矩阵形式的快速沃尔什变换法

汤 国 熙

提 要 本文用矩阵形式，建立系统地快速沃尔什变换（FWT）运算方法，它应用了二进制顺序 Walsh—paley，列率顺序 Walsh—Kaczmarz 和 Kronecker 顺序 Walsh—Hadamard 变换。由文献可知，某些FWT运算是借助于一定的矩阵编制计算程序的新方法，这样的变换法，可作为数字计算机的辅助设备。

快速 Walsh 变换，简记为 FWT.

设$\{X(m)\}$表示为如下形式的实数序列：

$$\{X(m)\}=\{X(0),X(1),\cdots,X(N-1)\}$$

其中$N=2^n$，$n=0,1,2,\cdots$.

$\{X(m)\}$的 Walsh 变换定义为：

$$\boldsymbol{Y}(n)=\frac{1}{N}\boldsymbol{W}_N\boldsymbol{X}(n) \tag{1}$$

式中$n=\log_2 N$，$\boldsymbol{W}_N$表示为二进制顺序、列率顺序和 Kronecker 顺序的 Walsh 矩阵，这些均为$N\times N$矩阵。

因为$\boldsymbol{W}_N$是正交和对称矩阵，所以逆变换定义为：

$$\boldsymbol{X}(n)=\boldsymbol{W}_N\boldsymbol{Y}(n) \tag{2}$$

由于Walsh函数的定义是多样形式，所以它所对应的变换形式也多种形式。这里仅介绍矩阵形式的FWT法。

一、一维序列的FWT法

1. $\boldsymbol{W}_N$分解成几个矩阵因子的乘积

在利用矩阵形式研究FWT时，对于任何一个顺序的 Walsh 函数来说，其FWT算法的基本想法是秩为$N=2^n$的Walsh 矩阵$\boldsymbol{W}_N$，恰好分解为相同秩数的n个矩阵因子的乘积：

$$\boldsymbol{W}_N=\boldsymbol{M}_N^{(1)}\boldsymbol{M}_N^{(2)}\cdots\boldsymbol{M}_N^{(n)} \tag{3}$$

本文1983年9月20日收到

式中$\boldsymbol{M}_N^{(i)}$ ($i=1,2,\cdots,n$)的每一列（或行）仅包含有两个非零元素。

这里取“稀疏”的优点是任何两个矩阵因子的乘积计算，是等效到 2^{n-1} 次加法和 2^{n-1} 次减法的运算。

根据线性代数知识：非奇异矩阵恒为若干初等矩阵之积。因此(3)这种分解法是正确的。

把(3)代入(1)得：

$$\boldsymbol{Y}(n)=\frac{1}{N}(\boldsymbol{M}_N^{(1)}\boldsymbol{M}_N^{(2)}\cdots\boldsymbol{M}_N^{(n)})\boldsymbol{X}(n) \tag{4}$$

其中 $\boldsymbol{X}'(n)=[X(0)\cdots X(N-1)]$为 $\boldsymbol{X}(n)$的转置矩阵。记

$$\left.\begin{aligned}
&\boldsymbol{X}^{(1)}(n)=\boldsymbol{M}_N^{(n)}\boldsymbol{X}(n)\\
&\boldsymbol{X}^{(2)}(n)=\boldsymbol{M}_N^{(n-1)}\boldsymbol{X}^{(1)}(n)\\
&\qquad\cdots\qquad\cdots\\
&\boldsymbol{X}^{(n-1)}(n)=\boldsymbol{M}_N^{(2)}\boldsymbol{X}^{(n-2)}(n)\\
&N\boldsymbol{Y}(n)=\boldsymbol{M}_N^{(1)}\boldsymbol{X}^{(n-1)}(n)
\end{aligned}\right\} \tag{5}$$

在(5)中，每个等式有$\frac{N}{2}$次加法和$\frac{N}{2}$次减法，共有 n 个等式，故总共有 $n\cdot 2^n=N\cdot\log_2 N$次运算，而如果直接进行计算，就必须进行$N\times(N-1)$次运算。

由上论述，这样的Walsh变换计算法比直接计算的速度快，故称之FWT，又因为Walsh函数仅取值$+1$和-1，在变换的运算过程中不考虑乘法，在同样情况下，对N个点信号进行 Fourier 变换时，需要 $N\log_2 N$ 次复数乘法和 $N\log_2 N$ 次加减法，所以FWT比FFT运算速度还快。

这种方法是迭代的方法，变换(4)由 n 次迭代而成的，因此，在信号流程图中分成 n 个部分。

下面首先研究 Hadamard 顺序的FWT的计算法。用 $\boldsymbol{W}_N^{(h)}$表示 Hadamard 顺序的Walsh 矩阵。

由上的论述可知，用矩阵的形式研究FWT时，问题在于如何把一个秩为N的矩阵分解成为具有相同的秩数的 n 个矩阵因子的乘积。

在一般情况下，$\boldsymbol{W}_N^{(h)}$可分解成：

$$\begin{aligned}
\boldsymbol{W}_N^{(h)}&=(\boldsymbol{H}_2\otimes\underbrace{\boldsymbol{I}_2\otimes\cdots\otimes\boldsymbol{I}_2}_{n-1\text{个}})(\boldsymbol{I}_2\otimes\boldsymbol{H}_2\otimes\underbrace{\boldsymbol{I}_2\otimes\cdots\otimes\boldsymbol{I}_2}_{n-2\text{个}})\cdots(\underbrace{\boldsymbol{I}_2\otimes\boldsymbol{I}_2\otimes\cdots\otimes\boldsymbol{I}_2}_{n-1\text{个}}\otimes\boldsymbol{H}_2)\\
&=\boldsymbol{M}_N^{(h,1)}\boldsymbol{M}_N^{(h,2)}\cdots\boldsymbol{M}_N^{(h,n)}
\end{aligned} \tag{6}$$

