QuantEcon Lunch Talk 35:

Discovering Faster Matrix Multiplication Algorithms with Human Intelligence

Shu Hu (The Australian National Univeristy and QuantEcon)

Motivations

Q1: What is "Human Intelligence"?

Q2: What is "Algorithm" and how to tell how "Fast" they are?

Q3: What is Matrix Multiplication and what Algorithms we have for computing it?

Q1: What is "Human Intelligence"?

Human Intelligence vs Artificial Intelligence

Q2: What is "Algorithm" and how to tell how "Fast" they are?

Computational Problem

Algorithm

Computational Complexity

Q3-1: What is Matrix Multiplication

Let $\mathbb{M}^{n\times n}(\mathcal{R})$ be a set of all $n \times n$ matrices over the field of real numbers.

If $A, B \in \mathbb{M}^{n\times n}(\mathcal{R})$

then $AB \in \mathbb{M}^{n\times n}(\mathcal{R})$, defined as

$$(AB)_{ij} = \sum_{k=1}^{n} A_{ik}B_{kj}$$

Q3-2 What matrix multiplication algorithms we have so far?

Naive Algorithm

Strassen Algorithm (1969)

And more

Q3-2 What matrix multiplication algorithms we have: Naive Algorithm

The pseudocode is

```
input A and B, both n by n matrices
initialize C to be an n by n matrix of all zeros``
for i from 1 to n:
    for j from 1 to n:
       for k from 1 to n:
            C[i][j] = C[i][j] + A[i][k]*B[k][j]
output C (as A*B)
```

Motivational Fact

Multiplication is inherently more costly than addition in terms of computational complexity

Q3-2 What matrix multiplication algorithms we have: Strassen Algorithm (1969)

Assumption: $A, B, C \in \mathbb{M}^{2^n \times 2^n}(\mathcal{R})$

Strassen Algorithm (1969): Assumption

All of these matrices have sizes that are powers of two, that is

$$A, B, C \in \mathbb{M}^{2^n \times 2^n}(\mathcal{R})$$

Strassen Algorithm (1969): Step 1

partitioin A, B, C into equally sized block matrices

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

$$B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$$

$$C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$$

with $M^{2^{n-1} \times 2^{n-1}}(\mathcal{R})$.

Naive Algorithm: detour

Given the partitions the naive algorithm would be rewritten as

$$\begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} = \begin{pmatrix} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{pmatrix}$$

Strassen Algorithm (1969): Step 2

Define the new matrices

$$M_1 = (A_{11} + A_{22})(B_{11} + B_{22})$$

$$M_2 = (A_{21} + A_{22})B_{11}$$

$$M_3 = A_{11}(B_{12} - B_{22})$$

$$M_4 = A_{22}(B_{21} - B_{11})$$

$$M_5 = (A_{11} + A_{12})B_{22}$$

$$M_6 = (A_{21} - A_{11})(B_{11} + B_{12})$$

$$M_7 = (A_{12} - A_{22})(B_{21} + B_{22})$$

Strassen algorithm (1969): Step 3

Express C_{ij} in terms of M_k:

$$\begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} = \begin{pmatrix} M_1 + M_4 - M_5 + M_7 & M_3 + M_5 \\ M_2 + M_4 & M_1 - M_2 + M_3 + M_6 \end{pmatrix}$$

Strassen algorithm (1969): Step 4:

Recursively iterate this division process until the submatrices degenerate into numbers (i.e., the elements of the ring $\mathcal{R}$)

Strassen algorithm (1969): pseudocode

```
strassen(A, B)
    n = A.rows
    let C be a new n*n matrix
    if n == 1
        c_11 = a_11 * b_11
    else partition A, B, C
        let S_1, S_2, ... and S_10 be 10 new n/2 * n/2 matrices
        let P_1, P_2, ... and P_7 be 7 new n/2 * n/2 matrices

        S_1 = B_12 - B_22
        S_2 = A_11 + A_12
        S_3 = A_21 + A_22
        S_4 = B_21 - B_11
        S_5 = A_11 + A_22
        S_6 = B_11 + B_22
        S_7 = A_12 - A_22
        S_8 = B_21 + B_22
        S_9 = A_11 - A_21
        S_10 = B_11 + B_12
```

Strassen algorithm (1969): pseudocode (cont'd)

```
        P1 = strassen(A_11, S_1)
        P2 = strassen(S_2, B_22)
        P3 = strassen(S_3, B_11)
        P4 = strassen(A_22, S_4)
        P5 = strassen(S_5, S_6)
        P6 = strassen(S_7, S_8)
        P7 = strassen(S_9, S_10)

        C_11 = P4 + P5 + P6 - P2
        C_12 = P1 + P2
        C_21 = P3 + P4
        C_22 = P1 + P5 - P3 - P7

    return C
```

Matrix Multiplication Exponent

usually denoted w, is the smallest real number for which any $n \times n$ matrix over a field can be multiplied togehter using $n^{w+o(1)}$ field operations.

Matrix Multiplication Algorithms developed by Human Intelligence so far:

Year	Bound on omega	Authors
1969	2.8074	Strassen
1978	2.796	Pan
1979	2.780	Bini, Capovani, Romani
1981	2.522	Schönhage
1981	2.517	Romani
1981	2.496	Coppersmith, Winograd
[illegible]	[illegible]	[illegible]

What's Next: Better Algorithms vs Parallel Programming