Politechnika Wrocławska
Faculty of Pure and Applied Mathematics
Field of study: Applied Mathematics
Specialty: Computational Mathematics

Master Thesis

APPLICATIONS OF WAVELETS IN DATA MINING

Kinga Kurowska

keywords:
wavelets
statistics
learning and adaptive systems

Short summary:
*** To be changed ***
The aim of this thesis is an analysis and comparison of statistical methods employing wavelets. Special attention will be paid to data mining and machine learning methods.

Supervisor	dr inż. Andrzej Giniewicz		
	Title, degree, name and surname	*Grade*	*Signature*

*For the purposes of archival thesis qualified to: **
a) Category A (perpetual files)
b) Category BE 50 (subject to expertise after 50 years)
** Delete as appropriate*

stamp of the faculty

Wrocław, 2018

Contents

Introduction

*** Few words about my thesis... Generally what is a data mining, what are the wavelets, what are the applications in data mining and finally, what is the main purpose - edge detection. ***

Chapter 1

Wavelets theory

What exactly is a wavelet? A wavelet means a small wave. This term says a lot about it nature. Wavelets are a family of functions which oscillates like wave and should be compactly supported. Additionally, the wavelet has zero mean.

Definition 1.1. Wavelets are created by scaling and shifting of the, so called, mother wavelet $\psi(t)$. The child wavelets are defined as

$$\psi^{(a,b)}(t) = |a|^{-\frac{1}{2}}\psi\left(\frac{t-b}{a}\right),\ a > 0. \tag{1.1}$$

There are plenty of different mother wavelets, for example

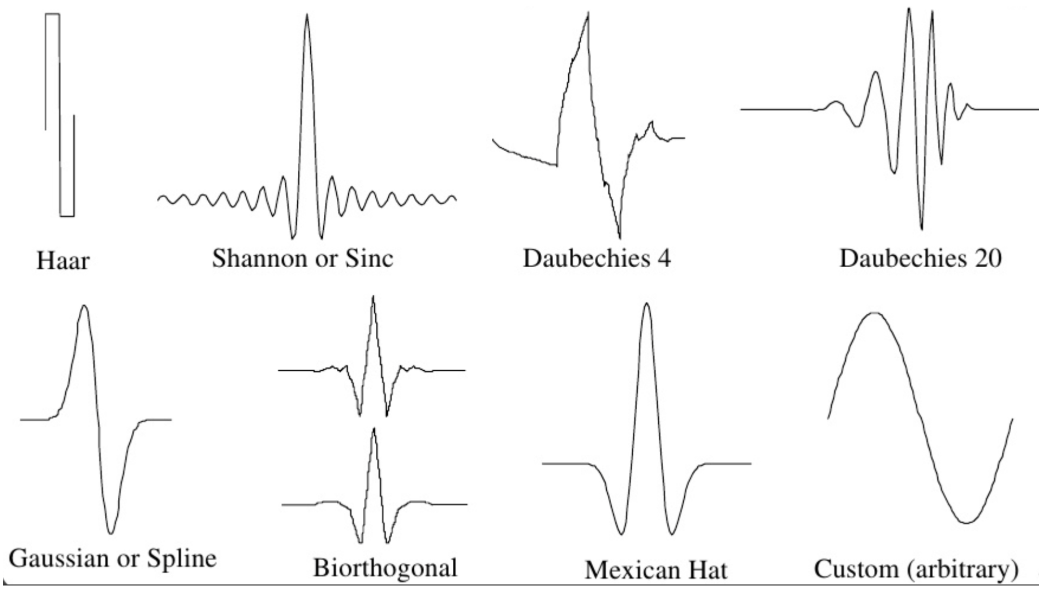


Figure 1.1: Different types of wavelets.

*** Add more about wavelets properties? ***

1.1. Daubechies wavelets

Each type of wavelet function is more suitable for different applications. The best for image analysis are the Daubechies wavelets.

Definition 1.2. Daubechies wavelets are collection of orthogonal and compactly supported functions. A denotation for those wavelets is dbN, where N means a maximal number of vanishing moments.

