Jim Lambers
ENERGY 281
Spring Quarter 2007-08
Lecture 10 Notes

1 FWT Example

We now illustrate the FWT with a simple example. Let the signal g be a signal defined on $[0,4)$ that is piecewise constant on each of the intervals $[x_j, x_{j+1})$, where $x_j = jh$, $j = 0, \ldots, 8$, and $h = 0.5$. The values of g on these eight subintervals are $\{0, 1, 0, 1.5, 0.5, 0, 0, 1\}$. For simplicity, we use the Haar wavelet as the mother wavelet, and the associated father wavelet, or smoothing function, is $\phi(x) = 1$ on the interval $[0, 1)$, and zero everywhere else.

One advantage of using the Haar wavelet to analyze a signal $g(x)$ is that the inner products

$$s_{m,n} = \langle \phi_{m,n}, g \rangle, \quad d_{m,n} = \langle \psi_{m,n}, g \rangle$$

can be simplified considerably, so that it is only necessary to integrate g. To see this, we consider $s_{m,n}$. Using substitutions, we obtain

$$\begin{aligned}
s_{m,n} &= \langle \phi_{m,n}, g \rangle \\
&= \int_{-\infty}^{\infty} \phi_{m,n}(x) g(x)\, dx \\
&= 2^{-m/2} \int_{-\infty}^{\infty} \phi\left(\frac{x - n2^m}{2^m}\right) g(x)\, dx \\
&= 2^{-m/2} \int_{-\infty}^{\infty} \phi\left(\frac{u}{2^m}\right) g(u + n2^m)\, du, \quad u = x - n2^m \\
&= 2^{m/2} \int_{-\infty}^{\infty} \phi\left(v\right) g((v+n)2^m)\, dv, \quad v = u/2^m \\
&= 2^{m/2} \int_0^1 g((v+n)2^m)\, dv \\
&= 2^{m/2} \int_n^{n+1} g(w2^m)\, dw, \quad w = v + n \\
&= 2^{-m/2} \int_{n2^m}^{(n+1)2^m} g(x)\, dx, \quad x = w2^m.
\end{aligned}$$

Similarly,

$$d_{m,n} = 2^{-m/2} \left[\int_{n2^m}^{(n+1/2)2^m} g(x)\, dx - \int_{(n+1/2)2^m}^{(n+1)2^m} g(x)\, dx \right].$$

When g is piecewise constant, as in this example, these integrals can be evaluated analytically.

In computing the FWT, the first question we should address is, which scales will be needed? To answer this, we observe that g is piecewise constant on intervals of width $1/2 = 2^{-1}$, which implies that $g \in V_{-1}$. From the relations between resolutions, g can be decomposed uniquely into a signal $g_0 \in V_0$ and a *detail* signal $d_0 \in W_0$, which captures the changes that are made to g_0 to obtain g. Then, because $V_0 = V_1 \oplus W_1$, g_0 can be decomposed into a signal $g_1 \in V_1$ and a detail $d_1 \in W_1$, and so on.

Because g contains eight values, we must compute eight coefficients altogether. To see where these coefficients come from, we note that $V_{-1} = W_0 \oplus W_1 \oplus \cdots \oplus W_M \oplus V_M$, where M is the index of the coarsest scale that we will use. Therefore, the eight coefficients will come from $M + 1$ details and one smoothed signal in V_M. The detail d_0 contains four coefficients, because the interval $[0, 4)$ is covered by the four wavelet functions $\psi_{0,n}$ for $n = 0, 1, 2, 3$. The next detail d_1 contains two coefficients, and the next one, d_2, contains one, for a total of seven. The eighth coefficient is obtained from the smoothed signal g_2. Therefore, $M = 2$ and three scales are needed. In general, for a signal with N points, $\log_2 N$ scales are needed to represent it completely.

