Distributed Statistical Estimation of Matrix Products with Applications

David Woodruff
CMU

Qin Zhang
IUB

Simons Institute
October 18, 2018

The Distributed Computation Model

p-norms,
heavy-hitters, ...

$A \in \{0,1\}^{m \times n}$ $\qquad$ $B \in \{0,1\}^{n \times m}$

Alice and Bob want to compute some function on
$C = A \times B$

Goal: minimize communication and number of rounds

Statistics of Matrix Products: p-Norms

- Alice holds $A \in \{0,1\}^{m \times n}$, Bob holds $B \in \{0,1\}^{n \times m}$
- Let $C = A \cdot B$. Alice and Bob want to approximate $\|C\|_p = \left(\sum_{i,j \in [n]} |C_{i,j}|^p\right)^{1/p}$

Statistics of Matrix Products: p-Norms

- Alice holds $A \in \{0,1\}^{m \times n}$, Bob holds $B \in \{0,1\}^{n \times m}$
- Let $C = A \cdot B$. Alice and Bob want to approximate $\|C\|_p = \left(\sum_{i,j \in [n]} |C_{i,j}|^p\right)^{1/p}$
  - $p = 0$: number of non-zero entries of C
    $\Rightarrow$ size of **set-intersection join**
    i-th row of A as set A_i, j-th column of B as set B_j, compute $\#(i,j)$ s.t. $A_i \cap B_j \neq \emptyset$

Statistics of Matrix Products: p-Norms

- Alice holds $A \in \{0,1\}^{m \times n}$, Bob holds $B \in \{0,1\}^{n \times m}$
- Let $C = A \cdot B$. Alice and Bob want to approximate $\|C\|_p = \left(\sum_{i,j \in [n]} |C_{i,j}|^p\right)^{1/p}$
  - $p = 0$: number of non-zero entries of C
    $\Rightarrow$ size of **set-intersection join**
    i-th row of A as set A_i, j-th column of B as set B_j,
    compute $\#(i,j)$ s.t. $A_i \cap B_j \neq \emptyset$
  - $p = 1$: sum of entries of C
    $\Rightarrow$ size of corresponding **natural join**
    compute $\#(i,k,j)$ s.t. $k \in A_i \cap B_j$

Statistics of Matrix Products: p-Norms

- Alice holds $A \in \{0,1\}^{m \times n}$, Bob holds $B \in \{0,1\}^{n \times m}$
- Let $C = A \cdot B$. Alice and Bob want to approximate $\|C\|_p = \left(\sum_{i,j \in [n]} |C_{i,j}|^p\right)^{1/p}$

size estimation is the key to database query optimization

  - $p = 0$: number of non-zero entries of C
    $\Rightarrow$ size of **set-intersection join**
    i-th row of A as set A_i, j-th column of B as set B_j, compute $\#(i,j)$ s.t. $A_i \cap B_j \neq \emptyset$
  - $p = 1$: sum of entries of C
    $\Rightarrow$ size of corresponding **natural join**
    compute $\#(i,k,j)$ s.t. $k \in A_i \cap B_j$

Statistics of Matrix Products: p-Norms

- Alice holds $A \in \{0,1\}^{m \times n}$, Bob holds $B \in \{0,1\}^{n \times m}$
- Let $C = A \cdot B$. Alice and Bob want to approximate

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

  size estimation is the key to database query optimization

  - $p = 0$: number of non-zero entries of C

    $\Rightarrow$ size of **set-intersection join**

    i-th row of A as set A_i, j-th column of B as set B_j, compute $\#(i,j)$ s.t. $A_i \cap B_j \neq \emptyset$
  - $p = 1$: sum of entries of C

    $\Rightarrow$ size of corresponding **natural join**

    compute $\#(i,k,j)$ s.t. $k \in A_i \cap B_j$
  - $p = \infty$: maximum entry of C

    $\Rightarrow$ most "similar" (A_i, B_j) pair

Application of set-intersection join

Applicant	Skills
A_1	S_1, S_4, S_9, S_{13}
A_2	S_2, S_9, S_{10}
A_m	S_6, S_7, S_8, S_{15}

$$\bowtie$$

Skills	Opening
S_2, S_3, S_4	B_1
S_3, S_4, S_9, S_{11}	B_2
S_4, S_8	B_m

Find all candidate (Applicant, Opening) pairs

Statistics of Matrix Products: Heavy Hitters

- Alice holds $A \in \{0,1\}^{m \times n}$, Bob holds $B \in \{0,1\}^{n \times m}$
- Let $C = A \cdot B$, and let

