of the main statistical quantities we consider is the  $p$ -norm  $\|C\|_p$  of the product  $C = A \cdot B$ , defined as

$$\|C\|_p = \left( \sum_{i,j \in [n]} |C_{i,j}|^p \right)^{1/p}.$$

Here the matrix product  $A \cdot B$  is the standard matrix product over the integers. Interpreting  $0^0$  as 0, we see that  $p = 0$  corresponds to the number of non-zero entries of  $C$ , which, interpreting the rows of  $A$  and columns of  $B$  as sets, corresponds to the set-intersection join size (see Section 1.1 for the formal definition). This can also be viewed as a matrix form of the well-studied distinct elements problem in the data stream literature (see, e.g., [8, 14, 21]). Again interpreting the rows of  $A$  and the columns of  $B$  as sets, the case  $p = 1$  corresponds to the size of the corresponding natural join (again see Section 1.1 for the formal definition). The  $p = 2$  case corresponds to the (squared) Frobenius norm of the matrix product  $A \cdot B$ , which is a norm of fundamental importance in a variety of distributed linear algebra prob-

lems, such as low rank approximation (for a recent survey, see [34]). The case  $p = \infty$  corresponds to the pair of sets of maximum intersection size. Estimating the largest entry in a Boolean matrix product has also been studied in the centralized setting. We refer readers to the recent paper [1] and references therein.

As a closely related problem, we also consider the  $\ell_0$ -sampling problem for which the goal is to sample each non-zero entry in  $C = AB$  with probability  $(1 \pm \epsilon) \frac{1}{\|C\|_0}$ , which corresponds to approximately outputting a random pair among the intersecting pairs of sets.  $\ell_0$ -sampling is also extensively studied in the data stream literature [15, 20, 30], and is used as a building block for sketching various dynamic graph problems (see [28] for a survey).

We also study the approximate heavy hitter problem defined as follows. Let

$$\text{HH}_\phi^p(C) = \{(i, j) \mid C_{i,j}^p \geq \phi \|C\|_p^p\}.$$

The  $\ell_p$ - $(\phi, \epsilon)$ -heavy-hitter ( $0 < \epsilon \leq \phi \leq 1$ ) problem asks to output a set  $S$  such that

$$\text{HH}_\phi^p(AB) \subseteq S \subseteq \text{HH}_{\phi-\epsilon}^p(AB).$$

As outputting the matrix product  $C$  requires outputting  $n^2$  numbers, it is natural to output the set  $S$  as a sparse approximation of  $C$ ; indeed this can be viewed as a matrix form of the well-studied compressed sensing problem.

As mentioned, these basic statistical problems, being interesting for their own sake, have strong relationships to fundamental problems in databases. We describe such relationships more formally below.

Despite a large amount of work on computing  $p$ -norms and heavy hitters on frequency vectors in the streaming literature (see, e.g., [31] for a survey), we are not aware of any detailed study of these basic statistical functions on matrix products. The purpose of this paper is to introduce a systematic study of statistical estimations on matrix products.

### 1.1 Motivation and Applications

Estimating the norm of a matrix product is closely related to two of the most important operations in relational databases – the *composition* and the *natural join*. Suppose we are given two relations  $\mathcal{A}$  and  $\mathcal{B}$ , where  $\mathcal{A}$  is defined over attributes  $(X, Y)$  and  $\mathcal{B}$  is defined over attributes  $(Y, Z)$ . Assume for simplicity that  $\text{dom}(X) = \text{dom}(Y) = \text{dom}(Z) = [n]$ . We thus have  $\mathcal{A} \subseteq [n] \times [n]$  and  $\mathcal{B} \subseteq [n] \times [n]$ . The

composition of  $\mathcal{A}$  and  $\mathcal{B}$  is defined to be

$$\mathcal{A} \circ \mathcal{B} = \{(i, j) \mid \exists k : (i, k) \in \mathcal{A} \wedge (k, j) \in \mathcal{B}\}.$$

The natural join is defined to be

$$\mathcal{A} \bowtie \mathcal{B} = \{(i, k, j) \mid (i, k) \in \mathcal{A} \wedge (k, j) \in \mathcal{B}\}.$$

It is easy to see that the natural join corresponds to the composition together with the requirement that all the “witnesses”  $k$  are output.

We further define “projection” sets  $A_i = \{k \mid (i, k) \in \mathcal{A}\}$  for each  $i \in [n]$ , and  $B_j = \{k \mid (k, j) \in \mathcal{B}\}$  for each  $j \in [n]$ . Then we can rewrite the composition and natural joins as follows:

$$\mathcal{A} \circ \mathcal{B} = \{(i, j) \mid A_i \cap B_j \neq \emptyset\},$$

$$\mathcal{A} \bowtie \mathcal{B} = \{(i, k, j) \mid k \in A_i \cap B_j\}.$$

We thus also refer to compositions as *set-intersection joins*, and natural joins as *set-intersection joins with witnesses*.

As an application of set-intersection joins, consider a job application scenario: we have  $n$  applicants, with the  $i$ -th applicant having a set of skills  $A_i$  from the universe  $\{1, \dots, n\}$ , and  $n$  jobs, with the  $j$ -th job requiring a set of skills  $B_j$ . Our goal is to find all the possible applicant-job matches, namely, those pairs  $(i, j)$  such that  $A_i \cap B_j \neq \emptyset$ . One may also be interested in the number of such matches (the  $\ell_0$ -norm) or the most qualified applicants (the entry realizing the  $\ell_\infty$ -norm, or the heavy hitters).

We can further relate set-intersection joins to *Boolean matrix multiplication*. Let  $A$  and  $B$  be two  $n \times n$  matrices such that each row  $A_{i,*}$  is the indicator vector of  $A_i$ , and each column  $B_{*,j}$  is the indicator vector of  $B_j$ . Then the non-zero entries of  $AB$  exactly correspond to the outputs of the set-intersection joins on  $\{A_1, \dots, A_n\}$  and  $\{B_1, \dots, B_n\}$ . If we are interested in estimates to the sizes of the joins, which are very useful for guiding query optimization since they can be computed using much less communication than computing the actual joins, then we have

- $\|AB\|_0 = |\mathcal{A} \circ \mathcal{B}|$ , that is, the  $\ell_0$ -norm of  $AB$  is the size of the composition of  $\mathcal{A}$  and  $\mathcal{B}$ ,
- $\|AB\|_1 = |\mathcal{A} \bowtie \mathcal{B}|$ , that is, the  $\ell_1$ -norm of  $AB$  is the size of the natural join of  $\mathcal{A}$  and  $\mathcal{B}$ .

Finally,  $\|AB\|_\infty$  corresponds to the pair  $(i, j)$  with the maximum overlap, and  $\{(i, j) \mid (AB)_{i,j} \geq \phi \|AB\|_p\}$  for a threshold  $\phi$  corresponds to the set of heavy hitters, i.e., those pairs of sets whose intersection size exceeds the threshold. These two problems have natural applications in inner product similarity joins on a set of vectors; we refer the reader to recent work [3] on inner product similarity joins and references therein.

**Remark 1** We note that all of these problems and the results in this paper can be straightforwardly modified to handle the general case where  $\text{dom}(X) = m_1$ ,  $\text{dom}(Z) = m_2$  and  $\text{dom}(Y) = n$ , which corresponds to  $AB$  where  $A \in \{0, 1\}^{m_1 \times n}$  and  $B \in \{0, 1\}^{n \times m_2}$ . See Section 6 for more discussions.

## 1.2 Our Results

For simplicity we use the notation  $\tilde{O}(\cdot)$  to hide  $\text{poly}(\log \frac{n}{\epsilon\delta})$  factors where  $\epsilon$  is the multiplicative approximation ratio and  $\delta$  is the error probability of a randomized communication algorithm. We say that  $X$  approximates  $Y$  within a factor of  $\alpha$  if  $X \in [\frac{Y}{\beta}, \gamma Y]$  where  $\beta, \gamma \geq 1$  and  $\beta\gamma \leq \alpha$ .

**Set-Intersection Join Size.** We give a 2-round  $\tilde{O}(n/\epsilon)$ -bit algorithm that approximates  $\|AB\|_p$ ,  $p \in [0, 2]$ , within a  $(1 + \epsilon)$  factor. For the important case of  $p = 0$ , this provides a significant improvement over the previous  $\tilde{O}(n/\epsilon^2)$  result in [16]. Also, due to the  $\Omega(n/\epsilon^2)$  lower bound in [16] for one-round algorithms (i.e., algorithms for which Alice sends a single message to Bob, who outputs the answer), this gives a separation in the complexity of this problem for one and two-round algorithms. As the algorithm in [16] is a direct application of an  $\tilde{O}(1/\epsilon^2)$  space streaming algorithm, our algorithm illustrates the power to go beyond streaming algorithms in this framework.

**Pair of Sets with Maximum Intersection Size.** We first give a constant round  $\tilde{O}(n^{1.5}/\epsilon)$ -bit algorithm that approximates  $\|AB\|_\infty$  within a  $(2 + \epsilon)$  factor. We complement our algorithm by showing a few different lower bounds that hold for algorithms with any (not necessarily constant) number of rounds. First, we show that any algorithm that approximates  $\|AB\|_\infty$  within a factor of 2 needs  $\Omega(n^2)$  bits of communication, thus necessitating our  $(2 + \epsilon)$  factor approximation. Moreover, we show that any algorithm achieving any constant factor approximation must use  $\tilde{\Omega}(n^{1.5})$  bits of communication, which shows that our  $(2 + \epsilon)$  factor approximation algorithm has optimal communication, up to polylogarithmic factors.

We next look at approximation algorithms that achieve approximation factors to  $\|AB\|_\infty$  that are larger than constant. We show it is possible to achieve a  $\kappa$ -approximation factor using  $\tilde{O}(n^{1.5}/\kappa)$  bits of communication. We complement this with an  $\Omega(n^{1.5}/\kappa)$  bit lower bound.

Finally we show that the fact that the matrices  $A$  and  $B$  are binary is crucial. Namely, we first show that for general matrices  $A$  and  $B$  with  $\text{poly}(n)$ -bounded integer entries, there is an  $\Omega(n^2)$  lower bound for any constant factor approximation. For general approximation factors  $\kappa$  that may be larger than constant, we show an upper and lower bound of  $\tilde{\Theta}(n^2/\kappa^2)$  communication. This shows an arguably surprising difference in approximation factor versus communication for binary and non-binary matrices.

**Heavy Hitters.** We give an  $O(1)$ -round protocol that computes  $\ell_p(\phi, \epsilon)$ -heavy-hitters,  $0 < \epsilon \leq \phi \leq 1$ , and  $p \in (0, 2]$ , with various tradeoffs depending on whether Alice and Bob’s matrices are arbitrary integer matrices, or whether they correspond to binary matrices. For arbitrary integer matrices, we achieve  $\tilde{O}(\frac{\sqrt{\phi}}{\epsilon} n)$  bits of communication for  $p = 1$ , and  $\tilde{O}(\frac{\phi}{\epsilon} n)$  bits of communication for every other  $p \in (0, 2] \setminus \{1\}$ .

We are able to significantly improve these bounds for binary matrices, which as mentioned above, have important applications to database joins. Here we show for every  $p \in (0, 2]$  an  $O(1)$ -round protocol with  $\tilde{O}(n + \frac{\phi}{\epsilon^2})$  bits of com-

munication.

### 1.3 Related Work

Early work on studying joins in a distributed model can be found in [29] (Section 5) and [24]. Here the goal is to output the actual join rather than its size, and such algorithms, in the worst case, do not achieve communication better than the trivial algorithm in which Alice sends her entire input to Bob for a centralized computation.

With the rise of the MapReduce-type models of computation, a number of works have been devoted to studying parallel and distributed computations of joins. Such works have looked at natural joins, multi-way joins, and similarity joins, in a model called the *massively parallel computation* model (MPC) [2, 9, 10, 17, 23, 25, 26]. Unlike our two-party communication model, in MPC there are multiple parties/machines, and the primary goal is to understand the round-load (maximum message size received by any server in any round) tradeoffs of the computation.

