The Input/Output Complexity of Sparse Matrix Multiplication

Rasmus Pagh, Morten Stöckel

IT University of Copenhagen

September 9 2014

Overview

- Let A and C be matrices over a semiring $\mathbb{R}$ with N nonzero entries in total.
- The problem: Compute matrix product $[AC]_{i,j} = \sum_k A_{i,k} C_{k,j}$ with Z nonzero entries.
- Central result: Can be done in (for most of parameter space) optimal $\tilde{O}\left(\frac{N\sqrt{Z}}{B\sqrt{M}}\right)$ I/Os.

Matrix multiplication, basics

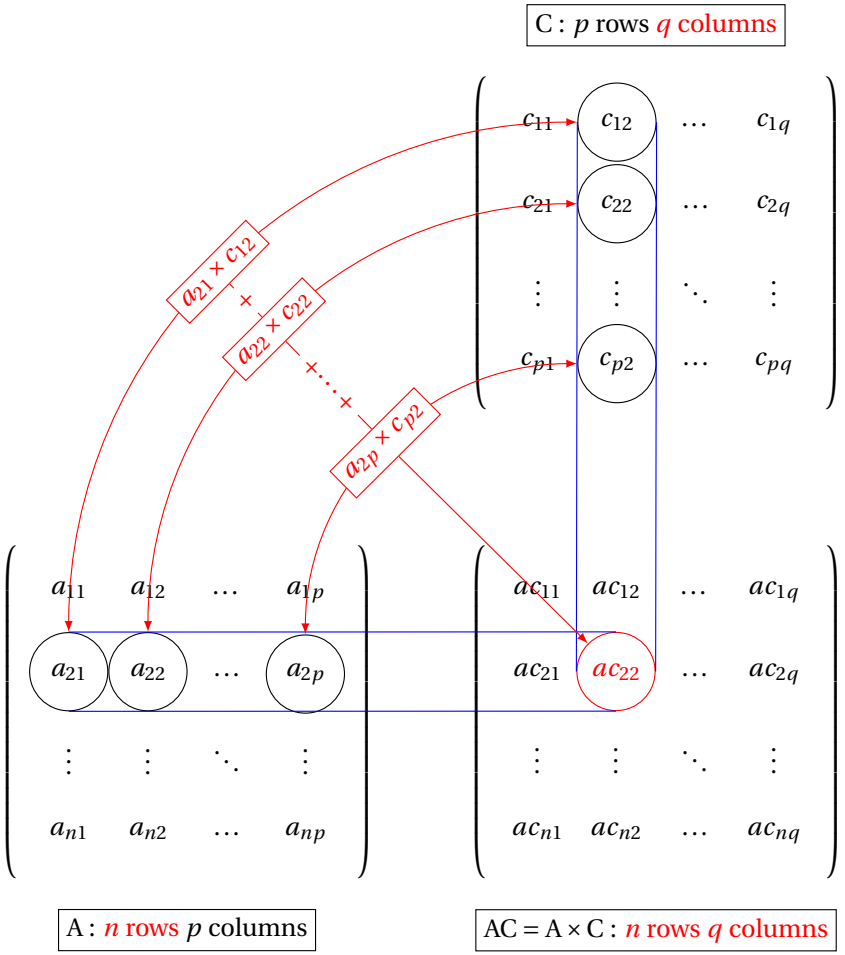
$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1p} \\ a_{21} & a_{22} & \dots & a_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{np} \end{pmatrix} \times \begin{pmatrix} c_{11} & c_{12} & \dots & c_{1q} \\ c_{21} & c_{22} & \dots & c_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ c_{p1} & c_{p2} & \dots & c_{pq} \end{pmatrix} = \begin{pmatrix} ac_{11} & ac_{12} & \dots & ac_{1q} \\ ac_{21} & ac_{22} & \dots & ac_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ ac_{n1} & ac_{n2} & \dots & ac_{nq} \end{pmatrix}$$

