信号的 Daubechies 小波分解与重构

Dezeming Family

2022 年 5 月 13 日

DezemingFamily 系列书和小册子因为是电子书，所以可以很方便地进行修改和重新发布。如果您获得了 DezemingFamily 的系列书，可以从我们的网站 [https://dezeming.top/] 找到最新版。对书的内容建议和出现的错误欢迎在网站留言。

目录

一　分解算法

本文是参照 [1] 来写的，但是该书在不少地方的标号和表述都出现了错误，非常影响读者理解。本文对其进行了改正，并修改了叙述方式，做到更通俗和易懂。

1 1　系数关系

V_{j+1} 空间的信号可以写为（系数长度为 N）：

$$f_{j+1}(t)=\sum_{k=0}^{N-1}a_{(j+1,k)}\phi_{(j+1,k)}(t) \tag{一.1}$$

解耦以后得到：

$$f_{j+1}(t)=\sum_{l\in\mathbb{Z}}a_{(j,l)}\phi_{(j,l)}(t)+\sum_{l\in\mathbb{Z}}c_{(j,l)}\psi_{(j,l)}(t) \tag{一.2}$$

其中：

$$\begin{aligned}a_{(j,l)}&=\sum_{k\in\mathbb{Z}}a_{(j+1,k)}h_{k-2l}\\c_{(j,l)}&=\sum_{k\in\mathbb{Z}}a_{(j+1,k)}g_{k-2l}\end{aligned} \tag{一.3}$$

由于仅在 $k\in[0,1,...,N-1]$ 时 $a_{(j+1,l)}\neq 0$，所以上式可以写为：

$$\begin{aligned}a_{(j,l)}&=\sum_{k=0}^{N-1}a_{(j+1,k)}h_{k-2l}\\c_{(j,l)}&=\sum_{k=0}^{N-1}a_{(j+1,k)}g_{k-2l}\end{aligned} \tag{一.4}$$

单独写一下 $a_{(j,l)}$：

$$a_{(j,l)}=a_{(j+1,0)}h_{-2l}+a_{(j+1,1)}h_{1-2l}+a_{(j+1,2)}h_{2-2l}+...+a_{(j+1,N-1)}h_{N-1-2l}$$

而另外，仅在 $m\in[0,1,2,...,L]$ 内 $h_m,g_m\neq 0$，也就是说要保证 $0\leq k-2l\leq L$，即 $\frac{k-l}{2}\leq l\leq\frac{k}{2}$（其实就是按照序号，$0-2l$ 要小于等于 L，而 $N-1-2l$ 要大于等于 0），最终整理可以得到 $-\frac{L}{2}\leq l\leq\frac{N-1}{2}$。注意 L 是一个奇数，所以其实 l 的范围应该写为（也是 $a_{(j,l)}$ 不为 0 的 l 下标值）：

$$l=\frac{-L+1}{2},\frac{-L+3}{2},...,0,...,\frac{N-2}{2} \tag{一.5}$$

1 2　矩阵形式

把信号的系数写为：

$$\mathbf{a}=\left[a_{(j+1,0)},a_{(j+1,1)},...,a_{(j+1,N-1)}\right]^T \tag{一.6}$$

$a_{(j,0)}$ 计算式为：

$$a_{(j,0)}=a_{(j+1,0)}h_0+a_{(j+1,1)}h_1+...+a_{(j+1,L)}h_L \tag{一.7}$$

但是，如果到后面，则需要 warp 操作。回忆 D6 小波（$L=5$），需要 Warp 的行数是 2 行。对于尺度滤波器和小波滤波器来说，需要 Warp 的行数是 $\frac{L-1}{2}$。由于 Warp，使得 $a_{(j+1,l)}$ 与不应该被归入的内容进行了乘加（我们使用 Warp 仅仅是为了构建正交矩阵，而不是真的需要 warp 操作!）。