In a recent paper [16] the authors and collaborators studied several join problems in the two-party communication model. The studied problems include set-intersection joins, *set-disjointness joins*, *set-equality joins*, and *at-least- $T$  joins*. Our results can be viewed as a significant extension to the results in [16], as well as a systematic study of classical data stream problems in the context of matrix products. In particular, [16] did not study estimating the  $p$ -norms of  $AB$ , for any  $p$  other than  $p = 0$ . For  $p = 0$ , they obtain an algorithm using  $\tilde{O}(n/\epsilon^2)$  communication, which we significantly improve to  $\tilde{O}(n/\epsilon)$  communication, and extend to any  $0 \leq p \leq 2$ . Moreover, we obtain the first bounds for approximating  $\|AB\|_\infty$ , where perhaps surprisingly, we are able to obtain an  $O(1)$ -approximation in  $\tilde{O}(n^{3/2})$  communication, beating the naïve  $n^2$  amount of communication. This leads us to the first algorithms for finding the frequent entries, or heavy hitters of  $AB$ .

While a number of recent works [6, 11, 22, 27, 36] look at distributed linear algebra problems (for a survey, see [34]), in all papers that we are aware of, the matrix  $C$  is distributed *additively*. What this means is that we want to estimate statistics of a matrix  $C = A + B$ , where  $A$  and  $B$  are held by Alice and Bob, respectively, who exchange messages with each other. In this paper, we instead study the setting for which we want to estimate statistics of a matrix  $C = A \cdot B$ , where  $A$  and  $B$  are again held by Alice and Bob, respectively, who exchange messages with each other. Thus, in our setting the underlying matrix  $C$  of interest is distributed *multiplicatively*. When  $C$  is distributed additively, a common technique is for the players to agree on a random linear sketching matrix  $S$ , and apply it to their inputs to reduce their size. For example, if Alice has matrix  $A$  and Bob has matrix  $B$ , then Alice can send  $S \cdot A$  to Bob, who can compute  $S(A+B)$ . A natural extension of it in the multiplicative case is for Alice to send  $S \cdot A$  to Bob, who can compute  $S \cdot A \cdot B$ . This is precisely how the algorithm for  $p = 0$  of [16] proceeds. We show by using the product structure of  $A \cdot B$  and more than one round, it is possible to obtain significantly less expensive algorithms than this direct sketching approach.

Finally, we would like to mention several papers considering similar problems but working in the centralized model.

In [12], Cohen uses exponential random variables and applies a minimum operation to obtain an unbiased estimator of the number of non-zero entries in each column of a matrix product  $C = AB$ . However, a direct adaptation of this algorithm to the distributed model would result  $\tilde{\Omega}(n/\epsilon^2)$  bits of communication and 1-round, which is the same as using the 1-round  $\ell_0$ -sketching protocol applied to each of the columns in earlier work [16]. In contrast we show that surprisingly, at least to the authors,  $\tilde{O}(n/\epsilon)$  bits of communication is possible with only 2 rounds. In [5], Amossen, Campagna, and Pagh improve the time complexity of [12], provided  $\epsilon$  is not too small. However, a direct adaptation of this algorithm to the distributed model would result an even higher communication cost of  $\Omega(n^2)$ .

In [13], the  $\ell_1$ -sampling problem is considered. In this paper we do not emphasize estimation of  $\|C\|_1$ , since this quantity can be computed exactly using  $O(n \log n)$  bits of communication, as stated in Remark 2. Similarly  $\ell_1$ -sampling can also be done in  $O(n \log n)$  bits of communication, as illustrated in Remark 3.

In [32], it is shown how to apply CountSketch to the entries of a matrix product  $C = AB$  where  $A, B \in \mathbb{R}^{n \times n}$ . The time complexity is  $O(\text{nnz}(A) + \text{nnz}(B) + n \cdot k \log k)$ , where  $\text{nnz}(A)$  denotes the number of non-zero entries of  $A$ , and  $k$  is the number of hash buckets in CountSketch which is at least  $1/\epsilon^2$ . This outperforms the naïve time complexity of first computing  $C$  and then hashing the entries of  $C$  one-by-one. While interesting from a time complexity perspective, it does not provide an advantage over CountSketch in a distributed setting. Indeed, for each of the hashes on Alice's side of the  $n$  outer products computed in [32], the size of the hash is  $\tilde{\Theta}(1/\epsilon^2)$ , and consequently communicating this to Bob takes  $\tilde{\Theta}(n/\epsilon^2)$  bits in total.

## 2. PRELIMINARIES

In this section we give background on several sketching algorithms that we will make use of, as well as some basic concepts in communication complexity. We will also describe some mathematical tools and previous results that will be used in the paper.

For convenience we use  $A \in \mathbb{Z}^{n \times n}$  to differentiate  $A$  from a binary matrix, but we will assume that all the input matrices have polynomially bounded integer entries. For all sketching matrices we will make use of, without explicitly stated, each of their entries can be stored in  $\tilde{O}(1)$  bits.

**Sketches.** A sketch  $sk(x)$  of a data object  $x$  is a summary of  $x$  of small size (sublinear or even polylogarithmic in the size of  $x$ ) such that if we want to perform a query (denoted by a function  $f$ ) on the original data object  $x$ , we can instead apply another function  $g$  on  $sk(x)$  such that  $g(sk(x)) \approx f(x)$ . Sketches are very useful tools in the development of space-efficient streaming algorithms and communication-efficient distributed algorithms. Many sketching algorithms have been developed in the data stream literature. In this paper we will make use of the following.

**Lemma 1 ([19, 21],  $\ell_p$ -Sketch ( $0 \leq p \leq 2$ ))** For  $p \in [0, 2]$  and a data vector  $x \in \mathbb{R}^n$ , there is a sketch  $sk(x) = Sx$  where  $S \in \mathbb{R}^{O(\frac{1}{\epsilon^2} \log \frac{1}{\delta}) \times n}$  is a random sketching matrix,

and a function  $g$  such that with probability  $1 - \delta$ ,  $g(sk(x))$  approximates  $\|x\|_p$  within a factor of  $(1 + \epsilon)$ .

**Communication Complexity.** We will use two-party communication complexity to prove lower bounds for the problems we study. In the two-party communication complexity model, there are parties Alice and Bob. Alice gets an input  $x \in \mathcal{X}$ , and Bob gets an input  $y \in \mathcal{Y}$ . They want to jointly compute a function  $f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathcal{Z}$  via a communication protocol. Let  $\Pi$  be a (randomized) communication protocol, and let  $r_A, r_B$  be the private randomness used by Alice and Bob, respectively. Let  $\Pi_{X,Y,r_A,r_B}$  denote the transcript (the concatenation of all messages) when Alice and Bob run  $\Pi$  on input  $(X, Y)$  using private randomness  $(r_A, r_B)$ , and let  $\Pi(X, Y, r_A, r_B)$  denote the output of the protocol. We say  $\Pi$  errs with probability  $\delta$  if for all  $(x, y) \in \mathcal{X} \times \mathcal{Y}$ ,

$$\Pr_{r_A, r_B}[\Pi_{X,Y,r_A,r_B} \neq f(x, y)] \leq \delta.$$

We define the randomized communication complexity of  $f$ , denoted by  $R_\delta(f)$ , to be  $\min_{\Pi} \max_{x,y,r_A,r_B} |\Pi_{X,Y,r_A,r_B}|$ , where  $|z|$  denotes the length of the transcript  $z$ .

We next introduce a concept called the distributional communication complexity. Let  $\mu$  be a distribution over the inputs  $(X, Y)$ . We say a deterministic protocol  $\Pi$  computes  $f$  with error probability  $\delta$  on  $\mu$  if

$$\Pr_{(X,Y) \sim \mu}[\Pi_{X,Y} \neq f(x, y)] \leq \delta.$$

The  $\delta$ -error distributional communication complexity under input distribution  $\mu$ , denoted by  $D_\delta^\mu(f)$ , is the minimum communication complexity of a deterministic protocol that computes  $f$  with error probability  $\delta$  on  $\mu$ . The following lemma connects distributional communication complexity with randomized communication complexity.

**Lemma 2 (Yao's Lemma)** For any function  $f$  and any  $\delta > 0$ ,  $R_\delta(f) \geq \max_{\mu} D_\delta^\mu(f)$ .

A standard method to obtain randomized communication complexity lower bounds is to first find a hard input distribution  $\mu$  for a function  $f$ , and then try to obtain a lower bound on the distributional communication complexity of  $f$  under inputs  $(X, Y) \sim \mu$ . By Yao's Lemma, this is also a lower bound on the randomized communication complexity of  $f$ .

We now introduce two well-studied problems in communication complexity.

**Set-Disjointness (DISJ).** In this problem Alice holds  $x = (x_1, \dots, x_t) \in \{0, 1\}^t$ , and Bob holds  $y = (y_1, \dots, y_t) \in \{0, 1\}^t$ . They want to compute

$$\text{DISJ}(x, y) = \bigvee_{i=1}^t (x_i \wedge y_i).$$

**Lemma 3 ([7])**  $R_{0.49}(\text{DISJ}) \geq \Omega(n)$ .

**Gap- $l_\infty$ .** In this problem Alice holds  $x = (x_1, \dots, x_t) \in [0, \kappa]^t$ , and Bob holds  $y = (y_1, \dots, y_t) \in [0, \kappa]^t$ , with the following promise: either  $|x_i - y_i| \leq 1$  for all  $i$ ; or for some  $i$ ,  $|x_i - y_i| \geq \kappa$ . Define  $\text{Gap-}l_\infty(x, y) = 1$  if  $\|x - y\|_\infty \geq \kappa$ , and  $\text{Gap-}l_\infty(x, y) = 0$  otherwise.

**Lemma 4 ([7])**  $R_{0.49}(\text{Gap-}l_\infty) \geq \Omega(n/\kappa^2)$ .

**Tools and Previous Results.** We will make use of the following results on distributed matrix multiplication and  $\ell_0$ -sampling on vectors.

**Lemma 5 ([16], Distributed Matrix Multiplication)** Suppose Alice holds a matrix  $A \in \mathbb{R}^{n \times n}$ , and Bob holds a matrix  $B \in \mathbb{R}^{n \times n}$ . There is an algorithm for Alice and Bob to compute  $C_A$  and  $C_B$  such that with probability  $1 - 1/n^{10}$ ,  $C_A + C_B = AB$ . The algorithm uses  $\tilde{O}(n\sqrt{\|AB\|_0})$  bits of communication and 2 rounds.

**Lemma 6 ([20],  $\ell_0$ -Sampling)** For a data vector  $x \in \mathbb{R}^n$ , there is a sketch  $sk(x) = Sx$  where  $S \in \mathbb{R}^{\tilde{O}(1) \times n}$  is a random sketching matrix, and a function  $g$  such that  $g(sk(x))$  returns  $i \in [n]$  for each coordinate  $x_i > 0$  with probability  $1/\|x\|_0$ . The process fails with probability at most  $1/n^{10}$ .

We will also need the standard Chernoff bound.

**Lemma 7 (Chernoff Bound)** Let  $X_1, \dots, X_n$  be independent Bernoulli random variables such that  $\Pr[X_i = 1] = p_i$ . Let  $X = \sum_{i \in [n]} X_i$ . Let  $\mu = \mathbf{E}[X]$ . It holds that  $\Pr[X \geq (1 + \delta)\mu] \leq e^{-\delta^2 \mu/3}$  and  $\Pr[X \leq (1 - \delta)\mu] \leq e^{-\delta^2 \mu/2}$  for any  $\delta \in (0, 1)$ .

### 3. $(1 + \epsilon)$ -APPROXIMATION OF $\ell_p$ ( $p \in [0, 2]$ )

For notational convenience (in order to unify  $\ell_0$  and  $\ell_p$  for constant  $p \in (0, 2]$ ), we define  $\|x\|_0^0 = \|x\|_0$  to be the number of non-zero entries of  $x$ .

Note that for a constant  $p$ , approximating  $\|C\|_p$  within a  $(1 + \epsilon)$  factor and approximating  $\|C\|_p^p$  within a  $(1 + \epsilon)$  factor are asymptotically equivalent – we can always scale the multiplicative error  $\epsilon$  by a factor of  $p$  (a constant), which will not change the asymptotic communication complexity. We will thus use these interchangeably for convenience.

**The Idea.** The high level idea of the algorithm is as follows. We first perform a *rough* estimation – we try to estimate the  $\ell_p$ -norm of each row of  $C$  within a  $(1 + \sqrt{\epsilon})$  factor. We then sample rows of  $C$  with respect to their estimated ( $p$ -th power of their)  $\ell_p$ -norm, obtaining a matrix  $C'$ . We finally use  $C'$  to obtain a *finer* estimation (i.e., a  $(1 + \epsilon)$ -approximation) of  $\|C\|_p^p$ .

**Algorithm.** Set parameters  $\beta = \epsilon^{1/2}$ ,  $\rho = 10^4 \beta^2 / \epsilon^2 = 10^4 / \epsilon$ . The algorithm for approximating  $\ell_p$ -norms for  $p \in [0, 2]$  is presented in Algorithm 1. We describe it in words below.

