Advanced Numerical Methods November 7, 2011

Lecture Notes 4: Fast Fourier Transfomation

Lecturer: Che-Rung Lee *Scribe:Chung Ming-Chia*

1 Problem Definition

We have two polynomials $p(x)$ and $q(x)$. We want to compute the result of $p(x)q(x)$.

$$\begin{aligned} p(x) &= a_{n-1}x^{n-1} + \cdots + a_1 x + a_0 \\ q(x) &= b_{n-1}x^{n-1} + \cdots + b_1 x + b_0 \\ r(x) &= p(x)q(x) \end{aligned}$$

We model the problem as $A\vec{x} = b$.

$$\begin{pmatrix} {x_2}^2 & x_2 & 1 \\ {x_1}^2 & x_1 & 1 \\ {x_0}^2 & x_0 & 1 \end{pmatrix} \begin{pmatrix} C_2 \\ C_1 \\ C_0 \end{pmatrix} = \begin{pmatrix} r(X_2) \\ r(X_1) \\ r(X_0) \end{pmatrix}$$

$$\therefore \vec{x} = A^{-1}\vec{b}$$

If A and $\vec{b}$ were known, we could find the coefficient of $r(x)$ by solving $\vec{x} = A^{-1}\vec{b}$. Since the computation of $inverse(A)$ is roughly $O(n^3)$, if the value of X are arbitrary values, the computational cost will also be $O(n^3)$. By carefully choosing the values of x, we can signaficantly reduce the computation cost.

2 Algorithm

Algorithm 1 Polynomial Multiplication

1. Find x_0,x_1,...,x_{2n} (Find A)
2. Evaluate $p(x_0), p(x_1), \ldots, p(x_{2n})$ ($A\vec{x}_p = \vec{b}_p$)
3. Evaluate $q(x_0), q(x_1), \ldots, q(x_{2n})$ ($A\vec{x}_q = \vec{b}_q$)
4. Compute $r(x_0), r(x_1), \ldots, r(x_{2n})$ (Use $r(x_i) = p(x_i)q(x_i)$ to get $\vec{b}_r$)
5. Solve $\vec{x}_r$ in $A\vec{x}_r = \vec{b}_r$

If we use some random points to substitute in, it's $O(n^3)$ time. Because there are n points and both of two polynomials. So, we use $\omega_j, j = 1, \ldots, 2n-1$. ω_j are the $2n-th$

root of 1. It can get $\vec{b}_p$ adn $\vec{b}_q$ in $O(n \lg n)$ time. We use $2n = 8(\omega = e^{i\frac{2\pi}{8}})$ in the following example.

$$A = W_8 = \left[\begin{array}{cccc|cccc} 1^0 & 1^1 & 1^2 & 1^3 & 1^4 & 1^5 & 1^6 & 1^7 \\ 1 & \omega^1 & \omega^2 & \omega^3 & \omega^4 & \omega^5 & \omega^6 & \omega^7 \\ 1 & \omega^2 & \omega^4 & \omega^6 & \omega^8 & \omega^{10} & \omega^{12} & \omega^{14} \\ 1 & \omega^3 & \omega^6 & \omega^9 & \omega^{12} & \omega^{15} & \omega^{18} & \omega^{21} \\ \hline 1 & \omega^4 & \omega^8 & \omega^{12} & \omega^{16} & \omega^{20} & \omega^{24} & \omega^{28} \\ 1 & \omega^5 & \omega^{10} & \omega^{15} & \omega^{20} & \omega^{25} & \omega^{30} & \omega^{35} \\ 1 & \omega^6 & \omega^{12} & \omega^{18} & \omega^{24} & \omega^{30} & \omega^{36} & \omega^{42} \\ 1 & \omega^7 & \omega^{14} & \omega^{21} & \omega^{28} & \omega^{35} & \omega^{42} & \omega^{49} \end{array}\right]$$

$$W_4 = \left[\begin{array}{cc|cc} 1^0 & 1^2 & 1^4 & 1^6 \\ 1 & \omega^2 & \omega^4 & \omega^6 \\ \hline 1 & \omega^4 & \omega^8 & \omega^{12} \\ 1 & \omega^6 & \omega^{12} & \omega^{18} \end{array}\right]$$

We want to find the relation between W_8 and W_4. Then, we can use Divide-and-Conquer. First, we collect all odd columns to the "front" and put all even columns to the "back". $\Leftrightarrow$ multiply a permutation matrix P.

$$P_8 = \left(\begin{array}{c|c|c|c|c|c|c|c} 1 & & & & & & & \\ \hline & & & & 1 & & & \\ \hline & 1 & & & & & & \\ \hline & & & & & 1 & & \\ \hline & & 1 & & & & & \\ \hline & & & & & & 1 & \\ \hline & & & 1 & & & & \\ \hline & & & & & & & 1 \end{array}\right)$$

