计算方法

北京师范大学数学科学学院
主讲人：纪光华
Email：ghji@bnu.edu.cn

第六章　　快速Fourier变换

本章主要内容

1. Fourier变换与离散Fourier变换
2. 快速Fourier变换
3. 应用

FFT

FFT产生20世纪60年代中期，它高效的实现了将一个N维数组转化为其离散Fourier变换的过程，将$O(N^2)$的计算复杂度减少到$O(N log_2 N)$,被评为20世纪十大算法之一。FFT在谱分析、卷积计算、求解微分方程等方面应用广泛。

Fourier变换

定义6.1

设$f(x)\in L^1(-\infty,+\infty)$,则$f(x)$的*Fourier*变换为

$$\hat{f}(k)=\int_{-\infty}^{+\infty}f(x)e^{-ikx}dx. \tag{1}$$

如果$f(x)\in L^1(-\infty,+\infty)\cap L^2(-\infty,+\infty)$，则$\hat{f}(k)$有*Fourier*逆变换

$$f(x)=\frac{1}{2\pi}\int_{-\infty}^{+\infty}\hat{f}(k)e^{ikx}dk \tag{2}$$

Fourier变换基本性质

1. 求导变系数：

$$\left(\widehat{f'(x)}\right)(k) = ik\hat{f}(k);$$

2. 平移性质：

$$\left(\widehat{f(x-a)}\right)(k) = e^{-ika}\hat{f}(k);$$

3. 卷积变乘积：

$$\left(\widehat{f*g}\right)(k) = \hat{f}(k)\hat{g}(k),$$

其中$(f*g)(x) = \int_{-\infty}^{+\infty} f(x-y)g(y)dy;$

4. Parseval等式：设$f \in L^1(-\infty,+\infty) \cap L^2(-\infty,+\infty)$,则

$$\int_{-\infty}^{+\infty} |f(x)|^2 dx = \frac{1}{2\pi}\int_{-\infty}^{+\infty} |\hat{f}(k)|^2 dk$$

离散Fourier变换（DFT）

定义6.2

设向量$a = (a_0, a_1, \cdots, a_{N-1})^T$,定义其离散Fourier变换（DFT）为$c = (c_0, c_1, \cdots, c_{N-1})^T \triangleq \hat{a}$,其中

$$c_k = \sum_{j=0}^{N-1} a_j e^{-jk\frac{2\pi i}{N}}, k = 0, 1, \cdots, N-1.$$

i为虚数单位，$\omega \triangleq e^{-\frac{2\pi i}{N}}$是$N$次基本单位根。

引理6.1

设ω是N次基本单位根，k是整数。则有

$$\sum_{j=0}^{N-1} \omega^{jk} = \begin{cases} n & \text{如果}\frac{k}{N}\text{是整数} \\ 0 & \text{其它} \end{cases} \tag{3}$$

定理6.1

向量a一定可由其离散Fourier变换c作离散Fourier逆变换得到，记作$a=\check{c}$,其中

$$a_j=\frac{1}{N}\sum_{k=0}^{N-1}c_k e^{jk\frac{2\pi i}{N}}, j=0,1,\cdots,N-1.$$

证明： 定义Fourier矩阵

$$F=\begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & \omega & \cdots & \omega^{N-1} \\ \vdots & \vdots & & \vdots \\ 1 & \omega^{N-1} & \cdots & \omega^{(N-1)^2} \end{pmatrix} \triangleq (\omega^{jk})_{j,k=0}^{N-1} \tag{4}$$

由DFT定义可知$c=Fa$. F为一个复的对称Vandermonde矩阵，记$F^{-1}=G$. 以下只需证明

$$G=\frac{1}{N}\begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & \omega^{-1} & \cdots & \omega^{-(N-1)} \\ \vdots & \vdots & & \vdots \\ 1 & \omega^{-(N-1)} & \cdots & \omega^{-(N-1)^2} \end{pmatrix}=\frac{1}{N}(\omega^{-jk})_{j,k=0}^{N-1}$$

