介绍性课程.

---

DFT 离散 Fourier 变换

FFT Fast Fourier Transform.

Cooley - Tukey 1965 Gauss 1805

IBM Princeton.

---

多项式乘法.

$f(x) = a_0 + a_1 x + \cdots + a_d x^d$ $\deg \le d$

$g(x) = b_0 + b_1 x + \cdots + b_d x^d$ $\deg \le d$.

$$f \cdot g = a_0 b_0 + (a_1 b_0 + b_1 a_0) x + \left( \sum_l a_l b_{k-l} \right) x^k + \cdots$$

"复杂度" 乘法 (加法) 次数. 更关注 $c \cdot d^2$ $O(d^2)$

FFT 降到 $O(d \log d)$

$f$ 可由 $x_0, x_1 \cdots x_d$ 的取值唯一确定.

$x_i \neq x_j, \forall i \neq j$

$$\begin{pmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_d) \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^d \\ 1 & x_1 & x_1^2 & \cdots & x_1^d \\ 1 & & & & \\ \vdots & & & & \\ 1 & x_d & x_d^2 & \cdots & x_d^d \end{pmatrix}} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_d \end{pmatrix}$$

矩阵 $M$, 范德蒙矩阵.

$|M| = \prod_{i<j} (x_j - x_i) \neq 0$. $M$ 可逆.

有 $$\begin{pmatrix} a_0 \\ \vdots \\ a_d \end{pmatrix} = M^{-1} \cdot \begin{pmatrix} f(x_0) \\ \vdots \\ f(x_d) \end{pmatrix}$$

$h(x) = f(x) \cdot g(x)$ $\deg \leq 2d$.

$(x_0, \cdots x_{2d})$

$h(x_i) = f(x_i)\, g(x_i)$ $O(d)$ 次运算.

问题: 代入值 (Evaluation)

左乘 $M \cdot \left( \quad \right)_{(2d+1)\times n} \begin{pmatrix} a_0 \\ \vdots \\ a_d \\ 0 \\ \vdots \\ 0 \end{pmatrix}_{(2d+1)\times 2}$

$O(d^2)$ $\qquad$ $O(dn)$

恢复 (recover) n 系数

左乘 $M^{-1} \cdot \left( \quad \right)_{n\times n} \left( \quad \right)$

$O(d^2)$

Improve $\boxed{O(d^2)}$ $\quad 2/n$

取 $x_1 \cdots x_{n/2}, \quad -x_1, \cdots -x_{n/2}$

$f(x)$ even function (偶函数)

$$f(x_i) = f(-x_i)$$

odd function (奇函数)

$$f(x_i) = -f(x_i)$$

$$f(x) = x^4 + 3x^3 + 2x^2 + x + 1$$

$$= \underbrace{(x^4 + 2x^2 + 1)}_{f_e(x^2)} + \underbrace{(3x^3 + x)}_{x \cdot f_o(x^2)}$$

$\deg f = d$

$\deg f_e, \deg f_o \le \frac{d}{2}$

$$f(x_i) = \overbrace{f_e(x_i^2)}^{\frac{n}{2} \cdot \frac{d}{2}} + x_i \overbrace{f_o(x_i^2)}^{\frac{n}{2} \cdot \frac{d}{2}}$$

$$f(-x_i) = f_e(x_i^2) - x_i f_o(x_i^2)$$

$$\left(\frac{1}{2} n d\right)$$

$f_e(x_i^2)$ | $f_o(x_i^2)$
--- | ---
$x_1^2 \cdots x_{\frac{n}{2}}^2$ | $x_1^2 \cdots x_{\frac{n}{2}}^2$

$f(x_i)$

$\pm x_1, \pm x_2 \cdots \pm x_{n/2}$ ±配对点的值

$x_1^2, \ldots x_{\frac{n}{2}}^2$ 不是 ±配对

$f$

$f_o$, $f_e$

引入复数 $\mathbb{C}$

$(f_o)_o$, $(f_o)_e$

$(f_e)_o$, $(f_e)_e$

$1 = x_1$ $\quad$ $-1 = -x_1$ $\quad$ $\sqrt{-1} = i = x_2$ $\quad$ $-\sqrt{-1} = i = -x_2$