其中$\boldsymbol{H}_2=\boldsymbol{W}_2^{(h)}$，$\boldsymbol{I}_2$ 表示 2 阶么阵，$\otimes$表示逻辑乘.

$$\boldsymbol{M}_N^{(h,1)}=\boldsymbol{H}_2\otimes\underbrace{\boldsymbol{I}_2\otimes\cdots\otimes\boldsymbol{I}_2}_{n-1\text{个}},\quad\cdots,\quad\boldsymbol{M}_N^{(h,n)}=\boldsymbol{I}_2\otimes\boldsymbol{I}_2\otimes\cdots\otimes\boldsymbol{I}_2\otimes\boldsymbol{I}_2$$

(6)式的成立是对于自然数 n ($2^{n+1}\leqslant 200$ 或 $n\geqslant\log_2(200)-1, N\equiv 0 \mod 4$) 的取值是正确的。

从(6)式可知，$\boldsymbol{W}_N^{(h)}$可以分解成 n 个矩阵乘积的形式，而每个矩阵因子恰好是 n 项

($n-1$ 个 $\boldsymbol{I}_2$ 和 $\boldsymbol{H}_2$) 所组成的。同时每个矩阵因子均满足“稀疏”条件。即矩阵的每列(行)仅包含两个非零元素。矩阵 $\boldsymbol{W}_N^{(h)}$ 和 $\boldsymbol{M}_N^{(i)}(i=1,2,\cdots,n)$ 均是对称矩阵。

对于 $\boldsymbol{W}_N^{(h)}$ 分解成相同阶数的 n 个矩阵因子的乘积的方法，还有：

①由于 Hadamard 矩阵的对称性，所以

$$\boldsymbol{W}_N^{(h)}=(\boldsymbol{W}_N^{(h)})'=(\boldsymbol{M}_N^{(h,n)})'(\boldsymbol{M}_N^{(h,n-1)})'\cdots(\boldsymbol{M}_N^{(h,1)})'=\boldsymbol{M}_N^{(h,n)}\cdots\boldsymbol{M}_N^{(h,1)} \tag{7}$$

②设 $\boldsymbol{M}_N^{(h)}=\boldsymbol{H}_2\otimes\boldsymbol{I}_2\otimes\cdots\otimes\boldsymbol{I}_2$，由线性代数的矩阵相似的性质，存在非奇异矩阵 $\boldsymbol{P}_N$，使

$$\boldsymbol{I}_2\otimes\boldsymbol{H}_2\otimes\boldsymbol{I}_2\otimes\cdots\otimes\boldsymbol{I}_2=\boldsymbol{P}_N\boldsymbol{M}_N^{(h)}\boldsymbol{P}_N^{-1}$$

$$\boldsymbol{I}_2\otimes\boldsymbol{I}_2\otimes\boldsymbol{H}_2\otimes\boldsymbol{I}_2\otimes\cdots\otimes\boldsymbol{I}_2=\boldsymbol{P}_N^2\boldsymbol{M}_N^{(h)}\boldsymbol{P}_N^{-2}$$

$$\cdots\quad\cdots\quad\cdots$$

$$\boldsymbol{I}_2\otimes\cdots\otimes\boldsymbol{I}_2\otimes\boldsymbol{H}_2=\boldsymbol{P}_N^{(n-1)}\boldsymbol{M}_N^{(h)}\boldsymbol{P}_N^{-(n-1)}$$

那么，

$$\begin{aligned}\boldsymbol{W}_N^{(h)}&=(\boldsymbol{M}_N^{(h)})(\boldsymbol{P}_N\boldsymbol{M}_N^{(h)}\boldsymbol{P}_N^{-1})\cdots(\boldsymbol{P}_N^{(n-1)}\boldsymbol{M}_N^{(h)}\boldsymbol{P}_N^{-(n-1)})\\&=(\boldsymbol{M}_N^{(h)}\boldsymbol{P}_N)(\boldsymbol{M}_N^{(h)}\boldsymbol{P}_N^{-1}\boldsymbol{P}_N^2)\cdots(\boldsymbol{M}_N^{(h)}\boldsymbol{P}_N^{-(n-1)}\boldsymbol{P}_N^n)\\&=(\boldsymbol{M}_N^{(h)}\boldsymbol{P}_N)^n\end{aligned} \tag{8}$$

这里 $\boldsymbol{P}_N^n=\boldsymbol{I}_N$，$\boldsymbol{I}_N$ 表示 N 阶么阵。

③由于 Hadamard 矩阵是对称的，所以

$$\boldsymbol{W}_N^{(h)}=(\boldsymbol{W}_N^{(h)})'=((\boldsymbol{M}_N^{(h)}\boldsymbol{P}_N)^n)'=((\boldsymbol{M}_N^{(h)}\boldsymbol{P}_N)')^n=(\boldsymbol{P}_N'\boldsymbol{M}_N^{(h)})^n \tag{9}$$

