



第二讲 特殊矩阵 (2)

李宇峰

liyf@nju.edu.cn

人工智能学院



7. 范德蒙矩阵与傅里叶矩阵

范德蒙矩阵的形式是：

$$\mathbf{A} = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{N-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{N-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^{N-1} \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_n \\ x_1^2 & x_2^2 & \cdots & x_n^2 \\ \vdots & \vdots & & \vdots \\ x_1^{N-1} & x_2^{N-1} & \cdots & x_n^{N-1} \end{bmatrix}$$

每行或每列元素组成一个等比序列的特殊矩阵。显而易见， \mathbf{A}^T 也是范德蒙矩阵。

7. 范德蒙矩阵与傅里叶矩阵

动机：很多时候需要对函数进行多项式近似表达，例如泰勒展式

泰勒展式

- **定义与公式**：泰勒展式是用一个函数在某一点的各阶导数来表示该函数在这一点附近的近似表达式。对于一个具有 n 阶导数的函数 $f(x)$ ，在 $x = a$ 处的泰勒展开式为

$$f(x) = f(a) + f'(a)(x - a) + \frac{f''(a)}{2!}(x - a)^2 + \cdots + \frac{f^{(n)}(a)}{n!}(x - a)^n + R_n(x)$$
，其中 $R_n(x)$ 是余项，表示泰勒展开式与原函数之间的误差。

- **在近似非线性函数中的作用**：泰勒展式是一种局部近似方法，它能够将一个复杂的非线性函数在某一点附近用一个多项式来近似表示。通过取不同阶数的泰勒展开式，可以得到不同精度的近似。一般来说，展开式的阶数越高，在展开点附近对原函数的近似效果就越好。例如，对于函数 $f(x) = e^x$ ，在 $x = 0$ 处的一阶泰勒展开式为 $e^x \approx 1 + x$ ，这在 $x = 0$ 附近对 e^x 有一定的近似效果；而取二阶泰勒展开式 $e^x \approx 1 + x + \frac{x^2}{2}$ ，则在 $x = 0$ 附近的近似精度更高。泰勒展式在分析非线性函数的局部性质、求解函数的极限、导数以及数值计算等方面都有广泛的应用。在数值计算中，常常利用泰勒展式来对非线性函数进行线性化或多项式近似，以便于进行计算和分析。同时，在物理、工程等领域，也经常用泰勒展式来近似复杂的非线性模型，从而简化问题的求解过程。

具体运用的时候一是仅能局部近似，二是依赖函数导数

7. 范德蒙矩阵与傅里叶矩阵

范德蒙矩阵应运而生，旨在为多项式近似表达提供便利

插值计算

- **构建插值多项式**：在数值分析的插值问题中，给定 $n + 1$ 个不同的节点 x_0, x_1, \dots, x_n 以及对应的函数值 y_0, y_1, \dots, y_n ，要构造一个 n 次插值多项式 $P(x) = a_0 + a_1x + \dots + a_nx^n$ 使得 $P(x_i) = y_i$, $i = 0, 1, \dots, n$ 。可以将其转化为线性方程组 $Va = y$ ，其中 V 是范德蒙矩阵， $a = (a_0, a_1, \dots, a_n)^T$ ， $y = (y_0, y_1, \dots, y_n)^T$ 。通过求解这个线性方程组，就能得到插值多项式的系数，从而实现插值计算。
- **误差估计**：利用范德蒙矩阵的条件数可以对插值误差进行估计。条件数反映了矩阵对数据扰动的敏感度，条件数越大，数据的微小变化可能导致解的较大变化，插值误差可能越大。

其他应用

- **信号处理**：在信号的频谱分析等领域，范德蒙矩阵可用于构建信号模型，通过对信号在不同频率点上的采样值，利用范德蒙矩阵求解信号的频谱系数，实现对信号频谱的分析。
- **编码理论**：在纠错编码中，范德蒙矩阵可用于构造某些线性码，利用其性质来实现对信息的编码和纠错，提高通信的可靠性。

范德蒙矩阵

在谐波恢复的扩展Prony方法中，信号模型假定是一组 p 个指数函数的叠加，这组指数函数有任意的幅值、相位、频率和阻尼因子。于是离散时间的数学模型是

$$\widehat{x}_n = \sum_{i=0}^p b_i z_i^n, \quad n = 0, 1, 2, \dots, N-1 \quad (2.1)$$

被用作拟合观测数据 x_1, x_2, \dots, x_{N-1} 的数学模型。

通常假定 b_i 和 z_i 为复数，并且 $b_i = A_i \exp(j\theta_i)$ ， $z_i = \exp[(\alpha_i + j2\pi f_i)\Delta t]$ ，其中 A_i 是幅值、 θ_i 是相位(rad)、 α_i 为阻尼因子、 f_i 为振荡频率(Hz)、 Δt 代表采样间隔(s)。

范德蒙矩阵

(2.1)式的矩阵形式是 $\Phi \mathbf{b} = \hat{\mathbf{x}}$ ，这里 $\mathbf{b} = (b_1, b_2, \dots, b_p)^T$ ， $\hat{\mathbf{x}} = (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_{N-1})^T$ ，而 Φ 是范德蒙矩阵

$$\Phi = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ z_1 & z_2 & z_3 & \dots & z_p \\ z_1^2 & z_2^2 & z_3^2 & \dots & z_p^2 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ z_1^{N-1} & z_2^{N-1} & z_3^{N-1} & \dots & z_p^{N-1} \end{bmatrix}$$

上式使平方误差 $\varepsilon = \sum_{n=0}^{N-1} |x_n - \hat{x}_n|^2$ 最小，便得到最小二乘解

$$\mathbf{b} = (\Phi^H \Phi)^{-1} \Phi^H \hat{\mathbf{x}}$$

完成对信号进行有效的多项式近似。

范德蒙矩阵

值得一提的是，上式中的 $\Phi^H \Phi$ 的计算存在简化方式，无须作范德蒙矩阵的乘法运算，就能直接利用

$$\Phi^H \Phi = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \cdots & \gamma_{1p} \\ \gamma_{21} & \gamma_{22} & \cdots & \gamma_{2p} \\ \vdots & \vdots & & \vdots \\ \gamma_{p1} & \gamma_{p2} & \cdots & \gamma_{pp} \end{bmatrix}$$

计算出 $\Phi^H \Phi$ ，其中

$$\gamma_{ij} = \sum_{k=0}^{N-1} (z_i^T z_j)^k = \frac{(z_i^T z_j)^N - 1}{z_i^T z_j - 1}$$

应用: 里德-所罗门码

In the original construction of Reed & Solomon(1960), 利用多项式系数对信号进行加密和解密。
the message $x = (x_1, \dots, x_k) \in F^k$ is mapped to the polynomial p_x with

$$p_x(a) = \sum_{i=1}^k x_i a^{i-1}.$$

As described above, the codeword is then obtained by evaluating p at n different points a_1, \dots, a_n of the field F . Thus the classical encoding function $C: F^k \rightarrow F^n$ for the Reed-Solomon code is defined as follows:

$$C(x) = (p_x(a_1), \dots, p_x(a_n)).$$

This function C is a linear mapping, that is, it satisfies $C(x) = x \cdot A$ for the following $(k \times n)$ -matrix A with elements from F :

$$A = \begin{bmatrix} 1 & \dots & 1 \\ a_1 & \dots & a_n \\ a_1^2 & \dots & a_n^2 \\ \vdots & \ddots & \vdots \\ a_n^{k-1} & \dots & a_n^{k-1} \end{bmatrix}$$

This matrix is the transpose of a Vandermonde matrix over F . In other words, the Reed-Solomon code is a linear code, and in the classical encoding procedure, its generator matrix is A .

