Denoising via Wavelet Transforms

Katie McCaffrey

Advisor: Pat Van Fleet

Overview

- Explanation of Denoising Algorithm
- Wavelet Shrinkage Methods
  - VisuShrink
  - SureShrink
- Mathematica Demo
- Other Possible Methods

Denoising Algorithm

Begin with a noisy vector $\mathbf{y} = \mathbf{v} + e$, where $\mathbf{v}$ is the true signal and $\mathbf{e}$ is Gaussian white noise.

Step 1: Compute i iterations of the wavelet transform on $\mathbf{y}$, obtaining the new vector $\mathbf{z}$, made of lowpass portion l and highpass portion d.

Step 2: Apply threshold rule to highpass portion d of $\mathbf{z}$, either “shrinking” the value or setting it equal to zero.

Step 3: Rejoin the modified highpass portion to the original lowpass, creating the modified vector $\check{\mathbf{z}}$**.**

Step 4: Compute i iterations of the inverse wavelet transform on $\check{\mathbf{z}}$ to obtain $\check{\upsilon}$.

The N-vector $\check{\upsilon}$ should be the denoised version of y.

Threshold Rules

- Hard Threshold Rule
  - Choose value for λ, the threshold value.
  - If $|x_k| < \lambda$, shrink x_k to 0.
  - If $|x_k| > \lambda$, keep x_k.
- Soft Threshold Rule
  - If $|x_k| < \lambda$, shrink x_k to 0.
  - If $|x_k| > \lambda$, shrink x_k by λ.
  - For $|x_k| > \lambda$, this is a linear function $f(\lambda) = |x_k| - \lambda$
- Let's try it on the vector [1, 1.2, 3, 6, 2.4, 5, 1.1] with $\lambda = 2$.
  - Hard Threshold Rule: [0, 0, 3, 6, 2.4, 5, 0]
  - Soft Threshold Rule: [0, 0, 1, 4, .4, 3, 0]

Accurate Shrinkage

- How do we choose an appropriate value of λ for a data set?
  - VisuShrink
  - SureShrink
  - Modifications?

VisuShrink

- Remember, **y** = **v** + **e**, where **e** is Gaussian white noise that is normally distributed with mean 0 and variance σ^2 and noise level σ.
- We want to minimize the mean squared error found by Donoho and Johnstone
- Universal threshold $\lambda^{univ} = \sigma\sqrt{2\ln(N)}$, where N is the number of values in **y**.
- As N -> ∞ the Median Absolute Deviation of the highpass values converges to .6745σ, so the estimate for σ is MAD(d)/.6745.
- Let's see this at work in Mathematica…

Threshold Changes

- Still using the VisuShrink method, I tested other possible threshold functions (other than linear)
  - Quadratic
  - Cubic
  - Firm Threshold
- Using 4 data sets that I added artificial noise to, I measured the error in each of the cases, resulting in the following values
- Results

	1	2	3	4
Linear	**.127**	.0420	.1051	**.0082**
Quad	.158	.0423	.1027	.0096
Cubic	.139	.0393	.1079	.0097
Firm	.163	**.0392**	**.0828**	.0088

SureShrink

- Stein's Unbiased Risk Estimator removes the error obtained by λ^{univ} which depends on the size of the data set
- Minimize the mean squared error by minimizing the function

$$f(\lambda) = N + ||g(x)||^2 + 2\Sigma d/dx_k(g_k(x))$$

where $g_k(x)$ is the threshold function minus the value for each value of x_k, $k = 1, 2, \ldots N$

- Once this value for λ^{sure} has been chosen, use it in the original threshold function to shrink the highpass portion and continue the process of denoising

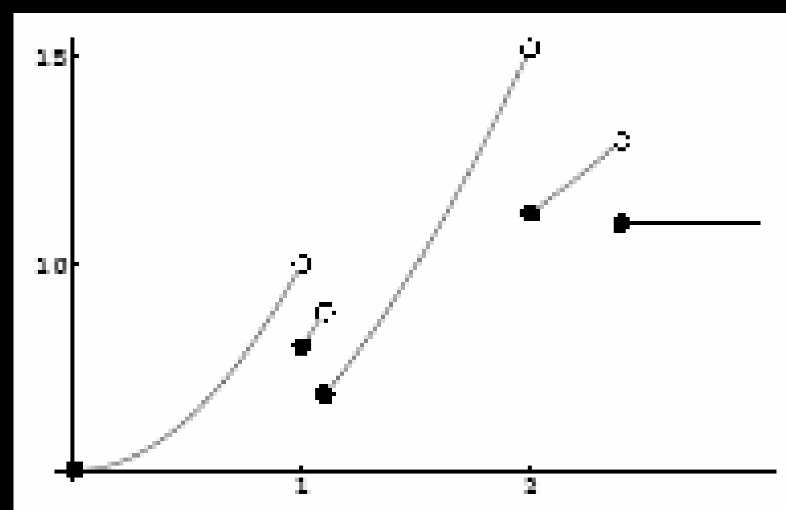
SureShrink

- With the linear soft threshold function, the function simplifies nicely as

$$f(\lambda) = N - 2 \cdot \#\{k : |x_k| \le \lambda\} + \Sigma \min(x_k^2, \lambda^2)$$