A : n rows p columns

C : p rows q columns

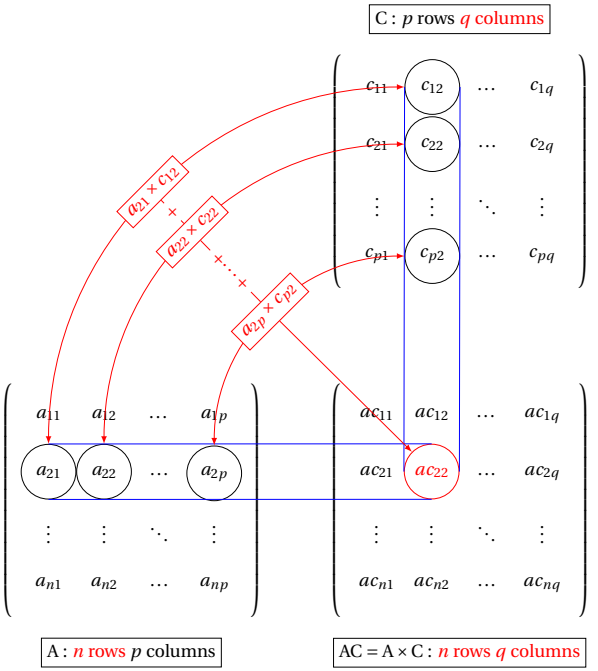
AC = A × C : n rows q columns

Matrix multiplication, basics



Cancellation of elementary products

We say that we have *cancellation* when two or more summands of $[AC]_{i,j} = \sum_k A_{i,k}C_{k,j}$ are nonzero but the sum is zero, e.g. $-2 * 3 + 1 * 6 + 0 * 4$.
Our algorithm handles such cases.



Motivation

Some applications:

- Computing determinants and inverses of matrices.
- Bioinformatics.
- Graphs: counting cycles, computing matchings.

The semiring I/O model, 1

- A word is big enough to hold a matrix element plus its coordinates.
- Internal memory that holds M words and disk of infinite size.
- One I/O: Transfer B words from disk to internal memory.
- Cost of an algorithm: Number of I/Os used.
- Operations allowed: Semiring operations, copy and equality check.

The semiring I/O model, 2

- We make no assumptions about cancellation.
- To produce output: must invoke `emit`$(.)$ on every nonzero output entry once.
- Matrices are of size $U \times U$.
- $\tilde{O}$ suppresses $\mathrm{polylog}$ factors in U and N.

Our results, 1

- Let A and C be $U \times U$ matrices over semiring $\mathbb{R}$ with N nonzero input and Z nonzero output entries. There exist algorithms 1 and 2 such that:
  1. emits the set of nonzero entries of AC with probability at least $1 - 1/U$, using $\tilde{O}\left(N\sqrt{Z}/(B\sqrt{M})\right)$ I/Os.
  2. emits the set of nonzero entries of AC, and uses $O\left(N^2/(MB)\right)$ I/Os.
- Previous best [Amossen & Pagh, 09]: $\tilde{O}\left(N\sqrt{Z}/(BM^{1/8})\right)$ I/Os (boolean matrices $\implies$ no cancellation).

Our results, 2

- Let A and C be $U \times U$ matrices over semiring $\mathbb{R}$ with N nonzero input and Z nonzero output entries. There exist algorithms 1 and 2 such that:
  1. emits the set of nonzero entries of AC with probability at least $1 - 1/U$, using $\tilde{O}\left(N\sqrt{Z}/(B\sqrt{M})\right)$ I/Os.
  2. emits the set of nonzero entries of AC, and uses $O\left(N^2/(MB)\right)$ I/Os.
- There exist matrices that require $\Omega\left(\min\left(\frac{N^2}{MB}, \frac{N\sqrt{Z}}{\sqrt{M}B}\right)\right)$ I/Os to compute all nonzero entries of AC.

Output size estimation

Size estimation tool: Given matrices A and C with N nonzero entries, compute ε-estimate of number of nonzeroes of each column of AC using $\tilde{O}(\varepsilon^{-3}N/B)$ I/Os.
Black boxed used [BBFJV,07]:

Fact

For dense $1 \times U$ vector y and sparse $U \times U$ matrix S we can compute yS in $O((\mathtt{nnz}(S)/B)\log_{M/B}(U/M)) = \tilde{O}((\mathtt{nnz}(S)/B)$ I/Os.