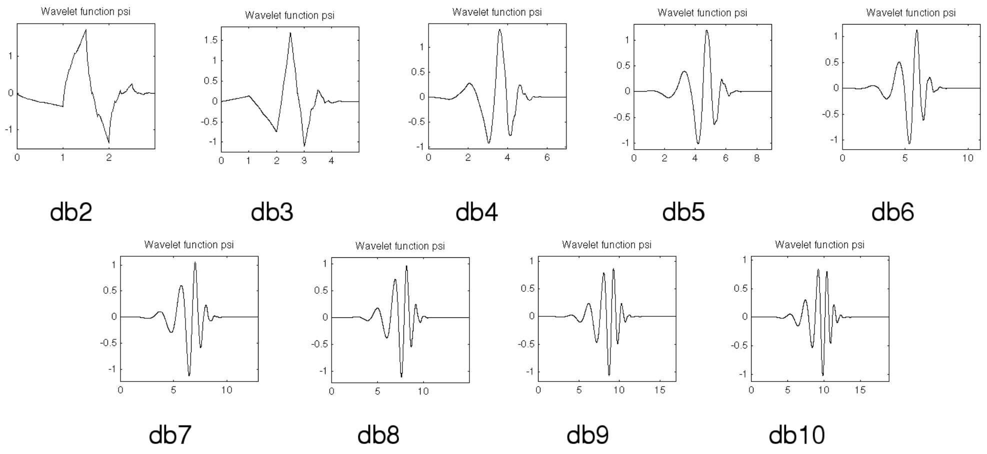


Figure 1.2: Daubechies wavelets.

1.2. Wavelet Transform

*** Add some short introduction? ***

Definition 1.3. Wavelet transform

$$W(a,b) = \int_{-\infty}^{\infty} y(t) a^{-\frac{1}{2}} \psi\left(\frac{t-b}{a}\right) dt, \tag{1.2}$$

where a is scale parameter, b translation parameter and $y(t)$ original signal.

1.2.1. Wavelet transform vs Fourier transform

Definition 1.4. Fourier transform

$$Y(f) = \int_{-\infty}^{\infty} y(t) e^{-i\omega t} dt, \tag{1.3}$$

where $y(t)$ is time domain signal and $Y(f)$ is frequency domain signal.

Wavelet transform	**Fourier transform**
Suitable for stationary and non--stationary signals	Suitable for stationary signals
High time and frequency resolution	Zero time resolution and very high frequency resolution
Very suitable for studying the local behaviours of the signal	No suitable
Sine and cosine waves	Scaled and translated mother wavelets

Table 1.1: Differences between Wavelet Transform and Fourier Transform.

What differs both transformations is the type of function. In Fourier case there are sine and cosine functions, wherein wavelet transform uses wavelets. Why use the Wavelet transform? Sine function oscillates on the whole real axis, thus it cannot represent abrupt changes. However, the Wavelet transform is localized in space and time, so it can be used to detect sudden changes in signals and images. Moreover, wide range of wavelet functions is a main advantage of wavelet analysis.

1.3. Discrete Wavelet transform

There are two types of the wavelet transform:
- Continuous Wavelet Transform (CWT),
- Discrete Wavelet Transform (DWT).

DWT is used to denoising and compression of signals and images. Also, DWT allows to detect smooth regions interrupted by edges or abrupt changes in contrast of images.
Scale and translation parameters are defined as

$$a = 2^j \text{ and } b = 2^j k, \;\; j, k = 1, 2, \ldots . \tag{1.4}$$

to avoid redundancy in coefficients.
The figure 1.3 on a page 9 shows how DWT works. Discrete Wavelet Transform splits signal with two filters: $g(n)$ - low pass filter (LPF) and $h(n)$ - high pass filter (HPF). The LPF captures a part with lower frequencies which is the main signal. Whereas, the HPF captures higher frequencies - a noise of the signal. Subsequently, both parts are downsampled by a factor of 2. This decomposition can be repeated on the LPF part of the signal. Hence, the next levels of DWT coefficients.

