Recap on complex numbers

Complex numbers: objects of the form $a + bi$, where $a, b \in \mathbb{R}$ and i stands for an 'imaginary' number such that $i^2 = -1$.

Modulus of a complex number: $|a + bi| = \sqrt{a^2 + b^2}$.

Complex numbers of modulus 1 can be written as $\cos\theta + i\sin\theta$ for a certain angle $\theta \in [0, 2\pi]$ (measured in radians!).

Alternative notation: $\cos\theta + i\sin\theta = e^{i\theta}$. Just a convenient notation, but it hides some nontrivial facts, for instance $e^{i(\theta_1+\theta_2)} = e^{i\theta_1}e^{i\theta_2}$ (follows from high-school trigonometry).

Geometric idea: Multiplication = rotation on the unit circle.

For instance, $e^{i0} = e^{i2\pi} = 1$, and $e^{i\pi} = -1$.

Recap on interpolation

Evaluating a polynomial $a(x) = a_0 + a_1 x + \cdots + a_{n-1} x^{n-1}$ on n given points $t_0, t_1, \ldots, t_{n-1}$: linear map (matrix) $\mathbb{C}^n \to \mathbb{C}^n$.

$$\underbrace{\begin{bmatrix} 1 & t_0 & t_0^2 & \ldots & t_0^{n-1} \\ 1 & t_1 & t_1^2 & \ldots & t_1^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & t_{n-1} & t_{n-1}^2 & \ldots & t_{n-1}^{n-1} \end{bmatrix}}_{:=V,\ \text{Vandermonde matrix}} \underbrace{\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix}}_{\mathbf{a}} = \begin{bmatrix} p(a_0) \\ p(a_1) \\ \vdots \\ p(a_{n-1}) \end{bmatrix} =: \underbrace{\begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-1} \end{bmatrix}}_{\mathbf{b}}$$

Evaluation: map coefficients $\mapsto$ values, $\mathbf{b} = V\mathbf{a}$.

Interpolation: map values $\mapsto$ coefficients, $\mathbf{a} = V^{-1}\mathbf{b}$.

Warning when n is large, often V is ill-conditioned.

Discrete Fourier transform

Let $z = e^{-i\frac{2\pi}{n}} = \cos\frac{2\pi}{n} - i\sin\frac{2\pi}{n}$.

Discrete Fourier transform (DFT): evaluation on the n points $1 = z^0, z, z^2, \ldots, z^{n-1}$.
Inverse DFT: interpolation on the same points.

Geometrically, these are n equispaced points on the unit circle; note that $z^n = 1$.
Fourier matrix:

$$V = F = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & z^1 & z^2 & \ldots & z^{n-1} \\ 1 & z^2 & z^4 & \ldots & z^{2n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & z^{n-1} & z^{2n-2} & \ldots & z^{(n-1)^2} \end{bmatrix}$$

Why is DFT interesting?

- It is perfectly well conditioned:

$$\overline{F^T}F = F\overline{F^T} = nI$$

(follows from $z^n = 1$ + geometric progression formula).

hence $\frac{1}{\sqrt{n}}F$ is unitary (complex analogue of orthogonal, conjugate+transpose instead of transpose) and $\kappa(F) = 1$.

- There is a specialized algorithm to compute $F\mathbf{a}$ and $F^{-1}\mathbf{b} = \frac{1}{n}\overline{F}^T\mathbf{b}$ in $O(n \log n)$, instead of $O(n^2)$ for a 'standard' matvec product.
- Applications: polynomial multiplication, structured matrix multiplication, time series analysis...

Fast Fourier Transform

We focus on evaluation, $\mathbf{b} = F\mathbf{a}$ (DFT). Interpolation (IDFT) is analogous, thanks to $F^{-1} = \frac{1}{n}\overline{F}^T$.

Divide-and-conquer: we will reduce one IDFT($2n$) to two IDFT(n).

Setup given a vector $\mathbf{a} = [a_0, a_1, \dots a_{2n-1}]$, we wish to evaluate

$$a(x) = a_0 + a_1 x + \cdots + a_{2n-1}x^{2n-1}$$

in $z = e^{i\frac{2\pi}{2n}}$ and its powers; we already know how to evaluate a polynomial of degree $< n$ in $e^{i\frac{2\pi}{n}} = z^2$ and its powers.

Fast Fourier Transform: the recursion

$$
\begin{aligned}
a(x) &= a_0 + a_1 x + \cdots + a_{2n-1} x^{2n-1} \\
&= \left(a_0 + a_2 x^2 + \cdots + a_{2n-2} x^{2(n-1)}\right) \\
&\qquad\qquad + x\left(a_1 + a_3 x^2 + \cdots + a_{2n-1} x^{2(n-1)}\right) \\
&= a_{ev}(x^2) + x a_{odd}(x^2).
\end{aligned}
$$

where a_{ev}, a_{odd} are the polynomials with coefficients taken from the vectors

$$
\begin{aligned}
\mathbf{a}_{ev} &= [a_0, a_2, \ldots, a_{2n-2}], \\
\mathbf{a}_{odd} &= [a_1, a_3, \ldots, a_{2n-1}].
\end{aligned}
$$

Fast Fourier transform: wrap-up

Input: vector a of length $2n$.