Alice and Bob first try to estimate the  $\ell_p$ -norm of each row in  $C$  within a factor of  $(1 + \beta)$ . This can be done by letting Bob send an  $\ell_p$ -sketch of  $B^T$  of size  $\tilde{O}(1/\beta^2)$  to Alice using the sketch in Lemma 1; Alice then computes  $\tilde{C} = (SB^T A^T)^T$ . With probability 0.99, we have that for all  $i \in [n]$ ,

$$\left\| \widetilde{C}_{i,*} \right\|_p^p \in \left[ \|C_{i,*}\|_p^p, (1 + \beta) \cdot \|C_{i,*}\|_p^p \right]. \quad (1)$$

We note that we can set  $\beta = \epsilon$  (instead of  $\beta = \sqrt{\epsilon}$ ) and directly get a  $(1 + \epsilon)$  approximation of  $\|C_{i,*}\|_p^p$  for each row

---

**Algorithm 1:**  $(1 + \epsilon)$ -Approximation for  $\ell_p$  ( $p \in [0, 2]$ )

---

**Input :** Alice has a matrix  $A \in \mathbb{Z}^{n \times n}$ , and Bob has a matrix  $B \in \mathbb{Z}^{n \times n}$ . Let  $C \leftarrow AB$

**Output:** A  $(1 + \epsilon)$ -approximation of  $\|C\|_p^p$

---

- 1 Let  $S$  be the sketching matrix in Lemma 1;
  - 2 Bob computes  $SB^T \in \mathbb{R}^{\tilde{O}(1/\beta^2) \times n}$  of  $B^T$  and sends it to Alice;
  - 3 Alice computes  $\tilde{C} \leftarrow (SB^T A^T)^T$ ;
  - 4 Alice partitions the  $n$  rows of  $\tilde{C}$  to (up to)  $L = \log_{1+\beta}(2n^{p+1}) = O(\frac{1}{\beta} \log n)$  groups  $G_1, \dots, G_L$ , such that  $G_\ell$  contains all  $i \in [n]$  for which  $(1 + \beta)^\ell \leq \|\tilde{C}_{i,*}\|_p^p < (1 + \beta)^{\ell+1}$ ;
  - 5 **foreach** group  $G_\ell$  ( $\ell \in [L]$ ) **do**
  - 6     Alice randomly samples each  $i \in G_\ell$  with probability  $p_\ell$ , where  $p_\ell = \frac{\rho}{|G_\ell|} \cdot \frac{\|\tilde{G}_\ell\|_p^p}{\|\tilde{C}\|_p^p}$  where  $\|\tilde{G}_\ell\|_p^p = \sum_{i \in G_\ell} \|\tilde{C}_{i,*}\|_p^p$ ; Alice sends  $p_\ell$  to Bob;
  - 7     Alice then replaces all non-sampled rows in  $A$  with the all-0 vector, obtaining  $A'$ , and sends  $A'$  to Bob;
  - 8 Bob computes  $C' \leftarrow A'B$ , and outputs  $\sum_{\ell \in [L]} \sum_{i \in G_\ell} \frac{1}{p_\ell} \|C'_{i,*}\|_p^p$ .
- 

$i$  (and thus  $\|C\|_p^p$ ). This is exactly what was done in [16]. However, the communication cost in this case is  $\tilde{O}(n/\epsilon^2)$ , which is higher than our goal by a factor of  $1/\epsilon$ .

Alice then sends Bob  $\|\tilde{C}_{i,*}\|_p^p$  for all  $i \in [n]$ . Both parties partition all the rows of  $\tilde{C}$  into up to  $L = O(1/\beta \cdot \log n)$  groups  $G_1, \dots, G_L$ , such that the  $\ell$ -th group  $G_\ell$  contains all  $i \in [n]$  for which

$$(1 + \beta)^\ell \leq \|\tilde{C}_{i,*}\|_p^p < (1 + \beta)^{\ell+1}. \quad (2)$$

By (1) and (2), we have that for each  $i \in G_\ell$ ,

$$(1 + \beta)^\ell \leq \|C_{i,*}\|_p^p < (1 + 3\beta) \cdot (1 + \beta)^\ell. \quad (3)$$

For a fixed group  $G_\ell$ , let  $\|G_\ell\|_p^p = \sum_{i \in G_\ell} \|C_{i,*}\|_p^p$  and  $\|\tilde{G}_\ell\|_p^p = \sum_{i \in G_\ell} \|\tilde{C}_{i,*}\|_p^p$ . For each  $\ell \in [L]$ , set

$$p_\ell = \frac{\rho}{|G_\ell|} \cdot \frac{\|\tilde{G}_\ell\|_p^p}{\|\tilde{C}\|_p^p}.$$

By (1) we have

$$p_\ell \in \left[ \frac{1}{2} \cdot \frac{\rho}{|G_\ell|} \cdot \frac{\|G_\ell\|_p^p}{\|C\|_p^p}, 2 \cdot \frac{\rho}{|G_\ell|} \cdot \frac{\|G_\ell\|_p^p}{\|C\|_p^p} \right] \quad (4)$$

For each  $\ell \in [L]$ , Alice randomly samples each  $i \in G_\ell$  with probability  $p_\ell$ . Alice then sends Bob  $A'$  which consists

of all the sampled rows of  $A$  with other rows being replaced by all-0 vectors. Bob then computes  $C' = A'B$ , and outputs  $\sum_{\ell \in [L]} \sum_{i \in G_\ell} \frac{1}{p_\ell} \|C'_{i,*}\|_p^p$  as the approximation to  $\|C\|_p^p$ .

We can show the following regarding Algorithm 1.

**Theorem 1** For any  $p \in [0, 2]$ , there is an algorithm that approximates  $\|AB\|_p$  for  $A, B \in \mathbb{Z}^{n \times n}$  within a  $(1 + \epsilon)$  factor with probability  $1 - 1/n^{10}$ , using  $\tilde{O}(n/\epsilon)$  bits of communication and 2 rounds.

**Correctness.** For each  $\ell \in [L]$ , and each  $i \in G_\ell$ , let  $X_i^\ell$  be a 0/1 random variable such that  $X_i^\ell = 1$  if  $i \in G_\ell$  is sampled by Alice, and  $X_i^\ell = 0$  otherwise. Define

$$Z^\ell = \frac{1}{p_\ell} \sum_{i \in G_\ell} \left( \|C_{i,*}\|_p^p - \frac{\|G_\ell\|_p^p}{|G_\ell|} \right) X_i^\ell.$$

It is clear that  $\mathbf{E}[Z^\ell] = 0$ . We now compute its variance.

$$\begin{aligned} \mathbf{Var}[Z^\ell] &= \frac{1}{p_\ell^2} \sum_{i \in G_\ell} \left( \left( \|C_{i,*}\|_p^p - \frac{\|G_\ell\|_p^p}{|G_\ell|} \right)^2 \mathbf{Var}[X_i^\ell] \right) \\ &\leq \frac{1}{p_\ell} \sum_{i \in G_\ell} \left( \|C_{i,*}\|_p^p - \frac{\|G_\ell\|_p^p}{|G_\ell|} \right)^2 \\ &\leq \frac{1}{p_\ell} \sum_{i \in G_\ell} \left( 3\beta \cdot \frac{\|G_\ell\|_p^p}{|G_\ell|} \right)^2 \quad (\text{by (3)}) \\ &= \frac{9\beta^2 \cdot (\|G_\ell\|_p^p)^2}{p_\ell |G_\ell|} \\ &\leq \frac{18\beta^2}{\rho} \cdot \|G_\ell\|_p^p \cdot \|C\|_p^p. \quad (\text{by (4)}) \end{aligned}$$

Define  $Z = \sum_{\ell \in [L]} Z^\ell$ . We then have  $\mathbf{E}[Z] = 0$ , and

$$\begin{aligned} \mathbf{Var}[Z] &= \sum_{\ell \in [L]} \mathbf{Var}[Z^\ell] \\ &\leq \frac{18\beta^2}{\rho} \cdot \|C\|_p^p \cdot \sum_{\ell \in [L]} \|G_\ell\|_p^p \\ &\leq \frac{18\beta^2}{\rho} (\|C\|_p^p)^2. \end{aligned}$$

By Chebyshev's inequality, we have

$$\Pr[|Z| \geq \epsilon \cdot \|C\|_p^p] \leq \frac{\mathbf{Var}[Z]}{(\epsilon \cdot \|C\|_p^p)^2} = \frac{18\beta^2}{\rho \epsilon^2} \leq 0.01.$$

We thus have  $\left| \sum_{\ell \in [L]} \sum_{i \in G_\ell} \frac{1}{p_\ell} \|C'_{i,*}\|_p^p - \|C\|_p^p \right| \leq \epsilon \|C\|_p^p$  with probability 0.99 (conditioned on (1) holding, which happens with probability 0.99 as well).

Finally note that we can always boost the success probability of the algorithm from 0.9 to  $(1 - 1/n^{10})$  using the standard median trick and paying another  $O(\log n)$  factor in the communication cost (which will be absorbed by the  $\tilde{O}(\cdot)$  notation).

**Complexity.** The communication cost of sending the  $\ell_p$ -sketch in the first round is  $O(n/\beta^2 \cdot \log n)$  words. The cost of sending the sampled rows is bounded by  $\sum_{\ell \in [L]} (p_\ell |G_\ell| \cdot n)$ . Thus the total communication cost is bounded by

$$\begin{aligned} & \sum_{\ell \in [L]} (p_\ell |G_\ell| \cdot n) + \left( \frac{n}{\beta^2} \cdot \log n \right) \\ &= \tilde{O}(n) \cdot \left( \rho + \frac{1}{\beta^2} \right) \\ &= \tilde{O}(n/\epsilon) \quad (\text{by our choices of } \rho \text{ and } \beta). \end{aligned}$$

It is clear that the whole algorithm finishes in 2 rounds of communication.

**Remark 2** We comment that for  $p = 1$ ,  $\|AB\|_1$  can actually be computed *exactly* using  $O(n \log n)$  bits of communication and 1 round: Alice simply sends  $\|A_{*,j}\|_1$  for each  $j \in [n]$  to Bob, and then Bob computes  $\sum_{j \in [n]} (\|A_{*,j}\|_1 \cdot \|B_{j,*}\|_1)$ , which is exactly  $\|AB\|_1$ .

**Remark 3** We can also perform  $\ell_1$ -sampling on  $C = AB$  using  $O(n \log n)$  bits of communication and 1 round. Alice sends for each  $j \in [n]$  the value  $\|A_{*,j}\|_1$  and a random sample from column  $A_{*,j}$ . Bob computes for each  $j \in [n]$  the value  $\|A_{*,j}\|_1 \cdot \|B_{j,*}\|_1$  as well as  $\sum_{j \in [n]} (\|A_{*,j}\|_1 \cdot \|B_{j,*}\|_1)$ , from which he samples a  $j \in [n]$  proportional to  $\|A_{*,j}\|_1 \cdot \|B_{j,*}\|_1$ . Finally, Bob samples a random entry  $b \in B_{j,*}$ , and if  $a \in A_{*,j}$  is the uniform sample in  $A_{*,j}$  that Alice sent to Bob, Bob outputs the pair  $(a, b)$  as the  $\ell_1$ -sample.

### 3.1 $\ell_0$ -Sampling

We now present a simple algorithm for  $\ell_0$ -sampling. Recall that the goal of  $\ell_0$ -sampling on matrix  $C = AB$  is to sample each non-zero entry in  $C$  with probability  $(1 \pm \epsilon) \frac{1}{\|C\|_0}$ .

The idea is fairly simple: we employ an  $\ell_0$ -sketch and  $\ell_0$ -samplers in parallel. We first use the  $\ell_0$ -sketch to sample a column of  $C$  proportional to its  $\ell_0$ -norm, and then apply the  $\ell_0$ -sampler to that column. For the first step, we use the one-way  $\ell_0$ -sketching algorithm in Lemma 1 to approximate the  $\ell_0$ -norm of each column of  $C$  within a factor of  $1 + \epsilon$ . For the second step, we use the one-way  $\ell_0$ -sampling algorithm for vectors in Lemma 6 for each column of  $C$ .

