Matrix Product Verification

By Saleh Almousa

Outline

- Introduction
- Deterministic Algorithms
- Probabilistic Algorithms
- An Improvement using Linear Independency

Introduction

- Two matrices A and B of sizes n by n:
  - $A_{n\times n}\, B_{n\times n} = C_{n\times n}$

Deterministic Algorithms

- Naïvely

$$c_{i,j} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

- Runs in $O(n^3)$

Deterministic Algorithms

Divide and Conquer

- Divide the two matrices into n/2 x n/2 smaller matrices, i.e.

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, B = \begin{bmatrix} e & f \\ g & h \end{bmatrix}$$

- The Product of A and B:

$$C = \begin{bmatrix} r & s \\ t & u \end{bmatrix} = \begin{bmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{bmatrix}$$

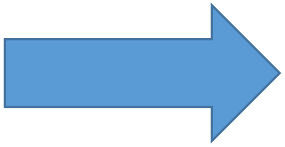
- Yet, the 8 multiplications can be done in 7 multiplications

Deterministic Algorithms

Divide and Conquer

$$C = \begin{bmatrix} r & s \\ t & u \end{bmatrix} = \begin{bmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{bmatrix}$$

$$p_1 = (f - h)\, a$$
$$p_2 = (a + b)\, h$$
$$p_3 = (c + d)\, e$$
$$p_4 = (g - e)\, d$$
$$p_5 = (a + d)\, (e + h)$$
$$p_6 = (b - d)\, (g - h)$$
$$p_7 = (a - c)\, (e + f)$$



$$r = p_5 + p_4 - p_2 + p_6$$
$$s = p_1 + p_2$$
$$t = p_3 + p_4$$
$$u = p_5 + p_1 - p_3 - p_7$$

Deterministic Algorithms
Divide and Conquer

- The overall complexity drops to

$$O(n^{\log 7}) \approx O(n^{2.81})$$

Matrix Product Verification

- Given A, B & C, Verify the following:

$$AB \stackrel{?}{=} C$$

Naïvely: O(n^3)

Divide and Conquer: O($n^{2.81}$)

Can we do better?

Probabilistic Algorithms
Freivalds

- Randomized
- Procedure
  1. Vector $r = [0,1]^n$ (randomly generated)
  2. Compute:
     1. Cr
     2. Br
     3. ABr
  3. If $ABr = Cr$
     1. Output “YES”
     2. Otherwise, output “NO”
- Runs in O(n^2)
- Requires n bits for the random vector. [[Sample Space]]

Probabilistic Algorithms
Freivalds

- E.g.

$A = \begin{bmatrix} 2 & 2 \\ 5 & 3 \end{bmatrix}$

$B = \begin{bmatrix} 1 & 3 \\ 0 & 9 \end{bmatrix}$

$C = \begin{bmatrix} 2 & 24 \\ 5 & 32 \end{bmatrix}$

Misleading

$r = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$r = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$r = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$	$r = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$
$Br = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$Br = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$Br = \begin{bmatrix} 3 \\ 9 \end{bmatrix}$	$Br = \begin{bmatrix} 4 \\ 9 \end{bmatrix}$
$ABr = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$ABr = \begin{bmatrix} 2 \\ 5 \end{bmatrix}$	$ABr = \begin{bmatrix} 24 \\ 42 \end{bmatrix}$	$ABr = \begin{bmatrix} 26 \\ 47 \end{bmatrix}$
$Cr = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$Cr = \begin{bmatrix} 2 \\ 5 \end{bmatrix}$	$Cr = \begin{bmatrix} 24 \\ 32 \end{bmatrix}$	$Cr = \begin{bmatrix} 26 \\ 37 \end{bmatrix}$
YES	YES	NO	NO

Probabilistic Algorithms
Freivalds

- Correctness ?!
  - If (AB = C) -> always correct
  - Otherwise,
    - If output "NO", it is **CORRECT**
    - If output "YES", it is $\frac{1}{2}$ **CORRECT**
  - Notice, if the algorithm runs for t constant times, then
    - If output "YES", it is $(1 - \frac{1}{2^t})$ **CORRECT**
    - For t=1,2,3, ..., 10, correctness is: %50, %75, %87.5, … , %99.9
    - Complexity is O($t\ n^2$)

Probabilistic Algorithms

Freivalds

- What if $r_1, r_2 \in [0,1]^n$ but they differ by one value at index i?
- E.g.

