Chapter25复数矩阵(埃尔米塔矩阵和酉矩阵)与傅里叶矩阵与变换

Section1.复数矩阵

Subsection1.概念

先介绍复数向量，我们不妨换一个字母符号来表示：$z=\begin{bmatrix}z_1\\z_2\\\vdots\\z_n\end{bmatrix}$，向量的每一个分量都是复数。此时$z$不再属于$\mathbb{R}^n$实向量空间，它现在处于$\mathbb{C}^n$复向量空间。

Subsection2.复数向量的模

实向量的模的计算方式:$|v|=\sqrt{v^Tv}$

类比到复向量:$z^Tz=\begin{bmatrix}z_1 & z_2 & \cdots & z_n\end{bmatrix}\begin{bmatrix}z_1\\z_2\\\vdots\\z_n\end{bmatrix}=z_1^2+z_2^2+\cdots+z_n^2$,这是错误的，因此平方后虚部为负，求模时本应相加的运算变成了减法。
（如向量$\begin{bmatrix}1 & i\end{bmatrix}$，右乘其转置后结果为$0$，但此向量的长度显然不是零。）

因此我们应该先取共轭复数，再做类似的转置乘法，即:

$$|z|=\sqrt{\bar{z}^Tz} \qquad \text{(复向量模公式)}$$

即$\begin{bmatrix}\bar{z}_1 & \bar{z}_2 & \cdots & \bar{z}_n\end{bmatrix}\begin{bmatrix}z_1\\z_2\\\vdots\\z_n\end{bmatrix}$，即使用向量共轭的转置乘以原向量即可。（如向量$\begin{bmatrix}1 & i\end{bmatrix}$，右乘其共轭转置后结果为$\begin{bmatrix}1 & -i\end{bmatrix}\begin{bmatrix}1\\i\end{bmatrix}=2$。）

我们把共轭转置乘以原向量$\bar{z}^Tz$记为z^Hz，H读作埃尔米特（人名为Hermite，形容词为Hermitian）

Subsection3.计算两向量的内积

有了复向量模的计算公式，同理可得，对于复向量，内积不再是实向量的y^Tx形式，复向量内积应为y^Hx。

Subsection4.对称矩阵(埃尔米特矩阵)

对于实矩阵，$A^T=A$即可表达矩阵的对称性。而对于复矩阵，我们同样需要求一次共轭$\bar{A}^T=A$。举个例子$\begin{bmatrix} 2 & 3+i \\ 3-i & 5 \end{bmatrix}$是一个复数情况下的对称矩阵。这叫做埃尔米特矩阵，有性质$A^H=A$。

Subsection5.正交特性(酉矩阵)

实向量我们对正交的特性为:

$$q_i^Tq_j=\begin{cases} 0 & i\neq j \\ 1 & i=j \end{cases}$$

同样的，复向量我们对正交的特性定义为:

$$\bar{q}_i^Tq_j=q_i^Hq_j=\begin{cases} 0 & i\neq j \\ 1 & i=j \end{cases} \tag{复向量正交}$$

标准正交矩阵：$Q=\begin{bmatrix} q_1 & q_2 & \cdots & q_n \end{bmatrix}$有$Q^TQ=I$。现在对于复矩阵则有$Q^HQ=I$，而复向量的正交矩阵我们称为酉矩阵，满足$Q^HQ=I$的性质。而后面提到的傅里叶矩阵就是一个酉矩阵。

Section2.傅里叶矩阵

我们定义一个矩阵：$F_n=\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w & w^2 & \cdots & w^{n-1} \\ 1 & w^2 & w^4 & \cdots & w^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w^{n-1} & w^{2(n-1)} & \cdots & w^{(n-1)^2} \end{bmatrix}$(区别于范德蒙矩阵)，其中:

$w^n=1$,$w=e^{i2\pi/n}$。易知w在复平面的单位圆上，$w=\cos\frac{2\pi}{n}+i\sin\frac{2\pi}{n}$。

欧拉公式

$$e^{ix}=cosx+isinx$$

在傅里叶矩阵中，当我们计算w的幂时，w在单位圆上的角度翻倍。比如在6阶情形下，$w=e^{2\pi/6}$，即位于单位圆上60°角处，其平方位于单位圆上120°角处，而w^6位于1处。从开方的角度看，它们是1的6个六次方根，而一次的w称为原根。

我们现在来看4阶傅里叶矩阵，先计算w有$w=i$，$w^2=-1$，$w^3=-i$，$w^4=1$，
$F_4=\begin{bmatrix}1&1&1&1\\1&i&i^2&i^3\\1&i^2&i^4&i^6\\1&i^3&i^6&i^9\end{bmatrix}=\begin{bmatrix}1&1&1&1\\1&i&-1&-i\\1&-1&1&-1\\1&-i&-1&i\end{bmatrix}$。