利用(8)和(9)式计算时，要比用(7)式简单些。但当使用数字计算机时，它们要求一个较大的存贮数量。

前面是对 Kronecker 顺序的 Hadamard 矩阵分解成 n 个矩阵乘积而引伸得来的变换法。对于其它顺序的矩阵，也可以如法泡制，但我们知道 Paley 矩阵 $\boldsymbol{W}_N^{(p)}$ 和 Kaczmarz 矩阵 $\boldsymbol{W}_N^{(K)}$ 均是 $N=2^n$ 阶矩阵，而且是由于同阶的 Hadamard 矩阵 $\boldsymbol{W}_N^{(h)}$ 交换行(列)而得到的。由线性代数知识得到，如果对于矩阵 $\boldsymbol{W}_N^{(h)}$ 的行(列)施以几次初等变换而得到 $\boldsymbol{W}_N^{(p)}$ (或 $\boldsymbol{W}_N^{(K)}$)，则必有非奇异矩阵 $\boldsymbol{P}_N$ (或 $\boldsymbol{Q}_N$)，使得

$$\boldsymbol{W}_N^{(P)}=\boldsymbol{P}_N\boldsymbol{W}_N^{(h)} \tag{10}$$

$$\boldsymbol{W}_N^{(K)}=\boldsymbol{Q}_N\boldsymbol{W}_N^{(h)} \tag{11}$$

由于 $\boldsymbol{W}_N^{(h)}$ 的对称性，所以 $\boldsymbol{W}_N^{(p)}$ 或 $\boldsymbol{W}_N^{(K)}$ 可以由一个确定的置换行(列)的初等矩阵去左乘矩阵 $\boldsymbol{W}_N^{(h)}$ 或右乘一个转置的初等矩阵。即

$$\boldsymbol{W}_N^{(P)}=\boldsymbol{W}_N^{(h)}\boldsymbol{P}_N' \quad 或 \quad \boldsymbol{W}_N^{(K)}=\boldsymbol{W}_N^{(h)}\boldsymbol{Q}_N'$$

在(10)中根据矩阵的乘法定理得：

$$Wal_p(k,r)=\sum_{j=0}^{N-1}P_{kj}Wal_H(j,r) \quad (k,r=0,1,\cdots,N-1)$$

其中 $Wal_p(k,r)$，$Wal_H(j,r)$ 分别表示 Paley 矩阵，Hadamard 矩阵的元素，P_{kj} 是 $\boldsymbol{P}_N$ 的第 k 行第 j 列元素。

根据不同顺序的 Walsh 函数之间的关系

$$P_{kj}=\begin{cases}1, & j=\langle k\rangle\\0, & 其它\end{cases} \tag{12}$$

其中 $\langle k\rangle$ 表示 $k=(k_{n-1}k_{n-2}\cdots k_0)_2$ 的逆顺序 $(k_0k_1\cdots k_{n-1})_2$。

同理可得：

$$Q_{kj}=\begin{cases}1, & j=\langle k\oplus\left[\dfrac{k}{2}\right]\rangle\\ 0, & \text{其它}\end{cases}$$

其中$\oplus$表示模 2 加，$\left[\frac{k}{2}\right]$表示不超过$\frac{k}{2}$的最大整数。Q_{kj} 是 $\boldsymbol{Q}_N$ 中第 k 行第 j 列的元素。

由(12)可知，$\boldsymbol{P}_N$ 是对称的，则

$$\boldsymbol{P}_N\boldsymbol{P}'_N=\boldsymbol{I}_N \tag{13}$$

事实上，由于

$$\boldsymbol{W}_N^{(p)}=\boldsymbol{P}_N\boldsymbol{W}_N^{(h)}=\boldsymbol{W}_N^{(h)}\boldsymbol{P}'_N$$

从而

$$\boldsymbol{W}_N^{(p)}\boldsymbol{W}_N^{(p)}=\boldsymbol{P}_N\boldsymbol{W}_N^{(h)}\boldsymbol{W}_N^{(h)}\boldsymbol{P}'_N=N\boldsymbol{P}_N\boldsymbol{P}'_N$$

另一方面，$\boldsymbol{W}_N^{(p)}\boldsymbol{W}_N^{(p)}=N\boldsymbol{I}_N$，故(13)式成立。

可是我们不能说，$\boldsymbol{Q}_N$ 是对称的，不过由定义它保持了 $\boldsymbol{Q}_N\boldsymbol{Q}_N'=\boldsymbol{I}_N$. 这样一来，由(10)和(11)我们建立起快速 Walsh-Paley 变换(FWPT)法和快速 Walsh-Kaczmarz 变换 (FWKT)法。

下面介绍 Paley 矩阵$\boldsymbol{W}_N^{(p)}$的第二个分解法。事实上，在上面的基础上，作了简单变换，使之比上面方法更适用于计算机计算。

设 $\boldsymbol{S}_N^{(i)}$ 是任意置换矩阵系，具有

$$\boldsymbol{S}_N^{(1)}=\boldsymbol{P}_N,\ \boldsymbol{S}_N^{(i)}\boldsymbol{S}_N^{(i)\prime}=\boldsymbol{I}_N \quad (i=1,2,\cdots,n)$$

由方程(10)得：

$$\begin{aligned}\boldsymbol{W}_N^{(p)}&=(\boldsymbol{S}_N^{(1)}\boldsymbol{M}_N^{(h,1)}\boldsymbol{S}_N^{(2)})(\boldsymbol{S}_N^{(2)}\boldsymbol{M}_N^{(h,2)}\boldsymbol{S}_N^{(3)})\cdots(\boldsymbol{S}_N^{(n)}\boldsymbol{M}_N^{(h,n)})\\&=\boldsymbol{M}_N^{(p,1)}\boldsymbol{M}_N^{(p,2)}\cdots\boldsymbol{M}_N^{(p,n)}\end{aligned}$$

令

$$\boldsymbol{S}_N^{(i)}=\boldsymbol{P}_{2^{n+1-i}}\otimes\boldsymbol{I}_{2^{i-1}} \quad (i=1,2,\cdots,n)$$