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

Statistics of Matrix Products: Heavy Hitters

- Alice holds $A \in \{0,1\}^{m \times n}$, Bob holds $B \in \{0,1\}^{n \times m}$
- Let $C = A \cdot B$, and let

$$\mathrm{HH}^p_\phi(C) = \{(i,j) \mid C_{i,j} \ge \phi \|C\|_p\}$$

  - ℓ_p-(ϕ, ϵ)-heavy-hitter ($0 < \epsilon \le \phi \le 1$): output a set $S \subseteq \{(i,j) \mid i,j \in [m]\}$ such that

$$\mathrm{HH}^p_\phi(C) \subseteq S \subseteq \mathrm{HH}^p_{\phi-\epsilon}(C)$$

Pairs (A_i, B_j) that are similar $\Rightarrow$ **similarity join**

Our Main Results – ℓ_p ($p \in [0, 2]$)

For simplicity, assume $m = n$

- For any $p \in [0, 2]$, a 2-round $\tilde{O}(n/\epsilon)$-bit algorithm that approximates $\|AB\|_p$ within a $(1 + \epsilon)$ factor

Our Main Results – ℓ_p ($p \in [0, 2]$)

For simplicity, assume $m = n$

- For any $p \in [0, 2]$, a 2-round $\tilde{O}(n/\epsilon)$-bit algorithm that approximates $\|AB\|_p$ within a $(1 + \epsilon)$ factor
  - For $p = 0$, this improves the previous result $\tilde{O}(n/\epsilon^2)$ (Van Gucht et al., PODS'15)
  - Same paper gives a lower bound of $\Omega(n/\epsilon^{2/3})$.

Our Main Results – ℓ_p ($p \in [0, 2]$)

For simplicity, assume $m = n$

- For any $p \in [0, 2]$, a 2-round $\tilde{O}(n/\epsilon)$-bit algorithm that approximates $\|AB\|_p$ within a $(1 + \epsilon)$ factor
  - For $p = 0$, this improves the previous result $\tilde{O}(n/\epsilon^2)$ (Van Gucht et al., PODS'15)
  - Same paper gives a lower bound of $\Omega(n/\epsilon^{2/3})$.

    If we restrict the communication to be one-way, then we have a lower bound $\Omega(n/\epsilon^2)$.

Our Main Results – ℓ_∞

- $O(1)$-round algorithms that approximate $\|AB\|_\infty$
  - within a factor of $(2+\epsilon)$ use $\tilde{O}(n^{1.5}/\epsilon)$ bits
  - within a factor of κ use $\tilde{O}(n^{1.5}/\kappa)$ bits

Our Main Results – ℓ_∞

- $O(1)$-round algorithms that approximate $\|AB\|_\infty$
  - within a factor of $(2+\epsilon)$ use $\tilde{O}(n^{1.5}/\epsilon)$ bits
  - within a factor of κ use $\tilde{O}(n^{1.5}/\kappa)$ bits

- Any algorithm (regardless of the #rounds used) that approximates $\|AB\|_\infty$ within a factor of
  - within a factor of 2 needs $\Omega(n^2)$ bits
  - within a factor of $\kappa \geq 4$ needs $\Omega(n^{1.5}/\kappa)$ bits

Our Main Results – ℓ_∞

- $O(1)$-round algorithms that approximate $\|AB\|_\infty$
  - within a factor of $(2+\epsilon)$ use $\tilde{O}(n^{1.5}/\epsilon)$ bits
  - within a factor of κ use $\tilde{O}(n^{1.5}/\kappa)$ bits

- Any algorithm (regardless of the #rounds used) that approximates $\|AB\|_\infty$ within a factor of
  - within a factor of 2 needs $\Omega(n^2)$ bits
  - within a factor of $\kappa \geq 4$ needs $\Omega(n^{1.5}/\kappa)$ bits

- The above results hold for **binary** matrices A and B. For **general** matrices $A, B \in \Sigma^{n \times n}$, the bound is $\tilde{\Theta}(n^2/\kappa^2)$ bits ($O(1)$-round for UB, any round for LB)