记向量

$$\begin{aligned} X_j &= (1, \omega^j, \cdots, \omega^{(N-1)j})^T, j = 0, 1, \cdots, N-1, \\ Y_k &= (1, \omega^{-k}, \cdots, \omega^{-(N-1)k})^T, k = 0, 1, \cdots, N-1. \end{aligned}$$

则

$$X_j^T \cdot Y_k = \sum_{l=0}^{N-1} \omega^{(j-k)l} = \begin{cases} 0, & j \neq k, \\ N, & j = k. \end{cases} \tag{5}$$

因为$j \neq k$时，$\sum_{l=0}^{N-1} \omega^{(j-k)l}$相当于

$$P(x) = 1 + x + \cdots + x^{N-1}$$

在ω^{j-k}处取值，由ω为N次单位根，$P(\omega) = 0$。
$G = \frac{1}{N} \bar{F} = F^{-1}$,记$F^* = \overline{F^T} = \bar{F} = NF^{-1}$

Remark:

1. 令$P(x) = a_0 + a_1 x + \cdots + a_{N-1} x^{N-1}$,则有
$$c_k = P(\omega^k), k = 0, 1, \cdots, N-1.$$
即求$a = (a_0, a_1, \cdots, a_{N-1})^T$的离散Fourier变换相当于求$P(x)$在$\omega^0, \omega, \cdots, \omega^{N-1}$这N个点上的取值，而求$c = (c_0, c_1, \cdots, c_{N-1})^T$的离散Fourier逆变换即相当于已知插值点求插值多项式。
2. 对$[0, 2\pi]$上的周期函数$f(x)$，给定点$x_j = \dfrac{2\pi j}{N}(j = 0, 1, \cdots, N-1)$上的值$a_j$，作三角插值
$$f_N(x) = \begin{cases} \sum_{k=-N/2+1}^{N/2} c_k e^{ikx}, & N \ even, \\ \sum_{k=-(N-1)/2}^{(N-1)/2} c_k e^{ikx}, & N \ odd, \end{cases} \tag{6}$$
使得$f_N(x_j) = a_j (j = 0, 1, \cdots, N-1)$. 令
$$\begin{cases} c_{\frac{N}{2}+k} \triangleq c_{-\frac{N}{2}+k}, & k = 1, 2, \cdots, \frac{N}{2} - 1, N \ even, \\ c_{\frac{N-1}{2}+k} \triangleq c_{-\frac{N+1}{2}+k}, & k = 1, 2, \cdots, \frac{N-1}{2}, N \ odd, \end{cases} \tag{7}$$
则$a = (Nc)^V$

DFT的性质

1. 卷积变乘积：

$$\widehat{(f*g)}_k = \hat{f}_k\hat{g}_k, k = 0, 1, \cdots, N-1,$$

其中$(f*g)_l = \sum_{j=0}^{N-1} f_{l-j}g_j (l = 0, 1, \cdots, N-1)$，而且$f_l$对指标$l$是以N为周期的，即$f_{-i} = f_{N-i}$。

2. Parseval等式：

$$N\sum_{j=0}^{N-1}|a_j|^2 = \sum_{k=0}^{N-1}|c_k|^2.$$

快速Fourier变换

n维向量DFT变换的复杂度为$O(n^2)$。
J.W.Cooley与J.W.Tukey（1965）把复杂度降到$O(n\log n)$

例6.1

设$\omega = e^{-i2\pi/4} = -i$。$DFT$变换为

$$c = \begin{pmatrix} w^0 & w^0 & w^0 & w^0 \\ w^0 & w^1 & w^2 & w^3 \\ w^0 & w^2 & w^4 & w^6 \\ w^0 & w^3 & w^6 & w^9 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix},$$

$$\begin{aligned}
w^0a_0 + w^0a_2 + w^0a_1 + w^0a_3 &= (w^0a_0 + w^0a_2) + w^0(w^0a_1 + w^0a_3) \\
w^0a_0 + w^2a_2 + w^1a_1 + w^3a_3 &= (w^0a_0 + w^2a_2) + w^1(w^0a_1 + w^2a_3) \\
w^0a_0 + w^4a_2 + w^2a_1 + w^6a_3 &= (w^0a_0 + w^0a_2) + w^2(w^0a_1 + w^0a_3) \\
w^0a_0 + w^6a_2 + w^3a_1 + w^9a_3 &= (w^0a_0 + w^2a_2) + w^3(w^0a_1 + w^2a_3)
\end{aligned}$$