这里$\boldsymbol{P}_{2^{n+1-i}}$由(12)确定，$\boldsymbol{I}_{2^{i-1}}$是 2^{i-1} 阶单位矩阵。

由(6)和(7)有

$$\boldsymbol{M}_N^{(p,i)}=(\boldsymbol{P}_{2^{n+1-i}}\otimes\boldsymbol{I}_{2^{i-1}})(\boldsymbol{I}_{2^{n-i}}\otimes\boldsymbol{H}_2\otimes\boldsymbol{I}_{2^{i-1}})(\boldsymbol{P}_{2^{n-i}}\otimes\boldsymbol{I}_{2^i})' \tag{14}$$

因为$(\boldsymbol{P}_{2^{n-i}}\otimes\boldsymbol{I}_{2^i})'=\boldsymbol{P}_{2^{n-i}}\otimes\boldsymbol{I}_{2^i}$，且利用关系：

$$(\boldsymbol{A}_m\otimes\boldsymbol{B}_n)(\boldsymbol{C}_m\otimes\boldsymbol{D}_n)=\boldsymbol{A}_m\boldsymbol{C}_m\otimes\boldsymbol{B}_n\boldsymbol{D}_n$$

式中 $\boldsymbol{A}_m,\boldsymbol{C}_m$；$\boldsymbol{B}_n,\boldsymbol{D}_n$ 分别是 m 和 n 阶方阵，那么

$$\begin{aligned}&(\boldsymbol{I}_{2^{n-i}}\otimes\boldsymbol{H}_2\otimes\boldsymbol{I}_{2^{i-1}})(\mathrm{P}_{2^{n-i}}\otimes\boldsymbol{I}_{2^i})'\\&=[\boldsymbol{I}_{2^{n-i}}\otimes(\boldsymbol{H}_2\otimes\boldsymbol{I}_{2^{i-1}})][\boldsymbol{P}_{2^{n-i}}\otimes\boldsymbol{I}_{2^i}]\\&=(\boldsymbol{I}_{2^{n-i}}\boldsymbol{P}_{2^{n-i}})\otimes((\boldsymbol{H}_2\otimes\boldsymbol{I}_{2^{i-1}})\boldsymbol{I}_{2^i})\\&=\boldsymbol{I}_{2^{n-i}}\boldsymbol{P}_{2^{n-i}}\otimes(\boldsymbol{H}_2\otimes\boldsymbol{I}_{2^{i-1}})\end{aligned}$$

把上式代入(14)得：

$$M_N^{(p,i)}=(P_{2^{n+1-i}}\otimes I_{2^{i-1}})(P_{2^{n-i}}\otimes H_2\otimes I_{2^{i-1}})$$

$$=P_{2^{n+1-i}}'(P_{2^{n-i}}\otimes H_2)\otimes(I_{2^{i-1}}I_{2^{i-1}})$$

$$=[P_{2^{n+1-i}}(P_{2^{n-i}}\otimes H_2)]\otimes I_{2^{i-1}}\qquad(i=1,2,\cdots,n)$$

如果令$S_N^{(i)}=I_{2^{i-1}}\otimes P_{2^{n+1-i}}\quad(i=1,2,\cdots,n)$，那么，可得到另一个相类似的结论：

$$M_N^{(p,i)}=I_{2^{i-1}}\otimes(P_{2^{n+1-i}}(H_2\otimes P_{2^{n-i}}))\quad(i=1,2,\cdots,n)$$

用同样方法，把 Kaczmarz 矩阵$W_N^{(K)}$分解成n个同阶矩阵的乘积，这里令

$$S_N^{(i)}=I_{2^{n+1-i}}\otimes P_{2^{i-1}}$$

式中$P_{2^{i-1}}$由(12)所确定。最后的矩阵因子：

$$M_N^{(K,i)}=I_{2^{n-i}}\otimes[(H_2\otimes P_{2^{i-1}})P'_{2^i}]$$

上述的讨论说明了FWT算法是多样性，具体算法最好配合计算机完成，所有算法具有相同的算术运算次数，一个FWT的结构，在一般计算机上不依赖于编制程序的窍门，而是仅仅与比较存贮的数量要求有关。根据FFT理论，(6)的方法，不要求存贮的辅助设备，能够妥当的完成。而(10)和(11)的方法是不允许进位计算。因此，要求附加$N=2^n$存贮单元系统。

在信号处理中，为了应用计算机技术，采用FWT的算法是比较好的。

2. 矩阵的分块法

首先研究Hadamard顺序的FWT法

设

$$W_N^{(h)}=H_N=\begin{bmatrix}H_{2^{n-1}} & H_{2^{n-1}}\\ H_{2^{n-1}} & -H_{2^{n-1}}\end{bmatrix}\tag{15}$$

由(1)得：

$$\begin{pmatrix}Y_h(0)\\ \vdots\\ Y_h(2^{n-1}-1)\\ \hdashline Y_h(2^{n-1})\\ \vdots\\ Y_h(N-1)\end{pmatrix}=\frac{H_N}{N}\begin{pmatrix}X(0)\\ \vdots\\ X(2^{n-1}-1)\\ \hdashline X(2^{n-1})\\ \vdots\\ X(N-1)\end{pmatrix}$$

$$=\frac{1}{N}\left(\begin{array}{c:c} H_{2^{n-1}} & H_{2^{n-1}} \\ \hdashline H_{2^{n-1}} & -H_{2^{n-1}} \end{array}\right)\left(\begin{array}{c} X(0) \\ \vdots \\ X(2^{n-1}-1) \\ \hline X(2^{n-1}) \\ \vdots \\ X(N-1) \end{array}\right)$$