Our Main Results – Heavy Hitters

- For **binary** matrices A and B, for any $p \in (0, 2]$, an $O(1)$-round $\tilde{O}(n + \frac{\phi}{\epsilon^2})$-bit algorithm that computes ℓ_p-(ϕ, ϵ)-heavy-hitters

Our Main Results – Heavy Hitters

- For **binary** matrices A and B, for any $p \in (0, 2]$, an $O(1)$-round $\tilde{O}(n + \frac{\phi}{\epsilon^2})$-bit algorithm that computes ℓ_p-(ϕ, ϵ)-heavy-hitters

- For **general** matrices A and B, for any $p \in (0, 2]$, we obtain $O(1)$-round $\tilde{O}(\frac{\sqrt{\phi}}{\epsilon} \cdot n)$ bits algorithms

Our Main Results – Heavy Hitters

- For **binary** matrices A and B, for any $p \in (0, 2]$, an $O(1)$-round $\tilde{O}(n + \frac{\phi}{\epsilon^2})$-bit algorithm that computes ℓ_p-(ϕ, ϵ)-heavy-hitters

- For **general** matrices A and B, for any $p \in (0, 2]$, we obtain $O(1)$-round $\tilde{O}(\frac{\sqrt{\phi}}{\epsilon} \cdot n)$ bits algorithms

All of our results above can be easily extended to rectangular matrices where $A \in \Sigma^{m \times n}$ and $B \in \Sigma^{n \times m}$

Previous Results

- As mentioned, Van Gucht et al. PODS'15 studied the estimation of $\|AB\|_0$

Previous Results

- As mentioned, Van Gucht et al. PODS'15 studied the estimation of $\|AB\|_0$

- A number of recent works look at distributed linear algebra problems (Balcan et al. KDD'16; Boutsidis et al. STOC'16; Woodruff&Zhong, ICDE'16; etc.)

  These works concern statistics estimation on $C = A + B$, compared with $C = A \cdot B$ studied in this paper

Previous Results

- As mentioned, Van Gucht et al. PODS'15 studied the estimation of $\|AB\|_0$

- A number of recent works look at distributed linear algebra problems (Balcan et al. KDD'16; Boutsidis et al. STOC'16; Woodruff&Zhong, ICDE'16; etc.)

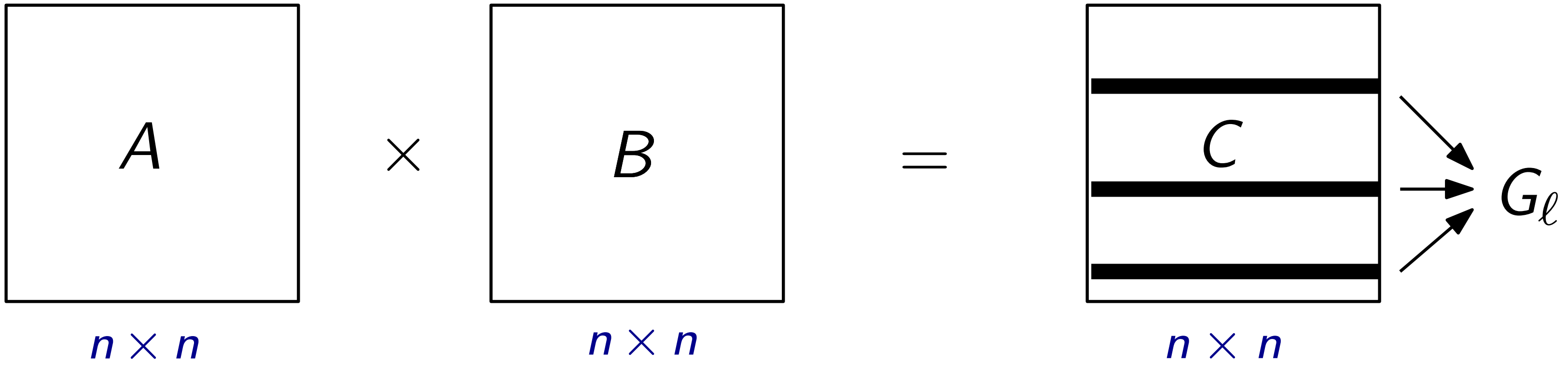
  These works concern statistics estimation on $C = A + B$, compared with $C = A \cdot B$ studied in this paper

- Similar problems have been studied in the RAM model (Cohen&Lewis, J. Algorithms, '99; Pagh TOCT'13; etc.)

