

Matrices and Strassen's Algorithm

1 Master Method

The master method depends on the master theorem

Let $T(n)$ be a function on the natural numbers defined by

$$T(n) = aT(n/b) + f(n)$$

(Interpret n/b as either $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$)

We will compare $f(n)$ with the function $n^{\log_b a}$

Let $T(n)$ be a function on the natural numbers defined by

$$T(n) = aT(n/b) + f(n)$$

for some constant $a \geq 1$ and $b > 1$

1. If $f(n) = O(n^{\log_b a - \epsilon})$ for some constant $\epsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$
2. If $f(n) = \Theta(n^{\log_b a})$ then $T(n) = \Theta(n^{\log_b a} \log n)$
3. If $f(n) = \Omega(n^{\log_b a + \epsilon})$ for some constant $\epsilon > 0$, and if there exist constant $c < 1$ and n_0 such that $af(n/b) \leq cf(n)$ for all $n \geq n_0$, then $T(n) = \Theta(f(n))$

2 Divide-and-conquer for matrix multiplication

2.1 Multiplying two matrices

We are interested in the following problem:

- Input: two square matrices $A, B \in \mathbb{R}^{n \times n}$
- Output: their product $C \in \mathbb{R}^{n \times n}$

We can do it in time $\Theta(n^3)$ by using the definition

$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}$$

(for simplicity, we assume n is a power of two)

Partition A , B and C into four parts:

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \quad B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}, \quad C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$$

Thus

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

2.2 Pseudo-code

Listing 1: R-MULT(A, B)

```

1  let C be a new  $n \times n$  matrix
2  if  $n=1$  then
3       $c_{11} \leftarrow a_{11}b + 11$ 
4  else
5      partition A, B and C
6       $C_{11} \leftarrow \text{R-MULT}(A_{11}, B_{11}) + \text{R-MULT}(A_{12}, B_{21})$ 
7       $C_{12} \leftarrow \text{R-MULT}(A_{11}, B_{12}) + \text{R-MULT}(A_{12}, B_{22})$ 
8       $C_{21} \leftarrow \text{R-MULT}(A_{21}, B_{11}) + \text{R-MULT}(A_{22}, B_{21})$ 
9       $C_{22} \leftarrow \text{R-MULT}(A_{21}, B_{12}) + \text{R-MULT}(A_{22}, B_{22})$ 
10 return C

```

2.3 Running time of simple approach

The running time of the R-MULT satisfies

$$T(N) = 8T(n/2) + \Theta(n^2)$$

Since we do 8 multiplications ($8T(n/2)$) and four additions (of time $\Theta(n^2)$ each)

By the master theorem $T(n) = \Theta(n^3)$

3 Strassen's algorithm

Strassen's algorithm uses a sophisticated divide-and-conquer approach. It has four steps:

1. Divide A, B and C into four parts as before
Running time $\Theta(1)$
2. Create 10 Matrices $S_1, \dots, S_{10} \in \mathbb{R}^{n/2 \times n/2}$, obtained by sums or differences of A_{11}, \dots, B_{22}
Running time: $\Theta(n^2)$
3. Using all the matrices available, recursively compute seven product matrices $P_1, \dots, P_7 \in \mathbb{R}^{n/2 \times n/2}$
Running time $7T(n/2)$
4. Compute C_{11}, \dots, C_{22} by adding and subtracting combinations of P_1, \dots, P_7
Running time $\Theta(n^2)$

3.1 Running time of Strassen's

Thus, the worst-case running time of Strassen's algorithm satisfies

$$T(n) \leq 7T(n/2) + \Theta(n^2)$$

By the master method we obtain

$$T(n) = \Theta(n^{\log_2 7}) = \Theta(n^{2.8074})$$

3.2 Description of S_1, \dots, S_{10}

$$\begin{aligned}
 S_1 &= B_{12} - B_{22} \\
 S_2 &= A_{11} + A_{12} \\
 S_3 &= A_{21} + A_{22} \\
 S_4 &= A_{21} - B_{11} \\
 S_5 &= A_{11} + A_{22} \\
 S_7 &= A_{12} - A_{22} \\
 S_8 &= B_{21} + B_{22} \\
 S_9 &= A_{11} - A_{21} \\
 S_{10} &= B_{11} + B_{12}
 \end{aligned}$$

3.3 Description of P_1, \dots, P_7

$$\begin{aligned}
 P_1 &= A_{11}S_1 = A_{11}B_{12} - A_{11}B_{22} \\
 P_2 &= S_2B_{22} = A_{11}B_{22} + A_{12}B_{22} \\
 P_3 &= S_3B_{11} = A_{21}B_{11} + A_{22}B_{11} \\
 P_4 &= A_{22}S_4 = A_{22}B_{21} - A_{22}B_{11} \\
 P_5 &= S_5S_6 = A_{11}B_{11} + A_{11}B_{22} + A_{22}B_{11} + A_{22}B_{22} \\
 P_6 &= S_7S_8 = A_{12}B_{21} + A_{12}B_{22} - A_{22}B_{21} - A_{22}B_{22} \\
 P_7 &= S_9S_{10} = A_{11}B_{11} + A_{11}B_{12} - A_{21}B_{11} - A_{21}B_{12}
 \end{aligned}$$

3.4 Expressing C

$$\begin{aligned}
 C_{11} &= P_5 + P_4 - P_2 + P_6 \\
 C_{12} &= P_1 + P_2 \\
 C_{21} &= P_3 + P_4 \\
 C_{22} &= P_5 + P_1 - P_3 - P_7
 \end{aligned}$$