Undergraduate Research Projects on Wavelet-Based Time Series Forecasting

John Merkel

Oglethorpe University
Atlanta, Georgia USA
jmerkel@oglethorpe.edu

Joint Mathematics Meetings, 2012

Outline

Outline

Outline

Outline

Outline

Outline

Wavelets Workshop

In July 2011, I attended the

Discrete Wavelet Module-Writing Workshop

Workshop Leaders:

- Patrick Van Fleet, University of St. Thomas
- Catherine Beneteau, University of South Florida
- Caroline Haddad, SUNY Geneseo
- David Ruch, Metropolitan State College of Denver.

Supported by: MAA, IMA, PREP and NSF

Module and Team

Module: Use wavelets to forecast oil prices

Da Team:

- Bruce Atwood (our coding hero)
- Caroline Haddad
- Helmut Knaust
- John Merkel

Student Project

- Apply techniques from workshop project to forecast stock prices.
- Main resources:
  - “Wavelet-based Prediction of Oil Prices" by Yousefi, Weinreich, Reinarz.
  - Bruce Atwood’s Mathematica Code.

Stationary Wavelet Transform (SWT)

- Let W = SWT
- $W\mathbf{x}$ has 2 parts:
  - H - lo pass (calculates averages)
  - G - hi pass (calculate differences)
- Let $\mathbf{x} = \langle 9, 8, 11, 9, 12, 14, 10, 11 \rangle$ (data, length 8)
  - $H\mathbf{x} = \langle 10, 8.5, 9.5, 10, 10.5, 13, 12, 10.5 \rangle$ (note 1^{st} *point*)
  - $G\mathbf{x} = \langle -1, -0.5, -1.5, -1, -1.5, -1, -2, -0.5 \rangle$

Stationary Wavelet Transform (SWT)

- Let W = SWT
- $W\mathbf{x}$ has 2 parts:
  - H - lo pass (calculates averages)
  - G - hi pass (calculate differences)
- Let $\mathbf{x} = \langle 9, 8, 11, 9, 12, 14, 10, 11 \rangle$ (data, length 8)
  - $H\mathbf{x} = \langle 10, 8.5, 9.5, 10, 10.5, 13, 12, 10.5 \rangle$ (note 1^{st} *point*)
  - $G\mathbf{x} = \langle -1, -0.5, -1.5, -1, -1.5, -1, -2, -0.5 \rangle$

Stationary Wavelet Transform (SWT)

- Let W = SWT
- $W\mathbf{x}$ has 2 parts:
  - H - lo pass (calculates averages)
  - G - hi pass (calculate differences)
- Let $\mathbf{x} = \langle 9, 8, 11, 9, 12, 14, 10, 11 \rangle$ (data, length 8)
  - $H\mathbf{x} = \langle 10, 8.5, 9.5, 10, 10.5, 13, 12, 10.5 \rangle$ (note 1^{st} *point*)
  - $G\mathbf{x} = \langle -1, -0.5, -1.5, -1, -1.5, -1, -2, -0.5 \rangle$

Stationary Wavelet Transform (SWT)

- Let W = SWT
- $W\mathbf{x}$ has 2 parts:
  - H - lo pass (calculates averages)
  - G - hi pass (calculate differences)
- Let $\mathbf{x} = \langle 9, 8, 11, 9, 12, 14, 10, 11 \rangle$ (data, length 8)
  - $H\mathbf{x} = \langle 10, 8.5, 9.5, 10, 10.5, 13, 12, 10.5 \rangle$ (note 1$^{\text{st}}$ *point*)
  - $G\mathbf{x} = \langle -1, -0.5, -1.5, -1, -1.5, -1, -2, -0.5 \rangle$

Stationary Wavelet Transform (SWT)

- Let W = SWT
- $W\mathbf{x}$ has 2 parts:
  - H - lo pass (calculates averages)
  - G - hi pass (calculate differences)
- Let $\mathbf{x} = \langle 9, 8, 11, 9, 12, 14, 10, 11 \rangle$ (data, length 8)
  - $H\mathbf{x} = \langle 10, 8.5, 9.5, 10, 10.5, 13, 12, 10.5 \rangle$ (note 1^{st}*point*)
  - $G\mathbf{x} = \langle -1, -0.5, -1.5, -1, -1.5, -1, -2, -0.5 \rangle$

