Fast Fourier Transform I

Motivation

Problem: Given two polynomials $f(x)$ and $g(x)$, find $f(x)g(x)$.

Example: $(2x+1)(x-3) = 2x^2 - 5x - 3$

We will assume throughout this lecture that the number of terms (including lower order 0 terms) in the polynomial is n and that n is a power of 2. In practice, if that's not true, you can pad the polynomial input with higher-order zeroes.

Monkey (FOIL): $\Theta(n^2)$

Lower bound: $\Omega(n)$

Applications: cryptography (Notice that $607 = 6 \cdot x^2 + 7$ and $921 = 9x^2 + 2x + 1$ when $x = 10$. Finding a better way to multiply polynomials together automatically yields a better way of multiplying large integers.)

Point-Value Representation

Coefficient form representation

$3x^2 - x + 2 = (3, -1, 2)$

The outside world generally likes polynomials in coefficient form.

Point-value representation (PV)

$3x^2 - x + 2 = \{(-1, 6), (0, 2), (1, 4)\}$

Can we convert from point-value form back to coefficient form?

$f(x) = ax^2 + bx + c \Rightarrow$

$a - b + c = 6, c = 2, a + b + c = 4 \Rightarrow$

$f(x) = 3x^2 - x + 2$

Why do we like point-value form?

Assume that we want to multiply the two polynomials $3x^2 - x + 2$ and $x^2 + 2x + 1$.

How many terms will be in the answer? 5

$3x^2 - x + 2 \to \{(-2, 16), (-1, 6), (0, 2), (1, 4), (2, 12)\}$

$x^2 + 2x + 1 \to \{(-2, 1), (-1, 0), (0, 1), (1, 4), (2, 9)\}$

The question: If $f(-2) = a$ and $g(-2) = b$, what is $f(-2)g(-2)$?

The answer: ab

$(3x^2 - x + 2)(x^2 + 2x + 1) =$

$\{(-2, 16), (-1, 0), (0, 2), (1, 16), (2, 108)\}$

So if we represent polynomials in point-value form, multiplication of polynomials takes time $\Theta(n)$!

The New Problem

Coefficient form of $f(x)$ and $g(x)$ $\xrightarrow{?}$
Point-value form of $f(x)$ and $g(x)$ $\xrightarrow{\Theta(n)}$
Point-value form of $f(x)g(x)$ $\xrightarrow{?}$
Coefficient form of $f(x)g(x)$

The new goal will be to figure out how to transition back and forth from coefficient form to point-value form quickly.

Definitions and Notation

Given two vectors v and w, define the result of componentwise multiplication to be $v * w$ so that $(v * w)_k = v_k w_k$.

Example: $(2, 1, 3) * (-1, 5, 0) = (-2, 5, 0)$

Define the concatenation of two vectors v and w to be $v.w$.

Example: $(2, 1, 3).(-1, 5, 0) = (2, 1, 3, -1, 5, 0)$

Math fact: $e^{ix} = \cos(x) + i\sin(x)$

Example: $\mathrm{e}^{i\frac{\pi}{4}} = \frac{1}{\sqrt{2}} + \frac{i}{\sqrt{2}}$

Computers are weird.

To compute the point-value form of a polynomial $f(x)$, a human being would prefer to plug in small integer values of x.

$f(x) = x^3 - x + 2 \rightarrow$

$f(x) = \{(-1, 2), (0, 2), (1, 2), (2, 8)\}$

The computer prefers the *roots of unity*. The nth roots of unity are the following numbers:

$\{e^{\frac{2\pi i 0}{n}}, e^{\frac{2\pi i 1}{n}}, e^{\frac{2\pi i 2}{n}}, \ldots, e^{\frac{2\pi i(n-1)}{n}}\}$

$f(x) = x^3 - x + 2 \rightarrow$

Because $n = 4$, use the 4th roots of unity: $\{1, i, -1, -i\}$

$f(x) = \{(1, 2), (i, 2 - 2i), (-1, 2), (-i, 2 + 2i)\}$

FFT Definitions

Assume that $P(x) = \sum_{i=0}^{n-1} a_i x^i$

$= a_0 + a_1 x + a_2 x^2 + \ldots a_{n-1} x^{n-1}$

$$\omega \equiv e^{\frac{2\pi i}{n}}$$

$$FFT(P(x)) \equiv (P(\omega^{n-1}), P(\omega^{n-2}), \ldots, P(\omega), P(1))$$

$$= \left(P(e^{\frac{2\pi i(n-1)}{n}}), \ldots, P(e^{\frac{2\pi i2}{n}}), P(e^{\frac{2\pi i1}{n}}), P(e^{\frac{2\pi i0}{n}})\right)$$

Define the following two polynomials.