$x_1^2 = 1$ $\qquad$ $x_2^2 = -x_1^2 = -1$

$\boxed{x_1^4} = 1.$

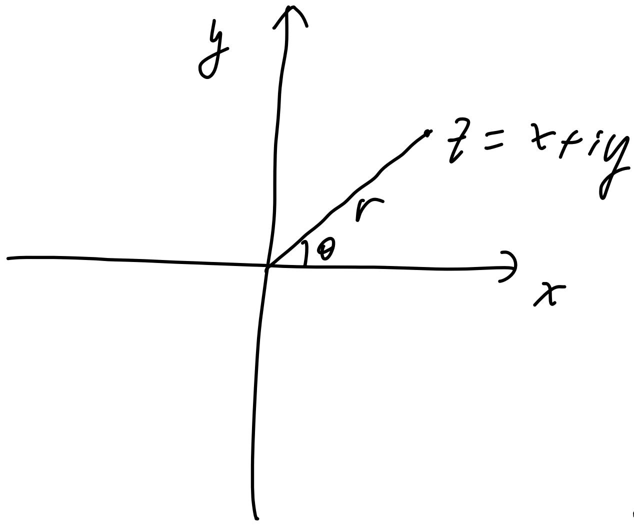
$1 \quad -1 \quad \sqrt{-1} \quad -\sqrt{-1} \quad e^{\frac{2\pi i}{8}}, \quad -e^{\frac{2\pi i}{8}} \quad e^{-\frac{2\pi i}{8}} \quad -e^{\frac{2\pi i}{8}}$

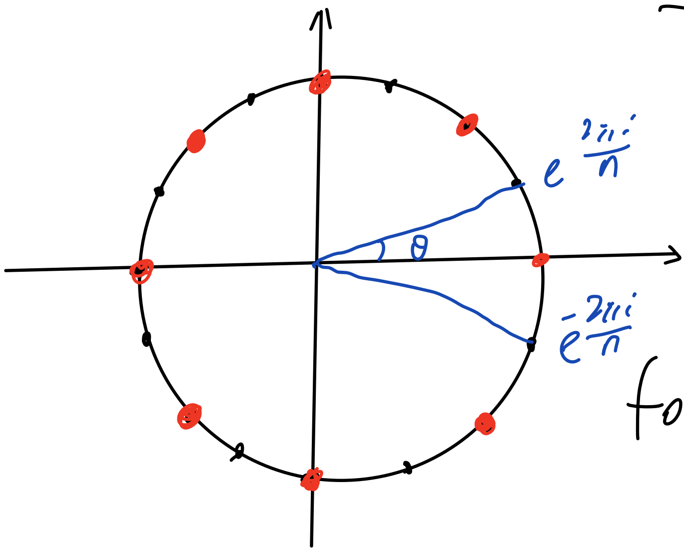
$1 \quad -1 \quad \sqrt{-1} \quad -\sqrt{-1}$

$1 \quad -1$

$1$

复数乘法. $z = r\cdot e^{i\theta} = r(\cos\theta + \sqrt{-1}\sin\theta)$




$\underline{f}$ 16个点

$$\boxed{e^{\frac{2\pi i}{16}\cdot k}}$$

$0 \le k \le 16-1$

$f_o$, $f_e$ 8个点.

归纳算法. $n = 2^k$ $w = e^{-\frac{2\pi i}{n}}$

$$\begin{pmatrix} f(1) \\ f(w) \\ f(w^2) \\ \vdots \\ f(w^{n-1}) \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & \cdots & & 1 \\ 1 & w & \cdots & & w^{n-1} \\ 1 & w^2 & w^4 & w^6 \cdots & w^{2(n-1)} \\ 1 & & & & \\ \vdots & & & & \\ 1 & & & & w^{(n-1)(n-1)} \end{pmatrix}}_{F_n.} \begin{pmatrix} a_0 \\ \vdots \\ \vdots \\ a_{n-1} \end{pmatrix}$$

$F_n$ 左乘 $\begin{pmatrix} a_0 \\ \vdots \\ a_{n-1} \end{pmatrix}$ 有 recursive 算法.

