Workshop on Computational Imaging

Introduction to sparse signals and their applications to image denoising

Anirudh Aatresh

22 January 2020

National Institute of Technology Karnataka

- A signal is a functional representation of data, which usually varies with respect to the independent variable(s).
- Examples of 1D signals $\Rightarrow x(t) = sin(t), y(t) = e^{-2t}$.

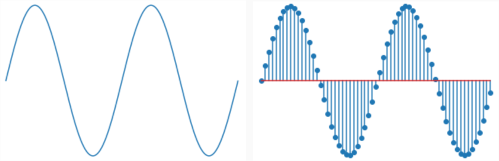


Figure 1: Examples of 1D signals.

2D Signals

- Signals that can be represented as a function of two independent variables.
- Common example – a grayscale image $\Rightarrow I(x, y)$. This is a signal that varies with position or spatial dimensions.

Figure 2: Sample grayscale image.

Sparse signals

- Sparse signals are those signals that have a limited number of non-zero elements or coefficients.

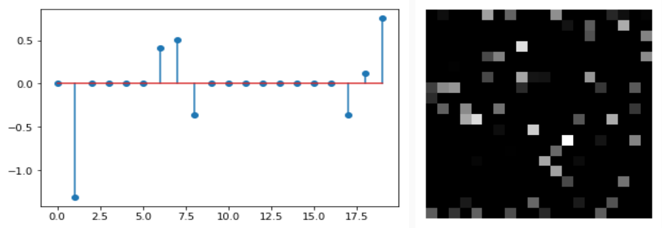


Figure 3: Sample sparse signals.

- Sparsity need not exist in the spatial or time domains, this can exist is transformed domains such as Fourier, wavelet, gradient etc.

Importance of Sparse signals

- Plays an important role in the design of signal compression algorithms such as compressed sensing.
- It has been observed that sparsity is highly desireable in neural networks. This has influenced the formulation and choice of activation functions.
- Signal denoising – sparsity in the gradient or wavelet domains is a desirable property in clean signals.

- Fourier transform yields the frequency domain representation of a signal, but is unable to localize the representation.

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} \tag{1}$$

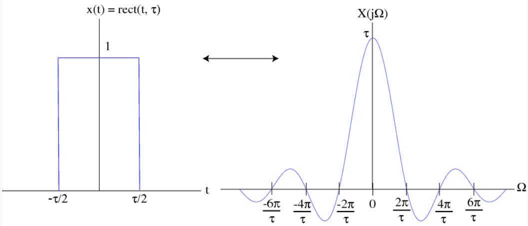


Figure 4: Fourier transform of a rectangular pulse. Source: cnx.org.

- Short time fourier transform is an alternative that suffers from the time-frequency trade off.

$$X(\omega, \tau) = \int_{-\infty}^{\infty} x(t)\phi(t-\tau)e^{-j\omega t} \tag{2}$$

$$\Delta t \Delta f \geq \frac{1}{4\pi} \tag{3}$$

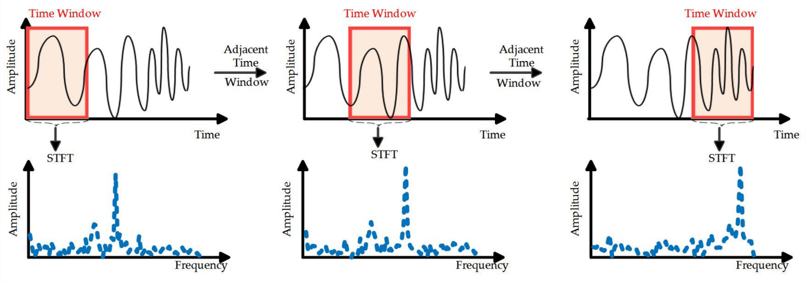
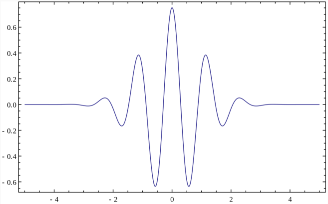


Figure 5: Short time fourier transform.

$$X(s,\tau) = \frac{1}{\sqrt{|s|}} \int_{-\infty}^{\infty} x(t)\psi^*(\frac{t-\tau}{s})dt \tag{4}$$



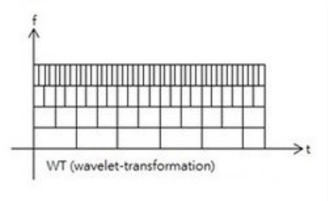


Figure 6: (Left) Morlet wavelet. (Right) Improvement provided by DWT.