- This gives different functions of λ between x_k and x_{k+1} *with a minimum at x_k each time*
- Example: $\mathbf{x} = [1, 1.1, 2, 2, 2.4]^T$
  - For $1 < \lambda \le 1.1$, $f(\lambda) = 5 - 2 \cdot 1 + 1^2 + 4\lambda^2$

$$f(\lambda) = \begin{cases} 5 + 5\lambda^2, & 0 \le \lambda < 1; \\ 4 + 4\lambda^2, & 1 \le \lambda < 1.1; \\ 3.21 + 3\lambda^2, & 1.1 \le \lambda < 2; \\ 7.21 + \lambda^2, & 2 \le \lambda < 2.4; \\ 10.97, & 2.4 \le \lambda \end{cases}$$



- Back to Mathematica…

Alternative Thresholds

- But what about different threshold functions?
  - Quadratic; $f(\lambda)$
  - Firm threshold; $h(\lambda_1, \lambda_2)$
- Goal: Find and minimize f and h

Finding Equations

- $f(\lambda) = N + ||g(x)||^2 + 2\Sigma d/dx_k(g_k(x))$
  - Quadratic: $s(x_k) = (|x_k| - \lambda)^2, \ |x_k| > \lambda;$

    $0, \ |x_k| < \lambda$
  - Firm: $0, \ |x_k| < \lambda_1;$

    $s(x_k) = \mathrm{sgn}(x_k)(\lambda_2/(\lambda_2 - \lambda_1))(|x_k| - \lambda_1), \ \lambda_1 < |x_k| < \lambda_2;$

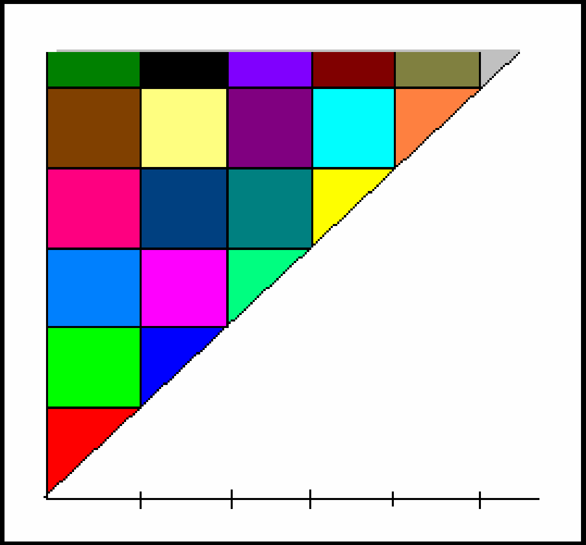
    $x_k, \ |x_k| > \lambda_2$

Currently…

- The example I used with the linear threshold function had 5 cases, one for each value of x_k in place for λ.
  - $h(\lambda_1, \lambda_2)$ has many more options because of the possible combinations of x_k's.
- Each combination provides an equation to minimize at the values of x_k
- Currently, I have found the cases for several vectors and am in the process of finding the minimum values for λ_1 and λ_2
- After all the work, it boils down to using Multivariable Calculus!

Current Place

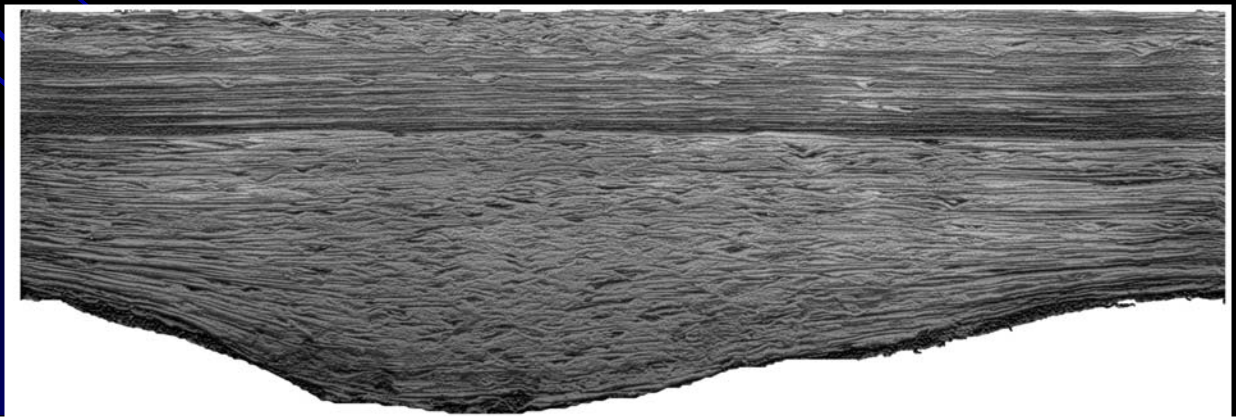
- Vectors:
  - [1, 2, 3, 4, 5] 21 cases
  - [1, 2, 3] 10 cases
  - [1, 1, 2, 3], [1, 2, 2, 3], [1, 2, 3, 3] 10 cases



- Current Questions:
  - In this graph, which cases are on the diagonal?
  - How do repeated vector values effect the resultant λ_1 and λ_2?
  - How do λ_1 and λ_2 from smaller repeated values differ from λ_1 and λ_2 from larger ones?

Other Applications

- Some other applications of denoising via wavelet transformations
  - De-noise audio files
  - Detect edges
  - Increase quality of digital images
- My applications in Geology
  - Cross-section of a sedimentary deposit from St. Anthony Falls Lab "Jurassic Tank"
  - How were the sediments deposited?
    - Patterns in cross-section, channel beds, etc.



Questions?