$(1+\epsilon)$-approximate ℓ_0

$(1+\epsilon)$-approximate ℓ_0

- Alice holds $A \in \{0,1\}^{n \times n}$, Bob holds $B \in \{0,1\}^{n \times n}$
- Let $C = A \cdot B$. Goal: $(1+\epsilon)$-approximate $\|C\|_0$



High level idea:

1. First perform a rough estimation of the number of non-zero entries in the rows of C
2. Use the rough estimation to partition the rows of C to groups s.t. rows in the same group have similar #non-zero entries
3. Sample rows in each group of C with a probability propotional to the (estimated) average #non-zero entries of rows of the group
4. Use sampled rows to estimate #non-zero entries of C

$(1+\epsilon)$-approximate ℓ_0 (cont.)

- Alice holds $A \in \{0,1\}^{n\times n}$, Bob holds $B \in \{0,1\}^{n\times n}$
- Let $C = A \cdot B$. Goal: $(1+\epsilon)$-approximate $\|C\|_0$

Algorithm. Set $\beta = \sqrt{\epsilon}, \rho = \Theta(1/\epsilon)$

1. (Bob $\to$ Alice) Use VanGucht et al.'s algo to get a $(1+\beta)$-approx (w.r.t. nnz of rows) of C (denoted by $\tilde{C}$)
2. Alice partitions the n rows of $\tilde{C}$ to $L = O(\log n/\beta)$ groups $G_1, \ldots, G_L$, s.t. G_ℓ contains all rows $i \in [n]$ with $(1+\beta)^\ell \le \left\|\widetilde{C_{\ell,*}}\right\|_0 \le (1+\beta)^{\ell+1}$
3. For each group ℓ, Alice samples each row $i \in G_\ell$ w.pr. $p_\ell = \frac{\rho}{\|\tilde{C}\|_0} \cdot \frac{\sum_{i\in G_\ell} \|\widetilde{C_{i,*}}\|_0}{|G_\ell|}$. A' : matrix containing sampled rows of A. Alice sends A' to Bob
4. Bob computes $C' \leftarrow A'B$, outputs $\sum_{\ell\in[L]} \sum_{i\in G_\ell} \frac{1}{p_\ell} \left\|C'_{i,*}\right\|_0$

$(1+\epsilon)$-approximate ℓ_0 (cont.)

- Alice holds $A \in \{0,1\}^{n \times n}$, Bob holds $B \in \{0,1\}^{n \times n}$
- Let $C = A \cdot B$. Goal: $(1+\epsilon)$-approximate $\|C\|_0$

Correctness: expectation + variance estimation

Algorithm. Set $\beta = \sqrt{\epsilon}, \rho = \Theta(1/\epsilon)$

1. (Bob $\rightarrow$ Alice) Use VanGucht et al.'s algo to get a $(1+\beta)$-approx (w.r.t. nnz of rows) of C (denoted by $\tilde{C}$)

2. Alice partitions the n rows of $\tilde{C}$ to $L = O(\log n/\beta)$ groups $G_1, \ldots, G_L$, s.t. G_ℓ contains all rows $i \in [n]$ with $(1+\beta)^\ell \le \left\|\widetilde{C_{\ell,*}}\right\|_0 \le (1+\beta)^{\ell+1}$

3. For each group ℓ, Alice samples each row $i \in G_\ell$ w.pr. $p_\ell = \frac{\rho}{\|\tilde{C}\|_0} \cdot \frac{\sum_{i \in G_\ell} \|\widetilde{C_{i,*}}\|_0}{|G_\ell|}$. A' : matrix containing sampled rows of A. Alice sends A' to Bob

4. Bob computes $C' \leftarrow A'B$, outputs $\sum_{\ell \in [L]} \sum_{i \in G_\ell} \frac{1}{p_\ell} \left\|C'_{i,*}\right\|_0$

$(1+\epsilon)$-approximate ℓ_0 (cont.)