"复杂度" $O(n\log n)$ 相对于直接算法 $O(n^2)$

恢复系数

$F_n^{-1}$ 左乘 $\begin{pmatrix} f(1) \\ \vdots \\ f(w^{n-1}) \end{pmatrix}$ 是否可以简化.

定理: $\underline{(\overline{F_n})^T \cdot F_n = n \cdot I_n}$.

$\overline{F_n}$ 是 $F_n$ 中每个元素作复共轭

证明: $F_n = (V_0, \cdots V_{n-1})$

$$(\overline{F_n})^T \cdot F_n = \begin{pmatrix} \overline{V_0}^T \\ \overline{V_1}^T \\ \vdots \\ \overline{V_{n-1}}^T \end{pmatrix} \begin{pmatrix} V_0 \cdots V_{n-1} \end{pmatrix}$$

$$= \left( \quad \bar{v}_i^T \cdot v_j \quad \right)$$

$$\bar{v}_i^T v_j = \sum_{k=0}^{n-1} (\bar{w}^i)^k \cdot (w^j)^k$$

$$= \sum_{k=0}^{n-1} (w^{-i} \cdot w^j)^k$$

$\bar{w} = w^{-1}$

$$= \sum_{k=0}^{n-1} (w^{j-i})^k$$

$\boxed{|w|^2 = \bar{w} \cdot w = 1}$

① $j = i$, $\bar{v}_i^T \cdot v_j = n$.

② $j \neq i$. 如果 $\underbrace{(j-i, n)}_{\text{最大公约数}} = 1$. $j-1$, $n$ 互素.

$(w^{j-1})^k$, $k = 0, \dots, n-1$, 互不相同.

是 $x^n - 1 = 0$ 所有根.

$$\sum_{k=0}^{n} (w^{j-i})^k = 0.$$

$w^{j-i}$ 是 $x^{\frac{n}{(j,i,n)}} - 1$ 的根.

$$\sum_{k=0}^{n} (w^{j-i})^k = 0.$$

推论: $F_n^{-1} = \frac{1}{n}(\overline{F_n}^T)$

$$F_n^{-1} \cdot \begin{pmatrix} f(1) \\ \vdots \\ f(w^{n-1}) \end{pmatrix} = \frac{1}{n} \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & \overline{w} & (\overline{w})^2 \cdots & (\overline{w})^{n-1} \\ 1 & (\overline{w})^2 & & \\ \vdots & \vdots & & \\ 1 & & \cdots & \end{pmatrix}$$

$\overline{w} = e^{\frac{2\pi i}{n}}$ Recursive 归纳算法

$O(n \log n)$

---

矩阵角度.

$$F_{2n} \begin{pmatrix} a_0 \\ \vdots \\ a_{2n-1} \end{pmatrix} = \underbrace{\begin{pmatrix} I_n & D_n \\ I_n & -D_n \end{pmatrix}}_{O(n)} \begin{bmatrix} F_n & \\ & F_n \end{bmatrix} \begin{pmatrix} f_{even} \\ f_{odd} \end{pmatrix}$$

$$D_n = \begin{pmatrix} 1 & & & \\ & w_{2n} & & \\ & & \ddots & \\ & & & w_{2n}^{n-1} \end{pmatrix} \qquad \boxed{w_{2n} = e^{-\frac{2\pi i}{2n}}}$$

置换矩阵

$$F_{2n} = \underbrace{\begin{pmatrix} I_n & D_n \\ I_n & -D_n \end{pmatrix}}_{O(n)} \underbrace{\begin{pmatrix} F_n & \\ & F_n \end{pmatrix}}_{\text{归纳}} \underbrace{\begin{pmatrix} P_{2n} \end{pmatrix}}$$