Distinct elements and matrix size

- Distinct elements: Given frequency vector x of size n where x_i denotes the number of times element i occurs, then $F_0 = \sum_i |x_i|^0$.
- Fundamental problem in streaming: Estimate F_0 without materializing x.
- Observation: The distinct elements of AC is $\texttt{nnz}(AC)$.

Linear distinct elements sketch, 1

Simple linear distinct elements sketch [Indyk slides, McGregor book].
Answer question: For a picked T, is $F_0 > (1+\varepsilon)T$?

1. Select sets $S_1, \ldots, S_k$ of coordinates s.t. $\mathbf{Pr}[i \in S_j] = 1/T$.
2. For each S_i: $s_j(x) = \sum_{i \in S_j} x_i$.
3. Answer yes if at most k/e of s_j are zero.

Analysis: For one set S_j we have $\mathbf{Pr}[s_j = 0] = (1-1/T)^{F_0} \approx e^{-F_0/T}$.
If $F_0 > (1+\varepsilon)T$ then $\mathbf{Pr}[s_j = 0] < 1/e - \varepsilon/3$. Repeat for $k = O(\varepsilon^{-2} \log \delta^{-1})$ independent sets to get probability $1-\delta$.

Linear distinct elements sketch, 2

- Can answer if $F_0 > (1+\varepsilon)T$ for some T.
- Repeat for $T = 1, (1+\varepsilon), (1+\varepsilon)^2, \ldots, n$, i.e. $O(\varepsilon^{-1} \log n)$ values.
- Total space: $O(\varepsilon^{-3} \log n \log \delta^{-1})$.
- Note: Random sets S_j form $k \times n$ projection matrix F and we maintain Fx.
- Linearity: $F(x + e_i) = Fx + Fe_i$

Output estimation

F is $\varepsilon^{-2} \log \delta^{-1} \times U$.
A and C are $U \times U$.
To get size estimate we must compute:

$$F \times A \times C$$

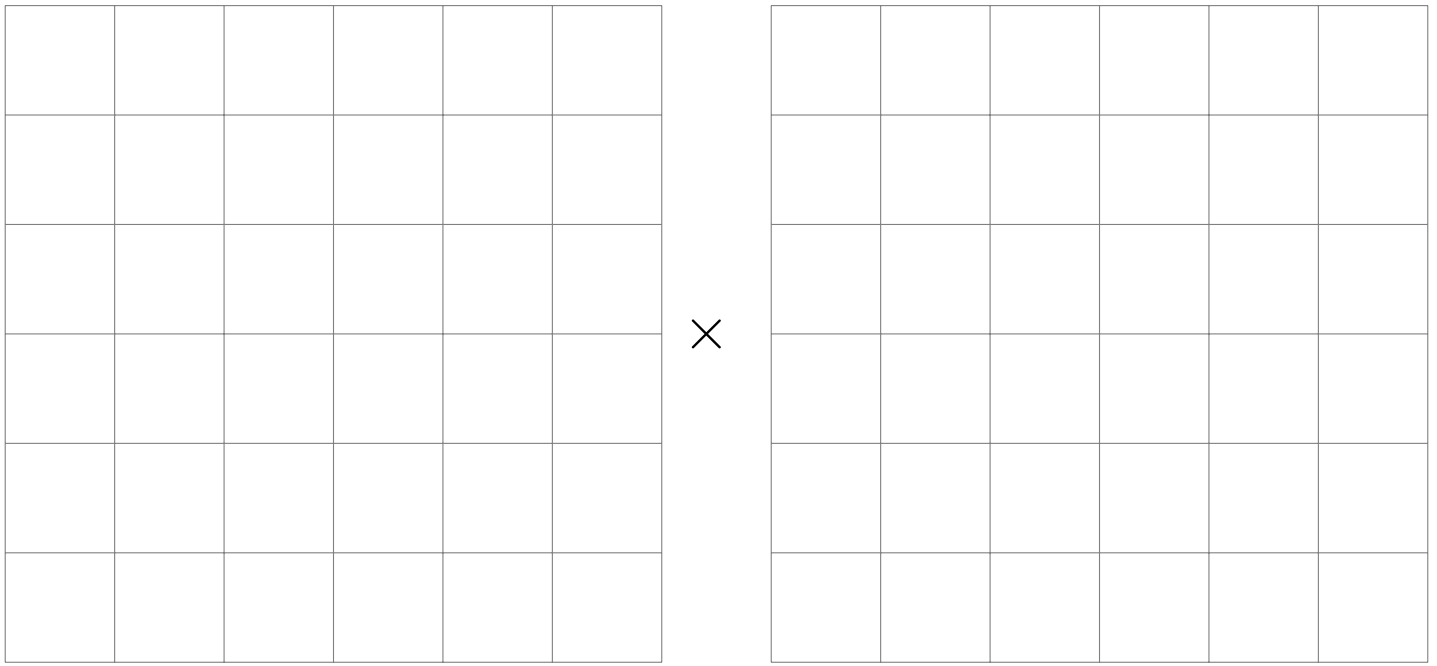
Output estimation

F is $\varepsilon^{-2} \log \delta^{-1} \times U$.
A and C are $U \times U$.
To get size estimate we must compute:

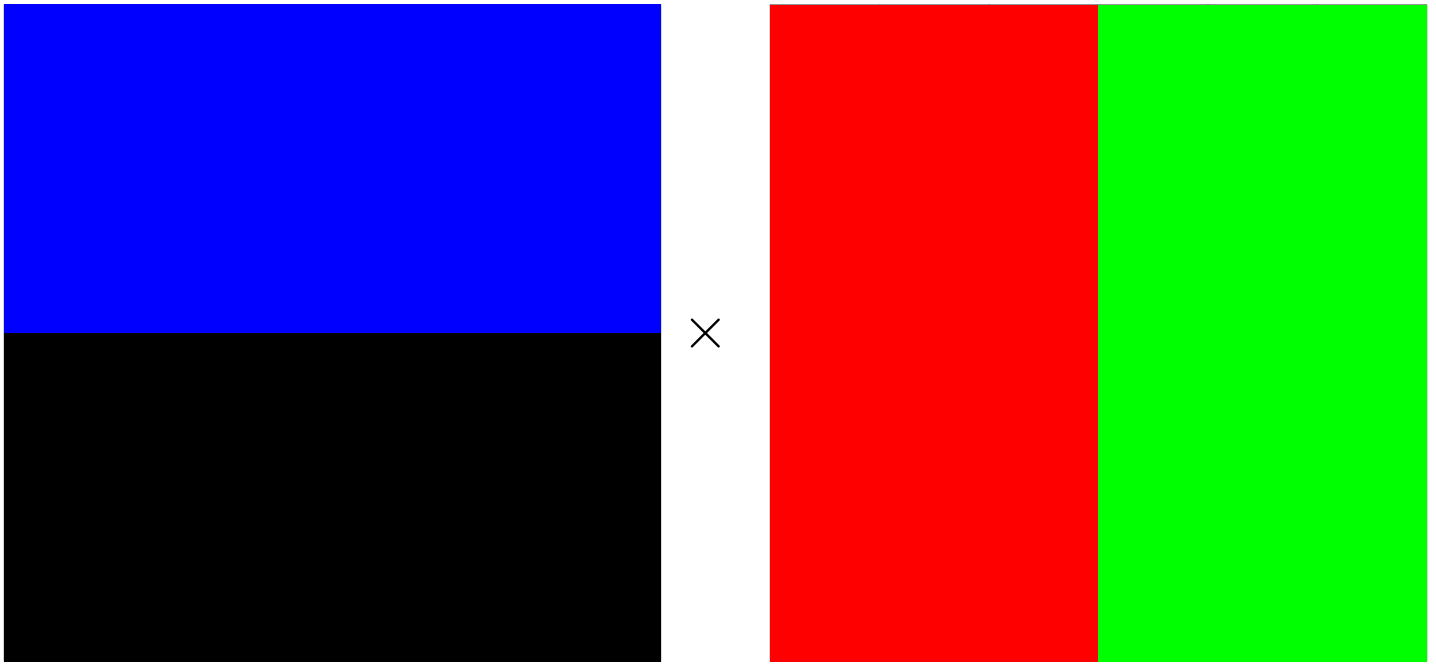
$$(F \times A) \times C$$

Due to associativity: Pick cheap order.
Analysis: $\varepsilon^{-2} \log \delta^{-1}$ invocations of dense vector sparse matrix black box: $\tilde{O}(\varepsilon^{-3} N/B)$ I/Os.
Note: Works with cancellation, contrary to previous size estimation.