因此，有

$$\begin{pmatrix} Y_h(0) \\ \vdots \\ Y_h(2^{n-1}-1) \end{pmatrix}=\frac{H_{2^{n-1}}}{N}\begin{pmatrix} X_1(0) \\ X_1(1) \\ \vdots \\ X_1(2^{n-1}-1) \end{pmatrix},\quad \begin{pmatrix} Y_h(2^{n-1}) \\ \vdots \\ Y_h(N-1) \end{pmatrix}=\frac{H_{2^{n-1}}}{N}\begin{pmatrix} X_1(2^{n-1}) \\ \\ X_1(N-1) \end{pmatrix}$$

这里

$$X_1(l)=X(l)+X(l+2^{n-1}) \quad (l=0,1,\cdots,2^{n-1}-1)$$
$$X_1(l)=X(l-2^{n-1})-X(l) \quad (l=2^{n-1},2^{n-1}+1,\cdots,N-1)$$

利用(15)继续递推，最后可得一般形式的结果：

$$X_k(l)=X_{k-1}(l)+X_{k-1}(l+2^{n-k}) \quad (l=i\cdot 2^{n-k},\cdots,(i+1)2^{n-k}-1<2^n-1)$$
$$X_k(l)=X_{k-1}(l-2^{n-k})-X_{k-1}(l) \quad (l=(i+1)2^{n-k},\cdots,(i+2)2^{n-k}-1\leqslant 2^n-1)$$

其中$i=0,2,\cdots,2(k-1),2k,k=1,2,\cdots,n$. 同时，规定$X_0(\cdot)=X(\cdot)$.

其次研究Paley顺序的FWT法。

由线性代数知识知，对于矩阵$W_N^{(h)}$，必存在一个非奇异初等矩阵R_N或P_N，使

$$W_N^{(p)}=W_N^{(h)}R_N, \qquad W_N^{(p)}=P_NW_N^{(h)} \tag{16}$$

由(12)知

$$R_{kj}X(j)=X(\langle k\rangle)$$

从而

$$R_NX(n)=X(\langle n\rangle)$$

记$\hat{X}(n)=X(\langle n\rangle)$，其中$\langle n\rangle$表示$n$的二进制顺序的逆顺序排列。

把(16)的左边等式代入(1)得：

$$NY_p(n)=W_N^{(p)}X(n)=W_N^{(h)}R_NX(n)=W_N^{(h)}\hat{X}(n) \tag{17}$$

从(17)我们看到，二进制顺序的FWT法与Kronecker顺序的FWT法是相类似地，只是在Kronecker顺序的FWT中的$X(n)$换成$\hat{X}(n)$。其它步骤完全一样。

最后研究Kaczmarz顺序的FWT法。

由线性代数的知识知：如果对于$W_N^{(h)}$的行和列施以若干次初等变换而得到$W_N^{(K)}$，则必有非奇异矩阵P_N和R_N存在，使

$$W_N^{(K)}=P_NW_N^{(h)}R_N$$

即

$$W_N^{(K)}=P_NW_N^{(p)}$$

从而得 Kaczmarz 顺序的 FWT

$$N\boldsymbol{Y}_K(n)=\boldsymbol{P}_N\boldsymbol{W}_N^{(h)}\boldsymbol{R}_N\boldsymbol{X}(n)=\boldsymbol{P}_N\boldsymbol{W}_N^{(h)}\hat{\boldsymbol{X}}(n)$$

或

$$N\boldsymbol{Y}_K(n)=\boldsymbol{P}_N\boldsymbol{W}_N^{(p)}\boldsymbol{X}(n)$$

二、二维序列的 FWT 法

给定$N_1\times N_2$ 矩阵

$$\boldsymbol{X}(n_2,n_1)=\begin{pmatrix} X(0,0) & \cdots & X(0,N_1-1)\\ X(1,0) & \cdots & X(1,N_1-1)\\ \cdots & & \cdots \\ X(N_2-1,0) & \cdots & X(N_2-1,N_1-1)\end{pmatrix}$$

其中$N_1=2^{n_1}$，$N_2=2^{n_2}$，n_1,n_2 均为非负整数。

二維 Walsh 变换定义为：

$$\boldsymbol{Y}(n_2,n_1)=\frac{1}{N_2N_1}\boldsymbol{W}_{N_2}(n_2)\boldsymbol{X}(n_2,n_1)\boldsymbol{W}_{N_1}(n_1) \tag{18}$$

其逆变换定义为

$$\boldsymbol{X}(n_2,n_1)=\boldsymbol{W}_{N_2}(n_2)\boldsymbol{Y}(n_2,n_1)\boldsymbol{W}_{N_1}(n_1) \tag{19}$$

其中

$$\boldsymbol{W}_{N_2}(n_2)=\begin{pmatrix} Wal(0,0) & Wal(0,1) & \cdots & Wal(0,N_2-1)\\ Wal(1,0) & Wal(1,1) & \cdots & Wal(1,N_2-1)\\ \cdots & \cdots & & \\ Wal(N_2-1,0) & Wal(N_2-1,1) & \cdots & Wal(N_2-1,N_2-1)\end{pmatrix} \tag{20}$$

而$\boldsymbol{W}_{N_1}(n_1)$与上矩阵相同，只是把上矩阵中$N_2$换成$N_1$而已。

下面讨论它的快速变换法：

在(18)中，记

$$\mathbf{Y}_*(n_2,q)=\frac{1}{N_2}\boldsymbol{W}_{N_2}(n_2)\boldsymbol{X}(n_2,q) \tag{21}$$

其中 $\boldsymbol{Y}_*'(n_2,q)=[Y_*(0,q)\cdots Y_*(N_2-1,q)]$，$\boldsymbol{X}'(n_2,q)=[X(0,q)X(1,q)\cdots X(N_2-1,q)]$，$q=0,1,\cdots,N_1-1$，$\boldsymbol{W}_{N_2}(n_2)$由(20)表示。

从而(18)可改写成：

$$\boldsymbol{Y}(n_2,n_1)=\frac{1}{N_1}\boldsymbol{Y}_*(n_2,q)\boldsymbol{W}_{N_1}(n_1) \tag{22}$$

其中$\boldsymbol{W}_{N_1}(n_1)$是将(20)式中$N_2,n_2$分别换成$N_1,n_1$. $\boldsymbol{Y}(n_2,n_1)=[Y(k,q)]$，$k=0,1,\cdots,N_2-1,q=0,1,\cdots,N_1-1$.