**Theorem 2** *There is an algorithm that performs  $\ell_0$ -sampling on  $C$  with success probability 0.9 using  $\tilde{O}(n/\epsilon^2)$  bits of communication and 1 round.*

**PROOF.** The size of the  $\ell_0$ -sampler (i.e., the sketching matrix  $S$ ) in Lemma 6 is bounded by  $\tilde{O}(n)$ , and the size of the  $\ell_0$ -sketch in Lemma 1 is bounded by  $\tilde{O}(n/\epsilon^2)$ . Thus the total number of bits of communication is bounded by  $\tilde{O}(n/\epsilon^2) + \tilde{O}(n) = \tilde{O}(n/\epsilon^2)$ . The algorithm finishes in 1 round since both the  $\ell_0$ -sketch and  $\ell_0$ -sampler can be computed in one round.

The success probability follows from a union bound on the success probabilities of the  $\ell_0$ -sketch and  $\ell_0$ -sampler for each of the  $n$  columns of  $C$ .  $\square$

---

### Algorithm 2: $(2 + \epsilon)$ -Approximation for $\ell_\infty$

---

**Input** : Alice has a matrix  $A \in \{0, 1\}^{n \times n}$ , and Bob has a matrix  $B \in \{0, 1\}^{n \times n}$ . Let  $C \leftarrow AB$

**Output**: A  $(2 + \epsilon)$ -approximation of  $\|C\|_\infty$

- 1 **foreach**  $\ell \leftarrow 0, 1, \dots, L$  **do**
  - 2     Alice samples each ‘1’ in  $A$  with probability  $p_\ell = 1/(1 + \epsilon)^\ell$  (and replaces those non-sampled 1’s by 0’s), obtaining matrix  $A^\ell$ ;
  - 3     Let  $C^\ell \leftarrow A^\ell B$ ;
  - 4 **foreach**  $\ell \leftarrow 0, 1, \dots, L$  **do**
  - 5     Alice and Bob compute  $\|C^\ell\|_1$  using Remark 2;
  - 6     Let  $\ell^*$  be the smallest index  $\ell \in \{0, 1, \dots, L\}$  for which  $\|C^\ell\|_1 \leq \gamma n^2$ ;
  - 7 **foreach**  $j \in [n]$  **do**
  - 8     Let  $u_j \leftarrow |\{i \in [n] \mid j \in A_i^{\ell^*}\}|$ , and  $v_j \leftarrow |\{i \in [n] \mid j \in B_i\}|$ ;
  - 9     **if**  $u_j \leq v_j$  **then**
  - 10         Alice sends  $I_j \leftarrow \{i \mid j \in A_i^{\ell^*}\}$  to Bob;
  - 11     **else**
  - 12         Bob sends  $I_j \leftarrow \{i \mid j \in B_i\}$  to Alice;
  - 13 Alice and Bob use the  $I_j$ ’s to compute matrices  $C_A$  and  $C_B$  respectively such that  $C^{\ell^*} = C_A + C_B$ ;
  - 14 Alice and Bob compute  $\|C_A\|_\infty$  and  $\|C_B\|_\infty$ , and output  $\max\{\|C_A\|_\infty / p_{\ell^*}, \|C_B\|_\infty / p_{\ell^*}\}$ .
- 

## 4. $(2 + \epsilon)$ -APPROXIMATION OF $\ell_\infty$

In this section we give almost tight upper and lower bounds for approximating  $\|C\|_\infty$ , that is, the maximum entry in the matrix product  $C$ . We first consider the product of binary matrices, and then consider the product of general matrices.

### 4.1 Upper Bounds for Binary Matrices

#### 4.1.1 An Upper Bound for $2 + \epsilon$ Approximation

**The Idea.** The high level idea is to scale down each entry of  $C$  so that  $\|C\|_1$  is as small as possible subject to the constraint that the largest entry of  $C$  is still approximately preserved (after scaling back). This down-scaling can be done by sampling each 1-entry of  $A$  with a certain probability (we replace the non-sampled 1’s by 0’s). Let  $A'$  be the matrix of  $A$  after applying sampling. Alice and Bob then communicate for each item  $j \in [n]$  the number of rows and columns in  $A'$  and  $B$  respectively that contain item  $j$  (i.e., those rows and columns with  $j$ -th coordinate equal to 1), and the one with the smaller number sends all the indices of those rows/columns to the other party. After this, Alice and Bob can compute matrices  $C_1$  and  $C_2$  independently such that  $C \approx C_1 + C_2$ , and then output  $\max\{\|C_1\|_\infty, \|C_2\|_\infty\}$  as an approximation to  $\|C\|_\infty$ .

**Algorithm.** Let  $L = \log_{1+\epsilon} \|A\|_1 = O(\frac{\log n}{\epsilon})$ . Set  $\gamma = \frac{10^4 \log n}{\epsilon^2}$ . We present the algorithm in Algorithm 2, and describe it in words below.

For  $\ell = 0, 1, \dots, L$ , Alice samples each 1-entry in  $A$  with

probability  $p_\ell = 1/(1 + \epsilon)^\ell$  (i.e., with probability  $(1 - p_\ell)$  the 1-entry is replaced by a 0-entry). Let  $A^\ell$  be the matrix after sampling  $A$  with probability  $p_\ell$ , and let  $C^\ell = A^\ell B$ .

For each  $\ell = 0, 1, \dots, L$ , Alice and Bob compute  $\|C^\ell\|_1$  using Remark 2. Let  $\ell^*$  be the smallest index  $\ell \in \{0, 1, \dots, L\}$  such that  $\|C^\ell\|_1 \leq \gamma n^2$ .

Let us focus on  $A^{\ell^*}$  and  $B$ , and consider each item  $j \in [n]$ . For convenience we identify the rows of  $A^{\ell^*}$  and columns of  $B$  as sets  $\{A_1^{\ell^*}, \dots, A_n^{\ell^*}\}$  and  $\{B_1, \dots, B_n\}$  respectively. Suppose  $j$  appears  $u_j$  times in Alice's sets, and  $v_j$  times in Bob's sets. Alice and Bob exchange the information of  $u_j$  and  $v_j$  for all  $j \in [n]$ . Then for each  $j \in [n]$ , if  $u_j \leq v_j$  then Alice sends all the indices of sets  $A_i^{\ell^*}$  containing  $j$  to Bob, otherwise Bob sends all the indices of sets  $B_i$  containing  $j$  to Alice.

At this point, Alice and Bob can form matrices  $C_A$  and  $C_B$  respectively so that  $C_A + C_B = C^{\ell^*}$ , where  $C_A$  corresponds to the portion of each entry of  $C^{\ell^*}$  restricted to the items  $j$  for which Alice knows the intersections (in other words, Alice knows the inner product defining the entry  $C^{\ell^*}$  restricted to a certain subset of items), and similarly define  $C_B$ . Finally Alice and Bob output  $\max\{\|C_A\|_\infty / p_{\ell^*}, \|C_B\|_\infty / p_{\ell^*}\}$  as the approximation of  $\|C\|_\infty$ .

We have the following theorem.

**Theorem 3** *Algorithm 2 approximates  $\|AB\|_\infty$  for two Boolean matrices  $A, B \in \{0, 1\}^{n \times n}$  within a  $(2 + \epsilon)$  factor with probability 0.9 using  $\tilde{O}(n^{1.5}/\epsilon)$  bits of communication and 3 rounds.*

**Correctness.** We first show that the claimed approximation holds. The following lemma is a key ingredient.

**Lemma 8** *With probability  $1 - 1/n^2$ ,  $\|C^{\ell^*}\|_\infty / p_{\ell^*}$  approximates  $\|C\|_\infty$  within a factor of  $1 + \epsilon$ .*

**PROOF.** We assume that  $\|C\|_1 > \gamma n^2$  since otherwise there is nothing to prove (in this case we have  $p_{\ell^*} = 1$  and  $C^{\ell^*} = C$ ).

We first define a few events.

$$\mathcal{E}_1: \|C^{\ell^*}\|_\infty \geq \frac{1}{2}\gamma.$$

$$\mathcal{E}_2: \text{For all pairs } (i, j), \text{ if } C_{i,j}^{\ell^*} \geq \frac{1}{8}\gamma, \text{ then } C_{i,j}^{\ell^*}/p_{\ell^*} \text{ approximates } C_{i,j} \text{ within a factor of } 1 + \epsilon.$$

$$\mathcal{E}_3: \text{For all pairs } (i, j), \text{ if } C_{i,j}^{\ell^*} < \frac{1}{8}\gamma, \text{ then } C_{i,j} < \frac{1}{4}\gamma/p_{\ell^*}.$$

In words,  $\mathcal{E}_1$  states that the maximum entry of  $C^{\ell^*}$  will be large.  $\mathcal{E}_2$  states that for all large entries  $(i, j)$  in  $C^{\ell^*}$ , the values  $C_{i,j}^{\ell^*}$ , after rescaling by a factor of  $1/p_{\ell^*}$ , can be used to approximate  $C_{i,j}$  within a factor of  $1 + \epsilon$ .  $\mathcal{E}_3$  states that for all small entries  $(i, j)$  in  $C^{\ell^*}$ , the corresponding values  $C_{i,j}$  cannot be the maximum in the matrix  $C$ .

It is not difficult to see that if all three events hold then Lemma 8 holds. Indeed, by  $\mathcal{E}_2$  we can approximate each  $C_{i,j}$  by  $C_{i,j}^{\ell^*}/p_{\ell^*}$  within a factor of  $1 + \epsilon$  as long as  $C_{i,j}^{\ell^*} \geq \frac{1}{8}\gamma$ , and by  $\mathcal{E}_1$  we have  $\|C^{\ell^*}\|_\infty \geq \frac{1}{2}\gamma$ . Therefore

$$\|C\|_\infty \geq \frac{1}{2}\gamma / (p_{\ell^*}(1 + \epsilon)) > \frac{1}{4}\gamma / p_{\ell^*}. \quad (5)$$

By  $\mathcal{E}_3$ , for all  $(i, j)$  with  $C_{i,j}^{\ell^*} < \frac{1}{8}\gamma$ , we have  $C_{i,j} < \frac{1}{4}\gamma/p_{\ell^*}$ ; by (5) we know that these entries  $(i, j)$  cannot be the maximum in  $C$ . We can thus conclude that  $\|C^{\ell^*}\|_\infty$  approximates  $\|C\|_\infty / p_{\ell^*}$  within a factor of  $1 + \epsilon$ .

In the rest of this section we show that each of  $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3$  holds with probability  $1 - 1/n^4$ . The success probability in Lemma 8 follows by a union bound.

For  $\mathcal{E}_1$ , we only need to show that  $\|C^{\ell^*}\|_1 \geq \frac{1}{2}\gamma n^2$ . Recall that  $\ell^*$  is the smallest index  $\ell \in \{0, 1, \dots, L\}$  such that  $\|C^\ell\|_1 \leq \gamma n^2$ . We thus have  $\|C^{\ell^*-1}\|_1 > \gamma n^2$ . We can view  $C^{\ell^*}$  as sampling each entry of  $C^{\ell^*-1}$  with probability  $1/(1 + \epsilon)$ . By a Chernoff bound, with probability  $1 - 1/n^{10}$  we have  $\|C^{\ell^*}\|_1 \geq \frac{1}{2}\gamma n^2$ . Consequently, we have  $\|C^{\ell^*}\|_\infty \geq \|C^{\ell^*}\|_1 / n^2 \geq \frac{1}{2}\gamma$ .

For  $\mathcal{E}_2$ , let us first focus on a particular pair  $(i, j)$ . Let  $z = C_{i,j}$ , and let  $k_1, \dots, k_z \in [n]$  be the indices for which  $A_{i,k_t}^{\ell^*} = B_{k_t,j} = 1$  for all  $t = 1, \dots, z$ . For each  $t \in [z]$ , define the random variable  $X_t$  such that  $X_t = 1$  if  $A_{i,k_t}^{\ell^*}$  is sampled in  $A^{\ell^*}$ , and  $X_t = 0$  otherwise. Let  $X = \sum_{t \in [z]} X_t$ . We thus have  $X = C_{i,j}^{\ell^*}$ , and

$$\mathbf{E}[X] = \sum_{t \in [z]} \mathbf{E}[X_t] = p_{\ell^*} \cdot z. \quad (6)$$

The claim is  $\mathbf{E}[X] \geq \frac{1}{16}\gamma$  with probability  $1 - 1/n^{10}$ . Suppose to the contrary that  $\mathbf{E}[X] < \frac{1}{16}\gamma$ . We can just consider the case that  $\mathbf{E}[X] \in [\frac{1}{32}\gamma, \frac{1}{16}\gamma)$  and argue that with probability  $1 - 1/n^{10}$  we have  $X < \frac{1}{8}\gamma$ , which contradicts the assumption of  $\mathcal{E}_2$  that  $X = C_{i,j}^{\ell^*} \geq \frac{1}{8}\gamma$ . Note that this is sufficient since if  $\mathbf{E}[X] < \frac{1}{32}\gamma$  then the probability that  $X < \frac{1}{8}\gamma$  will be even higher. In the case when  $\mathbf{E}[X] \in [\frac{1}{32}\gamma, \frac{1}{16}\gamma)$ , by a Chernoff bound we have

$$X \in [(1 - \epsilon)\mathbf{E}[X], (1 + \epsilon)\mathbf{E}[X]] \subseteq \left[ \frac{1}{64}\gamma, \frac{1}{8}\gamma \right)$$

with probability  $1 - 1/n^{10}$ .