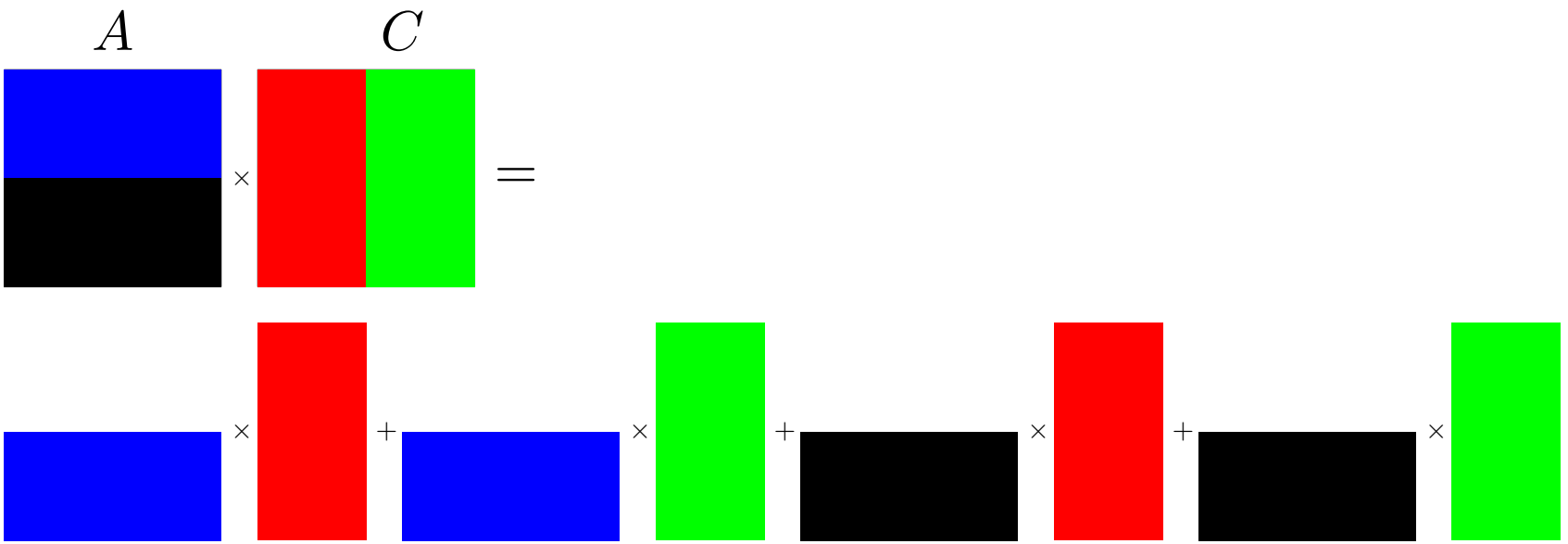
Matrix mult partitioning, 1



Matrix mult partitioning, 1

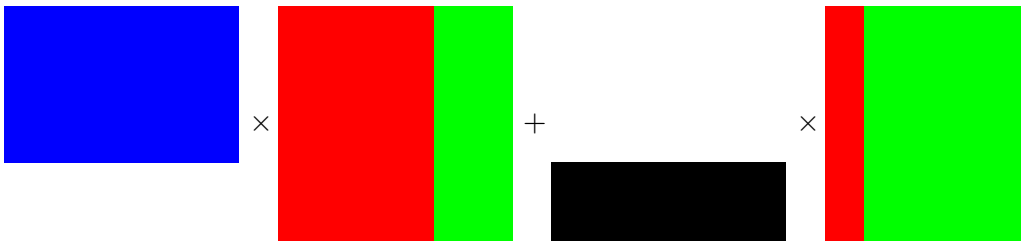


Matrix mult partitioning, 2



Partitioning the matrices

- What we want: Split matrices into disjoint colored groups s.t. every color combination has at most M nonzero output entries.
- Problem: Can't be done.
- Instead: Color rows of A using c colors. For each c groups of rows, do an independent coloring with c colors of columns of C.