上面的变换运算，说明二維变换或逆变换可以利用两次一維FWT来计算。第一次是关于$X(p,q)$的N_1个列的每一个列进行FWT．由(21)获得．第二次是在子序列

$\{Y_*(k,q)\}$的N_2个行的每一个行进行FWT，这个由(22)得到。

上述的运算，总共要进行N_1N_2 个FWT。如果直接进行计算,那么要进行$N_2N_1(N_2-1)+N_1N_2(N_1-1)$次加法和减法。如果用快速计算法，那么，二維 FWT 只要进行$N_1N_2(\log_2N_1+\log_2N_2)=N_1N_2\log_2N_1N_2$次加法和减法。

如果$N_1=N_2=N$，那么，(18)和(19)变成：

$$\mathbf{Y}(n,n)=\frac{1}{N^2}\boldsymbol{W}_N(n)\boldsymbol{X}(n,n)\boldsymbol{W}_N(n)$$

和

$$\boldsymbol{X}(n,n)=\boldsymbol{W}_N(n)\boldsymbol{Y}(n,n)\boldsymbol{W}_N(n)$$

例 设$n=3,N=8$,二維数据组序列为

$$\boldsymbol{X}(3,3)=\begin{vmatrix} X(0,0) & X(0,1)\cdots X(0,7) \\ X(1,0) & X(1,1)\cdots X(1,7) \\ \cdots & \cdots \\ X(7,0) & X(7,1)\cdots X(7,7)\end{vmatrix}$$

求二維 Kronecker 顺序的 FWT$\boldsymbol{Y}(3,3)$。

解 由(21)得：

$$\boldsymbol{Y}_1(3,q)=\frac{1}{8}\boldsymbol{H}_8\boldsymbol{X}(3,q),\ q=0,1,\cdots,7 \tag{23}$$

其中$\boldsymbol{H}_8$表示8×8 Hadamard 矩阵，表示为：

$$\boldsymbol{H}_8=\begin{bmatrix}\boldsymbol{H}_4 & \boldsymbol{H}_4\\ \boldsymbol{H}_4 & -\boldsymbol{H}_4\end{bmatrix}$$

根据(21)用FWT法求得变换：

$$\boldsymbol{Y}_1(k,q)=\frac{1}{8}\boldsymbol{H}_8X(k,q)\ \ (k=0,1,\cdots,7)$$

上等式可分解成：

$$\boldsymbol{Y}_1(l,q)=\frac{1}{8}\boldsymbol{H}_4\boldsymbol{X}_1(l,q)\quad (l=0,1,2,3)$$

$$\boldsymbol{Y}_1(m,q)=\frac{1}{8}\boldsymbol{H}_4\boldsymbol{X}_1(m,q)\quad (m=4,5,6,7)$$

其中$\boldsymbol{X}_1(k,q)$的元素$X_1(k,q)$和$\boldsymbol{X}(k,q)$的元素$X(k,q)$有如下的关系：

$$\left.\begin{aligned}X_1(l,q)&=X(l,q)+X(l+4,q)\quad (l=0,1,2,3)\\ X_1(l,q)&=X(l-4,q)-X(l,q)\quad (l=4,5,6,7)\end{aligned}\right\} \tag{24}$$

这里 $q=0,1,\cdots,7$，进一步计算有

$$\left.\begin{aligned}X_2(l,q)&=X_1(l,q)+X_1(l+2,q)\quad (l=0,1,4,5)\\ X_2(l,q)&=X_1(l-2,q)-X_1(l,q)\quad (l=2,3,6,7)\end{aligned}\right\} \tag{25}$$

从而

$$\left.\begin{aligned}8Y_1(l,q)&=X_2(l,q)+X_2(l+1,q)\quad (l=0,2,4,6)\\ 8Y_1(l,q)&=X_2(l-1,q)-X_2(l,q)\quad (l=1,3,5,7)\end{aligned}\right\} \tag{26}$$

这里 $q=0,1,\cdots,7$.

由(24),(25)和(26)可得:

$$8Y_1(k,q)=\sum_{i=0}^{7}(-1)^{\sum_{r=0}^{2}{}'\left[\frac{i}{2^{k_r}}\right]}X(i,q) \tag{27}$$

这里假定 $k=\sum_{r=0}^{2}{}'2^{k_r}$，"$\sum_{r=0}^{2}{}'$"表示部分项之和，如$k=5$时，$k=2^{k_0}+2^{k_2}$,而$k_0=0,k_1=1,k_2=2$. 在(27)式中,"$\sum'$"表示在$k$的表达式中,关于 k_r 那些项之和; $[x]$ 表示不超过 x 的最大整数; 如果 $k=0$,那么，令$[\cdot]=0$; $q=0,1,\cdots,7$.