应用: 里德-所罗门码

- 里德-所罗门码广泛应用于媒体数据存储，如CD和DVD
- 里德-所罗门码几乎被应用于所有二维条形码中，如二维码。

范德蒙矩阵的行列式

突出性质： A 的 n 个参数 x_1, x_2, \dots, x_n 不同，范德蒙矩阵行满秩。

范德蒙矩阵的行列式有规律，从阶为2, 3开始总结

$$A = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{N-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{N-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^{N-1} \end{bmatrix}$$

$$\det A_2 = \begin{vmatrix} 1 & x_1 \\ 1 & x_2 \end{vmatrix} = x_2 - x_1$$

$$\begin{aligned} \det A_3 &= \begin{vmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \end{vmatrix} \\ &= \begin{vmatrix} 1 & x_1 & x_1^2 \\ 0 & x_2 - x_1 & x_2^2 - x_1^2 \\ 0 & x_3 - x_1 & x_3^2 - x_1^2 \end{vmatrix} \\ &= \begin{vmatrix} x_2 - x_1 & x_2^2 - x_1^2 \\ x_3 - x_1 & x_3^2 - x_1^2 \end{vmatrix} \\ &= (x_2 - x_1)(x_3 - x_2)(x_3 - x_1). \end{aligned}$$

大胆猜测：

$$\det A_n = \prod_{1 \leq j < i \leq n} (x_i - x_j)$$

数学归纳法

范德蒙矩阵的行列式

证明：假设 k 阶范德蒙行列式为

$$\det A_k = \prod_{1 \leq j < i \leq k} (x_i - x_j)$$

我们要证明 $k + 1$ 阶范德蒙行列式也有相同的形式。将 A_{k+1} 的最末列替换为 $1, x, x^2, \dots, x^k$ ，

设此矩阵的行列式为 $f(x)$ ，亦即

$$f(x) = \begin{vmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^k \\ 1 & x_2 & x_2^2 & \cdots & x_2^k \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_k & x_k^2 & \cdots & x_k^k \\ 1 & x & x^2 & \cdots & x^k \end{vmatrix}$$

由最末列的余因子展开式可知 f 为变数 x 的 k 次多项式。若矩阵有相同的两列，其行列式等于零，故

$f(x_i) = 0$ ， $i = 1, 2, \dots, k$ ，也就是说 x_i ， $i = 1, 2, \dots, k$ ，为多项式 $f(x)$ 的 k 个根， $f(x)$ 可表示为

$$f(x) = \alpha(x - x_1)(x - x_2) \cdots (x - x_k)$$

上式中 α 为非零常数，剩下的问题是决定 α 。

范德蒙矩阵的行列式

由 $f(x)$ 的余因子展开式可得到 x^k 的系数，也就是 α 的值，

$$\begin{vmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{k-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{k-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_k & x_k^2 & \cdots & x_k^{k-1} \end{vmatrix}.$$

注意，上面的行列式即为 $\det A_k$ 。根据归纳法的假设，就有

$$f(x) = \left[\prod_{1 \leq j < i \leq k} (x_i - x_j) \right] (x - x_1)(x - x_2) \cdots (x - x_k).$$

还有一个不能错过的事实： $f(x_{k+1}) = \det A_{k+1}$ 。将 $x = x_{k+1}$ 代入上式，

$$\det A_{k+1} = \left[\prod_{1 \leq j < i \leq k} (x_i - x_j) \right] (x_{k+1} - x_1)(x_{k+1} - x_2) \cdots (x_{k+1} - x_k).$$

范德蒙矩阵的行列式

整理等号右端，得到

$$\det A_{k+1} = \prod_{1 \leq j < i \leq k+1} (x_i - x_j),$$

故证明所求。

直接推论： A 的 n 个参数 x_1, x_2, \dots, x_n 不同，范德蒙矩阵行满秩。

范德蒙矩阵

最小二乘解 $\mathbf{b} = (\Phi^H \Phi)^{-1} \Phi^H \hat{\mathbf{x}}$ 计算多项式系数一般需要 $O(n^3)$ 复杂度，
有没有高效算法？

通过数学性质，可以存在高效计算方式，介绍一种 $O(5n^2)$ 计算方法

范德蒙矩阵

4.6.1 多项式插值: $V^T a = f$

在许多近似和插值问题中都会遇到范德蒙德方程组. 实际上, 快速求解范德蒙德方程组的关键在于要认识到解 $V^T a = f$ 等价于多项式插值. 这是因为, 如果 $V^T a = f$ 且

$$p(x) = \sum_{j=0}^n a_j x^j, \quad (4.6.1)$$

那么对于 $i = 0:n$ 有 $p(x_i) = f_i$.

回忆一下, 如果 x_i 互异, 那么对插值点 $(x_0, f_0), \dots, (x_n, f_n)$ 可构造唯一的 n 次多项式. 因此, 只要 x_i 互异, V 就是非奇异矩阵. 我们假定这个条件在本节恒成立.

计算 (4.6.1) 中 a_j 的第一步是计算插值多项式 p 的牛顿表达式:

$$p(x) = \sum_{k=0}^n c_k \left(\prod_{i=0}^{k-1} (x - x_i) \right). \quad (4.6.2)$$

常数 c_k 是均差, 可如下确定:

范德蒙矩阵

```
c(0:n) = f(0:n)
for k = 0 : n-1
    for i = n : -1 : k+1
        c_i = (c_i - c_{i-1}) / (x_i - x_{i-k-1})
    end
end
end
```

(4.6.3)

参见 Conte and deBoor (1980).

接下来由牛顿表达式的系数 c_0, \dots, c_n 生成 (4.6.1) 的系数 a_0, \dots, a_n . 定义多项式 $p_n(x), \dots, p_0(x)$ 如下:

```
p_n(x) = c_n
for k = n-1 : -1 : 0
    p_k(x) = c_k + (x - x_k)p_{k+1}(x)
end
```

