Wavelet Transform

Wavelet transform

- General form $CWT_x^{\psi}(\tau, s) = \Psi_x^{\psi}(\tau, s) = \frac{1}{\sqrt{|s|}} \int x(t)\psi(\frac{t-\tau}{s})dt$

$$x(t) = \int CWT(\tau, s)\psi(\frac{t-\tau}{s})d\tau$$

Inverse wavelet transform

  – CWT: Continuous Wavelet Transform
  – *s*: Scale (1/frequency)
  – *τ*: Translation (shifting on time axis)
  – *ψ*: Wavelets; $\psi(\frac{t-\tau}{s})$ form an orthonormal basis (check with Haar wavelets!)

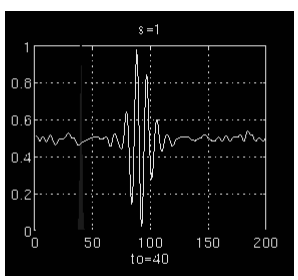
Wavelet Transform (cont'd)

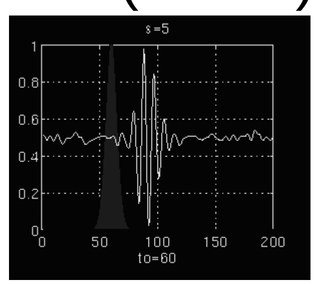
- Advertisement
  – WT does **time** *and* **frequency** localization (*F*ourier *T*ransform can only do frequency)
  – WT does multi-resolution analysis (*S*hort *T*ime *F*ourier *T*ransform only does one resolution at a time)
  – WT resolves **time** better at **high frequency**; and resolves **frequency** better at **lower frequency**

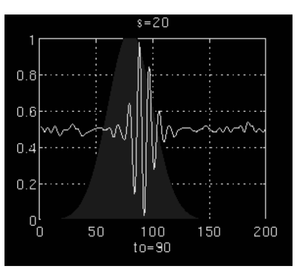
Wavelet Transform (cont'd)

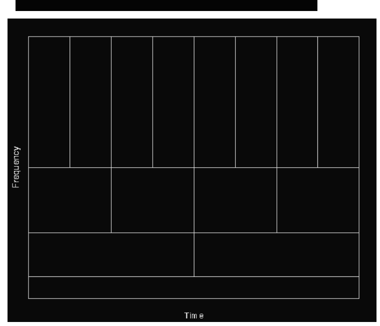
- Question (Jian): From 5.9 I know that both Wavelet & Fouries Transformation are aimed to approximate a function with a series basis functions. However, what is the main difference between these two transformations, and under what circumstances should we use one instead of the other?
  - Answer: see the previous (and the following) slides.

Wavelet Transform (cont'd)



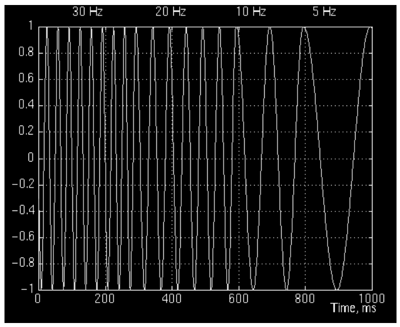


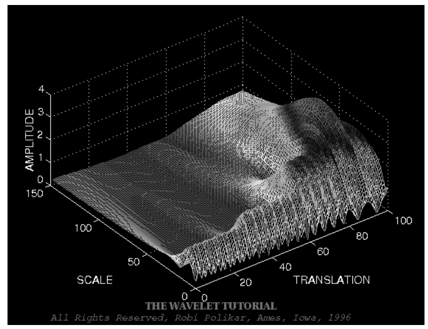




Images taken from (a great wavelet tutorial!):
http://engineering.rowan.edu/~polikar/WAVELETS/WTtutorial.html

Wavelet Transform (cont'd)





A great tutorial of wavelet transform is at:
http://engineering.rowan.edu/~polikar/WAVELETS/WTtutorial.html