直接计算需16次乘法和12次加法,但是在前两行与后两行中，括号内的项是重复的。
令$\mu = w^2, u_0 = \mu^0 x_0 + \mu^0 x_2, u_1 = \mu^0 x_0 + \mu^1 x_2, v_0 = \mu^0 x_1 + \mu^0 x_3, v_1 = \mu^0 x_1 + \mu^1 x_3$.
则，$u = (u_0, u_1)^T, v = (v_0, v_1)^T$是$n = 2$的DFT变换的输出。

$$u = F_2 \begin{bmatrix} x_0 \\ x_2 \end{bmatrix}, v = F_2 \begin{bmatrix} x_1 \\ x_3 \end{bmatrix}.$$

则，$c_0 = u_0 + w^0 v_0, c_1 = u_1 + w^1 v_1, c_2 = u_0 + w^2 v_0, c_3 = u_1 + w^3 v_1$.
只需要计算2个DFT(2)和一些加法和乘法。这一想法具有普遍意义，DFT(N)可以简化为两次DFT(N/2)和2N-1次运算（N-1次乘法和N次加法）。

FFT的计算次数

定理6.2

设n是2的幂次，则大学为n的FFT变换需要$n(2\log_2 n-1)+1$次加法和乘法运算，以及一次除以$\sqrt{n}$的运算。

Remark: DFT的快速算法可以直接用来计算逆DFT变换

$$F_n^{-1}y=\overline{F_n y}=\overline{F_n\bar{y}}.$$

Remark:

1. 充分利用合并同类项技术减少乘法运算；
2. 将变换个分量作为一个整体来考虑，充分减少重复计算。

基本算法： 假设以$N = 2^m$为例，记

$$P(x) = a_0 + a_1 x + \cdots + a_{N-1} x^{N-1}$$

因为

$$\begin{aligned}
P(x) &= (a_0 + a_2 x^2 + \cdots) + x(a_1 + a_3 x^2 + \cdots) \\
&= P_e(x^2) + x P_o(x^2), \\
P_e(t) &= a_0 + a_2 t + \cdots + a_{N-2} t^{\frac{N}{2}-1}, \\
P_o(t) &= a_1 + a_3 t + \cdots + a_{N-1} t^{\frac{N}{2}-1}.
\end{aligned}$$

令$\omega_k = e^{-\frac{2\pi i}{k}}$,则当$j = 0, 1, \cdots, \frac{N}{2} - 1$时，有

$$\begin{cases} c_j = P_e(\omega_N^{2j}) + \omega_N^j P_o(\omega_N^{2j}), \\ c_{\frac{N}{2}+j} = P_e(\omega_N^{2(\frac{N}{2}+j)}) + \omega_N^{\frac{N}{2}+j} P_o(\omega_N^{2(\frac{N}{2}+j)}) \end{cases} \tag{8}$$

由于$\omega_N^{2j} = \omega_{\frac{N}{2}}^j, \omega_N^{\frac{N}{2}+j} = -\omega_N^j, \omega_N^{N+2j} = \omega_{\frac{N}{2}}^j$得

$$c_j = v_j + \omega_N^j u_j, c_{j+\frac{N}{2}} = v_j - \omega_N^j u_j, j = 0, 1, \cdots, \frac{N}{2} - 1,$$

其中$v_j = P_e(\omega_{\frac{N}{2}}^j), u_j = P_o(\omega_{\frac{N}{2}}^j)$.