范德蒙矩阵

观察到 $p_0(x) = p(x)$, 可写出

$$p_k(x) = a_k^{(k)} + a_{k+1}^{(k)}x + \cdots + a_n^{(k)}x^{n-k},$$

令方程 $p_k = c_k + (x - x_k)p_{k+1}$ 中 x 的同次幂相等, 可得求系数 $a_i^{(k)}$ 的递推关系式:

$$a_n^{(n)} = c_n$$

for $k = n-1 : -1 : 0$

$$a_k^{(k)} = c_k - x_k a_{k+1}^{(k+1)}$$

for $i = k+1 : n-1$

$$a_i^{(k)} = a_i^{(k+1)} - x_k a_{i+1}^{(k+1)}$$

end

$$a_n^{(k)} = a_n^{(k+1)}$$

end

所以系数 $a_i = a_i^{(0)}$ 可如下计算:

$$a(0:n) = c(0:n)$$

for $k = n-1 : -1 : 0$

for $i = k : n-1$

$$a_i = a_i - x_k a_{i+1}$$

end

end

范德蒙矩阵

算法 4.6.1 给定 $x(0:n) \in \mathbb{R}^{n+1}$, 其元素互异, $f = f(0:n) \in \mathbb{R}^{n+1}$, 算法求出范德蒙德方程组 $V(x_0, \dots, x_n)^T \mathbf{a} = \mathbf{f}$ 的解 $\mathbf{a} = \mathbf{a}(0:n)$, 其覆盖 \mathbf{f} .

```
for k = 0 : n - 1
    for i = n : -1 : k + 1
        f(i) = (f(i) - f(i - 1)) / (x(i) - x(i - k - 1))
    end
end
for k = n - 1 : -1 : 0
    for i = k : n - 1
        f(i) = f(i) - f(i + 1) * x(k)
    end
end
```

这个算法需要 $5n^2/2$ 个 flop.

范德蒙矩阵

范德蒙矩阵

- 为复杂函数的多项式近似表达提供便利
- 很容易满秩，计算便利，行列式有规律
- 范德蒙方程可利用数学性质提高计算效率
- 应用广泛，如信号处理恢复，加密解密等

傅里叶矩阵

有没有办法进一步加速计算呢？答案是傅里叶矩阵，是一类特殊的范德蒙矩阵，除了矩阵计算，在图像压缩等领域中都有不可忽视的重要性

离散时间信号 x_0, x_1, \dots, x_{N-1} 的 Fourier 变换称为信号的离散 Fourier 变换 (DFT) 或频谱，定义为

$$X_k = \sum_{n=0}^{N-1} x_n e^{-j2\pi nk/N} = \sum_{n=0}^{N-1} x_n w^{nk}, \quad k = 0, 1, \dots, N-1 \quad (2.4.11)$$

式中， $w = e^{-j2\pi/N}$ 。离散 Fourier 变换的矩阵表示形式为

$$\begin{bmatrix} X_0 \\ X_1 \\ \vdots \\ X_{N-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & w & \dots & w^{N-1} \\ \vdots & \vdots & & \vdots \\ 1 & w^{N-1} & \dots & w^{(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{bmatrix} \quad (2.4.12)$$

或简记作

$$\hat{x} = Fx \quad (2.4.13)$$

傅里叶矩阵

$$F = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & \omega & \cdots & \omega^{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \cdots & \omega^{(N-1)(N-1)} \end{bmatrix}$$

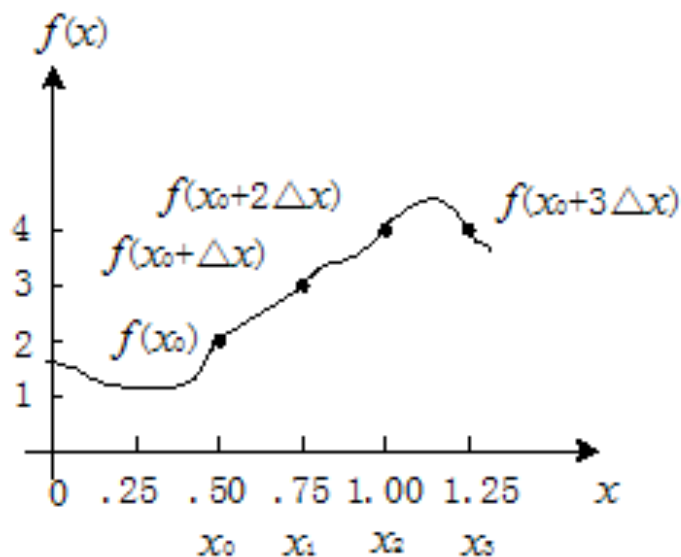
称为傅里叶矩阵，其 (i, k) 元素为 $F(i, k) = \omega^{(i-1)(k-1)}$ 。

显然，傅里叶矩阵是范德蒙矩阵，具有特殊结构：每一行和每一列的元素都分别组成各自的等比数列。

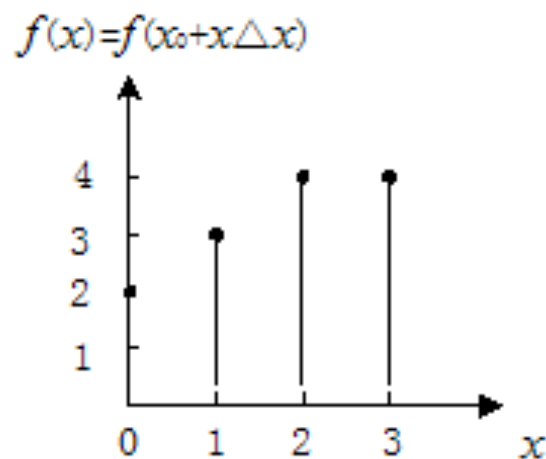
傅里叶矩阵

例，计算傅里叶变换 设一函数如图 (a) 所示，如果将此函数在自变量 $x_0 = 0.5, x_1 = 0.75, x_2 = 1.00, x_3 = 1.25$ 取样

并重新定义为图 (b) 离散函数，求其傅里叶变换。



(a)



(b)

$$F(0) = \frac{1}{4} \sum_{x=0}^3 f(x) \exp\{0\}$$

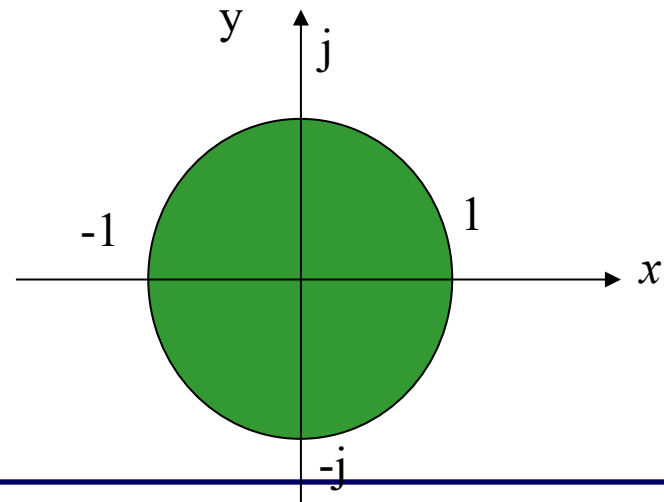
$$= (1/4)[f(0) + f(1) + f(2) + f(3)] = (1/4)[2 + 3 + 4 + 4] = 3.25$$

$$F(1) = \frac{1}{4} \sum_{x=0}^3 f(x) \exp\{-j2\pi x/4\} = \frac{1}{4} [2e^0 + 3e^{-j\pi/2} + 4e^{-j\pi} + 4e^{-j3\pi/2}] = \frac{1}{4} [-2 + j]$$

$$F(2) = \frac{1}{4} \sum_{x=0}^3 f(x) \exp\{-j4\pi x/N\} = \frac{1}{4} [2e^0 + 3e^{-j\pi} + 4e^{-j2\pi} + 4e^{-j3\pi}] = -\frac{1}{4} [1 + j0]$$