Filtering WT

Multi-resolution analysis

- General form
  - WT: $y^* = \underline{W_{N\times N}^T} y$

Projection matrix! Remember this? $W^T W\theta = W_{N\times N}^T y$

  - Inverse WT: $\hat{f} = W_{N\times N}\, \hat{\theta}_{N\times 1}$

Signal reconstruction

- SURE shrinkage (similar to lasso)

$$\min_{\theta} \|y - W\theta\|_2^2 + 2\lambda \|\theta\|_1$$

  - And W is *orthonormal*! We have

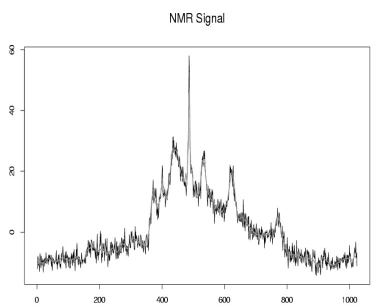
$$\hat{\theta}_j = \mathrm{sign}(y_j^*)(|y_j^*| - \lambda)_+$$

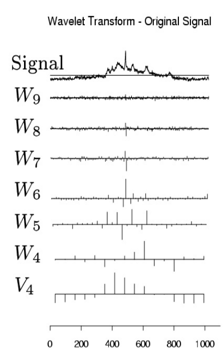
Discard the ones below λ; λ = 0 discards nothing

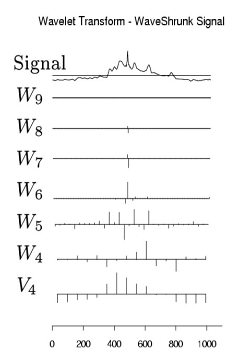
Filtering WT (cont'd)

- What value to choose for *λ*?
  - Answer: usually $\sigma\sqrt{2\log N}$
  - Why?
    - If y are white noise ($N(0;\sigma^2)$), after an orthonormal transformation (*W*), we still get white noise (y^*).
    - And for a set of *N* white noise signals, the max. expected value is $\sigma\sqrt{2\log N}$.
    - And we want to eliminate all of that!

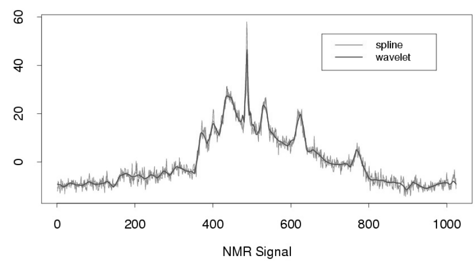
Filtering WT (cont'd)

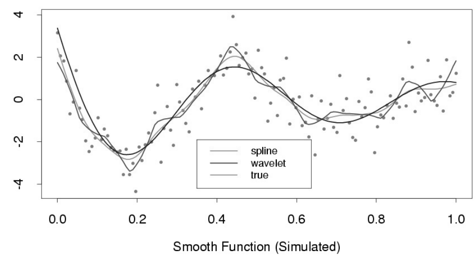






Smoothing Spines and Wavelets





Smoothing Spines and Wavelets (cont'd)

- Question (Jie): P155, first paragraph, "smoothing spline introduces detail everywhere in order to capture the detail in the isolated spikes", why did this happen? Could you give an intuitive interpretation of how wavelet smoothing localizes time and frequency to represent both smooth and/or locally bumpy functions?

Smoothing Spines and Wavelets (cont'd)

- Questions (Weng-keen):
  - In general, what are the pros and cons of using wavelets over other types of smoothing techniques? I can see how the multiresolution analysis allowed by using wavelets plus how the time and frequency localization are beneficial. However, it seems like wavelets aren't as compact to represent since you need to know all the coefficients at all the levels of resolution.

    Also, is wavelet smoothing much slower that other smoothing techniques?

  - Most of the applications of wavelets that I've seen are for one dimensional data. How does the complexity of wavelet smoothing change as you increase the dimensionality of the data? 5.9.2 mentions that wavelets are especially good when the data are measured on a uniform lattice. However, as dimensionality increases, the number of points on the lattice increases exponentially.