矩阵的四个列向量正交，我们验证一下第二列和第四列，$\bar{c_2}^Tc_4=1-0+1-0=0$，正交。不过我们应该注意到，F_4的列向量并不是标准的，我们可以给矩阵乘上系数$\frac{1}{2}$（除以列向量的长度）得到标准正交矩阵$F_4=\frac{1}{2}\begin{bmatrix}1&1&1&1\\1&i&-1&-i\\1&-1&1&-1\\1&-i&-1&i\end{bmatrix}$。此时有$F_4^HF_4=I$，于是该矩阵的逆矩阵也就是其共轭转置$F_4^H$。

Section3.快速傅里叶变换（Fast Fourier transform/FFT）

对于傅里叶矩阵，F_6，F_3、F_8，F_4、F_{64}，F_{32}之间有着特殊的关系。

举例来说:用一个列向量右乘F_{64}需要约64^2次计算，我们想要减少计算量，于是想要分解F_{64}，联系到F_{32}:

$$\begin{bmatrix}F_{64}\end{bmatrix}=\begin{bmatrix}I&D\\I&-D\end{bmatrix}\begin{bmatrix}F_{32}&0\\0&F_{32}\end{bmatrix}\begin{bmatrix}1&&\cdots&&&0&&\cdots&&\\0&&\cdots&&&1&&\cdots&&\\&1&\cdots&&&&0&\cdots&&\\&0&\cdots&&&&1&\cdots&&\\&&&\ddots&&&&&\ddots&\\&&&\ddots&&&&&\ddots&\\&&&\cdots&1&&&&\cdots&0\\&&&\cdots&0&&&&\cdots&1\end{bmatrix}$$

这几个矩阵我们分开来说:

1. 第一个矩阵由单位矩阵I和对角矩阵$D=\begin{bmatrix}1 & & & & \\ & w & & & \\ & & w^2 & & \\ & & & \ddots & \\ & & & & w^{31}\end{bmatrix}$组成，我们称这个矩阵为修正矩阵，显然其计算量来自$D$矩阵，对角矩阵的计算量约为32即这个修正矩阵的计算量约为32，单位矩阵的计算量忽略不计。
2. 第二个矩阵是两个F_{32}与零矩阵组成的，计算量约为2×32^2。
3. 第三个矩阵通常记为P矩阵，这是一个置换矩阵，其作用是讲前一个矩阵中的奇数列提到偶数列之前，将前一个矩阵从$\begin{bmatrix}x_0 & x_1 & \cdots\end{bmatrix}$变为$\begin{bmatrix}x_0 & x_2 & \cdots & x_1 & x_3 & \cdots\end{bmatrix}$，这个置换矩阵的计算量也可以忽略不计。（这里教授似乎在黑板上写错了矩阵，可以参考FFT、How the FFT is computed做进一步讨论。）(有待商榷,joe留，毕竟这个地方我觉得对于我来说不是重要的点)

所以我们把64^2复杂度的计算化简为$2\times 32^2+32$复杂度的计算，我们可以进一步化简F_{32}得到与F_{16}有关的式子$\begin{bmatrix}I_{32} & D_{32}\\ I_{32} & -D_{32}\end{bmatrix}\begin{bmatrix}I_{16} & D_{16} & & \\ I_{16} & -D_{16} & & \\ & & I_{16} & D_{16}\\ & & I_{16} & -D_{16}\end{bmatrix}\begin{bmatrix}F_{16} & & & \\ & F_{16} & & \\ & & F_{16} & \\ & & & F_{16}\end{bmatrix}\begin{bmatrix}P_{16} & \\ & P_{16}\end{bmatrix}\begin{bmatrix}P_{32}\end{bmatrix}$。而$32^2$的计算量进一步分解为$2\times 16^2+16$的计算量，如此递归下去我们最终得到含有一阶傅里叶矩阵的式子。

来看化简后计算量，$2\left(2\left(2\left(2\left(2\left(2(1)^2+1\right)+2\right)+4\right)+8\right)+16\right)+32$，约为$6\times 32=\log_2 64\times\frac{64}{2}$，算法复杂度为$\frac{n}{2}\log_2 n$。

于是原来需要n^2的运算现在只需要$\frac{n}{2}\log_2 n$就可以实现了。不妨看看$n=10$的情况，不使用FFT时需要$n^2=1024\times 1024$次运算，使用FFT时只需要$\frac{n}{2}\log_2 n=5\times 1024$次运算，运算量大约是原来的$\frac{1}{200}$。