We are all in this together

Almost there, gang -- one more week!

And we will get through this together.

- Use a laptop or desktop with a large screen so you can read these words clearly.
- To conserve bandwidth, please turn off your camera.
- Please mute your microphone unless I call on you.
- Please have the chat window open to ask questions.
- Reading for today: 11.2. (Ch. 11 not on final.)
- PS11 due ~~on last day of class~~. today (but all deadlines are flexible).
- **FINAL EXAM, FRI MAY 15, 9:45AM–NOON.**
- Final exam review, Wed May 13, 2:45–5:00pm.
- Today's DJ: Samer.

Final exam

- Fri May 15, 9:45am–noon. (Upload starts 11:45am.)
- Before exam, prepare 13 sheets of paper, numbered 1–13. If you put each problem on a separate sheet of paper, that will increase the probability that you get credit for all of your work.
- Sign into Zoom by 9:30–9:35am or so, video on and mic muted, and sign into Canvas at the same time.
- Exam will be emailed to you around 9:40am.
- Work on exam until 11:45am. (No upload until then.)
- Scan all 13 pages (including blank ones) into a single PDF and upload to Canvas between 11:45am and noon.

Please bring something analog to do.

Coverage: Roughly 30% Chs. 2-3, 40% Chs. 5-7, 30% Chs. 8-10?
Not on final exam: Ch. 4, Ch. 11

The Discrete Fourier Transform (DFT)

Definition

Fix $N \in \mathbf{N}$, let $\omega = e^{2\pi i/N}$ be the natural primitive Nth root of unity in $\mathbf{C}$, and let $f : \mathbf{Z}/(N) \to \mathbf{C}$ be a signal. We define the DFT of f to be $\hat{f} : \mathbf{Z}/(N) \to \mathbf{C}$ given by

$$\hat{f}(k) = \frac{1}{N} \sum_{n=0}^{N-1} f(n)\omega^{-nk}.$$

Previously: We saw that if we can compute the DFT quickly (faster than $O(N^2)$), all kinds of good stuff happens.

E.g., we can multiply large numbers faster!
A quantum computer is a billion-dollar box for computing DFT.

Example: $N = 12$

$w^{12} = 0$
exps (mod 12)

Writing out the definition of $\hat{f}(k)$ for $N = 12$, we get

$$\hat{f}(0) = \frac{1}{12}\left(f(0) + f(1) + f(2) + \cdots + f(11)\right)$$

$$\hat{f}(1) = \frac{1}{12}\left(f(0) + \omega^{-1} f(1) + \omega^{-2} f(2) + \cdots + \omega^{-11} f(11)\right)$$

$$\hat{f}(2) = \frac{1}{12}\left(f(0) + \omega^{-2} f(1) + \omega^{-4} f(2) + \cdots + \omega^{-10} f(11)\right)$$

$$\hat{f}(3) = \frac{1}{12}\left(f(0) + \omega^{-3} f(1) + \omega^{-6} f(2) + \cdots + \omega^{-9} f(11)\right)$$

$$\hat{f}(4) = \frac{1}{12}\left(f(0) + \omega^{-4} f(1) + \omega^{-8} f(2) + \cdots + \omega^{-8} f(11)\right)$$

$f(3) + w^{-4} f(4) + w^{-8} f(5) +$

0, -4, -8
0, -4, -8

$$\vdots$$

$$\hat{f}(11) = \frac{1}{12}\left(f(0) + \omega^{-11} f(1) + \omega^{-10} f(2) + \cdots + \omega^{-1} f(11)\right)$$

$O(N^2)$ time

Subgroups of C_N

$C_1 \le C_3 \le C_6 \le C_{12}$

$\le C_2 \le C_4 \le$

Theorem

Fix a positive integer N and let $\omega = e^{2\pi i/N}$. If $N = dq$ for positive $d, q \in \mathbf{Z}$, then C_d (the group of complex dth roots of unity) is precisely $\langle \omega^q \rangle$, the subgroup of C_N generated by ω^q.

Example: $N = 12$, $\omega = \omega_{12} = e^{2\pi i/12}$, so $\omega^{12} = 1$.