在一般情况下，设 $k=\sum_{r=0}^{t-1}{}'2^{k_r}$，"$\sum'$"表示关于 k_r 的部分项之和,且 $0=k_0<k_1<k_2<\cdots<k_{t-1}=n-1$，$k_r(r=0,1,\cdots,t-1)$是整数; $t>0$ 整数，那么，

$$NY_1(k,q)=\sum_{i=0}^{N-1}(-1)^{\sum_{r=0}^{t-1}{}'\left[\frac{i}{2^{k_r}}\right]}X(i,q)$$

这里"$\sum'$" 表示 k 的表达式中,关于 k_r 那些项之和; $[x]$表示不超过 x 的最大整数。

根据(22)有

$$\boldsymbol{Y}(3,3)=\frac{1}{8}\boldsymbol{F}_1(3,3)\boldsymbol{H}_8$$

上等式可分解为:

$$\begin{bmatrix} Y(0,0)\cdots Y(0,3)\\ \cdots\quad\cdots \\ Y(3,0)\cdots Y(3,3)\end{bmatrix}=\frac{1}{8}\begin{bmatrix} Y_2(0,0)\cdots Y_2(0,3)\\ \cdots\quad\cdots \\ Y_2(3,0)\cdots Y_2(3,3)\end{bmatrix}\boldsymbol{H}_4$$

$$\begin{bmatrix} Y(4,0)\cdots Y(4,3)\\ \cdots\quad\cdots \\ Y(7,0)\cdots Y(7,3)\end{bmatrix}=\frac{1}{8}\begin{bmatrix} Y_2(4,0)\cdots Y_2(4,3)\\ \cdots\quad\cdots \\ Y_2(7,0)\cdots Y_2(7,3)\end{bmatrix}\boldsymbol{H}_4$$

$$\begin{bmatrix} Y(0,4)\cdots Y(0,7)\\ \cdots\quad\cdots \\ Y(3,4)\cdots Y(3,7)\end{bmatrix}=\frac{1}{8}\begin{bmatrix} Y_2(0,4)\cdots Y_2(0,7)\\ \cdots\quad\cdots \\ Y_2(3,4)\cdots Y_2(3,7)\end{bmatrix}\boldsymbol{H}_4$$

$$\begin{bmatrix} Y(4,4)\cdots Y(4,7)\\ \cdots\quad\cdots \\ Y(7,4)\cdots Y(7,7)\end{bmatrix}=\frac{1}{8}\begin{bmatrix} Y_2(4,4)\cdots Y_2(4,7)\\ \cdots\quad\cdots \\ Y_2(7,4)\cdots Y_2(7,7)\end{bmatrix}\boldsymbol{H}_4.$$

其中

$$Y_2(l,m)=Y_1(l,m)+Y_1(l,m+4)\quad(m=0,1,2,3)$$
$$Y_2(l,m)=Y_1(l,m-4)-Y_1(l,m)\quad(m=4,5,6,7)$$

这里$l=0,1,\cdots,7$.

进一步计算得:

$$\left.\begin{aligned}Y_3(l,m)=Y_2(l,m)+Y_2(l,m+2)\quad(m=0,1,4,5)\\ Y_3(l,m)=Y_2(l,m-2)-Y_2(l,m)\quad(m=2,3,6,7)\end{aligned}\right\} \tag{28}$$

这里$l=0,1,\cdots,7$，从而得：

$$\left.\begin{aligned}8Y(l,m)&=Y_3(l,m)+Y_3(l,m+1) \quad (m=0,2,4,6)\\ 8Y(l,m)&=Y_3(l,m-1)-Y_3(l,m) \quad (m=1,3,5,7)\end{aligned}\right\} \tag{29}$$

这里$l=0,1,2,\cdots,7$.

将(28)代入(29)得：

$$8Y(l,m)=Y_2(l,m)+Y_2(l,m+1)+Y_2(l,m+2)+Y_2(l,m+3) \quad (m=0,4)$$
$$8Y(l,m)=Y_2(l,m-1)-Y_2(l,m)+Y_2(l,m+1)-Y_2(l,m+2) \quad (m=1,5)$$
$$8Y(l,m)=Y_2(l,m-2)+Y_2(l,m-1)-Y_2(l,m)-Y_2(l,m+1) \quad (m=2,6)$$
$$8Y(l,m)=Y_2(l,m-3)-Y_2(l,m-2)-Y_2(l,m-1)+Y_2(l,m) \quad (m=3,7)$$

最后得到结果：

$$8Y(k,m)=\sum_{q=0}^{7}(-1)^{\sum_{r=0}^{2}{}'\left[\frac{q}{2^{m_r}}\right]}Y_1(k,q) \tag{30}$$

这里假定$m=\sum_{r=0}^{2}{}'2^{m_r}$，“$\sum$”表示关于$m_r$的部分项之和。且$m_0=0,m_1=1,m_2=2$；$[x]$表示不超过$x$的最大整数；如果$m=0$，那么，令$[\cdot]=0$；$k=0,1,\cdots,7$.

根据(27)和(30)有

$$8^2Y(k,m)=\sum_{q=0}^{7}\sum_{p=0}^{7}(-1)^{\sum_{r=0}^{2}{}'\left[\frac{q}{2^{m_r}}\right]+\sum_{r=0}^{2}{}'\left[\frac{p}{2^{k_r}}\right]}X(p,q)$$

这里$k,m=0,1,\cdots,7$.