$\cancel{2\,O(\frac{n}{2})}$

$\underline{O(n)}$

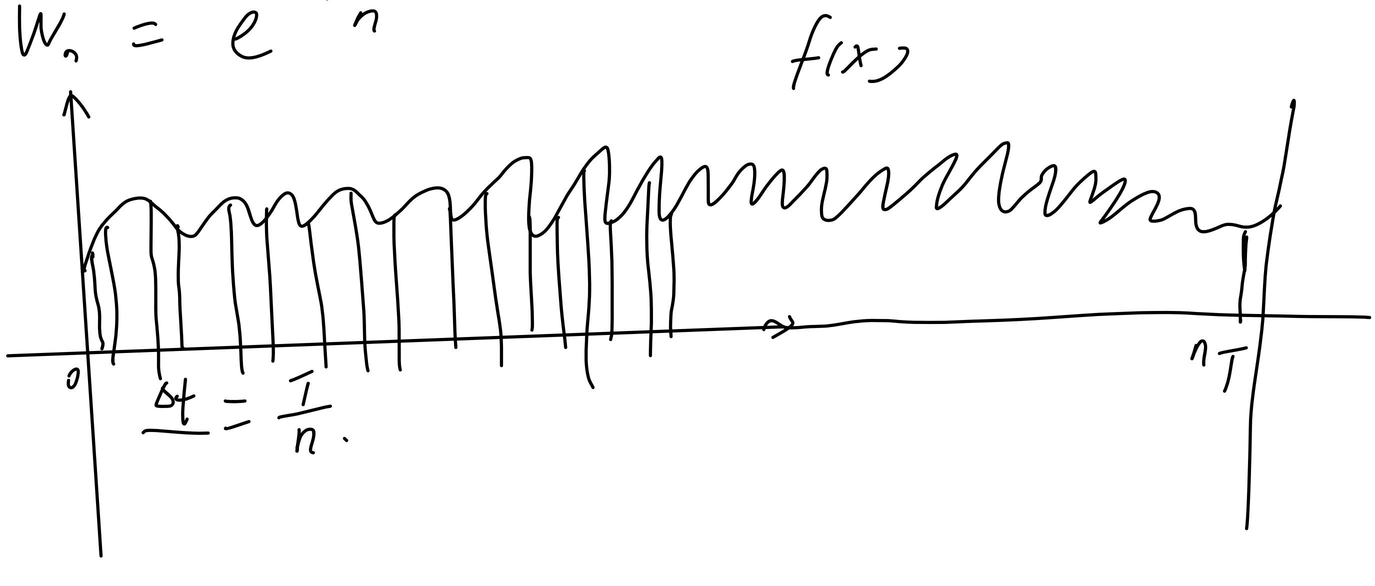
$\underline{O(n\log n)}$

---

## 通常 DFT.

Discrete Fourier Transform

$$w_n = e^{-\frac{2\pi i}{n}}$$



取样 $f(x_k)$, $x_k = \frac{k \cdot T}{n}$

$k = 0, \cdots, n-1.$

或者,

$\begin{pmatrix} f_0 \\ \vdots \\ f_{n-1} \end{pmatrix} \in \mathbb{R}^n, \ (\mathbb{C}^n)$

$\mathrm{DFT}: \mathbb{C}^n \to \mathbb{C}^n$

$$\begin{pmatrix} f_0 \\ \vdots \\ f_{n-1} \end{pmatrix} \mapsto F_n \cdot \begin{pmatrix} f_0 \\ \vdots \\ f_{n-1} \end{pmatrix} = \begin{pmatrix} \hat{f}_0 \\ \vdots \\ \hat{f}_{n-1} \end{pmatrix}$$