- Alice holds $A \in \{0,1\}^{n\times n}$, Bob holds $B \in \{0,1\}^{n\times n}$
- Let $C = A \cdot B$. Goal: $(1+\epsilon)$-approximate $\|C\|_0$

Algorithm. Set $\beta = \sqrt{\epsilon}, \rho = \Theta(1/\epsilon)$

Correctness: expectation + variance estimation

1. (Bob $\to$ Alice) Use VanGucht et al.'s algo to get a $(1+\beta)$-approx (w.r.t. nnz of rows) of C (denoted by $\tilde{C}$) $\tilde{O}(n/\beta^2) = \tilde{O}(n/\epsilon)$ bits
2. Alice partitions the n rows of $\tilde{C}$ to $L = O(\log n/\beta)$ groups $G_1, \ldots, G_L$, s.t. G_ℓ contains all rows $i \in [n]$ with $(1+\beta)^\ell \le \left\|\widetilde{C_{\ell,*}}\right\|_0 \le (1+\beta)^{\ell+1}$
3. For each group ℓ, Alice samples each row $i \in G_\ell$ w.pr. $p_\ell = \frac{\rho}{\|\tilde{C}\|_0} \cdot \frac{\sum_{i\in G_\ell} \|\widetilde{C_{i,*}}\|_0}{|G_\ell|}$. A' : matrix containing sampled rows of A. Alice sends A' to Bob $\tilde{O}(n \cdot \rho) = \tilde{O}(n/\epsilon)$ bits
4. Bob computes $C' \leftarrow A'B$, outputs $\sum_{\ell\in[L]} \sum_{i\in G_\ell} \frac{1}{p_\ell} \left\|C'_{i,*}\right\|_0$

A lower bound (Van Gucht, Williams, Woodruff, Z. '15)

Primitive problem 1: **Set Disjointness**

$\exists i, X_i \wedge Y_i = 1?$

$X \in \{0,1\}^n$ $Y \in \{0,1\}^n$

Lemma. [Razborov '90]
$\exists \mu, (X, Y) \sim \mu$, solving DISJ w.pr. 0.99 needs $\Omega(n)$ comm.

A lower bound (cont.)

Primitive problem 2: **Gap Hamming**

$X \in \{0,1\}^m$ $\qquad$ $Y \in \{0,1\}^m$

Let $W_i = X_i$ XOR Y_i. Goal: compute

$$\text{GAP-HAM}(X,Y) = \begin{cases} 0, & \text{if } \sum_{i \in [m]} W_i \leq \frac{m}{2} - \sqrt{m}, \\ 1, & \text{if } \sum_{i \in [m]} W_i \geq \frac{m}{2} + \sqrt{m}, \\ \text{don't care}, & \text{otherwise}, \end{cases}$$

Lemma. $\exists \nu, (X,Y) \sim \nu$, solving GAP-HAM w.pr. 0.99 needs to learn $\Omega(m)$ of W_i ($i \in [m]$) well $(I(W_i; \Pi) = \Omega(1))$

A lower bound (cont.)

- For each $i \in [m]$, choose $(A_i, B_i) \sim \mu$ where μ is a hard input distribution for set-disjointness.

  Define $SUM(A, B) = \sum_{i \in [m]} DISJ(A_i, B_i)$. W.h.p.

$$\ell_0(AB) = SUM(A, B) + m(m-1).$$

A lower bound (cont.)

- For each $i \in [m]$, choose $(A_i, B_i) \sim \mu$ where μ is a hard input distribution for set-disjointness.

  Define $SUM(A, B) = \sum_{i \in [m]} DISJ(A_i, B_i)$. W.h.p.

  $$\ell_0(AB) = SUM(A, B) + m(m-1).$$

- By composing GAP-HAM and DISJ, we can show that any rand. algo. that computes $SUM(A, B)$ w.pr. 0.99 up to an additive error $\sqrt{m}$ needs $\Omega(mn)$ comm.

A lower bound (cont.)

- For each $i \in [m]$, choose $(A_i, B_i) \sim \mu$ where μ is a hard input distribution for set-disjointness.

  Define $SUM(A, B) = \sum_{i \in [m]} DISJ(A_i, B_i)$. W.h.p.

  $$\ell_0(AB) = SUM(A, B) + m(m - 1).$$

- By composing GAP-HAM and DISJ, we can show that any rand. algo. that computes $SUM(A, B)$ w.pr. 0.99 up to an additive error $\sqrt{m}$ needs $\Omega(mn)$ comm.

- Set $m = 1/\epsilon^{2/3}$ to make error $\sqrt{m} = \epsilon \cdot \ell_0(AB)$, getting an LB $\Omega(n/\epsilon^{2/3})$.