$C_4 \not\le C_3$

$C_3 \not\le C_4$

I choose to look at chain C1, C3, C6, C12.

$$\langle \omega \rangle = \{1, \omega, \omega^2, \omega^3, \ldots, \omega^{10}, \omega^{11}\} = C_{12}$$

$$C_k = \langle \omega^{\frac{12}{k}} \rangle \quad \langle \omega^2 \rangle = \{1, \omega^2, \omega^4, \omega^6, \omega^8, \omega^{10}\} = C_6$$

$$\langle \omega^4 \rangle = \{1, \omega^4, \omega^8\} = C_3$$

$$\langle \omega^{12} \rangle = \{1\} = \{\omega^{12}\} = C_1$$

Also subgp, but not in our chosen chain:

$C_4 = \langle \omega^3 \rangle = \{1, \omega^3, \omega^6, \omega^9\}$

So $C_1 \le C_3 \le C_6 \le C_{12}$.

$C_1 = \langle \omega^{12} \rangle$, $C_3 = \langle \omega^4 \rangle$, $C_6 = \langle \omega^2 \rangle$, $C_{12} = \langle \omega \rangle$

As we go up the chain, we find bigger subgroups generated by smaller powers of omega.

The Fast Fourier Transform based on $H_0 \leq \cdots \leq H_n$

Us: $n=3$ $\quad C_1 \leq C_3 \leq C_6 \leq C_{12}$

$N=12$ $\quad$ 1 2 3 step in FFT

Fix $N \in \mathbf{N}$ and $\omega = e^{2\pi i/N}$. Let

$$C_1 = H_0 \leq H_1 \leq \cdots \leq H_{n-1} \leq H_n = C_N \tag{1}$$

As we go through algorithm:

$\mathbf{x} = \begin{bmatrix} x(0) \\ \vdots \\ x(N-1) \end{bmatrix}$ is current state, $\mathbf{y} = \begin{bmatrix} y(0) \\ \vdots \\ y(N-1) \end{bmatrix}$ new state.

Initialize: Let $\mathbf{x} = \begin{bmatrix} f(0) \\ \vdots \\ f(N-1) \end{bmatrix}$.

(I.e., we start with data for original signal f and step-by-step, turn it into the dat~~e~~a for the DFT $\hat{f}$.)

Main loop of the FFT

These are fudge factors that will appear

For $i = 1$ to n: Suppose $H_{i-1} = \langle \omega^m \rangle$, $H_i = \langle \omega^k \rangle$.
Since $H_{i-1} \leq H_i$, $m = kd$ for some $d \in \mathbf{Z}$. Use the standard transversal $1, \omega^k, \omega^{2k}, \dots, \omega^{(d-1)k}$ for H_{i-1} in H_i.
Fill subgroup: For $j = 0$ to $(N/k) - 1$ (i.e., ω^{jk} ranges over all elements of H_i), set

N=12

$$\begin{aligned} y(jk) &= \sum_{r=0}^{d-1} x(jm + rk)\omega^{-rkj} \\ &= x(jm) + x(jm + k)\omega^{-kj} + x(jm + 2k)\omega^{-2kj} + \dots. \end{aligned}$$

Translate: For $\ell = 1$ to $k - 1$ and $j = 0$ to $(N/k) - 1$, we set

$$y(jk + \ell) = \sum_{r=0}^{d-1} x(jm + rk + \ell)\omega^{-rkj}. \qquad (2)$$

Set current state to new state and loop. Set $\mathbf{x} = \mathbf{y}$ and loop.
Rescale. Divide every entry of $\mathbf{x}$ by N.

Example: $C_1 \leq C_3 \leq C_6 \leq C_{12} = \langle w \rangle$

Initialize: $\langle w^{12} \rangle \quad \langle w^{4} \rangle \quad \langle w^{2} \rangle$

$$x(0) = f(0)$$
$$x(1) = f(1)$$
$$x(2) = f(2)$$
$$x(3) = f(3)$$
$$x(4) = f(4)$$
$$x(5) = f(5)$$
$$x(6) = f(6)$$
$$x(7) = f(7)$$
$$x(8) = f(8)$$
$$x(9) = f(9)$$
$$x(10) = f(10)$$
$$x(11) = f(11)$$