Inverse. $\mathrm{iDFT}: \mathbb{C}^n \to \mathbb{C}^n$

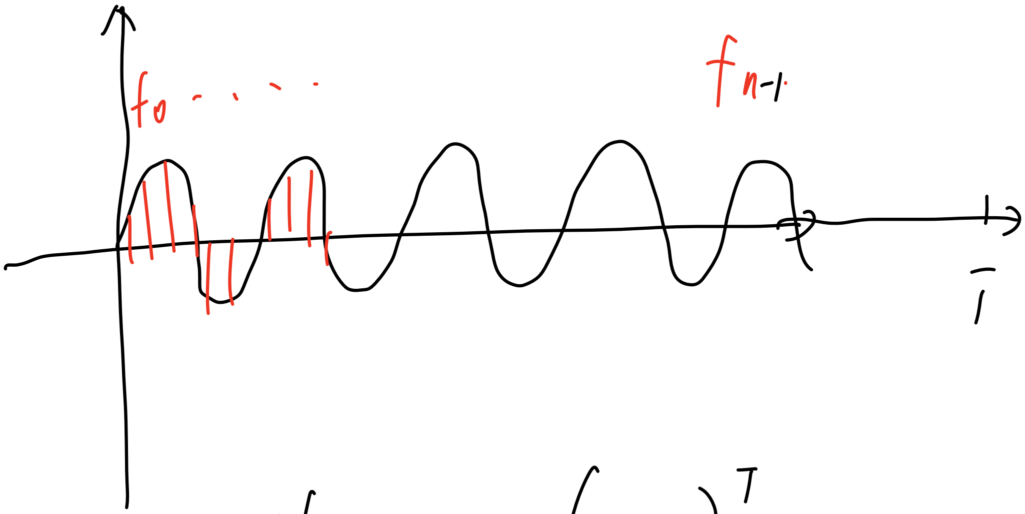
$$\begin{pmatrix} \hat{f}_0 \\ \vdots \\ \hat{f}_{n-1} \end{pmatrix} \mapsto \underline{(F_n^{-1})} \begin{pmatrix} \hat{f}_0 \\ \vdots \\ \hat{f}_n \end{pmatrix}$$

$\downarrow$

$\frac{1}{n} (\overline{F_n})^T$

$f = \sin(2\pi a t)$  $a$ 频率.

$= \frac{e^{2\pi i a t} - e^{-2\pi i a t}}{2i}.$




DFT $(f_0 \cdots f_{n-1})^T$

研究 $e^{2\pi i a t} = f(x)$

$$f_k = e^{2\pi i a \left(k \frac{T}{n}\right)}$$

$$\mathrm{DFT}(f_0, \cdots f_{n-1})^T$$

$$= (\hat{f}_0 \cdots \hat{f}_{n-1})$$

$$\hat{f}_k = \sum_{l=0}^{n-1} (w^k)^l \cdot e^{2\pi i \left(\frac{aT}{n} \cdot l\right)}$$

$$= \sum_{l=0}^{n-1} e^{2\pi i \left(\frac{aT}{n} - \frac{k}{n}\right) \cdot l}$$

$$= \sum_{l=0}^{n-1} e^{2\pi i \frac{(aT-k)l}{n}} = \begin{cases} n & k \equiv aT \pmod{n} \\ 0 & k \not\equiv aT \pmod{n} \end{cases}$$

$\boxed{aT\ 整数}$

$|\hat{f}_k|$ 大的 $k$.

$$\frac{k}{T} = a$$

$a > 0$

$f(t) = e^{-2\pi i a t}$

$k = n - aT$

如果 $f(t) = \overline{f(t)}$,

$\hat{f}_0 \cdots \hat{f}_{n-1}$ 有"对称性"