Remark:求向量a的N个分量的DFT可转化为求两个自向量a_e, a_o的$\frac{N}{2}$个分量的DFT，然后通过简单相加，相乘得到。这一算法称为Danielson-Lanczos算法，将其递归下去即得FFT。

算例

（1）分割；（2）组装；以$N=8$ 为例进行说明。

设$a=(a_0,a_1,\cdots,a_7)^T$，则

1. 分割（重排序）：
   - 第一步分割：$a_e=(a_0,a_2,a_4,a_6)^T, a_o=(a_1,a_3,a_5,a_7)^T$;
   - 第二步分割：$a_{ee}=(a_0,a_4)^T, a_{eo}=(a_2,a_6)^T, a_{oe}=(a_1,a_5)^T, a_{oo}=(a_3,a_7)^T$;
   - 第三步分割：$a_{eee}=a_0, a_{eeo}=a_4, a_{eoe}=a_2, a_{eoo}=a_6, a_{oee}=a_1, a_{oeo}=a_5, a_{ooe}=a_3, a_{ooo}=a_7$.
2. 组装：
   - 第一步组装：

$$
\begin{aligned}
c_{ee} &= (a_0+w_2^0a_4, a_0-w_2^0a_4)^T, c_{eo}=(a_2+w_2^0a_6, a_2-w_2^0a_6)^T,\\
c_{oe} &= (a_1+w_2^0a_5, a_1-w_2^0a_5)^T, c_{oo}=(a_3+w_2^0a_7, a_3-w_2^0a_7)^T;
\end{aligned}
$$

   - 第二步组装：

$$
c_e=\begin{pmatrix} c_{ee}+w_4\circ c_{eo} \\ c_{ee}-w_4\circ c_{eo}\end{pmatrix}, c_o=\begin{pmatrix} c_{oe}+w_4\circ c_{oo} \\ c_{oe}-w_4\circ c_{oo}\end{pmatrix};
$$

   - 第三步组装：$c=\begin{pmatrix} c_e+w_8\circ c_o \\ c_e-w_8\circ c_o\end{pmatrix}$.

其中$w_4 \triangleq (w_4^0, w_4^1)^T, w_8 \triangleq (w_8^0, w_8^1, w_8^2, w_8^3)^T$, $X \circ Y \triangleq (x_j y_j)_j$

$$a \xrightarrow{\text{重排序}} \begin{pmatrix} a_0 \\ a_4 \\ a_2 \\ a_6 \\ a_1 \\ a_5 \\ a_3 \\ a_7 \end{pmatrix} \xrightarrow{\text{第1步组装}} \begin{pmatrix} c_{ee} \\ c_{eo} \\ c_{eo} \\ c_{oo} \end{pmatrix} \xrightarrow{\text{第2步组装}} \begin{pmatrix} c_e \\ c_o \end{pmatrix} \xrightarrow{\text{第3步组装}} c$$

FFT的应用:计算卷积

设$h(x)=\int_0^{2\pi} f(x-y)g(y)dy$,其中$f(x), g(x)$是以$2\pi$为周期的连续函数。取$x_i = i\delta(i=0,1,\cdots,N-1, \delta=\frac{2\pi}{N})$，将积分作简单矩形离散:

$$h(x_i) \approx \sum_{j=0}^{N-1} f(x_i - x_j)g(x_j)\cdot\delta, i=0,1,\cdots,N-1. \tag{9}$$

令$f_i = f(x_i), g_i = g(x_i)$，并定义$f_i$对下标以$N$为周期。

令$h_i = \sum_{j=0}^{N-1} f_{i-j}g_j \cdot \delta, i=0,1,\cdots,N-1$，于是$h=(\hat{h})^V = [\delta\cdot(\hat{f}\circ\hat{g})]^V$,其中$h=(h_1,\cdots,h_{N-1})^T, g=(g_1,\cdots,g_{N-1})^T, f=(f_1,\cdots,f_{N-1})^T$

Remark:利用FFT可将原来N^2+N次乘法和$N(N-1)$次加法减至$\frac{3}{2}N\log_2 N + 2N$次乘法及$3N\log_2 N$次加法。