另一个问题是，虽然 $a_{(j+1,0)}$ 的下标是从 0 开始的，但根据式子：

$$a_{(j,l)}=\sum_{k=0}^{N-1}a_{(j+1,k)}h_{k-2l}$$

最低的下标出现在 $l = \frac{-L+1}{2}$ 时（注意跟 k 值无关，k 只是累加，$K \in [0, N-1]$ 时 $a_{(j+1,k)}$ 才有值），此时系数为 $a_{(j,1-L)}$，也就是说，a_j 中不为 0 的系数的下标是从 $1-L$ 开始的。

warp 最多的一行是最后一行，一共 warp 了 $L-1$ 个值：

$$\begin{bmatrix}\underbrace{h_2 \quad h_3 \quad ... \quad h_{L-1} \quad h_L}_{L-1} & 0 & ... & 0 & h_0 & h_1\end{bmatrix} \tag{一.8}$$

因此，我们在系数前添加 $L-1$ 个 0 就可以处理这种情况了。例如 $a_{(j,\frac{1-L}{2})}$：

$$a_{(j,\frac{1-L}{2})} = \sum_{k=0}^{N-1} a_{(j+1,k)} h_{k+L-1} \tag{一.9}$$

由于 h_k 仅在 $h = [0, 1, ..., L]$ 有值，所以上式写为：

$$a_{(j,\frac{1-L}{2})} = a_{(j+1,0)} h_{L-1} + a_{(j+1,1)} h_L \tag{一.10}$$

这跟补 0 以后相乘的方式刚好能对应上。

加上小波系数的变换，我们写成矩阵形式为：

$$\mathcal{W}_{N\times N} \left[\begin{array}{c} 0 \\ 0 \\ \vdots \\ \hline a_{(j+1,0)} \\ \vdots \\ a_{(j+1,N-1)} \end{array}\right] = \left[\begin{array}{c} a_{(j,\frac{1-L}{2})} \\ \vdots \\ a_{(j,0)} \\ \vdots \\ a_{(j,\frac{N}{2}-1)} \\ \hline c_{(j,\frac{1-L}{2})} \\ \vdots \\ c_{(j,0)} \\ \vdots \\ c_{(j,\frac{N}{2}-1)} \end{array}\right] \tag{一.11}$$

补 0 以后，$\mathbf{a}_{j+1}$ 的系数为 $N+L-1$ 个。投影以后，$\mathbf{a}_j$ 的系数是 $\frac{N}{2} + \frac{L-1}{2}$ 个。如果我们想继续做分解，就得再在 $\mathbf{a}_j$ 前面补充 $L-1$ 个 0，然后使用矩阵相乘。此外，如果 $\frac{N}{2}$ 是奇数，则在下次分解时有时会去掉最后一个系数，有时会补一个 0，具体应该怎么做还是看具体实现。

怎么进行连续迭代？就是我们已经知道想要分解几次，那么我们应该如何进行分解呢？我们需要注意这里面的系数关系，$\frac{1-L}{2}$ 到 $\frac{N}{2}-1$，如果 $L=3$（D4 尺度滤波器），则序号就是 -1 到 $\frac{N}{2}-1$，即：

$$a_{(j,-1)}, a_{(j,0)}, a_{(j,1)}, a_{(j,2)}, ..., a_{(j,\frac{N}{2}-1)} \tag{一.12}$$

之前 $a_{(j+1,k)}$ 的系数是从 0 开始的，因此，注意：

$$a_{(j,0)} = a_{(j+1,0)} h_0 + a_{(j+1,1)} h_1 + ... + a_{(j+1,L)} h_L \tag{一.13}$$

所以再进行迭代的时候，我们也应该在偶数这里对齐，也就是说，将 $\mathbf{a}_j$ 设置为：

$$[0, a_{(j,-1)}, a_{(j,0)}, a_{(j,1)}, a_{(j,2)}, ..., a_{(j,\frac{N}{2}-1)}] \tag{一.14}$$

在下一小节的迭代过程中我们会更详细地讲述这个过程。

1 3 迭代过程

V_{j+1} 空间的信号（系数长度为 N）：

$$f_{j+1}(t) = \sum_{k=0}^{N-1} a_{(j+1,k)} \phi_{(j+1,k)}(t) \tag{一.15}$$

为了投影写起来更方便，我们设 $N = 2^i M$。我们希望将信号投影到 V_{j-i} 和 $W_{j-m}, m = 0, 1, ..., i$ 中去（投影 $i+1$ 次），注意：

$$\begin{aligned} V_{j+1} &= W_j \oplus V_j = W_j \oplus W_{j-1} \oplus V_{j-1} \\ &= W_j \oplus W_{j-1} \oplus \cdots \oplus W_{j-i} \oplus V_{j-i} \end{aligned}$$