Partitioning the matrices, 2

Overview of how to partition matrices A and C:

1. Pick number of colors $c = \sqrt{\frac{\mathtt{nnz}(AC)\log U}{M}} + O(1)$
2. Recurse: Split A into A_1 and A_2 where it holds: $A_1 C \approx \mathtt{nnz}(AC)/2$ and $A_2 C \approx \mathtt{nnz}(AC)$.
3. After $\log c + O(1)$ recursive levels we have $O(c)$ disjoint colored groups of rows of A.
4. For each of those groups: Repeat procedure for columns of C.
5. The key point: $O(c^2)$ problems of size $\mathtt{nnz}(AC)/c^2 = O(M/\log U)$.

Getting the correct subproblem size

Say we can do splits of A into A_1, A_2 s.t.

1. $\mathtt{nnz}(A_1C) \in \left[(1-\log^{-1} U)\,\mathtt{nnz}(AC)/2; (1+\log^{-1} U)\,\mathtt{nnz}(AC)/2\right]$.
2. $\mathtt{nnz}(A_2C) \in \left[(1-\log^{-1} U)\,\mathtt{nnz}(AC)/2; (1+\log^{-1} U)\,\mathtt{nnz}(AC)/2\right]$.

Assume biggest possible positive error: after q recursions have problem output size $\mathtt{nnz}(AC)(1/2+1/(2\log U))^q$. Then after $\log c^2 + O(1)$ recursions:

$$\begin{aligned}\mathtt{nnz}(AC)\left(\frac{1}{2}+\frac{1}{2\log U}\right)^{\log c^2} &\leq \mathtt{nnz}(AC)2^{-\log c^2} e^{\frac{\log c^2}{\log U}} \\ &\leq \mathtt{nnz}(AC)O(1)/c^2 = O(M/\log U)\end{aligned}$$

How to compute the split

How to do relative error $1/\log U$ splits: Use size estimation tool:
For any set r of rows we have access to $\hat{z}_i$'s s.t.

$$(1-\log^{-1} U)\,\mathtt{nnz}\left(\sum_{i\in r}[AC]_{i*}\right) \leq \sum_{i\in r}\hat{z}_i \leq (1+\log^{-1} U)\,\mathtt{nnz}\left(\sum_{i\in r}[AC]_{i*}\right).$$

Splitting A into A_1 and A_2:

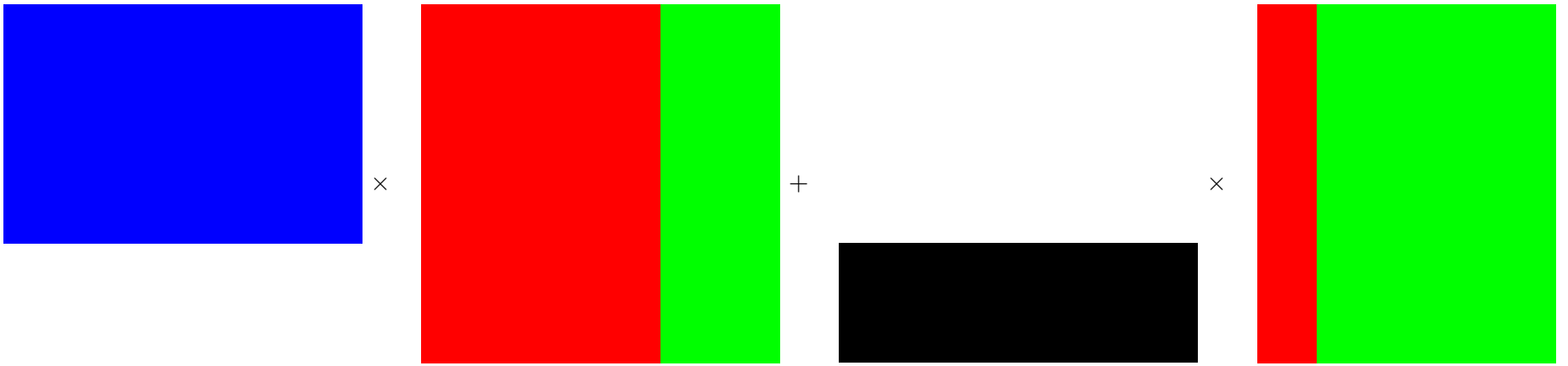
1. Let $\hat{Z} = \sum_i \hat{z}_i$.
2. Add rows from A to A_1 until $\sum_{i\in A_1}\hat{z}_i \geq \hat{Z}/2$.
3. The row that y overflows A_1: Compute $y \times C$ directly.
4. Add remaining rows to A_2

