

The Fast Fourier Transform

Andreas Klappenecker

Fourier Transform in Matrix Form

For a polynomial $A(x) = a_0 + a_1x + \dots + a_{n-1}x^{n-1}$ of degree $n-1$, a conversion from coefficient representation to p.v. representation at n distinct points $1, \omega, \dots, \omega^{n-1}$ can be done as follows:

$$\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \dots & \omega^{(n-1)^2} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix}$$

Fast Fourier Transform

Evaluate a degree $n-1$ polynomial $A(x) = a_0 + \dots + a_{n-1} x^{n-1}$ at its n^{th} roots of unity: $\omega^0, \omega^1, \dots, \omega^{n-1}$.

Divide. Divide the polynomial into even and odd powers.

$$A(x) = A_{\text{even}}(x^2) + x A_{\text{odd}}(x^2).$$

Conquer. Evaluate $A_{\text{even}}(x)$ and $A_{\text{odd}}(x)$ at the $\frac{1}{2}n^{\text{th}}$ roots of unity: $v^0, v^1, \dots, v^{n/2-1}$.

Combine.

$$A(\omega^k) = A_{\text{even}}(v^k) + \omega^k A_{\text{odd}}(v^k), \quad 0 \leq k < n/2$$

$$A(\omega^{k+n/2}) = A_{\text{even}}(v^k) - \omega^k A_{\text{odd}}(v^k), \quad 0 \leq k < n/2$$

Fourier Transform in Matrix Form

Split into even and odd powers (green and gray)

Each gray column vector is two copies of the same vector

$$\begin{pmatrix} A(1) \\ A(\omega) \\ \vdots \\ A(\omega^{n-1}) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \dots & \omega^{(n-1)^2} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix}$$

Realize that $1, \omega, \dots, \omega^{n/2-1}, -1, -\omega, \dots, -\omega^{n/2-1}$ (upper/lower part)

Reorganization

$$\begin{pmatrix} A(1) \\ A(\omega) \\ \vdots \\ A(\omega^{n-1}) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)^2} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix}$$

Idea: Put grey columns together, and green columns together

$$\begin{pmatrix} \text{Grey block} & \text{Green block} \end{pmatrix}$$

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ & & & \vdots & & & \\ 0 & 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 & 0 \\ & & & \vdots & & & \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix}$$

Reorganization

$$\left(\begin{array}{c|c} \text{Gray Box} & \text{Green Box} \end{array} \right) = \left(\begin{array}{c|c} F_{n/2} & \text{Green Box} \\ \hline F_{n/2} & \text{Green Box} \end{array} \right)$$

Reorganization

$$\begin{pmatrix} F_{n/2} & \text{ } \\ F_{n/2} & \text{ } \end{pmatrix} = \begin{pmatrix} F_{n/2} & DF_{n/2} \\ F_{n/2} & -DF_{n/2} \end{pmatrix}$$

$$D = \text{diag}(1, \omega, \omega^2, \dots, \omega^{n/2-1})$$

Fast Fourier Transform

$$F_n = \begin{pmatrix} I & D \\ I & -D \end{pmatrix} \begin{pmatrix} F_{n/2} & 0 \\ 0 & F_{n/2} \end{pmatrix} P$$

$$D = \text{diag}(1, \omega, \omega^2, \dots, \omega^{n/2-1})$$

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ & & & \vdots & & & \\ 0 & 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 & 0 \\ & & & \vdots & & & \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix}$$

Fast Fourier Transform

Evaluate a degree $n-1$ polynomial $A(x) = a_0 + \dots + a_{n-1} x^{n-1}$ at its n^{th} roots of unity: $\omega^0, \omega^1, \dots, \omega^{n-1}$.

Divide. Divide the polynomial into even and odd powers.

$$A(x) = A_{\text{even}}(x^2) + x A_{\text{odd}}(x^2).$$

Conquer. Evaluate $A_{\text{even}}(x)$ and $A_{\text{odd}}(x)$ at the $\frac{1}{2}n^{\text{th}}$ roots of unity: $v^0, v^1, \dots, v^{n/2-1}$.

Combine.

$$A(\omega^k) = A_{\text{even}}(v^k) + \omega^k A_{\text{odd}}(v^k), \quad 0 \leq k < n/2$$

$$A(\omega^{k+n/2}) = A_{\text{even}}(v^k) - \omega^k A_{\text{odd}}(v^k), \quad 0 \leq k < n/2$$

FFT Algorithm

```
FFT(n, a0, a1, ..., an-1) {  
    if (n == 1) return a0  
  
    (e0, e1, ..., en/2-1) ← FFT(n/2, a0, a2, a4, ..., an-2)  
    (d0, d1, ..., dn/2-1) ← FFT(n/2, a1, a3, a5, ..., an-1)  
  
    for k = 0 to n/2 - 1 {  
        ωk ← e2πik/n  
        yk ← ek + ωk dk  
        yk+n/2 ← ek - ωk dk  
    }  
  
    return (y0, y1, ..., yn-1)  
}
```


Summary

- The FFT evaluates a polynomial of degree $n-1$ at n -th roots of unity in $O(n \log n)$ steps.
- Inverse of FFT just as fast.
- Can multiply two polynomials of degree $n-1$ in $O(n \log n)$ time using FFT of length $2n$.
- Find more details in CLRS or Kleinberg/Tardos.