Denoise of Financial Time Series Data using Wavelet Transform [closed]

Asked 1 year, 4 months ago Active 1 year, 4 months ago Viewed 2k times

2

4

Closed. This question needs to be more focused. It is not currently accepting answers.

Want to improve this question? Update the question so it focuses on one problem only by editing this post.

Closed last year.

Improve this question

I need to denoise financial time-series data for a machine learning problem and don't understand how a wavelet transform is computed.

My understanding is that you need multiple points of a time signal to identify frequencies.

What does a wavelet transform do with the first point(s)? If there are not enough points, does it use future data to identify the frequencies? If yes, is it possible to give a wavelet transform to use data only from and in "t"?

finance quantitative-finance financial wavelet

edited Mar 2 '19 at 6:45
Emma
23.7k 7 22 42

asked Feb 24 '19 at 11:48
Jan
101 8

3 I'm voting to close this question as off-topic because it is not about programming as defined in the guidelines, plus it is too-broad (asking 3-4 different questions) and asking for external resources – desertnaut Feb 24 '19 at 12:02

2 Answers

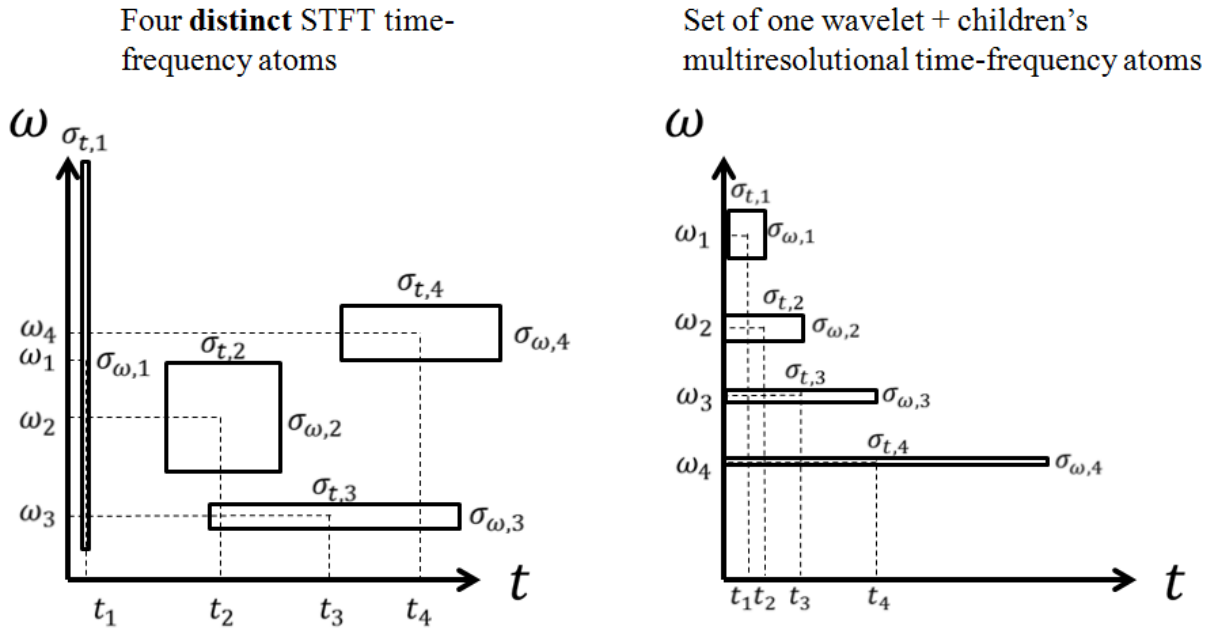
Active Oldest Votes

8

Generally, there are three types of methods to process your [financial] time-series data:

- Time domain methods (e.g., regression, statistical analysis on your financial time-series data such as mean, skewness, standard deviation, kurtosis, Black–Scholes model)
- Frequency domain methods (e.g., Fourier Transform, Power Spectral Density)
- Time-frequency domain methods (e.g., Short-Time Fourier Transform, Wavelet Transformation, Gabor Transform)

You may use `time-frequency methods` to denoise, categorize or classify financial time-series



.

[1-D] Frequency domain methods only return frequency information of your [1-D] financial time signals, which means that your [1-D] time data will be lost in exchange for having [1-D] frequency data.