FFT的应用：求解系数矩阵为循环矩阵的线性方程组

$$L = \begin{pmatrix} c_0 & c_{N-1} & \cdots & c_1 \\ c_1 & c_0 & \cdots & c_2 \\ \vdots & \vdots & & \vdots \\ c_{N-1} & c_{N-2} & \cdots & c_0 \end{pmatrix}$$

求解$Lx = b$,其
中$x = (x_0, x_1, \cdots, x_{N-1})^T, b = (b_0, b_1, \cdots, b_{N-1})^T$.
$(Lx)_i = \sum_{j=0}^{N-1} c_{i-j}x_j$.假设$c_i$对下标以N为周期。设$\lambda$为L的特征值，$c = (c_0, \cdots, c_{N-1})^T$, $Lx = \lambda x, i.e. c * x = \lambda x$,等式两端同时作DFT得$\hat{c} \circ \hat{x} = \lambda \hat{x}$,于是有$\lambda_k = \hat{c}_k$，相应的特征向量$\hat{x}_j^{(k)} = \delta_{kj}$

作逆变换得到：

$$
\begin{aligned}
x^{(0)} &= (1, 1, \cdots, 1)^T, \\
x^{(1)} &= (1, w^{-1}, \cdots, w^{-(N-1)})^T, \\
\cdots & \quad \cdots\cdots\cdots\cdot \\
x^{(N-1)} &= (1, w^{-(N-1)}, \cdots, w^{-(N-1)^2})^T
\end{aligned}
$$

于是

$$
\begin{aligned}
L &= (x^{(0)}, x^{(1)}, \cdots, x^{(N-1)}) diag(\lambda_0, \cdots, \lambda_{N-1}) (x^{(0)}, x^{(1)}, \cdots, x^{(N-1)}) \\
&\triangleq (NF^{-1})\Lambda(NF^{-1})^{-1} = F^{-1}\Lambda F
\end{aligned}
$$

求解$Lx = b$等价于求解$F^{-1}\Lambda Fx = b$。分三步走

- 求Fb，即b的FFT，的$\hat{b}$;
- 求Λ 即求c的FFT，得$\hat{c}$;
- 求$\hat{x}_k = \hat{b}_k / \hat{c}_k$，然后求$(\hat{x})^V$即得$x$.

FFT的应用：求解微分方程

设有一维Poisson方程：

$$\begin{cases} u''(x) = f(x), x \in (0, \pi), \\ u(0) = u(\pi) = 0. \end{cases}$$

$$u''(x) \approx \frac{u(x+h) - 2u(x) + u(x-h)}{h^2}$$

记$x_j = jh (j = 1, 2, \cdots, N-1; h = \frac{\pi}{N})$

$$u_{j+1} - 2u_j + u_{j-1} = h^2 f_j, j = 1, 2, \cdots, N-1,$$

将u作奇延拓成周期函数（Dirichlet边界条件），采用离散正弦变换DST和逆变换

$$\begin{aligned} U_k &= \sum_{j=1}^{N-1} u_j \sin\left(\frac{\pi}{N} jk\right), k = 1, 2, \cdots, N-1; \\ u_j &= \frac{1}{N} \sum_{k=1}^{N-1} U_k \sin\left(\frac{\pi}{N} jk\right), j = 1, 2, \cdots, N-1. \end{aligned}$$

Fast Sine Transform

$$\sum_{j=1}^{N-1} u_j \left[\sin(\frac{\pi}{N}(j+1)k) - 2\sin(\frac{\pi}{N}jk) + \sin(\frac{\pi}{N}(j-1)k)\right] = F_k h^2,$$

其中$F_k = \sum_{j=1}^{N-1} f_j \sin(\frac{\pi}{N}jk), (k = 1, 2, \cdots, N-1)$，于是令

$$U_k = -h^2 F_k / \left[4\sin^2\left(\frac{\pi}{2N}k\right)\right], k = 1, 2, \cdots, N-1$$

Remark:这里的$4\sin^2\left(\frac{\pi}{2N}k\right)$相当于离散情形时$-\frac{d^2}{dx^2}$的特征值。

计算步骤:

1. 对$f=(f_1,\cdots,f_{N-1})^T$作FST得$F_k(k=1,2,\cdots,N-1)$;
2. 求的$U_k=-h^2F_k/\left[4\sin^2\left(\frac{\pi}{2N}k\right)\right]$;
3. 求的$u=\check{U}$,其中$u=(u_1,u_2,\cdots,u_{N-1})^T, U=(U_1,U_2,\cdots,U_{N-1})^T$.