The open problem

The main problem left open by our work:

$O(n/\epsilon)$ UB vs. $\Omega(n/\epsilon^{2/3})$ LB

What is the right complexity?

The open problem

The main problem left open by our work:

$O(n/\epsilon)$ UB vs. $\Omega(n/\epsilon^{2/3})$ LB

What is the right complexity?

The difficulty: cannot set $m > 1/\epsilon^{2/3}$, since under the distribution μ we choose, w.h.p. each A_i will intersect each B_j ($j \neq i$), and the term $m(m-1)$ will "dominate" $\ell_0(AB)$. From another perspective,

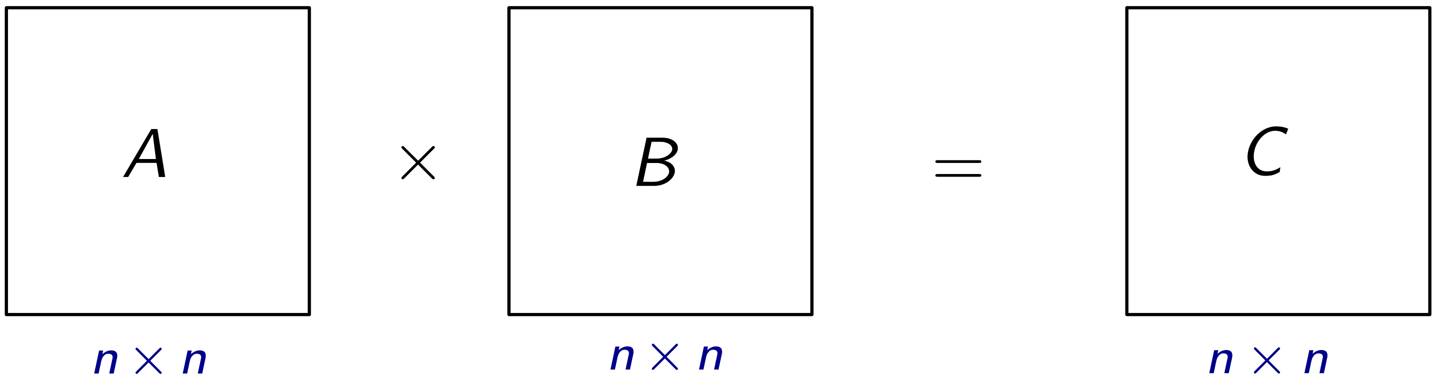
the primitive problems "overlap".

Need new techniques?

$(2+\epsilon)$-approximate ℓ_∞

$(2+\epsilon)$-approximate ℓ_∞

- Alice holds $A \in \{0,1\}^{n \times n}$, Bob holds $B \in \{0,1\}^{n \times n}$
- Let $C = A \cdot B$. Goal: $(2+\epsilon)$-approximate $\|C\|_\infty$



