Motivation

A noisy signal X, in our case it is BTC price, can be represented as a composition of Gaussian noise E and original signal S:

$$X = E + S, E \sim N(0, \sigma^2)$$

If wavelet transform is applied to X, then it produces one set of coefficients from S and other set from E. Large coefficients would likely be produced by original signal, and small ones – by noise.

Apart from being decomposed into noise and original signal, time series can be decomposed into low and high frequency components. It is claimed that high frequency component is more prone to manipulations activities. This is because manipulation causes prices to change dramatically, and fast change in a function is represented by high frequency component.

Manipulation usually triggers short-term price fluctuation around equilibrium price level. This oscillation is usually seen as data contamination. However, for investigation of time series it is more helpful to know true signal without noise. By applying discrete wavelet transform, we can receive true signal from high frequency component and remove additional noise. Discrete wavelet transform is a time-frequency representation of discretely sampled signal, which is continuous in time. This transform produces what is called detail and approximate coefficients, representing high and low frequency components, respectively.

Universal threshold method

Taking the detail coefficients, we can transform them via universal threshold method to remove noise from the series. For this, we need to find a threshold value, below which each coefficient will be set to zero. If the threshold value is too small, then considerable noise remains, and if the threshold value is too large, then some important feature of signal may be filtered out. The following formulas are used to determine threshold λ (W is detail coefficients, σ is average variance of noise and N is the length of the input signal – of time series):

$$\sigma = \frac{median\ (|W|)}{0.6745}$$

$$\lambda = \sigma\sqrt{2\ln\ (N)}$$

Then hard thresholding is applied to the detail coefficients. This method is used both by the authors of clustering and Hidden Markov models, but they apply it differently. In clustering model with DBSCAN and KPCA a more classical approach is used which is to remove small coefficients:

$$\widehat{W} = \begin{cases} W, & |W| \geq \lambda \\ 0, & |W| < \lambda \end{cases}$$

Since noise is usually represented by small coefficients, it is removed by hard thresholding. However, in paper on Hidden Markov model the authors remove large coefficients:

$$\widehat{W} = \begin{cases} W, & |W| \leq \lambda \\ 0, & |W| > \lambda \end{cases}$$

By this they claim to remove noise and low frequency components, which is not consistent with the theory. Noise is retained since it is represented with small coefficients, and low frequency component is not removed since the method is applied to high frequency component, which is a separate set of coefficients. So, from my limited experience with wavelets, the first approach seems to be more consistent with theory.

Next, the time series is reconstructed from obtained coefficients using inverse wavelet transform, so we receive a series of prices without noise in high frequency component. The authors do not use universal threshold method for low frequency component, possibly because it does not contain useful information for manipulation detection.

Difference with Fourier transform

Fourier transform does not capture time information, it only captures information about frequencies. On the other hand, wavelet transform captures frequency information and also tells when these frequencies occur in time. So, if a signal varies with time, Fourier transform does not capture it, whereas wavelet transform does. Besides, it is claimed that wavelet transform is better for using with finite, non-periodic and/or non-stationary signals, which is likely to characterize price series. What concerns mathematical representation, wavelet transform uses wavelet functions, which have limited domain, while Fourier transform uses sin(x) and cos(x) functions with unlimited domain, and this difference allows wavelet transform to capture time information.