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \qquad r_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \qquad r_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 10 & 11 & 12 \\ 13 & 14 & 15 \\ 16 & 17 & 18 \end{bmatrix} \qquad ABr_1 = \begin{bmatrix} 174 \\ 417 \\ 660 \end{bmatrix} \qquad ABr_2 = \begin{bmatrix} 84 \\ 201 \\ 318 \end{bmatrix}$$

$$C = \begin{bmatrix} 84 & 90 & 96 \\ 201 & 215 & 231 \\ 318 & 342 & 366 \end{bmatrix} \qquad Cr_1 = \begin{bmatrix} 174 \\ 416 \\ 660 \end{bmatrix} \qquad Cr_2 = \begin{bmatrix} 84 \\ 201 \\ 318 \end{bmatrix}$$

NO **YES**

Probabilistic Algorithms

Freivalds

- If the error in column i, there exist 2^{n-1} vectors that will pass the test with r(i) = 0
- We need a smart way for choosing the r vector.
- The random number sample space is proportional to the algorithm correctness

Probabilistic Algorithms
Generalized

- Procedure
  1. Let X = {1, …, r}
  2. Let F = {f_1, f_2, …, f_k}
  3. Choose x in X uniformly at random
  4. For each f in F:
     1. v = f(x)
     2. If $ABv \neq Cv$, then return ”NO”
  5. Return “YES”

- In Freivalds
  - r = 2n
  - f = binary(x) of size n

Probabilistic Algorithms
Generalized

- Algorithm Parameters
  - F
  - r

- Theorem: ”For any choice of F and r, if the probability of error is strictly less than 1, then $kr \geq n$”
  - Where k is number of functions in F
  - And r, the size of random number space X

Improvement

- Column space and basis
  - Reduce the random space and increase the correctness

Column Space and Basis

- A matrix M of size axb has a column space C(M) = $\{\overrightarrow{m_1}, \overrightarrow{m_2}, \ldots, \overrightarrow{m_b}\}$

$$M = \begin{matrix} \overrightarrow{m_1} & \overrightarrow{m_2} & \overrightarrow{m_3} & \overrightarrow{m_4} & \overrightarrow{m_5} \end{matrix} \begin{bmatrix} 1 & 0 & -1 & 0 & 4 \\ 2 & 1 & 0 & 0 & 9 \\ -1 & 2 & 5 & 1 & -5 \\ 1 & -1 & -3 & -2 & 9 \end{bmatrix}, C(M) = \{\overrightarrow{m_1}, \quad \overrightarrow{m_2}, \quad \overrightarrow{m_3}, \overrightarrow{m_4}, \quad \overrightarrow{m_5}\}$$

Column Spaces and Basis

$$M = \begin{bmatrix} 1 & 0 & -1 & 0 & 4 \\ 2 & 1 & 0 & 0 & 9 \\ -1 & 2 & 5 & 1 & -5 \\ 1 & -1 & -3 & -2 & 9 \end{bmatrix} \quad \begin{matrix} echelon\ form \\ \leftrightarrow \end{matrix} \quad R = \begin{bmatrix} 1 & 0 & -1 & 0 & 4 \\ 0 & 1 & 2 & 0 & 1 \\ 0 & 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Basis: $\overrightarrow{m_1}, \quad \overrightarrow{m_2}, \quad \overrightarrow{m_4}$

Basis: $\overrightarrow{r_1}, \quad \overrightarrow{r_2}, \quad \overrightarrow{r_4}$

Given the basis, the other vectors could be produced:

$$\overrightarrow{r_3} = 2\overrightarrow{r_2} - \overrightarrow{r_1}$$

$$\overrightarrow{r_5} = 4\overrightarrow{r_1} + \overrightarrow{r_2} - 3\overrightarrow{r_4}$$

Improvement

- Basis of a matrix can be found using Echelon Form (no multiplication involved)
- Given the same problem: is $AB = C$?
  1. Randomly choose x in X = {indices of column space basis of C}
  2. Vector v = binary(x)
  3. If $ABv \neq Cv$ output “NO”
  4. Otherwise output “YES”

Improvement

- The random space can get reduced if most of the columns in C are Linearly dependent.

- Theorem: If $ABv = Cv$ for all v from X, then $AB = C$.
  - Where X is the set of indices of all column basis of C

References

1. Chinn, D. D., & Sinha, R. K. (1993). Bounds on sample space size for matrix product verification. *Information processing letters*, *48*(2), 87-91.
2. Kimbrel, T., & Sinha, R. K. (1993). A probabilistic algorithm for verifying matrix products using o (n2) time and log2n+ o (1) random bits. *Information Processing Letters*, *45*(2), 107-110.