$A^0 = A$

$A^1 =$ subsample each entry of A w.pr. $\frac{1}{1+\epsilon}$

$A^2 =$ subsample each entry of A w.pr. $\frac{1}{(1+\epsilon)^2}$

$\cdots$

$C^0 = A^0 \times B$

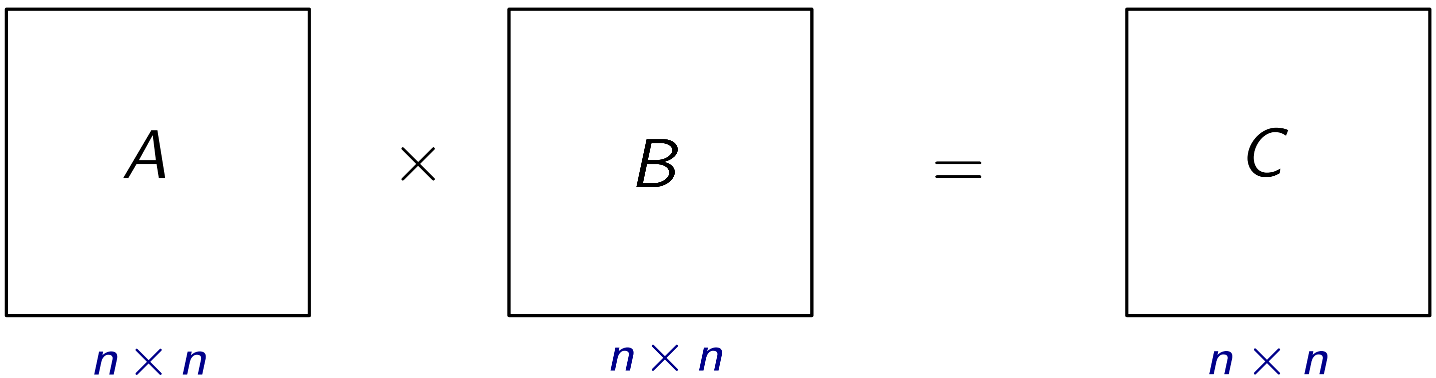
$C^1 = A^1 \times B$

$C^2 = A^2 \times B$

$\cdots$

$(2+\epsilon)$-approximate ℓ_∞

- Alice holds $A \in \{0,1\}^{n \times n}$, Bob holds $B \in \{0,1\}^{n \times n}$
- Let $C = A \cdot B$. Goal: $(2+\epsilon)$-approximate $\|C\|_\infty$



$A^0 = A$
$A^1 =$ subsample each entry of A w.pr. $\frac{1}{1+\epsilon}$
$A^2 =$ subsample each entry of A w.pr. $\frac{1}{(1+\epsilon)^2}$
...

$C^0 = A^0 \times B$
$C^1 = A^1 \times B$
$C^2 = A^2 \times B$
...

The idea: subsample C (via subsampling A) to a level ℓ s.t.
(1) ℓ is as large as possible, or, $\ell_1(C^\ell)$ is as small as possible
(2) $\ell_\infty(C^\ell) \cdot (1+\epsilon)^\ell$ still approximates $\ell_\infty(C)$ well.

$(2+\epsilon)$-approximate ℓ_∞ (cont.)

- Alice holds $A \in \{0,1\}^{n \times n}$, Bob holds $B \in \{0,1\}^{n \times n}$
- Let $C = A \cdot B$. Goal: $(2+\epsilon)$-approximate $\|C\|_\infty$

Algorithm Set $L = O(\log n/\epsilon), \gamma = \Theta(\log n/\epsilon^2)$

1. For $\ell = 0, 1, \ldots, L$, let $C^\ell \leftarrow A^\ell B$
   $A^\ell \Leftarrow$ sample each '1' in A w.pr. $p_\ell = \frac{1}{(1+\epsilon)^\ell}$.
2. Let ℓ^* be the smallest ℓ for which $\left\|C^\ell\right\|_1 \leq \gamma n^2$.
3. For each $j \in [n]$
   (a) u_j: #'1's in j-th column of A^{ℓ^*};
   v_j: #'1's in j-th row of B
   (b) If $u_j \leq v_j$, then Alice sends j-th column of A^{ℓ^*} to Bob;
   otherwise Bob sends j-th row of B to Alice
4. Alice and Bob use received information to compute matrices C_A and C_B respectively, s.t. $C_A + C_B = C^{\ell^*}$
5. Output $\max\left\{\frac{\|C_A\|_\infty}{p_{\ell^*}}, \frac{\|C_B\|_\infty}{p_{\ell^*}}\right\}$

$(2+\epsilon)$-approximate ℓ_∞ (cont.)

- Correctness

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

Simple Fact: If $C_A + C_B = C^{\ell^*}$, then

$$\frac{\left\| C^{\ell^*} \right\|_\infty}{2} \leq \max\left\{\|C_A\|_\infty, \|C_B\|_\infty\right\} \leq \left\| C^{\ell^*} \right\|_\infty$$

Put together, $\max\left\{\frac{\|C_A\|_\infty}{p_{\ell *}}, \frac{\|C_B\|_\infty}{p_{\ell *}}\right\}$ approximates $\|C\|_\infty$ within a factor of $2+\epsilon$.

$(2+\epsilon)$-approximate ℓ_∞ (cont.)

- Communication cost

  Bottleneck:

  For each $j \in [n]$

  (a) u_j: #'1's in j-th column of A^{ℓ^*};
  v_j: #'1's in j-th row of B

  (b) If $u_j \leq v_j$, then Alice sends j-th column of A^{ℓ^*} to Bob;
  otherwise Bob sends j-th row of B to Alice

$(2+\epsilon)$-approximate ℓ_∞ (cont.)