Iterating the SWT

- To apply SWT to $\mathbf{x}$ twice calculate $H(H\mathbf{x})$, $G(H\mathbf{x})$
- Three times, continue to work with lopass: $H(H^2\mathbf{x})$, $G(H^2\mathbf{x})$
- Each iteration yields a new set of lopass data and hipass data
- After n iterations we have $n+1$ data sets
    - hipass: $GH\mathbf{x}$, $GH^2w\mathbf{x}$, $\cdots$, $GH^n\mathbf{x}$
    - lopass: $H^n\mathbf{x}$

Procedure

1. Download closing ticker prices (10-20 years, e.g. Yahoo!)
2. To forecast close on date $d+1$:
    1. Grab sample of closing data $d-n$ to d ($n \simeq 100$ to 1000)
    2. Smooth data (optional, can use wavelets)
    3. Apply $m = 1$ to ? iterations of a wavelet transform (stationary or not)
    4. Extrapolate at each level (m hi-pass, 1 lo-pass), adding 1 data point
    5. Apply inverse wavelet transform to get time series w/ forecast on end
3. Repeat for $d+2, d+3, \cdots$

Procedure

1. Download closing ticker prices (10-20 years, e.g. Yahoo!)
2. To forecast close on date $d + 1$:
    1. Grab sample of closing data $d - n$ to d ($n \simeq 100$ to 1000)
    2. Smooth data (optional, can use wavelets)
    3. Apply $m = 1$ to ? iterations of a wavelet transform (stationary or not)
    4. Extrapolate at each level (m hi-pass, 1 lo-pass), adding 1 data point
    5. Apply inverse wavelet transform to get time series w/ forecast on end
3. Repeat for $d + 2, d + 3, \cdots$

Procedure

1. Download closing ticker prices (10-20 years, e.g. Yahoo!)
2. To forecast close on date $d + 1$:
   1. Grab sample of closing data $d - n$ to d ($n \simeq 100$ to 1000)
   2. Smooth data (optional, can use wavelets)
   3. Apply $m = 1$ to ? iterations of a wavelet transform (stationary or not)
   4. Extrapolate at each level (m hi-pass, 1 lo-pass), adding 1 data point
   5. Apply inverse wavelet transform to get time series w/ forecast on end
3. Repeat for $d + 2, d + 3, \cdots$

Procedure

1. Download closing ticker prices (10-20 years, e.g. Yahoo!)
2. To forecast close on date $d + 1$:
   1. Grab sample of closing data $d - n$ to d ($n \simeq 100$ to 1000)
   2. Smooth data (optional, can use wavelets)
   3. Apply $m = 1$ to ? iterations of a wavelet transform (stationary or not)
   4. Extrapolate at each level (m hi-pass, 1 lo-pass), adding 1 data point
   5. Apply inverse wavelet transform to get time series w/ forecast on end
3. Repeat for $d + 2, d + 3, \cdots$

Procedure

1. Download closing ticker prices (10-20 years, e.g. Yahoo!)
2. To forecast close on date $d + 1$:
   1. Grab sample of closing data $d - n$ to d ($n \simeq 100$ to 1000)
   2. Smooth data (optional, can use wavelets)
   3. Apply $m = 1$ to ? iterations of a wavelet transform (stationary or not)
   4. Extrapolate at each level (m hi-pass, 1 lo-pass), adding 1 data point
   5. Apply inverse wavelet transform to get time series w/ forecast on end
3. Repeat for $d + 2, d + 3, \cdots$

Procedure

1. Download closing ticker prices (10-20 years, e.g. Yahoo!)
2. To forecast close on date $d+1$:
   1. Grab sample of closing data $d-n$ to d ($n \simeq 100$ to 1000)
   2. Smooth data (optional, can use wavelets)
   3. Apply $m = 1$ to ? iterations of a wavelet transform (stationary or not)
   4. Extrapolate at each level (m hi-pass, 1 lo-pass), adding 1 data point
   5. Apply inverse wavelet transform to get time series w/ forecast on end
3. Repeat for $d+2, d+3, \cdots$

Procedure

1. Download closing ticker prices (10-20 years, e.g. Yahoo!)
2. To forecast close on date $d + 1$:
   1. Grab sample of closing data $d - n$ to d ($n \simeq 100$ to 1000)
   2. Smooth data (optional, can use wavelets)
   3. Apply $m = 1$ to ? iterations of a wavelet transform (stationary or not)
   4. Extrapolate at each level (m hi-pass, 1 lo-pass), adding 1 data point
   5. Apply inverse wavelet transform to get time series w/ forecast on end
3. Repeat for $d + 2, d + 3, \cdots$