$$F(3) = \frac{1}{4} \sum_{x=0}^3 f(x) \exp\{-j6\pi x/4\} = \frac{1}{4} [2e^0 + 3e^{-j3\pi/2} + 4e^{-j3\pi} + 4e^{-j9\pi/2}] = -\frac{1}{4} [2 + j]$$

$$F(u) = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \end{bmatrix}$$



傅里叶矩阵

$\bar{x} = Fx$ 可以看出，一个离散时间信号向量的离散傅里叶变换可以用矩阵 F 表示，因此称 F 为傅里叶矩阵。

根据定义可以验证 $FF^H = F^H F = NI_n$ 。傅里叶矩阵是一个范德蒙矩阵，它是非奇异的，是**正交阵**，于是由 $F^H F = NI_n$ 可知傅里叶矩阵的逆矩阵是

$$F^{-1} = \frac{1}{N} F^H = \frac{1}{N} F^*,$$

傅里叶矩阵

因此 $x = F^{-1}\bar{x} = \frac{1}{N}F^*\bar{x}$

$$\text{或写成} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{bmatrix} = \frac{1}{N} \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & \omega^* & \dots & (\omega^{N-1})^* \\ \vdots & \vdots & \dots & \vdots \\ 1 & (\omega^{N-1})^* & \dots & (\omega^{(N-1)(N-1)})^* \end{bmatrix} \begin{bmatrix} X_0 \\ X_1 \\ \vdots \\ X_{N-1} \end{bmatrix}$$

这就是离散傅里叶逆变换。

回顾一下范德蒙方程的解：最小二乘解 $\mathbf{b} = (\mathbf{F}^H \mathbf{F})^{-1} \mathbf{F}^H \hat{\mathbf{x}}$ 计算多项式系数一般需要 $O(n^3)$ 复杂度。现在因为 \mathbf{F} 是正交矩阵， $\mathbf{b} = \mathbf{F}^H \hat{\mathbf{x}}$ ，计算大为简化

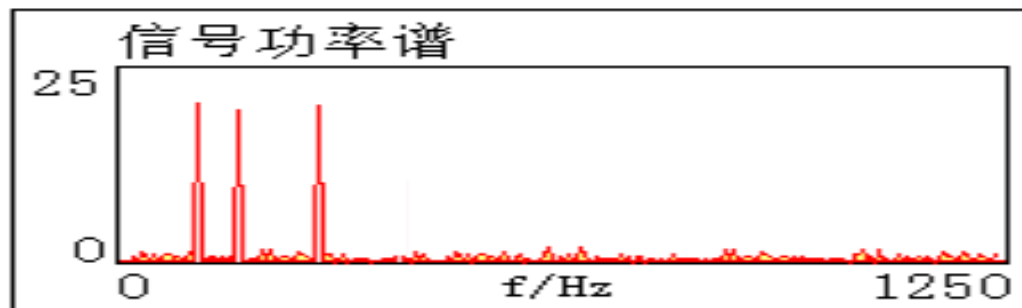
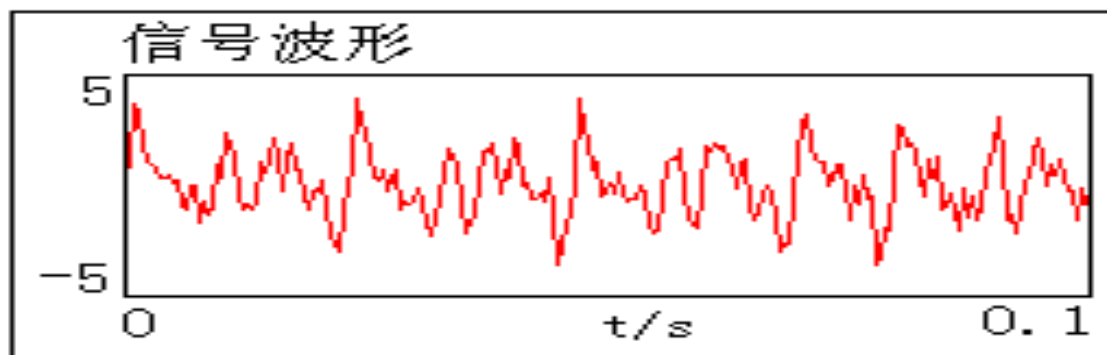
傅里叶矩阵在信号处理做出了巨大贡献

- **构建信号处理理论基础：** 傅里叶矩阵与傅里叶变换紧密相连，傅里叶变换是信号处理领域的核心理论之一。傅里叶矩阵是离散傅里叶变换（DFT）的矩阵表示形式，**通过傅里叶矩阵可以将离散信号从时域转换到频域**。这种转换使得我们能够从频域的角度分析信号的频率成分，理解信号的本质特征，为信号处理的理论研究提供了重要的工具和视角。
- **促进线性代数理论发展：** 傅里叶矩阵具有许多独特的数学性质，如正交性等。研究傅里叶矩阵的这些性质可以加深对线性代数中正交矩阵、特征值、特征向量等概念的理解。同时，傅里叶矩阵在矩阵分解、矩阵运算等方面也有特殊的应用，有助于推动矩阵理论的进一步发展。
- **图像处理领域**
 - **图像滤波：** 傅里叶矩阵可以将图像从空间域转换到频域，在频域中对图像的频率成分进行处理，如低通滤波、高通滤波等。低通滤波可以去除图像中的高频噪声，使图像变得平滑；高通滤波则可以增强图像的边缘和细节信息。
 - **图像压缩：** 基于傅里叶变换的图像压缩算法，如 JPEG 压缩标准，利用傅里叶矩阵将图像转换到频域后，根据人类视觉系统的特性，对图像的高频部分进行量化和编码，从而实现图像的压缩。这种压缩方法可以在保证一定图像质量前提下，大大减少图像的数据量，便于存储和传输。