- Communication cost

  Bottleneck:

  For each $j \in [n]$

  (a) u_j: #'1's in j-th column of A^{ℓ^*};
  v_j: #'1's in j-th row of B

  (b) If $u_j \le v_j$, then Alice sends j-th column of A^{ℓ^*} to Bob; otherwise Bob sends j-th row of B to Alice

  For each $j \in [n]$, we analyze two cases:

  - If $u_j, v_j > \sqrt{n}/\epsilon$, # such j is bounded by $\left\|C^{\ell^*}\right\|_1 / (\sqrt{n}/\epsilon)^2$

    The comm. cost can be bounded by $\tilde{O}\left(\frac{n^{1.5}}{\epsilon}\right)$.

  - If $\min\{u_j, v_j\} \le \sqrt{n}/\epsilon$, the communication is bounded by

$$\sum_{j:\min\{u_j,v_j\}\le\frac{\sqrt{n}}{\epsilon}} \min\{u_j, v_j\} \le n \times \frac{\sqrt{n}}{\epsilon} \le \frac{n^{1.5}}{\epsilon}.$$

A Lower Bound for ℓ_∞ (cont.)

A reduction from set-disjointness to $\ell_\infty(AB)$

1. Alice partitions $x \in \{0,1\}^{n^2/4}$ to $n/2$ chunks of size $n/2$ each, and uses them as rows to construct $A' \in \{0,1\}^{\frac{n}{2} \times \frac{n}{2}}$. Further, let

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

2. Similarly, Bob uses $y \in \{0,1\}^{n^2/4}$ to construct $B' \in \{0,1\}^{\frac{n}{2} \times \frac{n}{2}}$, and futher let

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

3. We have $\|A \cdot B\|_\infty = \|A' + B'\|_\infty$, which is 2 if $DISJ(x, y) = 1$, and 1 otherwise.

Concluding Remarks

Main results:

- $(1+\epsilon)$-approximating ℓ_p ($p \in [0,2]$) with $\Sigma = \mathbb{Z}$ using $\tilde{O}(n/\epsilon)$ comm. and 2 rounds.

- $(2+\epsilon)$-approximating ℓ_∞ with $\Sigma = \{0,1\}$ using $\tilde{O}(n^{1.5}/\epsilon)$ comm. and 4 rounds.

- ℓ_p-(ϕ,ϵ)-heavy-hitters with $\Sigma = \mathbb{Z}$ using $\tilde{O}(\frac{\sqrt{\phi}}{\epsilon} n)$ comm. and $O(1)$ rounds;
  that with $\Sigma = \{0,1\}$ using $\tilde{O}(n + \frac{\phi}{\epsilon^2})$ comm. and $O(1)$ rounds.

Concluding Remarks

Main results:

- $(1+\epsilon)$-approximating ℓ_p ($p \in [0,2]$) with $\Sigma = \mathbb{Z}$ using $\tilde{O}(n/\epsilon)$ comm. and 2 rounds.

  Open: close the gap between this UB and the $\Omega(n/\epsilon^{2/3})$ LB.

- $(2+\epsilon)$-approximating ℓ_∞ with $\Sigma = \{0,1\}$ using $\tilde{O}(n^{1.5}/\epsilon)$ comm. and 4 rounds.

  Open: better #rounds?

- ℓ_p-(ϕ,ϵ)-heavy-hitters with $\Sigma = \mathbb{Z}$ using $\tilde{O}(\frac{\sqrt{\phi}}{\epsilon} n)$ comm. and $O(1)$ rounds;
  that with $\Sigma = \{0,1\}$ using $\tilde{O}(n + \frac{\phi}{\epsilon^2})$ comm. and $O(1)$ rounds.

  Open: tight LBs?

Thank you!
Questions?

$\|C\|_1$ corresponds to natural join

U	V
1	2
1	4
2	1
2	2
2	3
⋯	

$\bowtie$

V	W
1	1
2	1
4	1
1	2
3	2
⋯	

$$\underbrace{\overset{V}{U\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ & \vdots & & \end{bmatrix}}}_{A} \times \underbrace{\overset{W}{V\begin{bmatrix} 1 & 1 & \cdots \\ 1 & 0 & \\ 0 & 1 & \\ 1 & 0 & \end{bmatrix}}}_{B}$$

$\|C\|_1$ corresponds to natural join

U	V
1	2
1	4
2	1
2	2
2	3
⋯	

$\bowtie$

V	W
1	1
2	1
4	1
1	2
3	2
⋯	

$\Rightarrow \quad (2, 1, 2)$