$P^{(0)}(x) \equiv \sum_{i=0}^{\frac{n}{2}-1} a_{2i} x^i$

$= a_0 + a_2 x + a_4 x^2 + \ldots + a_{n-2} x^{\frac{n}{2}-1}$

$P^{(1)}(x) \equiv \sum_{i=0}^{\frac{n}{2}-1} a_{2i+1} x^i$

$= a_1 + a_3 x + a_5 x^2 + \ldots + a_{n-1} x^{\frac{n}{2}-1}$

FFT Algorithm

$FFT(P(x))$

- If $n = 1$ return $P(x)$.
- Compute $FFT(P^{(0)}(x))$.
- Compute $FFT(P^{(1)}(x))$.
- $\omega \equiv e^{\frac{2\pi i}{n}}$
  $\Omega \equiv (\omega^{\frac{n}{2}-1}, \omega^{\frac{n}{2}-2}, \ldots, \omega, 1)$
  Return the following expression:
  $[FFT(P^{(0)}(x)) - \Omega * FFT(P^{(1)}(x))].[FFT(P^{(0)}(x)) + \Omega * FFT(P^{(1)}(x))]$

When $P(x)$ is a constant, does it work?

FFT Example

Assume that the goal is to multiply $3x - 4$ by $-x + 1$.
We will need to use $n = 4$.

$[FFT(P^{(0)}(x)) - \Omega * FFT(P^{(1)}(x))].[FFT(P^{(0)}(x)) + \Omega * FFT(P^{(1)}(x))]$

$\omega = i$	$0x^3 + 0x^2 + 3x - 4$			
$\omega = -1$	$0x + 3$		$0x - 4$	
$\omega = 1$	0	3	0	-4
$\omega = 1$	0	3	0	-4
$\omega = -1$	$(3, 3)$		$(-4, -4)$	
$\omega = i$	$(-4 - 3i, -7, -4 + 3i, -1)$			

Final row computation: $[(-4, -4) - (i, 1) * (3, 3)].[(-4, -4) + (i, 1) * (3, 3)] =$

$(-4 - 3i, -7, -4 + 3i, -1)$

$FFT(0, 0, 3, -4) = (-4 - 3i, -7, -4 + 3i, -1)$

FFT Example

Assume that the goal is to multiply $3x - 4$ by $-x + 1$. We will need to use $n = 4$.

$[FFT(P^{(0)}(x)) - \Omega * FFT(P^{(1)}(x))].[FFT(P^{(0)}(x)) + \Omega * FFT(P^{(1)}(x))]$

$\omega = i$	$0x^3 + 0x^2 - x + 1$			
$\omega = -1$	$0x - 1$		$0x + 1$	
$\omega = 1$	0	-1	0	1
$\omega = 1$	0	-1	0	1
$\omega = -1$	$(-1, -1)$		$(1, 1)$	
$\omega = i$	$(1 + i, 2, 1 - i, 0)$			

Final row computation: [(1,1)-(i,1)*(-1,-1)].[(1,1)+(i,1)*(-1,-1)]=(1+i,2,1-i,0)

$FFT(0, 0, -1, 1) = (1 + i, 2, 1 - i, 0)$

FFT Inversion

We now multiply our two answers together componentwise to get the point-value form of the answer.

$$\Rightarrow (-4-3i,-7,-4+3i,-1)*(1+i,2,1-i,0) = (-1-7i,-14,-1+7i,0)$$

Inversion: **In order to invert the FFT, perform the FFT using $\overline{\omega}$ in place of ω and then multiply by $\frac{1}{n}$.**

FFT Example

$$[FFT(P^{(0)}(x)) - \Omega * FFT(P^{(1)}(x))].[FFT(P^{(0)}(x)) + \Omega * FFT(P^{(1)}(x))]$$

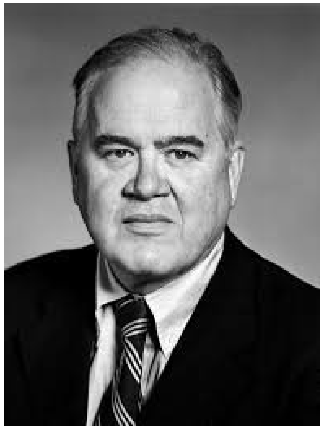
$\omega = -i$	$(-1-7i)x^3 - 14x^2 + (-1+7i)x + 0$			
$\omega = -1$	$(-1-7i)x + (-1+7i)$		$-14x + 0$	
$\omega = 1$	$-1-7i$	$-1+7i$	-14	0
$\omega = 1$	$-1-7i$	$-1+7i$	-14	0
$\omega = -1$	$(14i, -2)$		$(14, -14)$	
$\omega = -i$	$(0, -12, 28, -16)$			

Final row computation: [(14,-14)-(-i,1)*(14i,-2)].[(14,-14)+(-i,1)*(14i,-2)]=

(0,-12,28,-16)

To get the final answer: $\frac{1}{4}(0, -12, 28, -16) = (0, -3, 7, -4) \Rightarrow$
$(3x-4)(-x+1) = -3x^2 + 7x - 4$

FFT History



Tukey

Gauss

- John W. Tukey (Princeton) reportedly came up with the idea during a meeting of a US presidential advisory committee discussing ways to detect nuclear-weapon tests in the Soviet Union.

- Another participant at that meeting, Richard Garwin of IBM, recognized the potential of the method and put Tukey in touch with Cooley, who implemented it for a different (and less-classified) problem: analyzing 3d crystallographic data.

- In 1965, Cooley and Tukey published a paper describing how to perform the FFT conveniently on a computer.

- This algorithm, including its recursive application, was invented around 1805 by Carl Friedrich Gauss, who used it to interpolate the trajectories of the asteroids Pallas and Juno, but his work was not widely recognized (being published only posthumously and in neo-Latin).