傅里叶矩阵

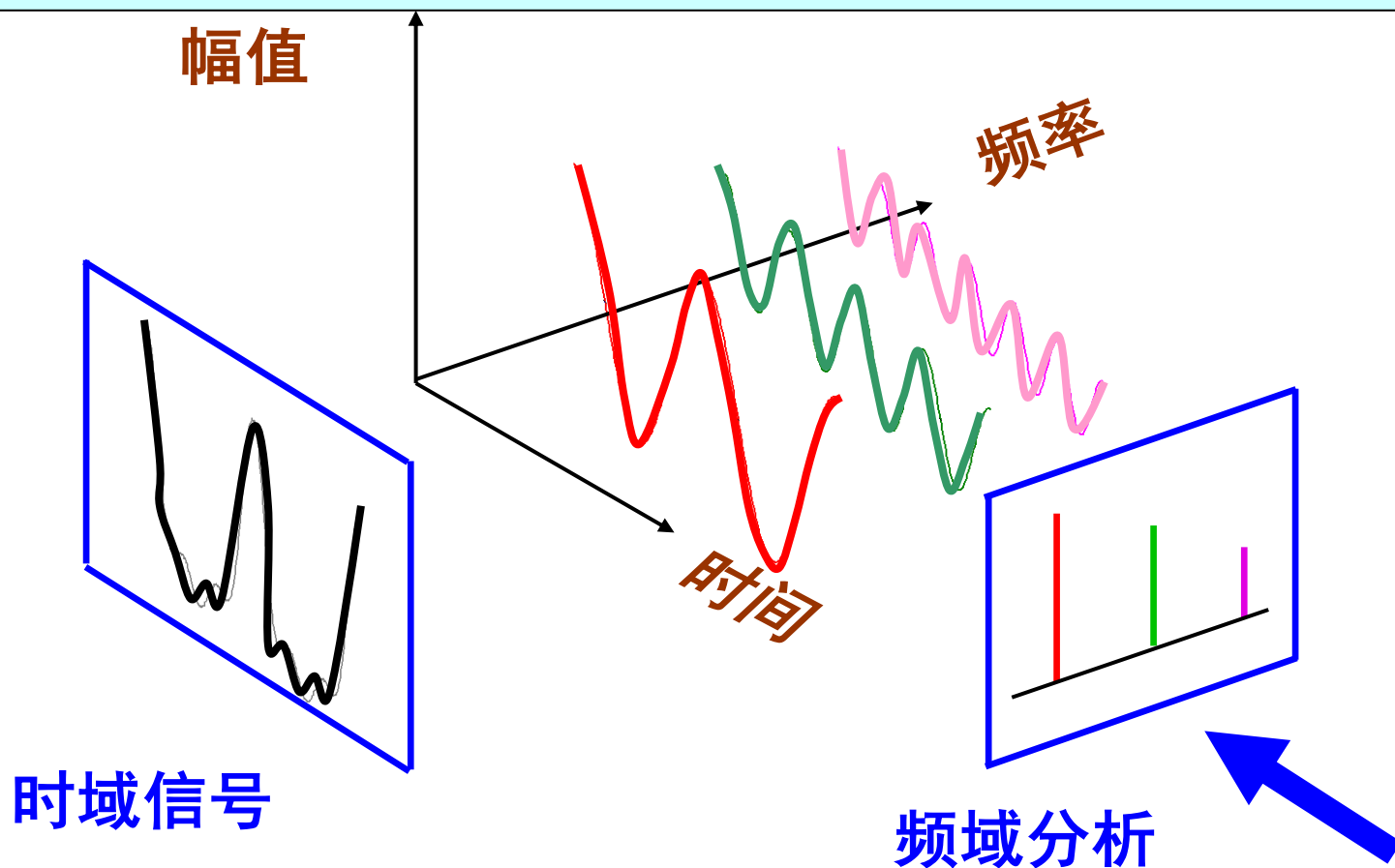
图像是连续信号的量化采样，信号包括丰富的频域信息

图例：受噪声干扰的多频率成分信号



傅里叶矩阵

信号频谱代表了信号在不同频率分量成分的大小，能够提供比时域信号波形更直观，丰富的信息。

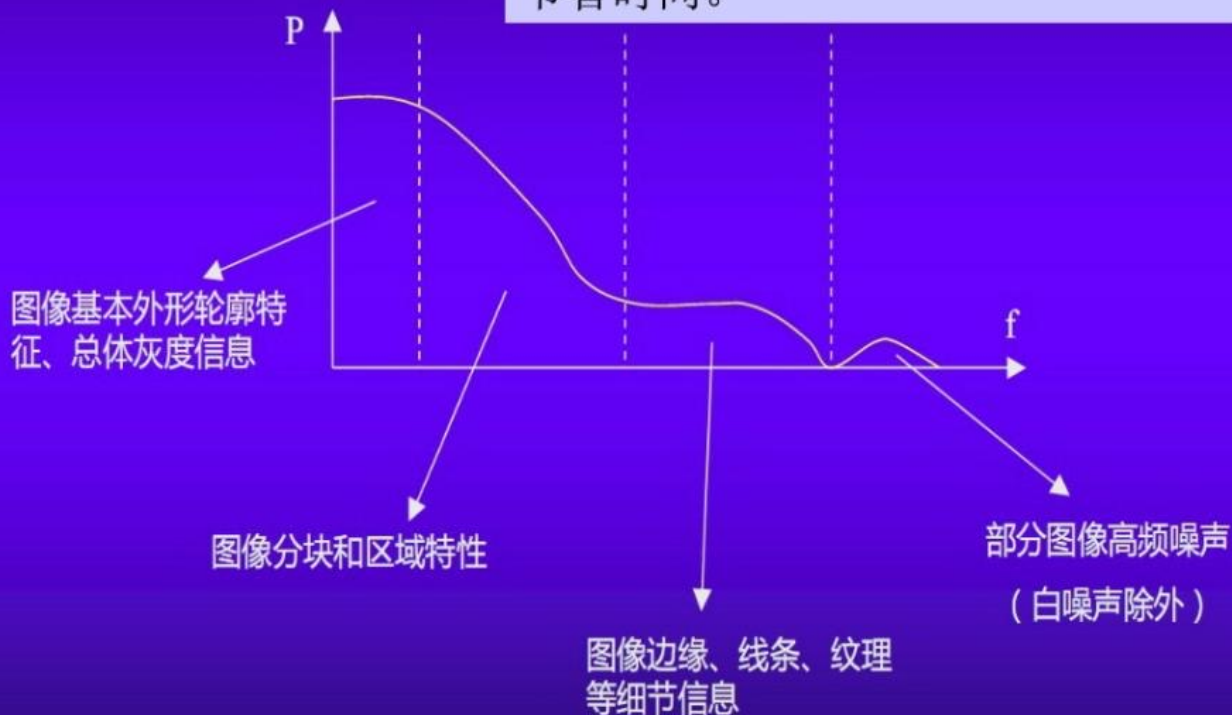


傅里叶矩阵

为什么要做图像变换？

图像信号的频域模型

❖ 变换后的图像，大部分能量都分布于低频谱段，这对以后图像的压缩、传输都比较有利。使得运算次数减少，节省时间。

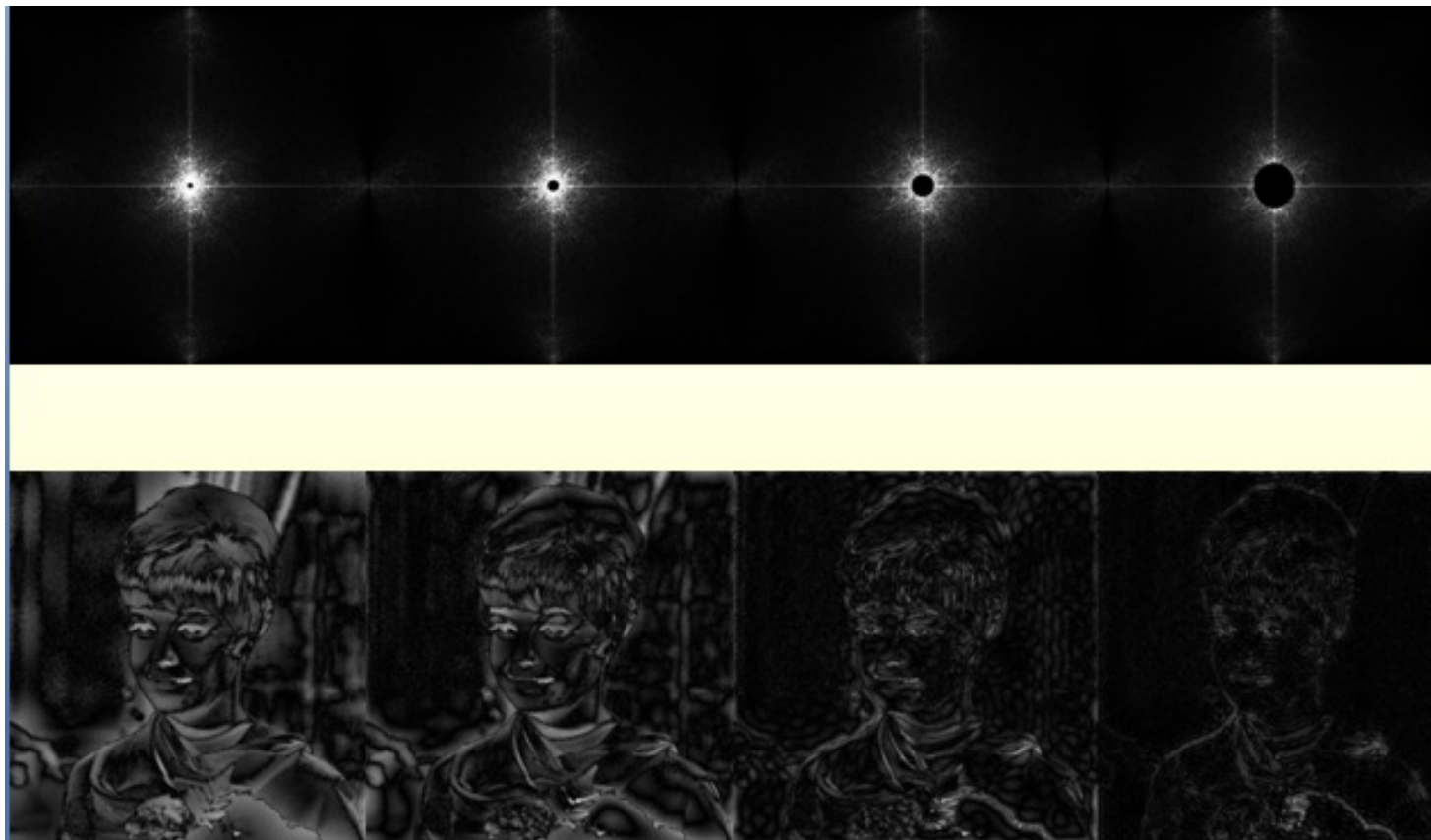


傅里叶矩阵