I/O cost of splitting

I/O cost:

- Initial size est: $\tilde{O}(N/B)$.
- Partition A: c dense-vector-sparse-matrix: $\tilde{O}(cN/B)$.
- For each of the c A partitions: Size est of total $\tilde{O}(N/B)$ and total c DVSM of $\tilde{O}(cN/B)$.
- Total: $\tilde{O}(cN/B) = \tilde{O}\left(\frac{N\sqrt{\mathbf{nnz}(AC)}}{B\sqrt{M}}\right)$ since $c = \sqrt{\frac{\mathbf{nnz}(AC)\log U}{M}}$.

Are we done?



Status

- Where we are: have $c^2 = \frac{\mathtt{nnz}(AC)\log U}{M}$ subproblems with output $\leq M/\log U$.
- Central cancellation difficulty: Intermediate results can be much larger than M.
- Our I/O aim: $\tilde{O}(cN/B)$, hence we can't pay for those cancelling inner products.
- Solution: Compute a particular polynomial and allow polynomially small error probability.

Compressed matrix mult sketches, 1

Idea from [Pagh,12]. Computes each outer product as a polynomial of degree $\approx M/\log U$.

- Let $r = \alpha 4M/\log U$ for small constant α.
- Let $h_t, h'_t : [U] \mapsto [r]$ be random hash functions.
- The polynomial:

$$p_t(x) = \sum_{k=1}^{U} \left(\sum_{i=1}^{U} A_{i,k} x^{h_t(i)} \right) \left(\sum_{j=1}^{U} C_{k,j} x^{h'_t(j)} \right)$$

- Can be computed in space $4r$.

Compressed matrix mult sketches, 2

$$p_t(x) = \sum_{k=1}^{U} \left(\sum_{i=1}^{U} A_{i,k} x^{h_t(i)} \right) \left(\sum_{j=1}^{U} C_{k,j} x^{h'_t(j)} \right)$$

- $\sum_{i=1}^{U} A_{i,k} x^{h_t(i)}$ can be computed in one pass over col k of A using space r
- Multiply and add to sum of products: additional $2r$ space.
- Central observation: Coefficient of $x^{h_t(i)+h'_t(j)}$ is $[AC]_{i,j}$ with constant probability.
- Left to do: Extract coordinates of coefficients and get high probability.
- Can be done in $O(\log U)$ repetitions [Pagh, 2012].

Algorithm summary

I/O cost of steps taken:

- Initial size est: $\tilde{O}(N/B)$.
- Partition into c^2 problems with output $M/\log U$: $\tilde{O}(cN/B)$.
- Compute and $\texttt{emit}(.)$ all subproblems: $\tilde{O}(cN/B)$.
- Total: $\tilde{O}(cN/B) = \tilde{O}\left(\frac{N\sqrt{\texttt{nnz}(AC)}}{B\sqrt{M}}\right)$ since $c = \sqrt{\frac{\texttt{nnz}(AC)\log U}{M}}$.

Concluding remarks

In the article:

- Size estimation: Supports cancellation and uses $\tilde{O}(\varepsilon^{-3}N/B)$ I/Os.
- Algorithm 1: $\tilde{O}\left(N\sqrt{Z}/(B\sqrt{M})\right)$ I/Os.
- Algorithm 2: $O\left(N^2/(MB)\right)$ I/Os.
- Lower bound: $\Omega\left(\min\left(\frac{N^2}{MB}, \frac{N\sqrt{Z}}{\sqrt{MB}}\right)\right)$ I/Os.

Open: Remove monte carlo (and log factors).

The Input/Output Complexity of Sparse Matrix Multiplication

Rasmus Pagh, Morten Stöckel

IT University of Copenhagen

September 9 2014