实数组的FFT实现

设$f = (f_1, f_2, \cdots, f_{N-1})^T$中元素全为实数，由DFT定义

$$F_k = \sum_{j=0}^{N-1} f_j e^{-jk\frac{2\pi i}{N}}, k = 0, 1, \cdots, N-1,$$

则

$$\begin{aligned} F_{N-k} &= \sum_{j=0}^{N-1} f_j e^{-j(N-k)\frac{2\pi i}{N}} \\ &= \sum_{j=0}^{N-1} f_j e^{jk\frac{2\pi i}{N}} = \bar{F}_k, k = 0, 1, \cdots, N-1. \end{aligned}$$

于是$F_0, F_{\frac{N}{2}}$(N为偶数时)为实数；由于只
有$F_0, F_1, \cdots, F_{\frac{N}{2}-1}, F_{\frac{N}{2}}(N\ even)$是真正需要的

为节约工作量及存储量

1. 令$h = (h_0, h_1, \cdots, h_{\frac{N}{2}-1})^T$，其中

$$h_j = f_{2j} + if_{2j+1}, j = 0, 1, \cdots, \frac{N}{2} - 1,$$

并求的$H \triangleq \hat{h}$，即有

$$H_n = \hat{h}_n = F_n^e + iF_n^o, n = 0, 1, \cdots, \frac{N}{2} - 1,$$

其中(H_n对下标以$N/2$为周期)：

$$F_n^e = \sum_{k=0}^{\frac{N}{2}-1} f_{2k} e^{-kn\frac{2\pi i}{N/2}}, F_n^o = \sum_{k=0}^{\frac{N}{2}-1} f_{2k+1} e^{-kn\frac{2\pi i}{N/2}}.$$

2. 由DFT定义知$F_n = F_n^e + e^{-\frac{2\pi i}{N}n} F_n^o$,于是

$$\bar{H}_{\frac{N}{2}-n} = \bar{F}^e_{\frac{N}{2}-n} - i\bar{F}^o_{\frac{N}{2}-n} = F_n^e - iF_n^o,$$

故$F_n = \frac{1}{2}(H_n + \bar{H}_{\frac{N}{2}-n}) - \frac{i}{2}(H_n - \bar{H}_{\frac{N}{2}-n})e^{-n\frac{2\pi i}{N}}$

FST

设$F_k = \sum_{j=1}^{N-1} f_j \sin\left(\frac{\pi}{N} jk\right), k = 0, 1, \cdots, \frac{N}{2} - 1$，定义$f_0 = 0$.

令$y_0 = 0, y_j = \sin(\frac{j\pi}{N})(f_j + f_{N-j}) + \frac{1}{2}(f_j - f_{N-j}), j = 1, \cdots, N-1.$

对向量$y = (y_0, \cdots, y_{N-1})^T$作实数组的FFT得：

$$\begin{aligned} Re_k &= \sum_{j=0}^{N-1} y_j \cos\left(\frac{2\pi}{N} jk\right) \\ &= F_{2k+1} - F_{2k-1}, k = 0, 1, \cdots, \frac{N}{2} - 1, \\ Im_k &= \sum_{j=1}^{N-1} y_j \sin\frac{2\pi jk}{N} = F_{2k}, k = 0, 1, \cdots, \frac{N}{2} - 1. \end{aligned}$$

于是有

$$F_{2k} = Im_k, F_{2k+1} = F_{2k-1} + Re_k, k = 0, 1, \cdots, \frac{N}{2} - 1,$$

其中$F_1 = -F_{-1}$即$F_1 = \frac{Re_0}{2}$.