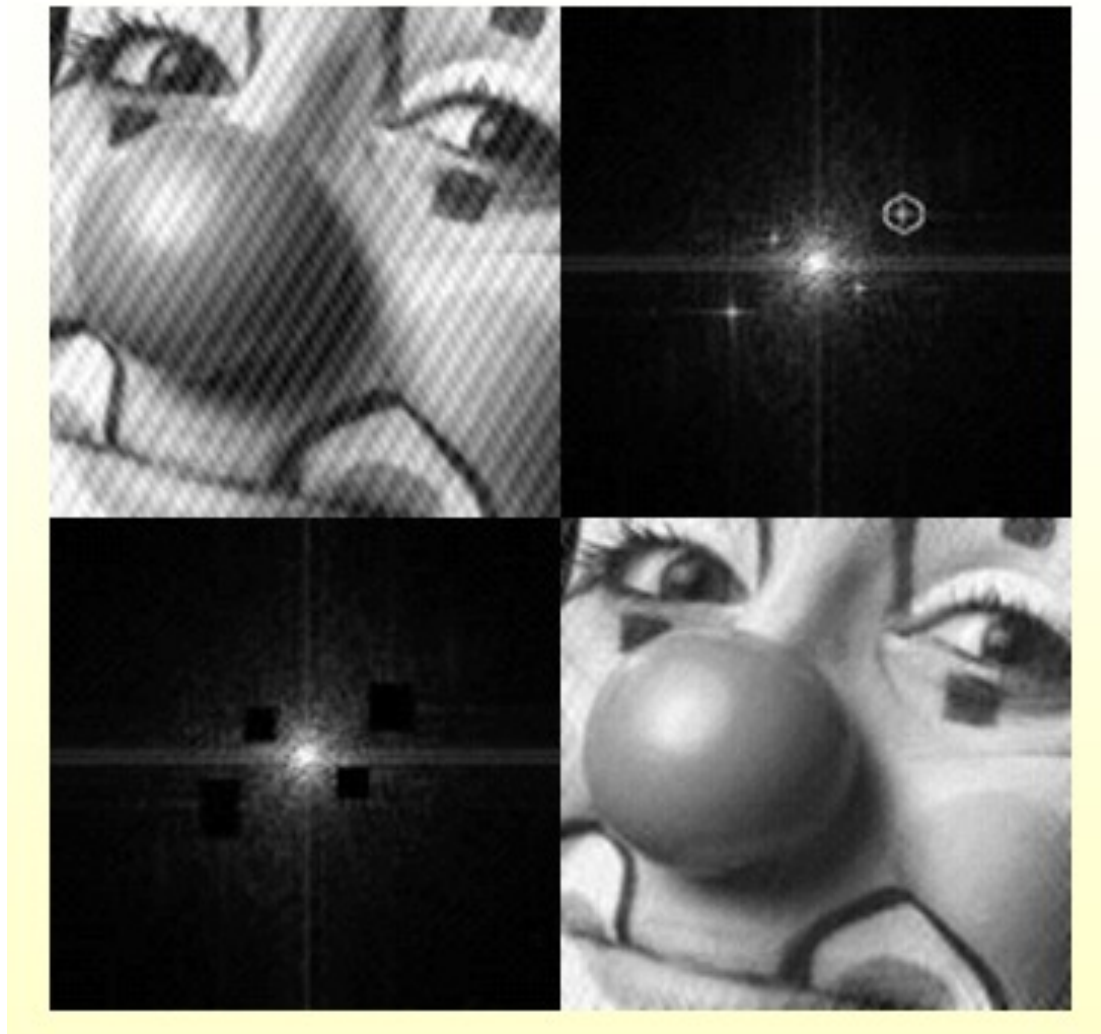
上图为傅里叶变化后的图像。高频部分（距离中心较远区域）越多，图像细节越丰富。反之图像越平滑，显得模糊

傅里叶矩阵



可用于边缘检测，上图为傅里叶变化后的图像。高频部分（距离中心较远区域）对应于图像的细节（边缘，线条等）。低频部分（中心部分）对应于图像的整体信息。

傅里叶矩阵



通过去除图像
频域谱的噪声
点，增强图像

傅里叶矩阵

傅里叶矩阵

- 高效计算范德蒙方程，可为多项式近似表达提供便利
- 实现某种分解（时域到频域），大幅简化计算
- 应用非常广泛，如信号处理恢复、图像压缩等等
- 傅里叶矩阵、傅里叶变换为信号处理做出了巨大贡献