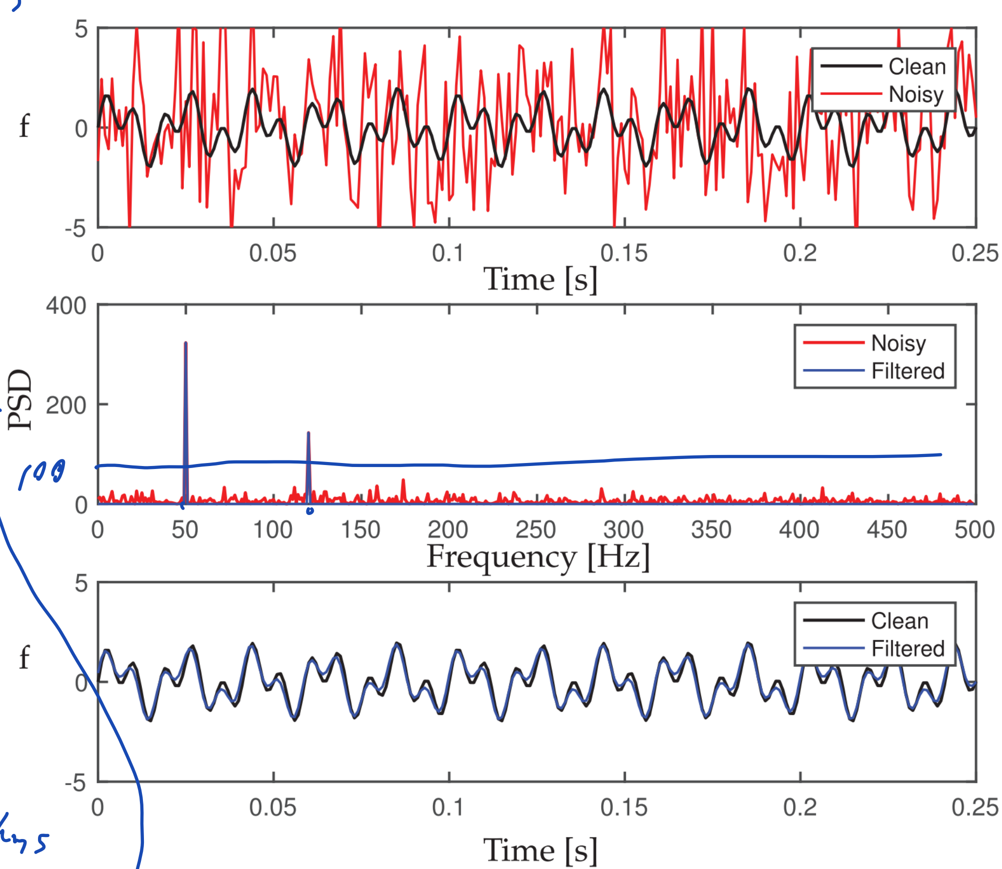
$$\hat{f}_k = \overline{\hat{f}_{n-k}}$$

$$f = \sin(2\pi f_1 t) + \sin(2\pi f_2 t)$$

$$f_1 = 50, \quad f_2 = 120$$

$$f + \text{Noise}, \quad T = 1, \quad n = 1000$$

This part is from Page 61 Data-Driven Science and Engineering Machine Learning Dynamical systems and Control by S.L. Brunton and J.N. Kutz




$\frac{k}{T}$ $s^{-1}$

$$PSD = \frac{|\hat{f}_k|^2}{n}$$

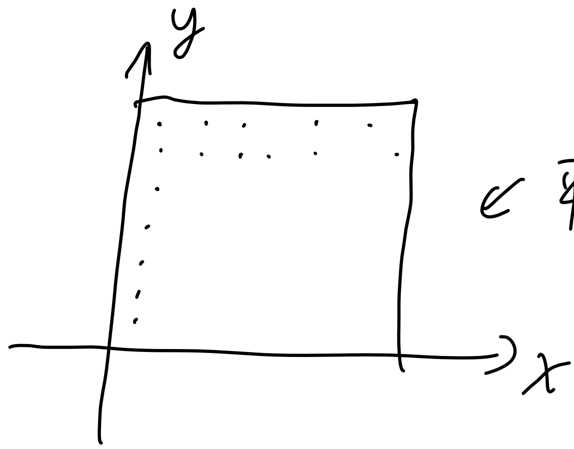
← f 不加 noise 有非0值 $|\hat{f}_k|^2$ 在 50, 120, 1000-50, 1000-120

$$\text{filtered } f = iDFT(\hat{f}_k \cdot \{PSD > 100\})$$

示性函数

① Denoise

② Compress




← 平面图像

$(x_k, y_l)$ 对 $f(x_k, y_l)$ 两次 DFT.

去掉 $|\hat{f}(k, l)|$ 小的值.