Now in the case that  $\mathbf{E}[X] \geq \frac{1}{16}\gamma$ , by another Chernoff bound we have  $X \in [(1 - \epsilon)\mathbf{E}[X], (1 + \epsilon)\mathbf{E}[X]]$  with probability  $1 - 1/n^{10}$ ; in other words,  $X/p_{\ell^*} (= C_{i,j}^{\ell^*}/p_{\ell^*})$  approximates  $\mathbf{E}[X]/p_{\ell^*} (= z = C_{i,j})$  within a factor of  $1 + \epsilon$ . Finally, by a union bound on at most  $n^2$  pairs  $(i, j)$ , the probability that  $\mathcal{E}_2$  holds is at least  $1 - 1/n^4$ .

For  $\mathcal{E}_3$ , we again focus on a particular pair  $(i, j)$ , and will reuse the notation in the analysis of  $\mathcal{E}_2$ . The observation is that if  $\mathbf{E}[X] \geq \frac{1}{4}\gamma$ , then  $X \geq (1 - \epsilon)\mathbf{E}[X] \geq \frac{1}{8}\gamma$  with probability  $1 - 1/n^{10}$ , contradicting the assumption of  $\mathcal{E}_3$ . We thus have  $C_{i,j} = z = \mathbf{E}[X]/p_{\ell^*} < \frac{1}{4}\gamma/p_{\ell^*}$  with probability  $1 - 1/n^{10}$ . Finally by a union bound on at most  $n^2$  pairs of  $(i, j)$ , the probability that  $\mathcal{E}_3$  holds is at least  $1 - 1/n^4$ .  $\square$

We now wrap up the correctness proof of the theorem. At the end of Algorithm 2 Alice and Bob obtain two matrices  $C_A$  and  $C_B$  such that  $C_A + C_B = C^{\ell^*}$ . We thus have  $\max\{\|C_A\|_\infty, \|C_B\|_\infty\} \geq \|C^{\ell^*}\|_\infty / 2$ . Combining

this with Lemma 8 we obtain

$$\frac{\|C\|_\infty}{2(1+\epsilon)} \leq \max \left\{ \frac{\|C_A\|_\infty}{p_{\ell^*}}, \frac{\|C_B\|_\infty}{p_{\ell^*}} \right\} \leq (1+\epsilon) \|C\|_\infty.$$

**Complexity.** By Remark 2, the step of computing  $\|C^\ell\|_1$  for all  $\ell = 0, 1, \dots, L$  costs  $\tilde{O}(L \cdot n) = \tilde{O}(n)$  bits. The exchanging of  $\{u_j, v_j \mid j \in [n]\}$  costs  $\tilde{O}(n)$  bits. The last step of computing  $\max\{\|C_A\|_\infty, \|C_B\|_\infty\}$  costs  $\tilde{O}(1)$  bits.

Now we consider the step of exchanging the indices of sets containing  $j$  for each  $j \in [n]$ . We analyze two cases. In the case that  $u_j, v_j > \sqrt{n}/\epsilon$ , there will be at most

$$\|C^{\ell^*}\|_1 \leq \frac{2\gamma n^2}{u_j \cdot v_j}$$

such items  $j$ . The total communication for such  $j$ 's is bounded by

$$\begin{aligned} \sum_{j: u_j, v_j > \sqrt{n}/\epsilon} \min\{u_j, v_j\} &\leq \sum_{\ell \geq 0} \frac{\gamma n^2}{n/\epsilon^2 \cdot 2^{2\ell}} \cdot \sqrt{n}/\epsilon \cdot 2^\ell \\ &= \tilde{O}(\gamma \epsilon n^{1.5}) = \tilde{O}(n^{1.5}/\epsilon). \end{aligned}$$

In the case that  $\min\{u_j, v_j\} \leq \sqrt{n}/\epsilon$ , we directly have

$$\sum_{j: \min\{u_j, v_j\} \leq \sqrt{n}/\epsilon} \min\{u_j, v_j\} \leq \sum_{j \in [n]} \sqrt{n}/\epsilon \leq n^{1.5}/\epsilon.$$

Summing up, the total communication cost is bounded by  $\tilde{O}(n^{1.5}/\epsilon)$ .

Finally we show that Algorithm 2 can be implemented in 3 rounds. In Round 1, for each level  $\ell$  Alice sends Bob  $\{\|A_{*,j}\|_1 \mid j \in [n]\}$  so that Bob can compute  $\|AB\|_1$  according to Remark 2, and consequently finds  $\ell^*$ . In Round 2, Bob sends  $\ell^*$  to Alice, together with all  $I_j$  corresponding to those  $j$  with  $u_j > v_j$ . In Round 3, Alice sends Bob all  $I_j$  corresponding to those  $j$  with  $u_j \leq v_j$ . Alice also forms  $C_A$ , computes and sends  $\|C_A\|_\infty$  to Bob. Finally Bob forms  $C_B$ , and computes  $\max\{\|C_A\|_\infty, \|C_B\|_\infty\}$  as the final output.

#### 4.1.2 An Upper Bound for General $\kappa$ -Approximation

**The Idea and Algorithm.** We next consider protocols obtaining a  $\kappa$ -approximation to  $\|C\|_\infty$  for a general approximation factor  $\kappa > 1$ . One way to do this is to exactly follow Algorithm 2. That is, we first scale down the entries of  $C$  by sampling the 1-entries in  $A$  to a level for which  $\|C^\ell\|_1 \leq \alpha n^2/\kappa$  where  $\kappa$  is the approximation ratio, and  $\alpha = \Theta(\log n)$ . If we continue to follow Algorithm 2, then we will get an  $\tilde{O}(n^{1.5}/\sqrt{\kappa})$  bound. We now show how to improve the bound to  $\tilde{O}(n^{1.5}/\kappa)$ .

The main change we make to Algorithm 2 is that we add a universe sampling step at the beginning. More precisely, we sample each column of  $A$  with probability  $q = \min\{\alpha/\kappa, 1\}$  where  $\alpha = 10^4 \log n$ , and then replace all non-sampled columns in  $A$  with all-0 vectors, obtaining a new matrix  $A'$ . Let  $D = A'B$ . Recall that  $C = AB$ . We compute  $\|C\|_1$  and  $\|D\|_1$ .

With this new universe sampling step it is possible to have  $\|D\|_1 = 0$ . If this happens then we also check  $\|C\|_1$ . If  $\|C\|_1 = 0$  then we simply output 0; otherwise we output

---

#### Algorithm 3: $\kappa$ -Approximation for $\ell_\infty$

---

**Input** : Alice has a matrix  $A \in \{0, 1\}^{n \times n}$ , and Bob has a matrix  $B \in \{0, 1\}^{n \times n}$ . Let  $C \leftarrow AB$

**Output**: A  $\kappa$ -approximation of  $\|C\|_\infty$

---

- 1 Set  $q = \min\{\alpha/\kappa, 1\}$  where  $\alpha = 10^4 \log n$ ;
  - 2 Alice samples each column of  $A$  with probability  $q$  (and replaces those non-sampled columns by the all-0 vector), obtaining  $A'$ . Let  $D \leftarrow A'B$ ;
  - 3 Alice and Bob compute  $\|D\|_1$  and  $\|C\|_1$ ;
  - 4 **if**  $\|D\|_1 = 0$  **then**
  - 5     **if**  $\|C\|_1 = 0$  **then** Output 0;
  - 6     **else** Output 1;
  - 7 **else**
  - 8     Follow Algorithm 2 and further sample  $A'$  with probability  $p_\ell = 1/2^\ell$  (instead of  $p_\ell = 1/(1+\epsilon)^\ell$ ) for  $\ell = 0, 1, \dots, \log_2 \|A'\|_1$ , and with the threshold  $\gamma n^2$  at Line 6 being replaced by  $\alpha/\kappa \cdot n^2$ . Finally output  $\max\{\|C_A\|_\infty/(q \cdot p_{\ell^*}), \|C_B\|_\infty/(q \cdot p_{\ell^*})\}$ .
- 

1. If  $\|D\|_1 > 0$ , then we follow Algorithm 2 to do further sampling on  $A'$ , obtaining  $A^1, A^2, \dots$ . Let  $C^\ell = A^\ell B$  for  $\ell = 1, 2, \dots$ . We again stop at the first level  $\ell^*$  for which  $\|C^{\ell^*}\|_1 \leq \alpha n^2/\kappa$ , and then exchange for each (surviving) universe item  $j$  the indices of sets that contain  $j$ , in exactly the same way as that in Algorithm 2.

The algorithm is presented in Algorithm 3. We have the following theorem.

**Theorem 4** *Algorithm 3 approximates  $\|AB\|_\infty$  for two Boolean matrices  $A, B \in \{0, 1\}^{n \times n}$  within a factor of  $\kappa$  for any  $\kappa \in [4, n]$  with probability 0.9 using  $\tilde{O}(n^{1.5}/\kappa)$  bits of communication and  $O(1)$  rounds.*

**Correctness.** For simplicity we assume that  $\alpha/\kappa \leq 1$  (and thus  $q = \alpha/\kappa$ ), since otherwise  $D = C$  and the arguments will follow those in Algorithm 2.

We define two events, and will show that each holds with probability  $1 - 1/n^4$ .

$\mathcal{E}_4$ : For all pairs  $(i, j)$ , if  $D_{i,j} \geq \frac{1}{8}\alpha$ , then  $D_{i,j}/q$  approximates  $C_{i,j}$  within a factor of 2.

$\mathcal{E}_5$ : For all pairs  $(i, j)$ , if  $D_{i,j} < \frac{1}{8}\alpha$ , then  $C_{i,j} < \frac{1}{4}\alpha/q$ .

We first assume that  $\|D\|_\infty > 0$ . Consider a pair  $(i, j)$ , if  $D_{i,j} < \frac{1}{8}\alpha$ , then we know by  $\mathcal{E}_5$  that  $C_{i,j} < \frac{1}{4}\alpha/q = \frac{1}{4}\kappa$ . Otherwise if  $D_{i,j} \geq \frac{1}{8}\alpha$  then by  $\mathcal{E}_4$  we know that  $D_{i,j}/q$  approximates  $C_{i,j}$  within a factor of 2. We thus conclude that  $\|D\|_\infty$  approximates  $\|C\|_\infty$  within a factor of  $\kappa/4$  if  $\|D\|_\infty > 0$ .

In the case that  $\|D\|_\infty = 0$ , by  $\mathcal{E}_5$  we know that all entries in  $C$  are less than  $\kappa/4$ . Then we can test whether  $\|C\|_1 > 0$ . If the answer is yes then we can output 1, which already approximates  $\|C\|_\infty$  within a factor of  $\kappa$ ; otherwise we know that  $C$  is the zero matrix, and we can output 0.



The proofs that each of  $\mathcal{E}_4$  and  $\mathcal{E}_5$  hold with probability  $1 - 1/n^4$  are analogous to those for  $\mathcal{E}_2$  and  $\mathcal{E}_3$  in the proof of Lemma 8.

**Complexity.** The analysis of the communication cost is again similar to that of Algorithm 2, and the bottleneck is still the exchange of the indices of sets containing  $j$  for each  $j \in [n]$ . We again analyze two cases. Note that after sampling we have  $\|C^{\ell^*}\|_1 = \tilde{O}(n^2/\kappa)$ , and the universe size is  $\tilde{O}(n/\kappa)$ .

- If  $\min\{u_j, v_j\} \leq \sqrt{n}$ , then since the universe size is  $\tilde{O}(n/\kappa)$ , the total communication is upper bounded by  $\tilde{O}(n/\kappa) \cdot \sqrt{n} = \tilde{O}(n^{1.5}/\kappa)$ .
- If  $\min\{u_j, v_j\} > \sqrt{n}$ , then since  $\|C^{\ell^*}\|_1 = \tilde{O}(n^2/\kappa)$ , the total communication is upper bounded by  $\|C^{\ell^*}\|_1 / \sqrt{n} = \tilde{O}(n^{1.5}/\kappa)$ .