We begin with $m = 0$ and apply the high-pass filter, computing

$$\begin{aligned}
d_{0,n} &= \langle \psi_{0,n}, g \rangle \\
&= \int_{-\infty}^{\infty} \psi_{0,n}(x) g(x)\, dx \\
&= \int_{-\infty}^{\infty} \psi(x - n) g(x)\, dx \\
&= \int_{n}^{n+1/2} g(x)\, dx - \int_{n+1/2}^{n+1} g(x)\, dx, \quad n = 0, 1, 2, 3.
\end{aligned}$$

Because ϕ is piecewise constant, we can easily evaluate these integrals exactly, and obtain

$$d_{0,0} = -0.5, \quad d_{0,1} = -0.75, \quad d_{0,2} = 0.25, \quad d_{0,3} = -0.5.$$

Next, we apply the low-pass filter, computing

$$\begin{aligned}
s_{0,n} &= \langle \phi_{0,n}, g \rangle \\
&= \int_{-\infty}^{\infty} \phi_{0,n}(x) g(x)\, dx \\
&= \int_{-\infty}^{\infty} \phi(x - n) g(x)\, dx
\end{aligned}$$

$$= \int_{n}^{n+1} g(x)\,dx, \quad n = 0, 1, 2, 3.$$

We obtain

$$s_{0,0} = 0.5, \quad s_{0,1} = 0.75, \quad s_{0,2} = 0.25, \quad s_{0,3} = 0.5.$$

It follows from the relation between resolutions, $V_{-1} = V_0 \oplus W_0$, that g has a unique decomposition into the sum of an element from V_0 and an element of W_0. Specifically,

$$g(x) = \sum_{n=0}^{3} s_{0,n}\phi_{0,n}(x) + \sum_{n=0}^{3} d_{0,n}\psi_{0,n}(x),$$

where the first sum is a smoothed, lower-resolution version of g, and the second sum represents the changes made to this smoothed version to obtain g.

We now continue the transform. Working with the smoothed signal

$$g_0 = \sum_{n=0}^{3} \phi_{0,n} s_{0,n} = \{0.5, 0.75, 0.25, 0.5\} \in V_0,$$

we apply the high-pass filter corresponding to $m = 1$, computing

$$\begin{aligned} d_{1,n} &= \langle \psi_{1,n}, g_0 \rangle \\ &= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2}} \psi\left(\frac{x-2n}{2}\right) g_0(x)\,dx \\ &= \frac{1}{\sqrt{2}} \int_{2n}^{2n+1} g(x)\,dx - \frac{1}{\sqrt{2}} \int_{2n+1}^{2n+2} g(x)\,dx, \quad n = 0, 1. \end{aligned}$$

We obtain

$$d_{1,0} = \frac{-0.25}{\sqrt{2}}, \quad d_{1,1} = \frac{-0.25}{\sqrt{2}}.$$

Next, we apply the low-pass filter to obtain

$$s_{1,0} = \frac{1}{\sqrt{2}} \int_{0}^{2} g(x)\,dx = \frac{1.25}{\sqrt{2}}, \quad s_{1,1} = \frac{1}{\sqrt{2}} \int_{2}^{4} g(x)\,dx = \frac{0.75}{\sqrt{2}}.$$

We now have

$$g(x) = \sum_{n=0}^{1} s_{1,n}\phi_{1,n}(x) + \sum_{n=0}^{1} d_{1,n}\psi_{1,n}(x) + \sum_{n=0}^{3} d_{0,n}\psi_{0,n}(x),$$

where the first sum is an even smoother approximation to g than g_0, the second sum describes the high-resolution changes made to this smoother signal to obtain g_0, and the third sum describes the higher-resolution detail added to g_0 to obtain g.

We continue with $m = 2$, and the smoothed signal $g_1 = \{0.625, 0.375\}$. The high-pass filter yields

$$\begin{aligned} d_{2,0} &= \langle \psi_{2,0}, g_1 \rangle \\ &= \int_{-\infty}^{\infty} \psi_{2,0}(x)\, g_1(x)\, dx \\ &= \int_{-\infty}^{\infty} \frac{1}{2}\psi\left(\frac{x}{4}\right) g_1(x)\, dx \\ &= \frac{1}{2}\int_0^2 g(x)\, dx - \frac{1}{2}\int_2^4 g(x)\, dx = 0.25, \end{aligned}$$

while the low-pass filter yields

$$\begin{aligned} s_{2,0} &= \langle \phi_{2,0}, g_1 \rangle \\ &= \int_{-\infty}^{\infty} \phi_{2,0}(x)\, g_1(x)\, dx \\ &= \int_{-\infty}^{\infty} \frac{1}{2}\phi\left(\frac{x}{4}\right) g_1(x)\, dx \\ &= \frac{1}{2}\int_0^4 g(x)\, dx = 1. \end{aligned}$$

This corresponds to the signal $g_2(x) = s_{2,0}\phi_{2,0}(x) = 0.5$ that is constant on $[0, 4)$.