在一般情况下，最后得到

$$N^2Y(k,m)=\sum_{p=0}^{N-1}\sum_{q=0}^{N-1}(-1)^{\sum_{r=0}^{t-1}{}'\left[\frac{p}{2^{k_r}}\right]+\sum_{r=0}^{t-1}{}'\left[\frac{q}{2^{m_r}}\right]}X(p,q)$$

这里的记号如上所述。

我们可用信号流程图说明第一步变换方向和过程，其中“⟶”表示前一个数乘1加到后一个数上，“--->”表示前一个数乘-1加到后一个数上。

对于第二步运算，在图中将$X(p,q)(p=0,1,\cdots,7)$相应地换成$Y_1(k,m)(k,m=0,1,\cdots,7)$，$X_i(p,q)(p,q=0,1,\cdots,7;\ i=1,2)$分别换成$Y_i(k,m)(k,m=0,1,\cdots,7;\ i=1,2)$，最后可得$Y(k,m)(k,m=0,1,\cdots,7)$.

由上例可以看到，利用两次一维的FWT法去计算二维Walsh变换法时，只需要进行$2N^2\log_2N$次运算就能达到结果。

在实际应用中，可用计算机来完成变换的计算，FWT程序控制要比FFT程序控制容易。

附信号流程图

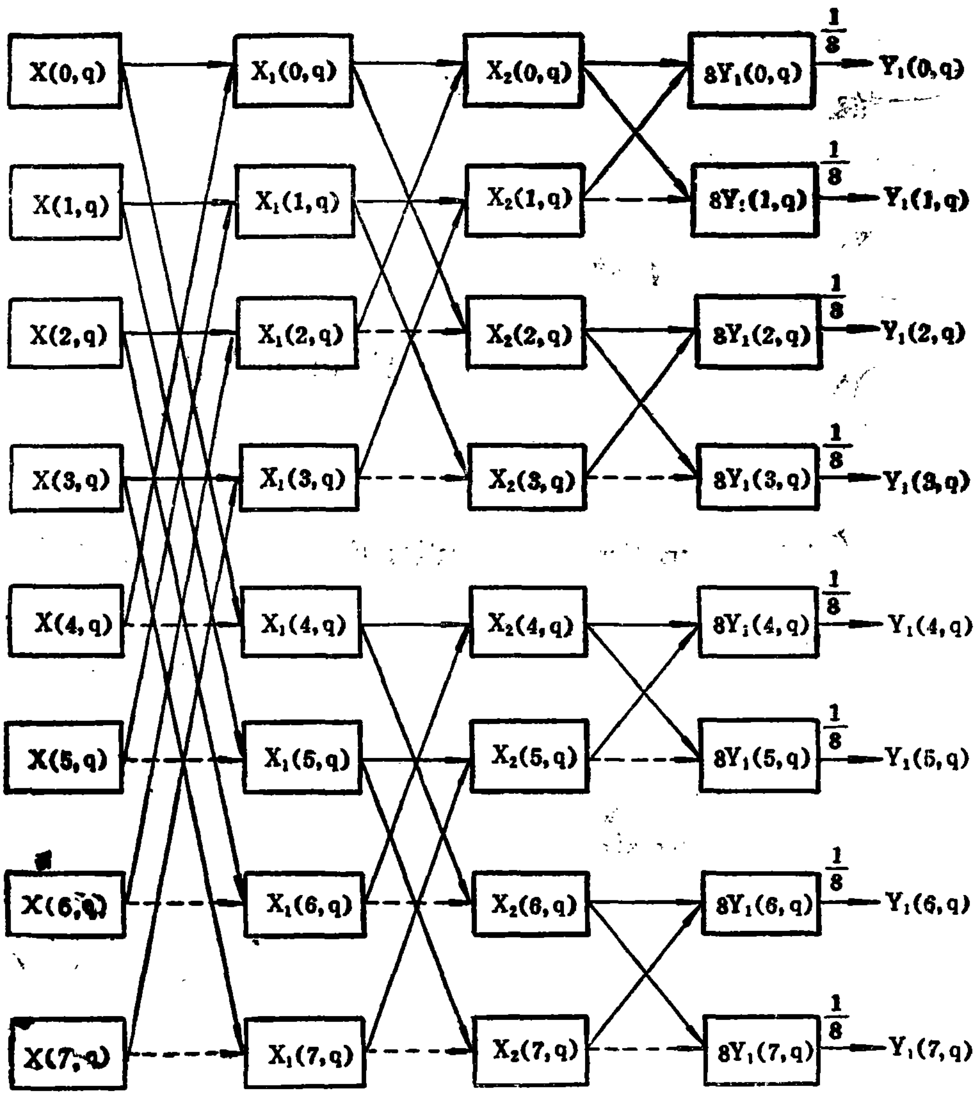

图 $N=8$时，FWT的信号流程图

参 考 文 献

[1] Rushforth, C.K.: Fast Fourier-Hadamard decoding of orthogonal codes. Information and control 15(1969),P.33—37

[2] Andrews, H.C.; Pratt, W.K.; Caspari, K.:Computer techniques in image precessing. Academic press New york/London 1970

[3] Kremer, H.:Praktische berechnung des spektrums mit der Schnellen Fourier—Transformation. elektronische datemverarbeitung 11 (1969), P.281—284

[4] Cochran, W.T. et al.: What is the Fast Fourier Transform? Proc.

IEEE 55(1967), P.1664—1674

[5] Kremer, H.: Representations and mutual relations of the different Systems walsh functions, proceedings Hatfield 1973

Fast Walsh Transform into Form of Matrix

Tang Guo-xi

Abstract

A method for systematically constructing Fast Walsh Transform algorithms is given. This method is applicable to binary-ordered Walsh—, Paley—, Sequency-ordered Walsh—Kaczmarz—and Kronecker—ordered Walsh-Hadamard Transformations. Using this method, some FWT algorithms already known from literature can be presented in a novel way by means of certain matrix factorizations. As the transformations are normally done with the aid of a digital computer, it is important to establish effective and time-saving computer algorithms.