Therefore the total communication is bounded by  $\tilde{O}(n^2/\kappa)$ . The number of rounds is clearly bounded by  $O(1)$ .

## 4.2 Lower Bounds for Binary Matrices

In this section we show that our algorithms for  $\ell_\infty$ -norm estimation in Section 4.1 are almost tight in the sense that (1)  $\Omega(n^2)$  bits of communication is needed if we want to go beyond a  $2+\epsilon$  approximation, and (2) for any approximation  $\kappa$  we need to use  $\Omega(n^{\frac{3}{2}}/\kappa)$  bits of communication.

### 4.2.1 A Lower Bound for 2-Approximation

**Theorem 5** Any algorithm that approximates  $\|AB\|_\infty$  for two Boolean matrices  $A, B \in \{0, 1\}^{n \times n}$  within a factor of 2 with probability 0.51 needs  $\Omega(n^2)$  bits of communication, even if we allow an unbounded number of communication rounds.

**PROOF.** We perform a reduction from the two-player set-disjointness (see Section 2) on strings of length  $(n/2)^2 = n^2/4$ , where Alice has  $x$  and Bob has  $y$ . Alice creates an  $n/2 \times n/2$  matrix  $A'$  indexed by the coordinates in  $x$ , that is, the  $i$ -th ( $i = 1, \dots, n/2$ ) row of  $A'$  consists of the  $((i-1)\frac{n}{2} + 1)$ -th,  $\dots$ ,  $\frac{in}{2}$ -th coordinates of  $x$ . Similarly, Bob creates an  $n/2 \times n/2$  matrix  $B'$  indexed by the coordinates in  $y$ . Next, Alice creates an  $n \times n$  input matrix

$$A = \begin{bmatrix} A' & I \\ \mathbf{0} & \mathbf{0} \end{bmatrix},$$

where  $I$  is an  $n/2 \times n/2$  identity matrix, and  $\mathbf{0}$  is an  $n/2 \times n/2$  all-0 matrix. Bob creates an  $n \times n$  input matrix

$$B = \begin{bmatrix} I & \mathbf{0} \\ B' & \mathbf{0} \end{bmatrix}.$$

Note that  $A$  and  $B$  are both binary matrices, as needed for the reduction to the  $\|AB\|_\infty$  problem.

The key is to observe that

$$A \cdot B = \begin{bmatrix} A' + B' & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \quad (7)$$

We thus have  $\|A \cdot B\|_\infty = \|A' + B'\|_\infty$ , which is 2 if  $x \cap y \neq \emptyset$ , and 1 otherwise. The claimed lower bound for approximating  $\|C\|_\infty$  within a factor of 2 follows from the  $\Omega(n^2)$  lower bounds for two-player set-disjointness on strings of length  $\Theta(n^2)$  for success probability 0.51 (Lemma 3).  $\square$

### 4.2.2 A Lower Bound for General $\kappa$ -Approximation

**Theorem 6** For any  $\kappa \in [1, n]$ , any randomized algorithm that approximates  $\|AB\|_\infty$  for two Boolean matrices  $A, B \in \{0, 1\}^{n \times n}$  within a factor of  $\kappa$  with probability 0.52 needs  $\tilde{\Omega}(n^{\frac{3}{2}}/\kappa)$  bits of communication, even if we allow an unbounded number of communication rounds.

The proof is again by a reduction from a communication problem which is highly structured. We first introduce a few simple communication problems which will be used as building blocks to construct the final communication problem that we will use for the reduction.

Set  $\beta = \sqrt{50 \log n/n}$ , and set  $k = 1/(4\kappa\beta^2)$  where  $\kappa$  is the approximation ratio.

*The AND Problem.* In this problem Alice holds a bit  $x$  and Bob holds a bit  $y$ . They want to compute  $\text{AND}(x, y) = x \wedge y$ .

Let  $X$  be Alice's input and  $Y$  be Bob's input. We define two input distributions for  $(X, Y)$ . Let  $W$  be a random bit such that  $\Pr[W = 0] = \Pr[W = 1] = 1/2$ ; let  $\lambda$  be the distribution of  $W$ .

$\nu_1$ : We first choose  $W \sim \lambda$ . If  $W = 0$ , we set  $(X, Y) = (0, 0)$  with probability  $1 - \beta$ , and  $(X, Y) = (0, 1)$  with probability  $\beta$ . If  $W = 1$ , we set  $(X, Y) = (0, 0)$  with probability  $1 - \beta$ , and  $(X, Y) = (1, 0)$  with probability  $\beta$ .

$\mu_1$ : Set  $(X, Y) = (0, 0)$  with probability  $1/2$ , and  $(X, Y) = (1, 1)$  with probability  $1/2$ .

*The DISJ Problem.* Recall the set-disjointness problem introduced in Section 2, where Alice holds  $x = (x_1, \dots, x_k) \in \{0, 1\}^k$ , and Bob holds  $y = (y_1, \dots, y_k) \in \{0, 1\}^k$ , and they want to compute  $\text{DISJ}(x, y) = \bigvee_{i=1}^k \text{AND}(x_i, y_i)$ .

Let  $X = (X_1, \dots, X_k)$  be Alice's input, and  $Y = (Y_1, \dots, Y_k)$  be Bob's input. We again define two input distributions for  $(X, Y)$ .

$\nu_k$ : Set  $(X_i, Y_i) \sim \nu_1$  for each  $i \in [k]$ .

$\mu_k$ : We first set  $(X_i, Y_i) \sim \nu_k$ , and then pick  $M$  uniformly at random from  $\{1, \dots, k\}$ , and reset  $(X_M, Y_M) \sim \mu_1$ .

*The SUM Problem.* In this problem Alice holds  $u = (u_1, \dots, u_n)$  where  $u_i \in \{0, 1\}^k$  for each  $i \in [n]$ , and Bob holds  $v = (v_1, \dots, v_n)$  where  $v_i \in \{0, 1\}^k$  for each  $i \in [n]$ . They want to compute  $\text{SUM}(u, v) = \sum_{i=1}^n \text{DISJ}(u_i, v_i)$ .

Let  $U = (U_1, \dots, U_n)$  be Alice's input, and  $V = (V_1, \dots, V_n)$  be Bob's input. We define the following input distribution for  $(U, V)$ .

$\phi$ : We first set  $(U_i, V_i) \sim \nu_k$ , and then pick a  $D$  uniformly at random from  $\{1, \dots, n\}$ , and reset  $(U_D, V_D) \sim \mu_k$ .

Note that under  $(U, V) \sim \phi$ ,  $\Pr[\text{SUM}(U, V) = 0] = \Pr[\text{SUM}(U, V) = 1] = 1/2$ . Using the standard information complexity machinery (which we omit here; and can be found in for example [18, 35]) we can show the following.

**Theorem 7** *Any deterministic algorithm solving  $\text{SUM}(U, V)$  correctly with probability 0.51 under  $(U, V) \sim \phi$  needs  $\Omega(\beta kn)$  bits of communication.*

*Input Reduction.* We now perform a reduction from  $\text{SUM}$  to the  $\ell_\infty$ -norm estimation problem. Given  $(U, V) \sim \phi$ , we construct matrices  $A$  and  $B$  as follows. We set  $A = [A^1, \dots, A^{n/k}]$  where  $A^1 = \dots = A^{n/k}$ , and for each  $A^z$  ( $z \in [n/k]$ ) we have  $A_{i,*}^z = U_i$  for all  $i \in [n]$ . Similarly, we set  $B = [B^1, \dots, B^{n/k}]^T$  where  $B^1 = \dots = B^{n/k}$ , and for each  $B^z$  ( $z \in [n/k]$ ) we have  $B_{*,i}^z = V_i$  for all  $i \in [n]$ . Let  $\psi$  denote the resulting distribution of  $(A, B)$ . We have the following lemma.

**Lemma 9** *For any  $\kappa$ , any deterministic algorithm that approximates  $\|AB\|_\infty$  within a factor of  $\kappa$  with probability  $\delta$  under  $(A, B) \sim \psi$  can be used to compute  $\text{SUM}(U, V)$  with probability  $(\delta + 0.01)$  under  $(U, V) \sim \phi$ .*

PROOF. Let  $(U, V) \sim \phi$ , and let  $(A, B)$  be constructed using  $(U, V)$  as described in the input reduction above. Let  $C = AB$ . We first compute the value of  $\|C\|_\infty$ .

We analyze two cases. When  $\text{SUM}(U, V) = 0$ , we have  $\text{DISJ}(U_i, V_i) = 0$  for all  $i \in [n]$ . Consider a pair  $(i, j)$  ( $i, j \in [n], i \neq j$ ). We analyze the inner product  $\langle A_{i,*}, B_{*,j} \rangle$ . For each  $t \in [k]$ , the probability that  $A_{i,t} = B_{t,j} = 1$  is at most  $\beta^2$ . We thus have

$$\mathbb{E}[\langle A_{i,*}, B_{*,j} \rangle] \leq \beta^2 n.$$

By a Chernoff bound we have  $\langle A_{i,*}, B_{*,j} \rangle \leq 2\beta^2 n$  with probability  $1 - e^{-\beta^2 n/3} \geq 1 - 1/n^{10}$ . By a union bound on all pairs  $(i, j)$  ( $i \neq j$ ), we have that with probability  $1 - 1/n^8$ ,  $C_{i,j} = \langle A_{i,*}, B_{*,j} \rangle \leq 2\beta^2 n$  for all  $(i, j)$  ( $i \neq j$ ). Consequently,

$$\|C\|_\infty \leq 2\beta^2 n. \quad (8)$$

When  $\text{SUM}(U, V) = 1$ , we have  $\text{DISJ}(U_i, V_i) = 0$  for all  $i \in [n] \setminus D$ , and  $\text{DISJ}(U_D, V_D) = 1$ . We thus have

$$\|C\|_\infty \geq n/k. \quad (9)$$

By our choices of parameters  $\beta$  and  $k$ , we have

$$(n/k)/(2\beta^2 n) = 2\kappa > \kappa.$$

The lemma thus follows from (8) and (9).  $\square$

Theorem 6 follows from Lemma 9, Theorem 7, our choices of  $\beta$  and  $k$ , and Yao's minimax lemma.

### 4.3 General Matrices

Finally we observe that the communication complexity for approximating  $\|AB\|_\infty$  for non-binary matrices  $A, B$  is significantly different than that for binary matrices.

**Theorem 8** *Let  $A \in \mathbb{Z}^{n \times n}$  and  $B \in \mathbb{Z}^{n \times n}$ . In the two-party communication model we have:*

1. *There is an algorithm that computes  $\|AB\|_\infty$  within a factor  $\kappa$  using  $\tilde{O}(n^2/\kappa^2)$  bits of communication and one round.*
2. *Any algorithm that approximates  $\|AB\|_\infty$  within a factor  $\kappa$  needs  $\tilde{\Omega}(n^2/\kappa^2)$  bits of communication, even if we allow an arbitrary number of communication rounds.*

For the upper bound, we first recall a simple algorithm for sketching  $\|x\|_\infty$  ( $x \in \mathbb{Z}^n$ ).<sup>1</sup> We first partition the vector  $x$  into  $n/\kappa^2$  blocks each of size  $\kappa^2$ , and then use the AMS sketching algorithm [4] for  $\ell_2$ -norm estimation for each block; the sketch size is  $\tilde{O}(1)$  if we target an  $O(1)$ -approximation and  $1 - 1/n^{10}$  success probability. Since for each vector  $y \in \mathbb{Z}^{\kappa^2}$  we have  $\|y\|_\infty \in \left[ \frac{\|y\|_2}{\kappa}, \|y\|_2 \right]$ , we obtain a sketch of size  $\tilde{O}(n/\kappa^2)$  for estimating  $\|x\|_\infty$  within a factor of  $\kappa$ . Denote this sketching matrix by  $S \in \mathbb{R}^{\tilde{O}(n/\kappa^2) \times n}$ .

In the matrix product setting Alice simply applies  $S$  to  $A$  and sends  $SA \in \mathbb{R}^{\tilde{O}(n/\kappa^2) \times n}$  to Bob. Bob then estimates the  $\ell_\infty$ -norm of each column of  $C (= AB)$  using  $SA$  and  $B$  (and computing  $SA \cdot B$ ), and then outputs  $\max_{j \in [n]} \|C_{*,j}\|_\infty$ .