8. Hadamard 矩阵

傅里叶变换涉及大量浮点实数乘法计算，有没有办法进一步加速计算呢？答案是哈达玛矩阵，是一类特殊的正交矩阵，广泛用于图像压缩等领域中

$H_n \in R^{n \times n}$ 称为 *Hadamard* 矩阵是指该矩阵的所有元素取+1或者取-1，且满足

$$H_n H_n^T = H_n^T H_n = nI_n$$

Hadamard 矩阵具有以下基本性质：

- (1) 只有当 $n=2$ 或者 n 是 4 的整数倍时，*Hadamard* 矩阵才存在；
- (2) $\frac{1}{\sqrt{n}} H_n$ 为标准正交矩阵，即满足

$$\left(\frac{1}{\sqrt{n}} H_n\right)^T \left(\frac{1}{\sqrt{n}} H_n\right) = \left(\frac{1}{\sqrt{n}} H_n\right) \left(\frac{1}{\sqrt{n}} H_n\right)^T = I$$

8. Hadamard 矩阵

(3) $n \times n$ 的Hadamard矩阵 H_n 的行列式 $\det(H_n) = \pm n^{\frac{n}{2}}$

可以看出，用-1乘Hadamard矩阵的任一行或者任一列，得到的结果仍然是一个Hadamard矩阵。于是得到第1列和第1行所有元素为+1的Hadamard矩阵，这称为规范化Hadamard矩阵。

令 $n = 2^k$, $k = 1, 2, \dots$, 则规范化的标准正交 Hadamard 矩阵具有通用构造公式

$$\bar{H}_n = \frac{1}{\sqrt{2}} \begin{bmatrix} \bar{H}_{n/2} & \bar{H}_{n/2} \\ \bar{H}_{n/2} & -\bar{H}_{n/2} \end{bmatrix} \quad (2.5.2)$$

其中

$$\bar{H}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (2.5.3)$$

8. Hadamard 矩阵

例2.2 当 $n = 2^3 = 8$ 时, *Hadamard*矩阵为?

令 $n = 2^k$, $k = 1, 2, \dots$, 则规范化的标准正交 Hadamard 矩阵具有通用构造公式

$$\bar{H}_n = \frac{1}{\sqrt{2}} \begin{bmatrix} \bar{H}_{n/2} & \bar{H}_{n/2} \\ \bar{H}_{n/2} & -\bar{H}_{n/2} \end{bmatrix} \quad (2.5.2)$$

其中

$$\bar{H}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (2.5.3)$$

8. Hadamard 矩阵

例2.2 当 $n = 2^3 = 8$ 时, *Hadamard*矩阵为

$$H_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

8. Hadamard 矩阵

Hadamard 矩阵为 H_8 的 1~8 行的波形函数 $\phi_0(t), \phi_1(t) \dots, \phi_7(t)$

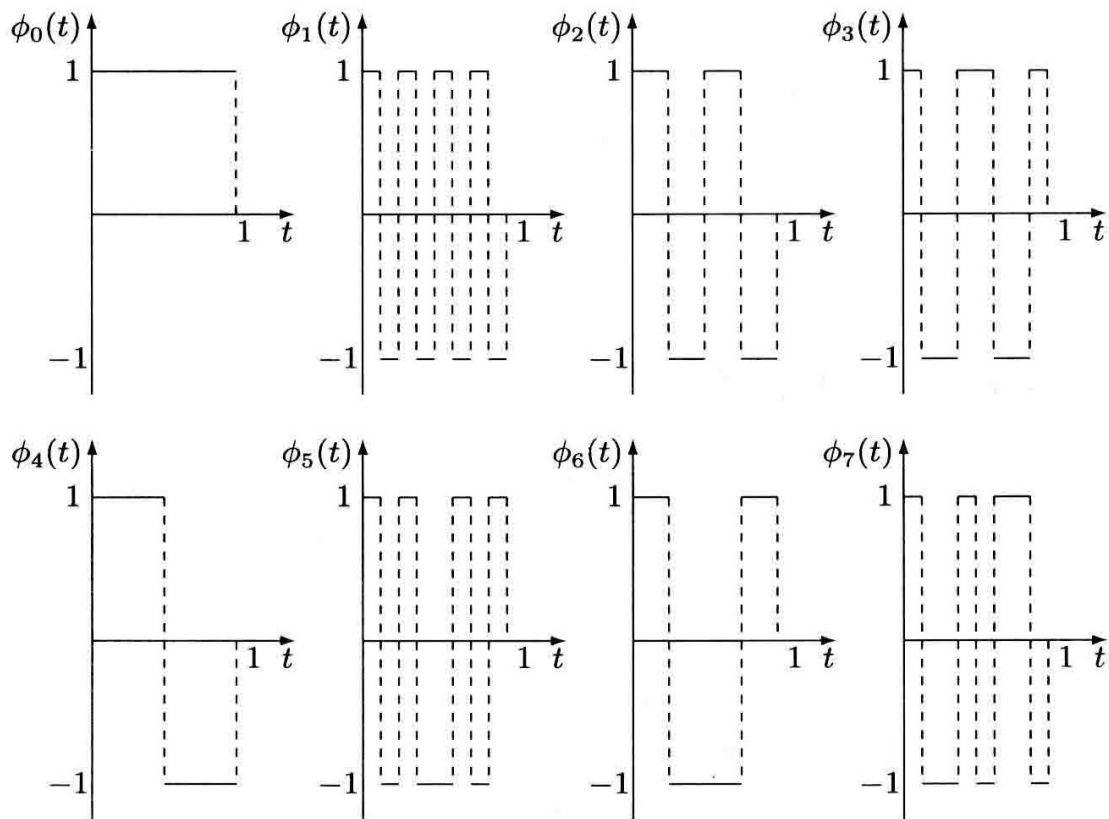


图 2.5.1 Hadamard 矩阵每行的波形

8. Hadamard 矩阵

由上图容易看出， H_8 的8个矩形脉冲波形函数 $\phi_0(t), \phi_1(t) \dots, \phi_7(t)$ 相互正交，即

$$\int_0^1 \phi_i(t) \phi_j(t) dt = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

使用Hadamard矩阵 H 的线性变换 $Y = HX$ 称为矩阵 X 的Hadamard变换。

8. Hadamard 矩阵

由于Hadamard矩阵是正交矩阵，并且元素只取+1或-1，故Hadamard变换是唯一一只使用加法和减法的正交变换。

Hadamard变换的典型应用有：

- (1) 在移动通信中用作编码，得到的码称为Hadamard码（或称Walsh-Hadamard码）
- (2) Hadamard矩阵的行向量可以用来仿真码分多址中各个用户的扩频波形向量。
- (3) 视频编码（比如H. 264）

8. Hadamard 矩阵

- 计算下列图像的哈达玛变换，计算 $H * F * H$ 。

$$(1) f_1 = \begin{bmatrix} 1 & 3 & 3 & 1 \\ 1 & 3 & 3 & 1 \\ 1 & 3 & 3 & 1 \\ 1 & 3 & 3 & 1 \end{bmatrix} \quad (2) f_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

8. Hadamard 矩阵

- 计算下列图像的哈达玛变换。

$$(1) f_1 = \begin{bmatrix} 1 & 3 & 3 & 1 \\ 1 & 3 & 3 & 1 \\ 1 & 3 & 3 & 1 \\ 1 & 3 & 3 & 1 \end{bmatrix} \quad (2) f_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