At the end, we want x(k) to be equal to $\hat{f}(k)$

$= (\text{san})$

$i = 1$: $m = 12$, $k = 4$, $d = 3$

$\{1, w^4, w^8\}$ $= \frac{12}{4}$

$H_0 \leq L_1 = \langle w^{12} \rangle$

$L_1 = C_3 = \langle w^4 \rangle$

$j = 0$: $y(0) = x(0) + x(4) + x(8)$

$y(1) = x(1) + x(5) + x(9)$

$y(2) = x(2) + x(6) + x(10)$

$y(3) = x(3) + x(7) + x(11)$

$j = 1$: $y(4) = x(0) + x(4)w^{-4} + x(8)w^{-8}$

$k_j = 4$

$y(5) = x(1) + x(5)w^{-4} + x(9)w^{-8}$

$y(6) = x(2) + x(6)w^{-4} + x(10)w^{-8}$

$y(7) = x(3) + x(7)w^{-4} + x(11)w^{-8}$

$j = 2$: $y(8) = x(0) + x(4)w^{-8} + x(8)w^{-4}$

$k_j = 8$

$-2(8) = -16 = -4$

After i = 1 stage of the main loop, our current state is:

$x(0) = f(0) + f(4) + f(8)$
$x(1) = f(1) + f(5) + f(9)$
$x(2) = f(2) + f(6) + f(10)$
$x(3) =$
$x(4) = f(0) + f(4)\, w^{-4} + f(8)\, w^{-8}$
$x(5) =$
$x(6) =$
$x(7) =$
$x(8) = f(0) + f(4)\, w^{-8} + f(8)\, w^{-4}$
$x(9) =$
$x(10) =$
$x(11) =$

translates

$i = 2$: $m = 4$, $k = 2$, $d = 2$ $\quad H_1 = C_3 = \langle w^4 \rangle$

find g: $\langle w^2 \rangle$ $\quad i_1 4/2 \quad H_2 = C_6 = \langle w^2 \rangle$

$j=0$: $y(0) = x(0) + x(2)$

$y(1) = x(1) + x(3)$

$j=1$: $y(2) = x(4) + x(6)\, w^{-2}$

$y(3) = x(5) + x(7)\, w^{-2}$

$j=2$: $y(4) = x(8) + x(10)\, w^{-4}$

etc.

$j=3$: $y(6) = x(0) + x(2)\, w^{-6}$

$j=4$ $y(8) = x(4) + x(6)\, w^{-8}$

$j=5$ $y(10) = x(8) + x(10)\, w^{-10}$

$r=0 \qquad r=1$

After i = 2 step, our current state looks like:

$x(0) = f(0) + f(4) + f(8) + f(2) + f(6) + f(10)$
$= f(0) + f(2) + f(4) + f(6) + f(8) + f(10)$ (Like DFT, but 2,4,...,10)
($x(1)$ is translate of $x(0)$)

$$x(2) = f(0) + f(4)w^{-4} + f(8)w^{-8} + \left(f(2) + f(6)w^{-4} + f(10)w^{-8}\right)w^{-2}$$

$$= f(0) + f(4)w^{-4} + f(8)w^{-8} + f(2)w^{-2} + f(6)w^{-6} + f(10)w^{-10}$$

$$= f(0) + f(2)w^{-2} + f(4)w^{-4} + f(6)w^{-6} + f(8)w^{-8} + f(10)w^{-10}$$

Like DFT!!!!

Somehow, this miracle continues, and gives us the DFT for N=12 at the end.

Only way to really understand this is to do it yourself.

How much time does all this take?

Simplify: Suppose $N = 2^n$, $\overbrace{C_1 \leq C_2 \leq C_4 \leq \cdots \leq C_N}^{n \text{ steps}}$, $d = 2$ at every stage. (Standard case in engineering.)

- Number of steps in big loop is $n = \log_2(N)$.
- In one iteration of big loop, for each of the N entries of $\mathbf{y}$, we do d multiplications and $d - 1$ additions, a number **independent of** N.
- So total time taken is big O of n times N, or $O(N \log N)$.

$O(N \log N)$ **is way better than** $O(N^2)$.
WIN.

`https://youtu.be/dfe8tCcHnKY?t=19`

Thank you!!!

This can't possibly actually work all the time, can it?

If we had another week, we could get into the proof of this — you now have enough background knowledge to understand the proof. Very briefly:

- In each step of subgroup chain

$$\{1\} = H_0 \leq H_1 \leq \cdots \leq H_{n-1} \leq H_n = C_N,$$

  we have a transversal of d elements.
- By making a choice of one of d elements at each of the n stages, we can obtain each element of C_N exactly once.
- This decomposition idea, plus the fact that the "fudge factors" ω^{-rj} are actually *homomorphisms* (!!), mean that we can compute the DFT by this "divide and conquer" method.