For the lower bound, we again use the technique in Section 4.2.1 to convert a matrix product to a matrix sum, and then perform a reduction from the  $\ell_\infty$ -norm estimation problem (see Section 2). Given two vectors  $x, y \in [0, \kappa]^{n^2/4}$ , we construct  $A', B'$  and  $A, B$  exactly the same way as that in Section 4.2.1. We then have  $\|A \cdot B\|_\infty = \|A' + B'\|_\infty$ , which evaluates to  $\kappa$  if  $\text{Gap-}l_\infty(x, y) = 1$ , and evaluates to at most 1 if  $\text{Gap-}l_\infty(x, y) = 0$ . The lower bound follows from Lemma 4.

## 5. APPROXIMATE HEAVY HITTERS

In this section we consider the  $\ell_p$ - $(\phi, \epsilon)$ -heavy-hitter problem described in the introduction. We first propose an algorithm for products of general matrices, and then consider the problem for binary matrices.

### 5.1 General Matrices

We first consider  $p = 1$ . General  $p \in (0, 2]$  can be handled in a similar way.

**The Idea.** The idea for computing approximate heavy hitters is similar to our ideas for the  $\ell_\infty$ -norm, that is, we sample 1-entries in  $A$  to scale down the values of entries in  $C$  to a level such that the heavy-hitter entries are still non-zero, while there are not many non-zero entries corresponding to non-heavy-hitter entries. Let  $C'$  denote the matrix  $C$  after we scale down. Since there cannot be many heavy hitters, the number of non-zero entries in  $C'$  is small. We can thus perform a sparse recovery algorithm on  $C'$  to find all the heavy hitters.

**Algorithm.** We present the algorithm in Algorithm 4, and describe it in words below.

Alice and Bob first compute  $\|C\|_1$  using Remark 2. Next, similar to Algorithm 2 for approximating  $\|C\|_\infty$ , we sample the 1-entries in matrix  $A$ . The sampling is simpler in this

<sup>1</sup>This algorithm was described in [33].

---

**Algorithm 4:** Computing  $\ell_1$ - $(\phi, \epsilon)$ -Heavy-Hitters

---

**Input :** Alice has a matrix  $A \in \tilde{O}(n/\kappa^2)^{n \times n}$ , and Bob has a matrix  $B \in \tilde{O}(n/\kappa^2)^{n \times n}$ . Let  $C \leftarrow AB$

**Output:**  $\ell_1$ - $(\phi, \epsilon)$ -Heavy-Hitters of  $C$

- 1 Alice and Bob compute  $\|C\|_1$ ;
  - 2 Set the sampling rate  $\beta \leftarrow \min \left\{ \frac{10^4 \log n}{\left(\frac{\epsilon}{\phi}\right)^2 \cdot \frac{\phi}{8} \|C\|_1}, 1 \right\}$ ;
  - 3 Alice samples each 1-entry in  $A$  with probability  $\beta$  (and replaces all the non-sampled 1's by 0's), obtaining matrix  $A^\beta$ ; let  $C^\beta \leftarrow A^\beta B$ ;
  - 4 Alice and Bob then use Lemma 5 to recover all the non-zero entries of  $C^\beta$ ; the recovered matrix  $C^\beta$  is distributed at Alice's side and Bob's side, denoted by  $C_A$  and  $C_B$  where  $C^\beta = C_A + C_B$ ;
  - 5 Alice creates  $C'_A$  consisting of all entries in  $C_A$  that are larger than  $\frac{\epsilon\beta}{8} \|C\|_1$ , and sends  $C'_A$  to Bob. Bob outputs all entries in  $C' = C'_A + C_B$  that are at least  $\beta \cdot (\phi - \frac{\epsilon}{2}) \|C\|_1$ .
- 

case since we only need to sample the entries at the fixed ratio  $\beta$ . Let  $C^\beta$  be the resulting matrix after sampling.

Alice and Bob then use Lemma 5 to recover all the non-zero entries in  $C^\beta$ ; the entries of the recovered  $C^\beta$  are distributed across the two parties, denoted by  $C_A$  and  $C_B$  where  $C^\beta = C_A + C_B$ . Alice then sends all "heavy" entries in  $C_A$ , that is, those whose values are larger than  $\frac{\epsilon\beta}{8} \|C\|_1$ , to Bob. Bob then outputs all the heavy hitters in  $C'$  which is constructed by adding the heavy entries of  $C_A$  (received from Alice) to  $C_B$ .

**Theorem 9** *Algorithm 4 computes the  $\ell_1$ - $(\phi, \epsilon)$ -heavy-hitters ( $0 < \epsilon \leq \phi \leq 1$ ) of  $AB$ , where  $A, B \in \mathbb{Z}^{n \times n}$ , with probability 0.9 and using  $\tilde{O}(\frac{\sqrt{\phi}}{\epsilon} n)$  bits of communication and  $O(1)$  rounds.*

We will assume that  $\|C\|_1 \geq \frac{10^4 \log n}{\left(\frac{\epsilon}{\phi}\right)^2 \cdot \frac{\phi}{8}} = \frac{8 \cdot 10^4 \phi \log n}{\epsilon^2}$ , since otherwise  $\beta = 1$ , and then  $C^\beta = C$ , in which case the proof is only simpler.

**Correctness.** We define two events.

$\mathcal{E}_6$ : For all pairs  $(i, j)$ , if  $C_{i,j} \geq \frac{\phi}{8} \|C\|_1$ , then  $C_{i,j}^\beta / \beta$  approximates  $C_{i,j}$  within a factor of  $1 + \frac{\epsilon}{4\phi}$ .

$\mathcal{E}_7$ : For all pairs  $(i, j)$ , if  $C_{i,j} < \frac{\phi}{8} \|C\|_1$ , then  $C_{i,j}^\beta / \beta < \frac{\phi}{4} \|C\|_1$ .

The correctness of Theorem 9 holds if both  $\mathcal{E}_6$  and  $\mathcal{E}_7$  hold. To see this, first consider those pairs  $(i, j)$  for which  $C_{i,j} < \frac{\phi}{8} \|C\|_1$ . By  $\mathcal{E}_7$  we have

$$C'_{i,j} \leq C_{i,j}^\beta \leq \beta \cdot \frac{\phi}{4} \|C\|_1 < \beta \cdot (\phi - \frac{\epsilon}{2}) \|C\|_1.$$

Thus pair  $(i, j)$  will not be output in Step 5 of Algorithm 4.

We next consider those pairs  $(i, j)$  with  $C_{i,j} \geq \frac{\phi}{8} \|C\|_1$ . By  $\mathcal{E}_6$  we have that  $C_{i,j}^\beta \in \left[ \frac{\beta C_{i,j}}{1 + \frac{\epsilon}{4\phi}}, \beta(1 + \frac{\epsilon}{4\phi}) C_{i,j} \right]$ . Now we consider two cases.

1. If  $C_{i,j} \geq \phi \|C\|_1$ , then

$$\begin{aligned} C'_{i,j} &\geq C_{i,j}^\beta - \frac{\epsilon\beta}{8} \|C\|_1 \\ &\geq \frac{\beta C_{i,j}}{1 + \frac{\epsilon}{4\phi}} - \frac{\epsilon\beta}{8} \|C\|_1 \\ &\geq \frac{\beta\phi \|C\|_1}{1 + \frac{\epsilon}{4\phi}} - \frac{\epsilon\beta}{8} \|C\|_1 \\ &\geq \beta \left( \phi - \frac{\epsilon}{2} \right) \|C\|_1. \end{aligned}$$

Thus pair  $(i, j)$  will be output.

2. If  $C_{i,j} \leq (\phi - \epsilon) \|C\|_1$ , then

$$\begin{aligned} C'_{i,j} \leq \beta C_{i,j}^\beta &\leq \beta \left( 1 + \frac{\epsilon}{4\phi} \right) C_{i,j} \\ &\leq \beta \left( 1 + \frac{\epsilon}{4\phi} \right) (\phi - \epsilon) \|C\|_1 \\ &\leq \beta \left( \phi - \frac{\epsilon}{2} \right) \|C\|_1. \end{aligned}$$

Thus pair  $(i, j)$  will not be output.

In the following we show that both  $\mathcal{E}_6$  and  $\mathcal{E}_7$  hold with probability  $1 - 1/n^4$ .

For  $\mathcal{E}_6$ , for a fixed pair  $(i, j)$ , by sampling we have

$$\mathbf{E}[C_{i,j}^\beta] = \beta \cdot C_{i,j} \geq \beta \cdot \frac{\phi}{8} \|C\|_1.$$

By a Chernoff bound we have

$$\begin{aligned} \Pr \left[ \left| C_{i,j}^\beta - \mathbf{E}[C_{i,j}^\beta] \right| \right] &\geq \frac{\epsilon}{4\phi} \cdot \mathbf{E}[C_{i,j}^\beta] \\ &\leq 2 \cdot e^{-\left(\frac{\epsilon}{4\phi}\right)^2 \beta \frac{\phi}{8} \|C\|_1 / 3} \\ &\leq 1/n^{10}. \end{aligned}$$

By a union bound over the at most  $n^2$   $(i, j)$  pairs, we have that with probability  $1 - 1/n^4$ ,  $C_{i,j}^\beta / \beta$  approximates  $C_{i,j}$  within a factor of  $(1 + \frac{\epsilon}{4\phi})$  for all pairs  $(i, j)$ .

For  $\mathcal{E}_7$ , consider a fixed pair  $(i, j)$ . If  $C_{i,j} < \frac{\phi}{8} \|C\|_1$ , then  $\mathbf{E}[C_{i,j}^\beta] < \beta \cdot \frac{\phi}{8} \|C\|_1$ . By a Chernoff bound we have that  $C_{i,j}^\beta \leq 2\beta \cdot \frac{\phi}{8} \|C\|_1$  with probability  $1 - 1/n^{10}$ . Thus the probability that  $\mathcal{E}_7$  holds is at least  $1 - 1/n^4$  by a union bound over all  $(i, j)$  pairs.

**Complexities.** Step 1 can be done using  $\tilde{O}(n)$  bits (Remark 2). By a Chernoff bound, it holds with probability  $1 - 1/n^{10}$  that  $\|C^\beta\|_1 \leq 2\beta \|C\|_1 = O\left(\frac{\phi}{\epsilon^2} \log n\right)$ . Consequently we have  $\|C^\beta\|_0 \leq \|C^\beta\|_1 = O\left(\frac{\phi}{\epsilon^2} \log n\right)$ . By Lemma 5 we have that with probability  $1 - 1/n^{10}$  Alice and Bob can recover all non-zero entries of  $C^\beta$  in Step 4 using  $\tilde{O}(\frac{\sqrt{\phi}}{\epsilon} n)$  bits of communication and 2 rounds. The communication in Step 5 is bounded by  $\tilde{O}(1/\epsilon)$ . We thus can bound the total communication by  $\tilde{O}(\frac{\sqrt{\phi}}{\epsilon} n)$ .

Finally, it is easy to see that the algorithm terminates in  $O(1)$  rounds.

The above analysis can be straightforwardly extended to  $\ell_p$ -norms for all constants  $p \in (0, 2]$  simply by replacing the sampling probability  $\beta$  by  $\beta^p$  at Line 2, and replacing  $\|C\|_1$  and matrix entries  $M_{i,j}$  by  $\|C\|_p^p$  and  $|M_{i,j}|^p$  respectively at Lines 1, 2 and 5. At Line 1 one can use Algorithm 1 to estimate  $\|C\|_p^p$  up to a factor of  $(1 + \frac{\epsilon}{4\phi})$ , which costs  $\tilde{O}(\frac{\phi}{\epsilon}n)$  bits of communication by Theorem 1, and is the bottleneck here.

**Corollary 1** *For two matrices  $A, B \in \mathbb{Z}^{n \times n}$ , there is an algorithm that computes the  $\ell_p$ - $(\phi, \epsilon)$ -heavy-hitters ( $0 < \epsilon \leq \phi \leq 1, p \in (0, 2]$ ) of  $AB$  with probability 0.9 using  $\tilde{O}(\frac{\phi}{\epsilon}n)$  bits of communication and  $O(1)$  rounds.*

## 5.2 Binary Matrices

In this section we show that we can do better for binary matrices by employing the idea we use for  $\ell_\infty$ -norm estimation. Again Alice holds  $A \in \{0, 1\}^{n \times n}$  and Bob holds  $B \in \{0, 1\}^{n \times n}$ , and let  $C = AB$ . Due to the similarity of the approach compared with the  $\ell_\infty$ -norm case (Section 4.1), we do not repeat some of the details.