假设我们要投影 i 次，则我们会在 $\mathbf{a}_{j+1}$ 前填充 $i(L-1)$ 个 0。但是，我们需要明确的是 $N + i(L-1)$ 可能不会被 2^i 整除（尽管 $N = 2^i M$ 可以被 2^i 整除）。所以，事实上我们会在前面填充 $k(L-1)$ 个 0，从而保证 $N + k(L-1)$ 能够被 2^i 整除。

举例说明一下：假设 $\mathbf{a_3}$ 的长度是 24，我们使用 D4 尺度滤波器，因此 $L = 3$。$24 = 2^3 \times 3$，所以我们投影三次。

$24 + k(L-1) = 24 + 2k$ 可以整除 $2^3 = 8$，所以 k 是 4 的倍数。假如我们选 $k = 4$，我们会得到 32 个系数的 $'\mathbf{a_3}$，这里用 $'$ 来表示补 0 以后的 $\mathbf{a_3}$。迭代中会出现什么情况？第一次迭代：

$$\begin{bmatrix} h_0 & h_1 & h_2 & h_3 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & h_0 & h_1 & h_2 & h_3 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_0 & h_1 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ & \cdots & & \cdots & & \cdots & & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & h_0 & h_1 & h_2 & h_3 \\ h_2 & h_3 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & h_0 & h_1 \end{bmatrix} \begin{bmatrix} \underbrace{0 \;\; 0 \;\; ... \;\; 0}_{8} & \underbrace{a_{(3,0)} \;\; ... \;\; a_{(3,23)}}_{24} \end{bmatrix}^T \tag{一.16}$$

乘完以后，系数前面还剩 3 个 0，因为矩阵前三排乘以 $'\mathbf{a_3}$ 是 0，后面都是有值的数了。所以，最终得到：

$$\begin{bmatrix} \underbrace{0 \;\; 0 \;\; 0}_{3} & a_{(2,-1)} & a_{(2,0)} & a_{(2,1)} & ... & a_{(2,11)} \end{bmatrix}^T \tag{一.17}$$

前面只有 3 个 0 了，注意因为第 3 个 0 的系数坐标是偶数位的。继续迭代，则问题来了，迭代完以后前面就没有为 0 的值了，因此由于 warp 的问题，第三次迭代就是错误的。

假如我们选 $k = 8$，我们会得到 40 个系数的 $'\mathbf{a_3}$,：

$$\begin{bmatrix} \underbrace{0 \;\; 0 \;\; ... \;\; 0}_{16} & \underbrace{a_{(3,0)} \;\; ... \;\; a_{(3,23)}}_{24} \end{bmatrix}^T \tag{一.18}$$

第一次迭代以后，前面会有 7 个 0；第二次迭代以后，前面会剩下 2 个 0，因此最后一次迭代则是正确的。注意 $L = 3$ 时，第一次迭代得到的坐标最小的不为 0 的 $a_{(j,l)}$ 是 $l = \frac{1-L}{2} = -1$ 是一个奇数；第二次迭代得到的坐标最小的不为 0 的 $a_{(j-1,m)}$ 是 $m = -2$（变回了偶数）；第三次迭代之前，前面恰好有 2 个 0，是非常完美的。

本节的内容相对比较绕，大家最好自己在纸上用笔写写画画，否则很容易搞晕。

1 4 塔式分解算法

上述过程，也就是我们通常所说的塔式分解算法。通过多次迭代，将信号不断地进行分解。

然而，该过程相对来说比较慢，尤其是当 N 很大时，计算过于麻烦。实际计算中，有时会把信号与滤波器的卷积放在频域中进行计算。

另外，通过双正交小波方法，可以有效避免填充 0，我们会以后再进行介绍。

二 重构算法

重构的过程其实就是分解的逆过程，乘以 $\mathcal{W}_{N\times N}^T$ 即可。可以想到的是，越乘得到的 0 就会越多，最终得到原始的补 0 后的系数。过程不再多说。

参考文献

[1] Ruch D K , Fleet P V . Wavelet Theory: An Elementary Approach with Application[M]. John Wiley & Sons, 2009.

[2] https://zh.m.wikipedia.org/zh/多贝西小波