[1-D] Time domain methods only return [1-D] time analysis of your financial signals, which also cannot help you to capture the frequency information.

You may use a `Continuous Wavelet Transform` or a `Discrete Wavelet Transform` to denoise financial time-series data.

There are many tools/languages that might help you to do so: `MatLab` , `Python` , and such. If you might have a programmer around you, s/he can probably help you in a few hours or a day to pass your [1-D] financial time data through one of these [2-D] time-frequency methods and visualize the outputs.

Your question is primarily about `sampling rate` . If your sampling rate is too low, thus frequency domain method may not return accurate resolution for you (regardless of Nyquist theorem). However, if you use such method for denoising, it normally means you have high-frequency data and [usually] you may want to down-sample or filter your data.

I suggest you to read about mathematics of wavelets with respect to mother and child (e.g., Morlet, Daubechies, etc.) which will help you to understand how a base function maps throughout your financial time-series data, the transformation occurs, and a new time and frequency representation of your initial financial time-series data result in.

As you know, Wavelet is a mathematical transform. As you wish, you may give almost any input data to the transform equation and will transform it for you. You may initially pick a window size. Imagine, you have a `1X1000` vector of `[0,1,0.3,1.2,-1,...]` equity or derivative information as your window, or any other larger size window `1X1,000,000` . It may not matter, if your data is from the past or you predict from the future and transform it via wavelet.

As you know, financial [chart] data usually as time passes, have an additional [data point] record, either a real data point or a forecasted one. In that case, It is absolutely fine, and you

`delata` time can be any fraction of time. You may just need to also consider computation, later for scale-up, which may not be an issue for you at this time.

My general view about your approach, not knowing many assumptions, is that you are in a challenging yet really great direction.

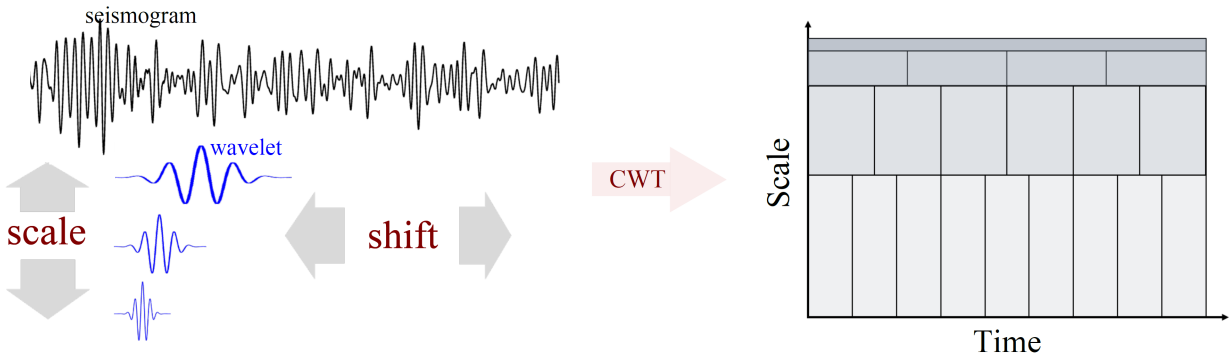


Image Courtesy: Harvard University

Good project, best wishes, thank you for your question and welcome to stackoverflow.com!

edited Feb 24 '19 at 19:48

answered Feb 24 '19 at 14:20

Emma
23.7k 7 22 42

2

By now, I found an answer to my initial question and want to share what I found out. This paper was a great source: https://www.sciencedirect.com/science/article/pii/S0022169418303317?via%3Dihub .

1

The MODWT (maximal overlap discrete wavelet transformation) is the only transformation that you can use for real-world applications as other WTs use future data points in every timestep t. Due to the "boundary constraint" when using MODWT, the first L = (2 ** J - 1) * (L - 1) + 1, where J equals the filter level and L the number of wavelet coefficients of a particular wavelet, values have to be removed.

I saved the deleted MODWT values and add them back to my predictions after training my neural network. Then, I can perform the inverse MODWT and ensure that the points at the end don't become distorted. When calculating the metrics, I delete the first L values again.

answered Mar 10 '19 at 13:21

Jan
101 8