Procedure

1. Download closing ticker prices (10-20 years, e.g. Yahoo!)
2. To forecast close on date $d + 1$:
   1. Grab sample of closing data $d - n$ to d ($n \simeq 100$ to 1000)
   2. Smooth data (optional, can use wavelets)
   3. Apply $m = 1$ to ? iterations of a wavelet transform (stationary or not)
   4. Extrapolate at each level (m hi-pass, 1 lo-pass), adding 1 data point
   5. Apply inverse wavelet transform to get time series w/ forecast on end
3. Repeat for $d + 2, d + 3, \cdots$

Data: Caterpillar - NYSE: CAT

2635	2.49
2636	2.56
2637	2.57
2638	2.57
2639	2.6
2640	2.58
2641	2.6
2642	2.63
2643	2.65
2644	2.67
2645	2.67
2646	2.66
2647	2.68
2648	2.71
2649	2.71
2650	2.78
2651	2.84
2652	2.8
2653	2.78
2654	2.75

To forecast 2650th point grab data from 2050-2649 (600 points)

Smooth Data (optional)

- Apply m iterations of SWT to $\mathbf{x}$ (data)
- Zero out coefficients in the m hipass data sets that are below some (small) threshold value (i.e. get rid of the noise)
- Apply inverse SWT to recover smoothed data
- Mathematica has a built in thresholding function. I used the *universal thresholding* option

Smooth Data (optional)

- Apply m iterations of SWT to $\mathbf{x}$ (data)
- Zero out coefficients in the m hipass data sets that are below some (small) threshold value (i.e. get rid of the noise)
- Apply inverse SWT to recover smoothed data
- Mathematica has a built in thresholding function. I used the *universal thresholding* option

Smooth Data (optional)

- Apply m iterations of SWT to $\mathbf{x}$ (data)
- Zero out coefficients in the m hipass data sets that are below some (small) threshold value (i.e. get rid of the noise)
- Apply inverse SWT to recover smoothed data
- Mathematica has a built in thresholding function. I used the *universal thresholding* option

Smooth Data (optional)

- Apply m iterations of SWT to $\mathbf{x}$ (data)
- Zero out coefficients in the m hipass data sets that are below some (small) threshold value (i.e. get rid of the noise)
- Apply inverse SWT to recover smoothed data
- Mathematica has a built in thresholding function. I used the *universal thresholding* option

Apply SWT, Extrapolate

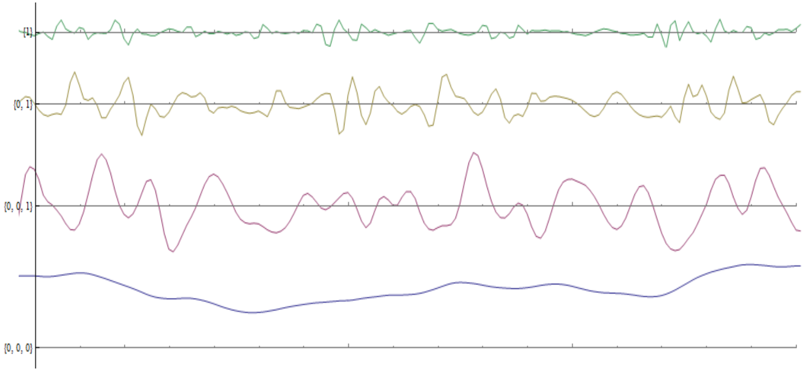

Figure: Wavelet Decomposition

We extrapolated using interpolating polynomial of degree $p = 1, 2, 3$ (a parameter we can play with)

Apply Inverse SWT

Put it all back together, including extrapolated point (forecast), via inverse SWT

Trading Strategy

Are there actionable results?
Can we make some money here?

Trading Strategy:

- If forecast value is higher than current value, go long
- Otherwise exit market

Paremeters

We have the following parameters to tune:

- Smoothing Wavelet - Harr, D4, D6, etc.
- Smoothing Iterations
- Sample Size (i.e. look-back)
- Forecast Wavelet
- Forecast Iterations
- Polynomial Interpolation Order

Paremeters

We have the following parameters to tune:

- Smoothing Wavelet - Harr, D4, D6, etc.
- Smoothing Iterations
- Sample Size (i.e. look-back)
- Forecast Wavelet
- Forecast Iterations
- Polynomial Interpolation Order

Paremeters

We have the following parameters to tune:

- Smoothing Wavelet - Harr, D4, D6, etc.
- Smoothing Iterations
- Sample Size (i.e. look-back)
- Forecast Wavelet
- Forecast Iterations
- Polynomial Interpolation Order

Paremeters

We have the following parameters to tune:

- Smoothing Wavelet - Harr, D4, D6, etc.
- Smoothing Iterations
- Sample Size (i.e. look-back)
- Forecast Wavelet
- Forecast Iterations
- Polynomial Interpolation Order

Paremeters

We have the following parameters to tune:

- Smoothing Wavelet - Harr, D4, D6, etc.
- Smoothing Iterations
- Sample Size (i.e. look-back)
- Forecast Wavelet
- Forecast Iterations
- Polynomial Interpolation Order

Paremeters

We have the following parameters to tune:

- Smoothing Wavelet - Harr, D4, D6, etc.
- Smoothing Iterations
- Sample Size (i.e. look-back)
- Forecast Wavelet
- Forecast Iterations
- Polynomial Interpolation Order

Less is More

I examined CAT and Nasdaq
Optimal Parameters:

- Smoothing Wavelet: D6 $<$ D4 $<$ D2 $< \emptyset$
- Smoothing Iterations: $3 < 2 < 1 < 0$
- Sample Size: 500-600
- Forecast Wavelet: D6 $<$ D4 $<$ D2
- Forecast Iterations: $3 < 2 < 1$
- Polynomial Interpolation Order: $3 < 2 < 1$

Less is More

I examined CAT and Nasdaq
Optimal Parameters:

- Smoothing Wavelet: D6 < D4 < D2 < $\emptyset$
- Smoothing Iterations: 3 < 2 < 1 < 0
- Sample Size: 500-600
- Forecast Wavelet: D6 < D4 < D2
- Forecast Iterations: 3 < 2 < 1
- Polynomial Interpolation Order: 3 < 2 < 1

Less is More

I examined CAT and Nasdaq
Optimal Parameters:

- Smoothing Wavelet: D6 < D4 < D2 < $\emptyset$
- Smoothing Iterations: 3 < 2 < 1 < 0
- Sample Size: 500-600
- Forecast Wavelet: D6 < D4 < D2
- Forecast Iterations: 3 < 2 < 1
- Polynomial Interpolation Order: 3 < 2 < 1

Less is More

I examined CAT and Nasdaq
Optimal Parameters:

- Smoothing Wavelet: D6 < D4 < D2 < $\emptyset$
- Smoothing Iterations: 3 < 2 < 1 < 0
- Sample Size: 500-600
- Forecast Wavelet: D6 < D4 < D2
- Forecast Iterations: 3 < 2 < 1
- Polynomial Interpolation Order: 3 < 2 < 1

Less is More

I examined CAT and Nasdaq
Optimal Parameters:

- Smoothing Wavelet: D6 < D4 < D2 < $\emptyset$
- Smoothing Iterations: 3 < 2 < 1 < 0
- Sample Size: 500-600
- Forecast Wavelet: D6 < D4 < D2
- Forecast Iterations: 3 < 2 < 1
- Polynomial Interpolation Order: 3 < 2 < 1

Less is More

I examined CAT and Nasdaq
Optimal Parameters:

- Smoothing Wavelet: D6 < D4 < D2 < $\emptyset$
- Smoothing Iterations: 3 < 2 < 1 < 0
- Sample Size: 500-600
- Forecast Wavelet: D6 < D4 < D2
- Forecast Iterations: 3 < 2 < 1
- Polynomial Interpolation Order: 3 < 2 < 1

Returns - CAT

Time Period	B/H	Wavelet	Notes
1982-2011	11.6%	20.6%	Yea!

Returns - CAT

Time Period	B/H	Wavelet	Notes
1982-2011	11.6%	20.6%	Yea!
1982-1997	10.5%	30.6%	Yea!

Returns - CAT

Time Period	B/H	Wavelet	Notes
1982-2011	11.6%	20.6%	Yea!
1982-1997	10.5%	30.6%	Yea!
1998-2011	5.7%	4.6%	Huh?

Returns - Nasdaq

Time Period	B/H	Wavelet	Notes
1979-2011	9.8%	20.0%	Yea!

Returns - Nasdaq

Time Period	B/H	Wavelet	Notes
1979-2011	9.8%	20.0%	Yea!
1979-1997	14.6%	35.3%	Yea!

Returns - Nasdaq

Time Period	B/H	Wavelet	Notes
1979-2011	9.8%	20.0%	Yea!
1979-1997	14.6%	35.3%	Yea!
1998-2011	3.7%	1.9%	Huh?

Resources

- Patrick Van Fleet's Wavelets Page
  http://cam.mathlab.stthomas.edu/wavelets/index.php
- David Ruch's Wavelets in Undergraduate Education WUE project page
  http://math.mscd.edu/WUE