1. Using two DFTs of half the size n, compute

$$\mathbf{c} = \mathrm{DFT}(\mathbf{a}_{ev}) = \left[a_{ev}(1), a_{ev}(z^2), a_{ev}((z^2)^2), \ldots, a_{ev}((z^2)^{n-1})\right],$$

$$\mathbf{d} = \mathrm{DFT}(\mathbf{a}_{odd}) = \left[a_{odd}(1), a_{odd}(z^2), a_{odd}((z^2)^2), \ldots, a_{odd}((z^2)^{n-1})\right],$$

2. For each $k = 1, 2, \ldots, 2n-1$, compute $a(z^k) = a_{even}(z^{2k}) + z^k a_{odd}(z^{2k})$, i.e.,

$$\mathbf{b} = [\mathbf{c} + \mathbf{z} \odot \mathbf{d}, \mathbf{c} - \mathbf{z} \odot \mathbf{d}],$$

where $\mathbf{z} = [1, z, z^2, \ldots, z^{n-1}]$, and $\odot$ is entry-by-entry product.

Remarks

- This gives rise naturally to an algorithm for $n = 2^k$ nodes. Algorithms for non-powers-of-two are possible, but more cumbersome. Usually in applications one can get away by padding the vectors with zeros.
- Variants that use real arithmetic only are possible, but more cumbersome.
- Usually, roots of unity z^k are precomputed.
- Complexity: $\mathrm{DFT}(2n) = 2\,\mathrm{DFT}(n) + 4n$. Standard application of the 'master theorem' gives $O(n \log n)$.

Applications of FFT

Application #1: fast product of polynomials ($O(n \log n)$):
Input vectors $\mathbf{a}$ and $\mathbf{c}$ that contain the coefficients of

$$a(x) = a_0 + a_1 x + \cdots + a_n x^n,$$
$$c(x) = c_0 + c_1 x + \cdots + a_m x^m,$$

Output the vector of coefficients of $a(x)c(x)$, i.e.,

$\mathbf{e} = [a_0 c_0, a_1 c_0 + a_0 c_1, a_2 c_0 + a_1 c_1 + a_0 c_2, \ldots, a_n c_{m-1} + a_{n-1} c_m, a_n c_m]$.

1. Choose a number of nodes $N \geq n + m$ for DFT (why?).
2. Evaluate $\mathbf{b} = DFT(\mathbf{a}) = [a(z^0), a(z^1), a(z^2), \ldots, a(z^{N-1})]$ and $\mathbf{d} = DFT(\mathbf{c})$.
3. Compute $\mathbf{b} \odot \mathbf{d} =$ $[a(z^0)c(z^0), a(z^1)c(z^1), a(z^2)c(z^2), \ldots, a(z^{N-1})c(z^{N-1})]$.
4. Compute $\mathbf{e} = IDFT(\mathbf{b} \odot \mathbf{d})$.

Applications of DFT

Application #1b: fast convolution / moving averages: it's basically the same thing, i.e.,

$$[a_0c_0, a_1c_0 + a_0c_1, a_2c_0 + a_1c_1 + a_0c_2, \dots, a_nc_{m-1} + a_{n-1}c_m, a_nc_m].$$

Application #1c: product with a triangular Toeplitz matrix:

$$\begin{bmatrix} a_0 & a_1 & \dots & a_{n-1} \\ & \ddots & \ddots & \vdots \\ & & a_0 & a_1 \\ & & & a_0 \end{bmatrix} \begin{bmatrix} c_{n-1} \\ \vdots \\ c_1 \\ c_0 \end{bmatrix} = \begin{bmatrix} a_0c_{n-1} + \dots + a_{n-1}c_0 \\ \vdots \\ a_0c_1 + a_1c_0 \\ a_0c_0 \end{bmatrix}$$

(Non-triangular Toeplitz matrices can be seen as the sum of a lower + an upper triangular one.)

Signal processing

Quick derivation to connect our presentation with the usage of FFT in signal processing.

Problem: signal processing

Given n equispaced samples of a certain 'signal' (for instance: a sound wave)

$$\mathbf{b} = [b_0, b_1, b_2, \dots, b_{n-1}],$$

decompose this signal into a sum of sines/cosines.

Put these numbers along the points $z^0, z^1, \dots, z^{n-1}$, where $z = e^{-i\frac{2\pi}{n}} = \cos\frac{2\pi}{n} - i\sin\frac{2\pi}{n}$, and perform IDFT: this returns a polynomial $a(x)$ such that $b_k = a(z^k)$, i.e., the b_k are obtained by 'sampling' this polynomial on n equispaced points on the unit circle.

Signal processing

Actually, instead of a polynomial

$$a(x) = a_0 + a_1 x + a_1 x^2 + \cdots + a_{\frac{n}{2}} x^{\frac{n}{2}} + a_{\frac{n}{2}+1} x^{\frac{n}{2}+1} + \cdots + a_{n-1} x^{n-1},$$

it's better to think about a rational function

$$f(x) = a_0 + a_1 x + a_1 x^2 + \cdots + a_{\frac{n}{2}} x^{\frac{n}{2}} + a_{\frac{n}{2}+1} x^{-(\frac{n}{2}-1)} + \cdots + a_{n-1} x^{-1}.$$

$a(x)$ and $f(x)$ take the same values on $x = z^k$ because $x^n = 1$ on these values of x.

Signal processing

Now, make a change of variable: $x = e^{-i\theta}$. As θ ranges over $[0, 2\pi]$, x ranges over the unit circle. So the elements of $\mathbf{b}$ are obtained by sampling at equispaced points on $[0, 2\pi]$ the function

$$f(e^{i\theta}) = a_0 + a_1(\cos\theta - i\sin\theta) + a_2(\cos 2\theta - i\sin 2\theta) + \dots$$
$$+ a_{2n-2}(\cos(-2\theta) - i\sin(-2\theta)) + a_{2n-1}(\cos(-\theta) - i\sin(-\theta))$$

We can simplify this expression using $\cos(-\theta) = \cos(\theta)$, $\sin(-\theta) = -\sin(\theta)$; so the coefficient of $\cos(\theta)$ is $a_1 + a_{2n-1}$, that of $\sin(\theta)$ is $i(a_{2n-1} - a_1)$, ...