

Convolution with the Hartley transform

M. Rule

July 13, 2020

The [Hartley transform](#) can be computed by summing the real and imaginary parts of the Fourier transform.

$$\begin{aligned}\mathcal{F}\mathbf{a} &= \mathbf{x} + i\mathbf{y} \\ \mathcal{H}\mathbf{a} &= \mathbf{x} + \mathbf{y},\end{aligned}\tag{1}$$

where \mathbf{a} , \mathbf{x} , and \mathbf{y} are real-valued vectors, \mathcal{F} is the Fourier transform, and \mathcal{H} is the Hartley transform. It has several useful properties.

- It is unitary, and also an involution: it is its own inverse.
- Its output is real-valued, so it can be used with numerical routines that cannot handle complex numbers.
- It can be computed in $\mathcal{O}(n \log(n))$ time using standard Fast Fourier Transform (FFT) libraries.

Disadvantages are that the Hartley spectrum is less intuitive than the Fourier spectrum, and that convolutions are trickier to compute. With the Fourier transform, convolutions can be computed using the convolution theorem:

$$\mathbf{a} * \mathbf{b} = \mathcal{F}^{-1}[(\mathcal{F}\mathbf{a}) \circ (\mathcal{F}\mathbf{b})]\tag{2}$$

where \mathbf{a} and \mathbf{b} are vectors, \mathcal{F} is the Fourier transform, and \circ denotes element-wise multiplication.

Convolutions for real-valued inputs can also be computed using the Hartley transform, albeit with a bit more work. For the Fourier transform real-valued inputs, the negative frequencies are the complex conjugate of positive frequencies. That is, if the Fourier coefficient for $+\omega$ is $x + iy$, then the coefficient for $-\omega$ is $x - iy$.

For real-valued inputs, this means that positive frequencies of the Hartley transform take on values $x + y$ and negative values $x - y$. These two components are orthogonal, and can be used to recover the original real and imaginary parts of the Fourier transform.

$$\begin{aligned}[\Re(\mathcal{F}\mathbf{a})](\omega) &= \frac{1}{2} \{[\mathcal{H}\mathbf{a}](\omega) + [\mathcal{H}\mathbf{a}](-\omega)\} \\ [\Im(\mathcal{F}\mathbf{a})](\omega) &= \frac{1}{2} \{[\mathcal{H}\mathbf{a}](\omega) - [\mathcal{H}\mathbf{a}](-\omega)\}\end{aligned}\tag{3}$$

To convolve real-valued signals using the Hartley transform, convert the Hartley spectrum to the Fourier spectrum, and use the standard convolution theorem. This simplifies to:

$$\mathbf{a} * \mathbf{b} = \frac{1}{2} \mathcal{H}^{-1} \{ [\mathcal{H}\mathbf{a}](\omega)[\mathcal{H}\mathbf{b}](-\omega) + [\mathcal{H}\mathbf{a}](-\omega)[\mathcal{H}\mathbf{b}](\omega) + [\mathcal{H}\mathbf{a}](\omega)[\mathcal{H}\mathbf{b}](-\omega) - [\mathcal{H}\mathbf{a}](-\omega)[\mathcal{H}\mathbf{b}](\omega) \} \quad (4)$$

Here's an example in numpy/scipy:

```

from pylab import *

# reference convolution
p = exp(-0.5*linspace(-9,9,100)**2)
q = randn(100)

fp = fft(p,norm='ortho')
fq = fft(q,norm='ortho')
z = real(iffit(fp*fq,norm='ortho'))

# ~ fht convolution
def fht(*args):
    f = fft(*args,norm='ortho')
    return real(f) + imag(f)

reverse = lambda a:roll(a[::-1],len(a)+1)

hp = fht(p)
hq = fht(q)
rp = reverse(hp)
rq = reverse(hq)
hz = fht(hp*hq + rp*hq + hp*rq - rp*rq)/2
print(max(abs(hz - z)))

>>> 1.6653345369377348e-16

```