- The discrete wavelet transform is equivalent to passing the original signal through a number of filters or filter banks.

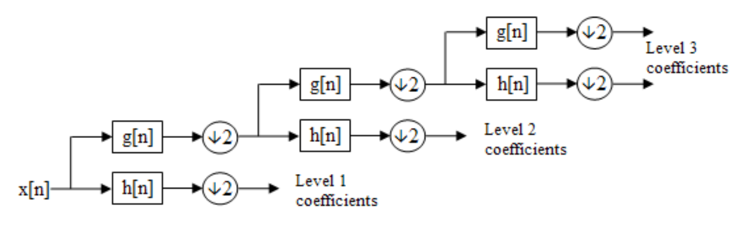


Figure 7: 1D DWT.

2D DWT

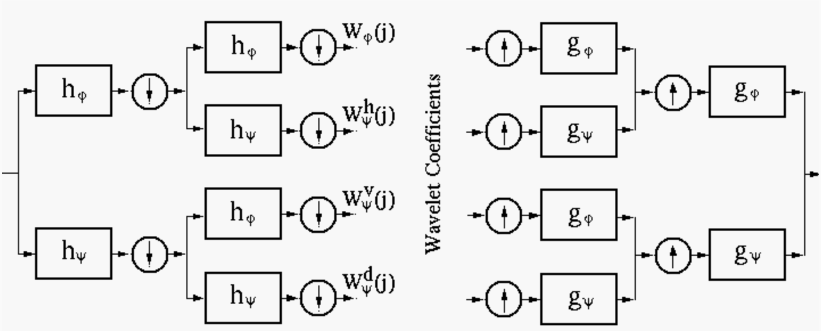


Figure 8: 2D DWT and IDWT. Source: fourier.eng.hmc.edu/

How do we enforce the sparsity?

- Thresholding with a carefully chosen threshold will cause the signal to become sparse.
- Hard-threshold:

$$\textit{hard threshold}(X) = \begin{cases} X & |X| \geq T \\ 0 & |X| < T \end{cases}$$

- The abruptness of hard thresholding causes artifacts in the solution that is not desired.
- Effect of hard thresholding is also difficult to control using the threshold value.

- Better alternative – soft-thresholding:

$$soft\ threshold(X) = \begin{cases} sgn(X)(|X| - T) & |X| \geq T \\ 0 & |X| < T \end{cases}$$

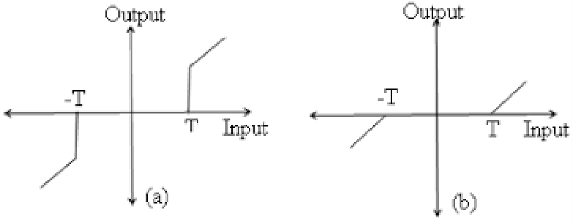


Figure 9: (Left) Hard and (Right) soft thresholding functions. Source: scielo.org.

Hard and soft thresholding continued

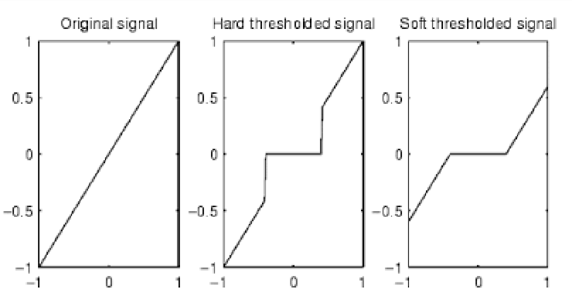


Figure 10: Hard and soft thresholding functions. Source: wthresh wavelet toolbox.

