FAST FOURIER TRANSFORM

REU SUMMER 2005

Abstract. This note is taken from the lecture notes of Akos Magyar.

If N is a natural number, then let $\mathcal{F}_N$ denote the Fourier transform of functions defined on $\mathbb{Z}_N$, that is:

$$\mathcal{F}_N f(m) = \sum_{n=0}^{N-1} f(n)\omega_N^{-nm} \tag{1}$$

where $\omega_N = e^{\frac{2\pi i}{N}}$ is the N-th root of unity.

If N is large an important practical problem arises which is to compute the Fourier transform using as few elementary operations (such as multiplications and additions) as possible.

If $M(N)$ denotes the minimum number of multiplications needed to compute $\mathcal{F}_N f$ of any function $f : \mathbb{Z}_N \to \mathbb{C}$, then a naive count tells us that $M(N) \leq N^2$. Our aim is to discuss the following

Theorem 1. (Cooley-Tukey 1965) *Let $N = 2^k$ be a power of two. Then the Fourier transform $\mathcal{F}_N f$ of any function $f : \mathbb{Z}_N \to \mathbb{C}$ can be computed by performing at most: $M(N) \leq N(\log_2 n - 1)$ multiplications.*

The algorithm behind the above theorem is the so-called Fast Fourier Transform (FFT), and has turned out to be extremely useful in applications, such as in signal processing. It has been implicitly used by many mathematicians, arguably even by Gauss in 1805!

It is based on the following

Lemma 2. *One has $M(2N) \leq 2M(N) + 2N$.*

Proof. For $f : \{0, 1, \ldots, 2N-1\} \to \mathbb{C}$ lets denote its restrictions to even and odd numbers by: $f_e, f_o : \{0, 1, \ldots, N-1\} \to \mathbb{C}$ where

$$f_e(n) = f(2n)\ , \quad f_o(n) = f(2n+1)$$

Also since $\omega_{2N}^2 = \omega_N$ one has by definition if $0 \leq m < N$:

$$\mathcal{F}_{2N} f(m) = \sum_{n=0}^{N-1} f(2n)\omega_{2N}^{-2nm} + \sum_{n=0}^{N-1} f(2n+1)\omega_{2N}^{-(2n+1)m} = \tag{2}$$

$$\sum_{n=0}^{N-1} f_e(n)\omega_N^{-nm} + \omega_{2N}^{-m} \sum_{n=0}^{N-1} f_o(n)\omega_N^{-nm} = \mathcal{F}_N f_e(m) + \omega_{2N}^{-m}\mathcal{F}_N f_o(m)$$

while for $N \leq m' < 2N$ by writing $m' = N + m$ one gets exactly the same way (using $\omega_{2N}^N = -1$)

$$\mathcal{F}_{2N} f(m') = \mathcal{F}_N f_e(m) - \omega_{2N}^{-m}\mathcal{F}_N f_o(m) \tag{3}$$

Now to compute $\mathcal{F}_N f_e$ and $\mathcal{F}_N f_o$, one needs $2M(N)$ multiplications and then $2N$ additional multiplications are needed to compute ω_{2N}^{-m} and the products $\omega_{2N}^{-m}\mathcal{F}_N f_o(m)$. This proves the lemma. □

Proof of Theorem1. Let $N = 2^k$ and proceed by induction on k.

For $k = 1$ one has: $\mathcal{F}_2 f(0) = f(0) + f(1)$ and $\mathcal{F}_2 f(1) = f(0) - f(1)$ so $M(2) = 0$

For the induction step $k \to k+1$ one has by the above lemma:

$$M(2^{k+1}) \leq 2M(2^k) + 2^{k+1}$$

$$\frac{M(2^{k+1})}{2^{k+1}} \leq \frac{M(2^k)}{2^K} + 1 \leq k - 1 + 1 = k$$

and this is what we wanted to show. □

Next we discuss a "cheap" way of multiplying polynomials of large degree and also large numbers using the FFT.

Let $p(x) = \sum_{n=0}^{P} a_n x^n$ and $q(x) = \sum_{m=0}^{Q} b_m x^m$ be polynomials of degrees P and Q. Then their product $r(x) = p(x)q(x)$ is of the form: $r(x) = \sum_{k=0}^{R} c_k x^k$ with $R = P + Q$ and for $0 \leq k \leq R$ one has

$$c_k = \sum_n a_n b_{k-n} \quad \text{where} \quad 0 \leq n \leq P \quad \text{and} \quad 0 \leq k-n \leq Q \tag{4}$$

Again a by naive count, to compute the coefficients c_k one would first compute all the products $a_n b_m$ using $(P+1)(Q+1)$ multiplications. The idea is that formula (4) looks like the convolutions of functions on $\mathbb{Z}_N$ which is transformed into point-wise multiplication by the Fourier transform.

Let N be a power of 2 such that: $P + Q < N \leq 2(P+Q)$ and define the functions: $f, g : \mathbb{Z}_N \to \mathbb{C}$ by:

$$f(n) = \begin{cases} a_n & \text{if } 0 \leq n \leq P \\ 0 & \text{if } P < n \leq N-1 \end{cases}$$

and similarly

$$g(m) = \begin{cases} b_m & \text{if } 0 \leq n \leq Q \\ 0 & \text{if } Q < n \leq N-1 \end{cases}$$

Then it is easy to check that

$$c_k = f * g(k) = \sum_{n=0}^{N-1} f(n)g(k-n)$$

Note that the difference $k-n$ is computed in $\mathbb{Z}_N$ that is $(mod\, N)$, so if $k < n$ then $k-n := N-(n-k)$. Using the fact that: $\widehat{f * g}(m) = \hat{f}(m)\hat{g}(m)$ one computes the convolution $f * g$ by applying the FFT twice to get $\hat{f}$ and $\hat{g}$, then using N multiplications one gets $\widehat{f * g}$ and by one more application of the FFT gives $f * g$. Thus we have

Corollary 3. *The product polynomial $r(x) = p(x)q(x)$ can be computed by using no more than*

$$M \leq 3N \log_2 N \; - 2N \tag{5}$$

multiplications, where $N \leq 2(P+Q)$.