图像压缩

$$\square \Rightarrow F_1 = \frac{1}{4} H f_1 H = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 3 & 3 & 1 \\ 1 & 3 & 3 & 1 \\ 1 & 3 & 3 & 1 \\ 1 & 3 & 3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 8 & 0 & 0 & -4 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad F_2 = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

8. Hadamard 矩阵

哈达玛矩阵

- 离散整数正交矩阵，高效计算
- 应用非常广泛，如信号处理恢复、图像压缩等等

实际应用意义

- 通信领域
 - **编码与解码**：在通信系统的纠错编码中，哈达玛矩阵可用于构造哈达玛码。由于其特殊的正交性质，能够实现高效的编码和解码过程，提高通信系统的抗干扰能力和信息传输的准确性，降低误码率。
 - **扩频通信**：在扩频通信技术中，哈达玛矩阵被用作扩频码序列。利用其良好的自相关性和互相关性，不同用户的信号在接收端可以通过相关检测进行分离，有效提高了通信系统的容量和抗多址干扰能力。
- 信号处理领域
 - **快速变换**：基于哈达玛矩阵的哈达玛变换是一种快速的正交变换，类似于傅里叶变换，可用于信号的频谱分析、数据压缩等。与其他变换相比，哈达玛变换只涉及加法和减法运算，计算复杂度低，在实时信号处理和大数据量信号处理中具有明显优势。
 - **图像识别与处理**：在图像识别和处理中，哈达玛矩阵可用于图像的特征提取和编码。通过将图像与哈达玛矩阵进行运算，能够提取图像的特定特征，实现图像的压缩、加密和特征匹配，提高图像识别和处理的效率和准确性。

9. Toeplitz 矩阵

任何一条对角线的元素取相同值的特殊矩阵

$$A = \begin{bmatrix} a_0 & a_{-1} & a_{-2} & \cdots & a_{-n} \\ a_1 & a_0 & a_{-1} & \cdots & a_{-n+1} \\ a_2 & a_1 & a_0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & a_{-1} \\ a_n & a_{n-1} & \cdots & a_1 & a_0 \end{bmatrix} = [a_{i-j}]_{i,j=0}^n$$

称为*Toeplitz*矩阵。显然一个 $(n+1) \times (n+1)$ 的*Toeplitz*矩阵由其第一行元素 $a_0, a_{-1}, \dots, a_{-n}$ 和第一列元素 a_0, a_1, \dots, a_n 完全确定

9. Toeplitz 矩阵

最常见的是**对称**Toeplitz矩阵 $A = [a_{|i-j|}]_{i,j=0}^n$ ，即其元素还满足对称关系 $a_{-i} = a_i (i = 1, 2, \dots, n)$ 。对称Toeplitz矩阵仅由其第1行元素就可以完全描述。因此常将 $(n+1) \times (n+1)$ 对称 Toeplitz 矩阵简单记为

$$A = \text{Teop}[a_0, a_1, \dots, a_n]。$$

Toeplitz矩阵具有如下性质：

- (1) Toeplitz矩阵的线性组合仍然是Toeplitz矩阵；
- (2) 若Toeplitz矩阵A的元素 $a_{ij} = a_{|i-j|}$ ，则称A为对称Toeplitz矩阵；
- (3) Toeplitz矩阵A的转置仍然为Toeplitz矩阵；
- (4) Toeplitz矩阵的元素相对于交叉对角线对称。

9. Toeplitz 矩阵

例子：在时间序列分析与功率谱估计中，需要求解Yule-Walker方程

$$\begin{bmatrix} R(0) & R(-1) & \dots & R(-m+1) \\ R(1) & R(0) & \dots & R(-m+2) \\ \vdots & \vdots & \ddots & \vdots \\ R(m-1) & R(m-2) & \dots & R(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} R(1) \\ R(2) \\ \vdots \\ R(m) \end{bmatrix}$$

其中 $R(-i) = E\{x^*(n)x(n-i)\}$ 是观测信号 $x(n)$ 的自相关函数， a_i 表示Model的第 i 个系数。

$$E_m = \sum_{i=0}^m a_i^{(m)} R(-i) \quad (2.6.4)$$

是 m 阶预测误差滤波器的误差输出功率。

9. Toeplitz 矩阵

Yule-Walker 方程式 (2.6.3) 为 Toeplitz 线性方程组, 因为自相关矩阵为 $(m+1) \times (m+1)$ Toeplitz 矩阵。

Toeplitz 线性方程组 (2.6.3) 可以利用经典 Levinson 递推公式

$$a_i^{(m)} = a_i^{(m-1)} + \beta_m a_{m-i}^{*(m-1)}, \quad i = 0, 1, \dots, m-1 \quad (2.6.5)$$

$$\beta_m = a_m^{(m)} \quad (2.6.6)$$

$$E_m = (1 - |\beta_m|^2) E_{m-1} \quad (2.6.7)$$

求解, 其计算复杂度为 $O(m^2)$ 。

9. Toeplitz 矩阵

4.7.2 三个问题

假设存在标量 r_1, \dots, r_n 使得对于 $k = 1:n$,

$$\mathbf{T}_k = \begin{bmatrix} 1 & r_1 & \cdots & r_{k-2} & r_{k-1} \\ r_1 & 1 & \ddots & & r_{k-2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ r_{k-2} & & \ddots & \ddots & r_1 \\ r_{k-1} & r_{k-2} & \cdots & r_1 & 1 \end{bmatrix}$$

阅读
思考
实践

是正定矩阵. (不失一般性, 其对角线元素已单位化.) 下面介绍三个重要的算法.

- 解 Yule-Walker 问题 $\mathbf{T}_n \mathbf{y} = -[r_1, \dots, r_n]^T$ 的 Durbin 算法.
- 解一般右端项问题 $\mathbf{T}_n \mathbf{x} = \mathbf{b}$ 的 Levinson 算法.
- 计算 $\mathbf{B} = \mathbf{T}_n^{-1}$ 的 Trench 算法.

9. Toeplitz 矩阵

Toeplitz矩阵

- 对于循环性、周期性的、时序性的任务应用广泛
 - 当其为正交矩阵时，求解高效
-
- **图像处理**
 - **图像去噪与恢复**：在图像去噪和恢复算法中，Toeplitz 矩阵可用于构建图像的先验模型。基于图像的局部相关性和 Toeplitz 矩阵的结构特点，能够有效地去除图像中的噪声，恢复图像的细节信息，提高图像的质量和清晰度。
 - **图像特征提取**：在图像特征提取过程中，Toeplitz 矩阵可以用于构建图像的纹理特征描述子。通过对图像的灰度值或其他特征进行 Toeplitz 矩阵建模，提取出具有代表性的纹理特征，用于图像的分类、检索和识别等任务。

特殊矩阵

- Hermitian matrix
- 置换与互换矩阵
- 广义置换与选择矩阵
- 正交矩阵与酉矩阵
- 三角矩阵
- Vandermonde matrix
 - Fourier matrix
- Hadamard matrix
- Toeplitz matrix