Then,

$$W_8 \times P_8 = \begin{pmatrix} M_1 & M_3 \\ M_2 & M_4 \end{pmatrix}$$

$$= \left[\begin{array}{cccc|cccc} 1^0 & 1^1 & 1^2 & 1^3 & 1^4 & 1^5 & 1^6 & 1^7 \\ 1 & \omega^2 & \omega^4 & \omega^6 & \omega^1 & \omega^3 & \omega^5 & \omega^7 \\ 1 & \omega^4 & \omega^8 & \omega^{12} & \omega^2 & \omega^6 & \omega^{10} & \omega^{14} \\ 1 & \omega^6 & \omega^{12} & \omega^{18} & \omega^3 & \omega^9 & \omega^{15} & \omega^{21} \\ \hline 1 & \omega^8 & \omega^{16} & \omega^{24} & \omega^4 & \omega^{12} & \omega^{20} & \omega^{28} \\ 1 & \omega^{10} & \omega^{20} & \omega^{30} & \omega^5 & \omega^{15} & \omega^{25} & \omega^{35} \\ 1 & \omega^{12} & \omega^{24} & \omega^{36} & \omega^6 & \omega^{18} & \omega^{30} & \omega^{42} \\ 1 & \omega^{14} & \omega^{28} & \omega^{42} & \omega^7 & \omega^{21} & \omega^{35} & \omega^{49} \end{array}\right]$$

$$M_1 = M_2 = W_4$$

Let M_x be $\left(\begin{array}{cc|cc} 1^0 & 0 & 0 & 0 \\ 0 & \omega^1 & 0 & 0 \\ \hline 0 & 0 & \omega^2 & 0 \\ 0 & 0 & 0 & \omega^3 \end{array}\right)$ Let M_y be $\left(\begin{array}{cc|cc} \omega^4 & 0 & 0 & 0 \\ 0 & \omega^5 & 0 & 0 \\ \hline 0 & 0 & \omega^6 & 0 \\ 0 & 0 & 0 & \omega^7 \end{array}\right)$

Then, $M_3 = M_x \times M_1$, $M_4 = M_y \times M_2$

$\because M_y = (\omega^4) \times M_x$
$\therefore M_y = -M_x$
Let D_4 be M_x.
$\because P_8$ is an orthogonal matrix.
$\therefore P_8 \times P_8^{-1} = I$.
Then,

$$\begin{aligned} W_8\vec{x} &= W_8 P_8 {P_8}^{-1}\vec{x} \\ &= \begin{pmatrix} W_4 & D_4W_4 \\ W_4 & -D_4W_4 \end{pmatrix} {P_8}^{-1}\vec{x} \end{aligned}$$

Then we can reduce the problem(W_8) to the subproblem(W_4), whose size is half of original one.

3 iFFT Algorithm

Algorithm 2 iFFT Algorithm

$\vec{c}$ =BitReverse($\vec{x}$)
for $s = 0 : \lg n - 1$ **do**
  $m \leftarrow 2^s$
  $\omega \leftarrow e^{\frac{-i\theta}{m}}$
  Set D_m
  for $k = 0 : 2m : n - 1$ **do**
    $C_1 \leftarrow C(k : k + m - 1)$
    $C_2 \leftarrow D_m C(k + m : k + 2m - 1)$
    $C(k : k + 2m - 1) \leftarrow [C_1 + C_2, C_1 - C_2]^T$
  end for
end for

4 Conclusion

At the first, we can use FFT to get $\vec{b}_p$ and $\vec{b}_q$ in $O(n \lg n)$ time. Then, we compute the array-wise multiplication($\vec{b}_r$) of $\vec{b}_p$ and $\vec{b}_q$. Finally, we use iFFT to get $\vec{x}_r$ in $O(n \lg n)$ time.

5 Appendix

Definition 1 (Euler Formula)**.**

$$e^{i\theta} = \cos\theta + i\sin\theta$$

Proof. O(1) Multiplication
Let $x = e^{i\theta}$.

Let $y = e^{i\phi}$.
Then,

$$\begin{aligned} x * y &= e^{(i\theta)+(i\phi)} \\ &= e^{i(\theta+\phi)} \\ &= \cos(\theta+\phi) + i\sin(\theta+\phi). \end{aligned}$$

□

We can compute x^n in $O(1)$ time, if $x = e^{i\theta}$.

Definition 1 (Orthgonal Matrix)**.**

$$A = (\vec{a_1}, \vec{a_2}, \ldots, \vec{a_n}), \| \vec{a_i} \| = 1$$

$$\vec{a_i}^T \vec{a_j} \begin{cases} 0 & \textit{if } i \neq j \\ 1 & \textit{otherwise} \end{cases}$$

Proof. The inverse of an orthogonal matrix A is A^T.

$\because A = (\vec{a_1}, \vec{a_2}, \ldots, \vec{a_n}) \therefore A^T = \begin{pmatrix} \vec{a_1}^T \\ \vec{a_2}^T \\ \vdots \\ \vec{a_n}^T \end{pmatrix}$

$\Rightarrow A^T A = \begin{pmatrix} \vec{a_1}^T\vec{a_1}(1) & \vec{a_1}^T\vec{a_2}(0) & \ldots & \vec{a_1}^T\vec{a_n}(0) \\ \vec{a_2}^T\vec{a_1}(0) & \ddots & & \vec{a_1}^T\vec{a_n}(0) \\ \vdots & & \ddots & \vec{a_1}^T\vec{a_n}(0) \\ \vec{a_n}^T\vec{a_1}(0) & \vec{a_n}^T\vec{a_2}(0) & \ldots & \vec{a_1}^T\vec{a_n}(1) \end{pmatrix}$

$\because A^T A = I \therefore A^{-1} = A^T$ □