The signal g_2 can be represented by a single wavelet function in W_3 provided a suitable extension for g_2 beyond the interval $[0, 4)$ is used. If we extend the signal g_2 so that it is equal to -0.5 on the interval $[4, 8)$, then we can apply the high-pass filter one more time and obtain

$$d_{3,0} = \int_{-\infty}^{\infty} \frac{1}{2\sqrt{2}}\psi\left(\frac{x}{8}\right) g_2(x)\, dx = \sqrt{2}$$

while the low-pass filter yields

$$s_{3,0} = \int_{-\infty}^{\infty} \frac{1}{2\sqrt{2}}\phi\left(\frac{x}{8}\right) g_2(x)\, dx = 0.$$

We conclude that with the extension to $[4, 8)$, $g_2 \in W_3$, and therefore $g \in W_0 \oplus W_1 \oplus W_2 \oplus W_3$, and can be expressed as

$$g(x) = \sum_{m=0}^{3} \sum_{n=-\infty}^{\infty} d_{m,n}\psi_{m,n}(x),$$

using the inverse DWT.

2 Compression and De-Noising

As the previous example shows, a discrete signal with N samples can be represented by N members of a wavelet family. However, if any of the detail coefficients are negligible, they can be omitted from the MRA, allowing the signal to be compressed without significant loss of accuracy. Also, detail coefficients at finer scales may represent the noise in a signal, especially if their magnitude is signifcant compared to what would be expected in view of the smoothness of the signal at coarser scales, so dropping them has the effect of de-noising the signal.

3 Working with Wavelets in MATLAB

In order to work with wavelets in MATLAB, you must ensure that your installation includes the Wavelet Toolbox. We now describe a few of the most useful functions from this toolbox.

- `dwt` performs a DWT. In its simplest usage, `dwt` accepts two arguments, a vector `X` consisting of the signal to be transformed, and a string `WNAME` indicating the wavelet family to use. For information about available wavelet families, consult the documentation on the `waveinfo` command. The output of `dwt` consists of two vectors, the approximation coefficients `CA` and the detail coefficients `CD`. If `X` $\in V_j$ for some j, then `CA` $\in V_{j+1}$ and and `CD` $\in W_{j+1}$. Essentially `dwt` performs one iteration of the MRA in Section 1, corresponding to a single scale. It should be noted that `dwt` uses different scaling constants than we use in these notes, but other than that the computations are identical.

- `idwt` performs inverse DWT. Given a vector of approximation coefficients `CA`, a vector of detail coefficients `CD`, and a wavelet family indicated by the string `WNAME`, `idwt` reconstructs and returns the original signal `X`.

- `wavedec` performs a complete MRA on a given signal `X` which is specified as the first argument. The second argument is a positive integer `N` indicating the number of scales that should be contained in the wavelet decomposition, and the third argument is a string `WNAME` specifies the name of the wavelet family to use. The first output argument are a vector `C` containing the approximation coefficients at the coarsest scale, followed by the detail coefficients at all requested scales, beginning from the coarsest and ending with the finest (which is one scale coarser than `X` itself). The second output argument is a vector `L` that indicates the indices within `C` at which each set of coefficients begins, for bookkeeping purposes. If you use `wavedec` with a signal `X` of length $2^{\mathtt{N}}$, with `WNAME` equal to the string `'haar'`, then the coefficients in the output vector `C` are identical to those obtained using the procedure described in Section 1, except that due to the different scaling constants used, the detail coefficients in `C` are multiplied by $\sqrt{2}$ and the approximation coefficient is multiplied by $2^{N/2}$.

- `wdencmp` performs de-noising or compression on a given signal `X`, using a given wavelet family `WNAME` and a given number of scales `N` to consider. Additional input arguments are the thresholds to use for de-noising or compression, as well as a string indicating whether the same thresholds should be used for all scales, or whether different thresholds should be used at each scale. If you do not know how to select the thresholds yourself, you can use the function `ddencmp` to select them for you.