- Using the soft-thresolding function, we can apply the iterative shrinkage thresholding algorithm (ISTA) – repeated application of soft-thresholding in the appropriate domain.

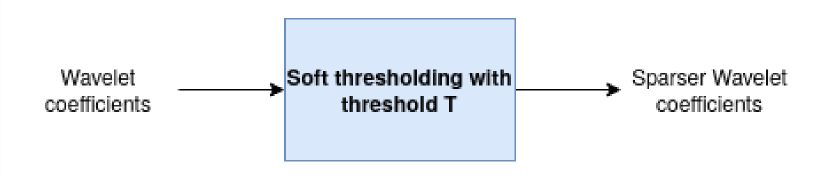


Figure 11: Method to obtain sparsity in wavelet domain.

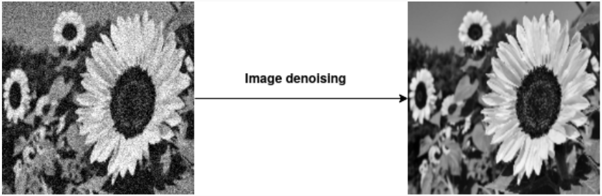


Figure 12: Image denoising overview.

- If $\mathbf{y}$ is the observed noisy image, we can model it as: $\mathbf{y} = \mathbf{x} + \mathbf{e}$, $\mathbf{e}$ is drawn from $\mathcal{N}(0, \sigma^2\mathbf{I})$. The denoising process can be defined as:

$$J(\mathbf{x}) = \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{W}\mathbf{x}\|_1 \tag{5}$$

$$\mathbf{x}^* = \text{argmin}_{\mathbf{x}}\{\frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{W}\mathbf{x}\|_1\} \tag{6}$$

- Consider a cost function $J(w)$ as a quadratic function in w

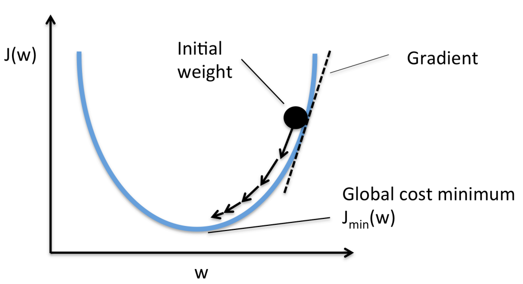


Figure 13: 1D gradient descent in action. Source: rasbt.github.io.

- $w^* = \text{argmin}_w\{J(w)\}$.
- At the kth update step:

$$w_k = w_{k-1} - \alpha \frac{\partial J(w_{k-1})}{\partial w_{k-1}} \quad (7)$$

1. Initialize x_0,N,α.
2. for $k = \{1, 2, 3, ..., N\}$:
   2.1 Calculate $g = \frac{\partial J(x_{k-1})}{\partial x_{k-1}}$.
   2.2 Weight update step 1: $z_k = x_{k-1} - \alpha g$.
   2.3 Convert z_k to wavelet domain W_{z_k}. $W_{z_k} = \mathrm{DWT}(z_k)$.
   2.4 Compute $q_k = \text{soft threshold}(W_{z_k})$.
   2.5 Weight update step 2: $x_k = \mathrm{IDWT}(q_k)$.

Questions?

A. Beck and M. Teboulle.
A fast iterative shrinkage-thresholding algorithm for linear inverse problems.
SIAM J. Img. Sci., 2(1):183–202, Mar. 2009.

A. Chambolle, R. A. De Vore, Nam-Yong Lee, and B. J. Lucier.
Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelet shrinkage.
IEEE Transactions on Image Processing, 7(3):319–335, 1998.

I. Daubechies, M. Defrise, and C. Mol.
An iterative thresholding algorithm for linear inverse problems with a sparsity constraint.
Communications on Pure and Applied Mathematics, 57:1413–1457, 2003.

M. A. T. Figueiredo and R. D. Nowak.
An em algorithm for wavelet-based image restoration.
IEEE Transactions on Image Processing, 12(8):906–916, 2003.