We first assume that  $\|AB\|_p^p \geq 100\phi \log n / \epsilon^2$ , and will consider the other case later. The algorithm is as follows.

*Step 1:* Alice and Bob first estimate  $L_p = \|C\|_p$  within a factor of 2, denoted by  $L'_p$ .

*Step 2:* Alice samples each column of  $A$  with probability  $\beta = \min\left\{\frac{\alpha}{\phi^{1/p}L'_p}, 1\right\}$  for  $\alpha = (10^4 \log n)^{1/p}$ , obtaining  $A'$ . Let  $C' = A'B$ . Alice and Bob then exchange the indices of sets containing  $j$  for each surviving item  $j \in [n]$  as Step 7-12 in Algorithm 2, obtaining  $C_A$  and  $C_B$  for which  $C' = C_A + C_B$ .

*Step 3:* Alice and Bob try to verify for each non-zero entry in  $C_A$  or  $C_B$  whether it is indeed a heavy hitter. Let  $S_A, S_B$  consist of all the entries  $(i, j)$  in  $C_A, C_B$  for which  $(C_A)_{i,j}^p \geq \beta^p \phi (L'_p)^p / 20$  or  $(C_B)_{i,j}^p \geq \beta^p \phi (L'_p)^p / 20$ , respectively. Then for each entry  $(i, j) \in S_A \cup S_B$ , Alice and Bob try to estimate  $C_{i,j}$  within a  $(1 + \epsilon/(2\phi))$  factor by sampling  $\tilde{O}(1/(\epsilon/\phi)^2)$  coordinates of their corresponding row and column in  $A$  and  $B$ .

By Chernoff bounds, one has that after sampling we have with probability  $(1 - 1/n^{10})$  that (1) the number of sampled columns of  $A$  (or, the number of surviving universe items) is bounded by  $\tilde{O}(\beta n)$ , and (2)  $\|C'\|_1 = \tilde{O}(\beta L_1)$ .

The correctness proof is identical to that for the  $\ell_\infty$ -norm estimation algorithms in Section 4.1. We next turn to analyzing the communication cost.

The first step costs  $\tilde{O}(n)$  bits of communication by Theorem 1. For the second step, reusing the notation  $u_j, v_j$  for each universe item  $j$  in Algorithm 2, we analyze two cases:

- If  $\min\{u_j, v_j\} \leq \sqrt{L_1/n}$ , then since there are at most  $\tilde{O}(\beta n)$  surviving universe items, the total communication is upper bounded by

$$\tilde{O}(\beta n) \cdot \sqrt{\frac{L_1}{n}} = \tilde{O}\left(\frac{\sqrt{n}}{\phi^{1/p}} \cdot \frac{\sqrt{L_1}}{L_p}\right).$$

- If  $\min\{u_j, v_j\} > \sqrt{L_1/n}$ , then since  $\|C'\|_1 = \tilde{O}(\beta L_1)$ , the total communication is upper bounded by

$$\tilde{O}\left(\frac{\beta L_1}{\sqrt{L_1/n}}\right) = \tilde{O}\left(\frac{\sqrt{n}}{\phi^{1/p}} \cdot \frac{\sqrt{L_1}}{L_p}\right).$$

It is easy to see that the third step costs  $\tilde{O}((\phi/\epsilon)^2 \cdot 1/\phi) = \tilde{O}(\phi/\epsilon^2)$  bits of communication since there can be at most  $\tilde{O}(1/\phi)$  entries whose  $p$ -th powers are at least  $\beta^p \phi L_p^p / 20$ . Summing up, the total communication is bounded by  $\tilde{O}(Z)$  where

$$\begin{aligned} Z &= n + \frac{\sqrt{n}}{\phi^{1/p}} \cdot \frac{\sqrt{L_1}}{L_p} + \frac{\phi}{\epsilon^2} \\ &\leq n + \frac{\phi}{\epsilon^2} + \frac{n^{\frac{1}{2}}}{\phi^{1/p}} \cdot \frac{\sqrt{L_1}}{L_2/(n^{\frac{1}{2}-\frac{1}{p}})} \\ &\leq n + \frac{\phi}{\epsilon^2} + \frac{n^{1-\frac{1}{p}}}{\phi^{1/p}} \quad (\sqrt{L_1} \leq L_2) \\ &\leq 2\left(n + \frac{\phi}{\epsilon^2}\right) \cdot \left(\frac{\phi}{\epsilon^2} \geq \frac{1}{\phi}\right) \end{aligned}$$

In the case that  $\|AB\|_p^p < 100\phi \log n / \epsilon^2$ , we can just omit the subsampling in Step 2 of the algorithm. A similar analysis gives a communication cost of  $\tilde{O}(n + \frac{\sqrt{\phi n}}{\epsilon} + \frac{1}{\epsilon}) = \tilde{O}(n + \frac{\phi}{\epsilon^2})$ .

**Theorem 10** *There is an algorithm that computes the  $\ell_p$ - $(\phi, \epsilon)$ -heavy-hitters ( $0 < \epsilon \leq \phi \leq 1, p \in (0, 2]$ ) of  $AB$ , where  $A, B \in \{0, 1\}^{n \times n}$ , with probability 0.9 and using  $\tilde{O}(n + \frac{\phi}{\epsilon^2})$  bits of communication and  $O(1)$  rounds.*

## 6. CONCLUDING REMARKS

In this paper we studied a set of basic statistical estimation problems of matrix products in the distributed model, including the  $\ell_p$ -norms, distinct elements,  $\ell_0$ -sampling and heavy hitters. These problems have a number of applications in database joins.

We would like to mention again that our algorithms for square matrices can be straightforwardly modified to handle rectangular matrices where  $A \in \Sigma^{m \times n}$  ( $m \geq n$ ) and  $B \in \Sigma^{n \times m}$ . We briefly list here how our main upper bounds look like on rectangular matrices. All the algorithms remain the same (we of course have to change some occurrences of  $n$  to  $m$  in several places).

- The communication cost for  $(1 + \epsilon)$ -approximating  $\ell_p$  ( $p \in [0, 2]$ ) with  $\Sigma = \mathbb{Z}$  remains  $\tilde{O}(n/\epsilon)$ .
- The communication cost for  $(2 + \epsilon)$ -approximating  $\ell_\infty$  with  $\Sigma = \{0, 1\}$  becomes  $\tilde{O}(m^{1.5})$ , and that for  $\kappa$ -approximating  $\ell_\infty$  with  $\Sigma = \{0, 1\}$  becomes  $\tilde{O}(m^{1.5}/\kappa)$ .
- The communication cost for  $\ell_p$ - $(\phi, \epsilon)$ -heavy-hitters with  $\Sigma = \mathbb{Z}$  remains  $\tilde{O}(\frac{\phi}{\epsilon}n)$ , and that for  $\ell_p$ - $(\phi, \epsilon)$ -heavy-hitters with  $\Sigma = \{0, 1\}$  remains  $\tilde{O}(n + \frac{\phi}{\epsilon^2})$ .

## 7. REFERENCES

- [1] A. Abboud and A. Rubinfeld. Distributed PCP theorems for hardness of approximation in P. *CoRR*, abs/1706.06407, 2017.
- [2] F. N. Afrati and J. D. Ullman. Optimizing multiway joins in a map-reduce environment. *IEEE Trans. Knowl. Data Eng.*, 23(9):1282–1298, 2011.
- [3] T. D. Ahle, R. Pagh, I. P. Razenshteyn, and F. Silvestri. On the complexity of inner product similarity join. In *PODS*, pages 151–164, 2016.
- [4] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. *J. Comput. Syst. Sci.*, 58(1):137–147, 1999.
- [5] R. R. Amossen, A. Campagna, and R. Pagh. Better size estimation for sparse matrix products. *Algorithmica*, 69(3):741–757, 2014.
- [6] M. Balcan, Y. Liang, L. Song, D. P. Woodruff, and B. Xie. Communication efficient distributed kernel principal component analysis. In *KDD*, pages 725–734, 2016.
- [7] Z. Bar-Yossef, T. S. Jayram, R. Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. *J. Comput. Syst. Sci.*, 68(4):702–732, 2004.
- [8] Z. Bar-Yossef, T. S. Jayram, R. Kumar, D. Sivakumar, and L. Trevisan. Counting distinct elements in a data stream. In *RANDOM*, pages 1–10, 2002.
- [9] P. Beame, P. Koutris, and D. Suci. Communication steps for parallel query processing. In *PODS*, pages 273–284, 2013.
- [10] P. Beame, P. Koutris, and D. Suci. Skew in parallel query processing. In *PODS*, pages 212–223, 2014.
- [11] C. Boutsidis, D. P. Woodruff, and P. Zhong. Optimal principal component analysis in distributed and streaming models. In *STOC*, pages 236–249, 2016.
- [12] E. Cohen. Structure prediction and computation of sparse matrix products. *J. Comb. Optim.*, 2(4):307–332, 1998.
- [13] E. Cohen and D. D. Lewis. Approximating matrix multiplication for pattern recognition tasks. *J. Algorithms*, 30(2):211–252, 1999.
- [14] P. Flajolet and G. N. Martin. Probabilistic counting algorithms for data base applications. *J. Comput. Syst. Sci.*, 31(2):182–209, 1985.
- [15] G. Frahling, P. Indyk, and C. Sohler. Sampling in dynamic data streams and applications. *Int. J. Comput. Geometry Appl.*, 18(1/2):3–28, 2008.
- [16] D. V. Gucht, R. Williams, D. P. Woodruff, and Q. Zhang. The communication complexity of distributed set-joins with applications to matrix multiplication. In *PODS*, pages 199–212, 2015.
- [17] X. Hu, Y. Tao, and K. Yi. Output-optimal parallel algorithms for similarity joins. In *PODS*, pages 79–90, 2017.
- [18] Z. Huang, B. Radunovic, M. Vojnovic, and Q. Zhang. Communication complexity of approximate matching in distributed graphs. In *STACS*, pages 460–473, 2015.
- [19] P. Indyk. Stable distributions, pseudorandom generators, embeddings and data stream computation. In *FOCS*, pages 189–197, 2000.
- [20] H. Jowhari, M. Saglam, and G. Tardos. Tight bounds for  $\ell_p$  samplers, finding duplicates in streams, and related problems. In *PODS*, pages 49–58, 2011.
- [21] D. M. Kane, J. Nelson, and D. P. Woodruff. An optimal algorithm for the distinct elements problem. In *PODS*, pages 41–52, 2010.
- [22] R. Kannan, S. Vempala, and D. P. Woodruff. Principal component analysis and higher correlations for distributed data. In *COLT*, pages 1040–1057, 2014.
- [23] B. Ketsman and D. Suci. A worst-case optimal multi-round algorithm for parallel computation of conjunctive queries. In *PODS*, pages 417–428, 2017.
- [24] D. Kossmann. The state of the art in distributed query processing. *ACM Comput. Surv.*, 32(4):422–469, 2000.
- [25] P. Koutris, P. Beame, and D. Suci. Worst-case optimal algorithms for parallel query processing. In *ICDT*, pages 8:1–8:18, 2016.
- [26] P. Koutris and D. Suci. Parallel evaluation of conjunctive queries. In *PODS*, pages 223–234, 2011.
- [27] Y. Liang, M. Balcan, V. Kanchanapally, and D. P. Woodruff. Improved distributed principal component analysis. In *NIPS*, pages 3113–3121, 2014.
- [28] A. McGregor. Graph stream algorithms: a survey. *SIGMOD Record*, 43(1):9–20, 2014.
- [29] P. Mishra and M. H. Eich. Join processing in relational databases. *ACM Comput. Surv.*, 24(1):63–113, 1992.
- [30] M. Monemizadeh and D. P. Woodruff. 1-pass relative-error  $\ell_p$ -sampling with applications. In *SODA*, pages 1143–1160, 2010.
- [31] S. Muthukrishnan. Data streams: Algorithms and applications. *Foundations and Trends in Theoretical Computer Science*, 1(2), 2005.
- [32] R. Pagh. Compressed matrix multiplication. *TOCT*, 5(3), 2013.
- [33] M. E. Saks and X. Sun. Space lower bounds for distance approximation in the data stream model. In *STOC*, pages 360–369, 2002.
- [34] D. P. Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science*, 10(1-2):1–157, 2014.
- [35] D. P. Woodruff and Q. Zhang. An optimal lower bound for distinct elements in the message passing model. In *SODA*, pages 718–733, 2014.
- [36] D. P. Woodruff and P. Zhong. Distributed low rank approximation of implicit functions of a matrix. In *ICDE*, pages 847–858, 2016.