Daubechies wavelets

Joël Karel

DAUBECHIES WAVELET FILTERS

When generating wavelets from filter banks, not all sets of filter coefficients are feasible: the coefficients $f_0, f_1, \ldots, f_n$ need to satisfy certain requirements to make that the scaling function $\phi(t)$ and the wavelet function $\psi(t)$ exhibit their characteristic properties. Note that the coefficients f_k relate to the coefficients c_k in the lecture notes as $f_k = \frac{1}{\sqrt{2}} c_k$. Common conditions are the following:

(i) The *integral of the scaling function* $\phi(t)$ must be equal to 1:

$$f_0 + f_1 + \ldots + f_{n-1} + f_n = 1.$$

(ii) The *integral of the wavelet function* $\psi(t)$ must be equal to 0:

$$f_0 - f_1 + \ldots + (-1)^{n-1} f_{n-1} + (-1)^n f_n = 0.$$

(iii) *Orthogonality* between the bases $\{\phi(t-k)\}$, $\{\psi(t-k)\}$ and the scaled bases $\{2^{-j/2}\psi(2^j t - k)\}$ derived from it in the accompanying multi-resolution structure. This is achieved with the *double-shift orthogonality* property of the filter coefficients:

$$\sum_{k=0}^{n} f_k f_{k-2\ell} = 0, \qquad \text{for all integers } \ell \neq 0.$$

Ortho*normality* of these bases (which makes that the energy, i.e. the L_2-norm, of the basis functions is equal to 1) requires that in addition the following condition is satisfied:

$$\sum_{k=0}^{n} f_k^2 = \frac{1}{2}.$$

However, it is possible to demonstrate that the latter condition is already implied (jointly) by the previous ensemble of conditions.

For the *Daubechies wavelet* family the following additional requirement holds, which uses all the freedom there is left in choosing the coefficients:

(iv) The low-pass filter $F(z) = f_0 + f_1 z^{-1} + f_2 z^{-2} + \ldots + f_n z^{-n}$ has *as many zeros as possible at* $z = -1$.

Note that condition (ii) is also of this type; it already implies that at least 1 zero of $F(z)$ is located at $z = -1$. This fourth condition means that the low-pass filter $F(z)$ has the property of *'maximal flatness'* at the high frequencies, which is widely regarded as an attractive feature for bandpass filtering.
The Daubechies db1 wavelet is generated when $F(z)$ is to be supplied with 1 zero at $z = -1$. Then $n = 1$ and there are 2 filter coefficients: f_0 and f_1. The Daubechies db2 wavelet emerges when $F(z)$ has two zeros at $z = -1$, the Daubechies db3 wavelet corresponds to three zeros at $z = -1$, and so on. For db2 one has $n = 3$ and there are 4 filter coefficients; for db3 one has $n = 5$ and there are 6 filter coefficients, and so on.

EXERCISE 1

a) Determine the filter coefficients f_0 and f_1 of the Daubechies db1 wavelet.[1] Show that they are the same as the filter coefficients of the Haar wavelet. Indeed: only the conditions (i) and (ii) apply in this case, requirement (iii) is automatically satisfied and requirement (iv) is limited to requirement (ii).

b) Determine the filter coefficients f_0, f_1, f_2, f_3 of the Daubechies db2 wavelet. In addition to the conditions (i) and (ii), now also the conditions (iii) and (iv) must hold (with *two* zeros at $z = -1$). To save you some work, condition 4 is already programmed. Uncomment the code.

c) Compute the filter coefficients $f_0, f_1, f_2, f_3, f_4, f_5$ of the Daubechies db3 wavelet. Now the requirements (iii) and (iv) each yield two conditions (three zeros of $F(z)$ are to be located at $z = -1$).

EXERCISE 2

a) Load the data in the file `dataset1.mat`. Use the function `discreteWaveletTransform` (you still have to complete it) or your own version `dwlt`; but watch out with proper scaling! to calculate a single step in the Daubechies 2 wavelet transform. This function takes the signal, the low-pass filter (e.g. the filter coefficients f) and the high-pass filter (alternating flip of f) as an input and returns the approximation coefficients and detail coefficients.

b) Calculate four scales in the multiresolution analysis of the dataset, using a "logarithmic tree" of Daubechies 2 filter banks. You can also use your version of the `wldecom` function. The output of this function should be an array C of the form: $\texttt{C} = [\texttt{app.coef.}(\texttt{J} - \texttt{j})|\texttt{det.coef.}(\texttt{J} - \texttt{j})|\ldots|\texttt{det.coef.}(\texttt{J})]$

c) Use the function `plotWaveletDecompositionMap.m` to display the wavelet decomposition of the dataset. This function takes C, the number of levels in the decomposition and the length of the dataset as an input argument.

d) Repeat b) and c) for `dataset2.mat`.

EXERCISE 3

Now you should be able to build arbitrary Daubechies wavelets. Build several Daubechies wavelet filter banks of different filter lengths and several polynomial signals and verify how polynomial signals are affected by these filter banks.

Also ensure that you can do the 2D wavelet transform with Daubechies wavelets. This sometimes pops up during exams!

[1] You can do this by constructing a function that returns an array with the deviation from each condition. This function is then used as an input argument to `fsolve` which searches the coefficients that minimize this deviation.