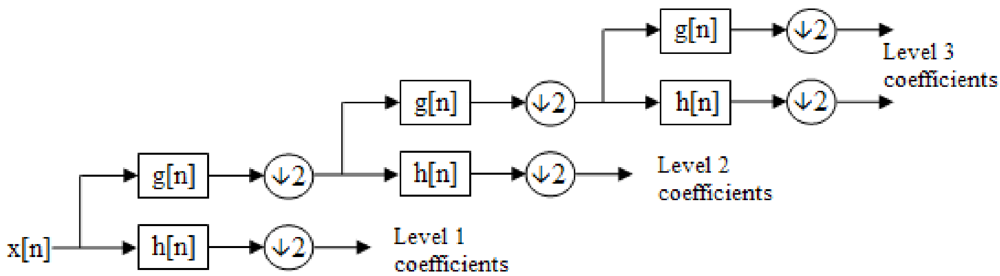


Figure 1.3: Discrete Wavelet transform on a signal $x(n)$.

1.4. 2-D Discrete Wavelet transform

2-Dimensional Discrete Wavelet Transform works similar way as 1-D with High Pass Filter, Low Pass Filter and downsampling, except that one level of the decomposition includes double filtering, on columns and rows. The figure 1.4 shows an image decomposition. Firstly, the DWT is applied on columns of the input image and then on the rows of the both outputs. Ultimately, there are four results:

- LL - result of LPF applied on both, columns and rows,
- LH - result of LPF applied on columns and HPF on rows,
- HL - result of HPF applied on columns and LPF on rows,
- HH - result of HPF applied on both, columns and rows.

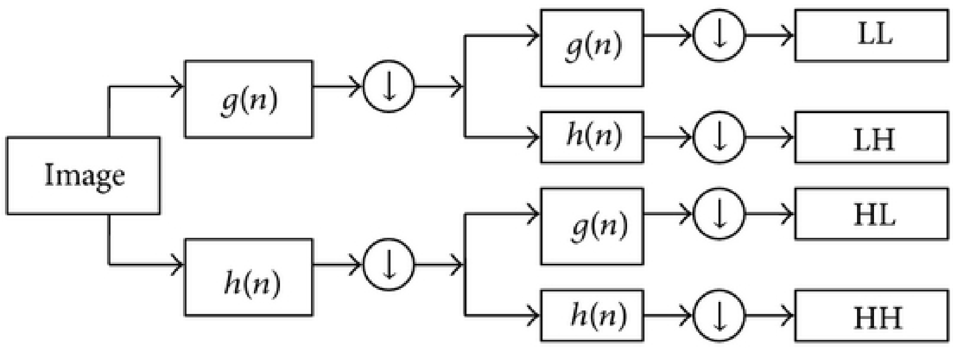


Figure 1.4: 2-D Discrete Wavelet transform on an image.

Recall that the outcome of Low Pass Filter in the previous case was the main signal (without a noise). Thus, a 2-Dimensional equivalent is an approximation of an analysed image. The High Pass Filter captures high frequencies, then for an image the outcome are sudden changes in the image contrast. Now, lets focus on what exactly each result represent. First one, the LL is just an approximation of the initial image. Next, the LH shows abrupt changes in a horizontal direction, whereas the HL part present similar issues but in a vertical direction. The HH shows sudden changes in a diagonal direction. In conclusion, the output of 2-D DWT gives us an approximation of the image and three parts with abrupt changes in different directions. What information gives as these sudden variations? Thanks to those we are able to find a places where two smooth regions meets. This kind of image anomaly could be interpreted as edges.

Chapter 2

Edge detection

In this chapter let us focus on the main goal of the thesis, which is edge detection. What is an edge? It is a place where image brightness changes rapidly. There are various methods to identify such discontinuities. The most popular are gradient based (e.g. Canny, Prewitt, Sobel) and Laplacian based. However, there is also another method, which provides similar results and can be more efficient in terms of computation. This method is based on the 2-Dimensional Discrete Wavelet Transform described in section 1.4.

What should be done? *** Link to article Edge detection ***

1. Convert image to grey scale.
2. Apply 2-D DWT on an image.
3. Remove the LL part.
4. Denoise the LH, HL and HH coefficients.
5. Reconstruct the initial image.
6. Post-processing - modify contrast to emphasize obtained edges.

2.1. Implementation of an algorithm

The algorithm is implemented in `Python` using library/package (?) `PyWavelets`. There are used also auxiliary libraries like: `numpy`, `matplotlib`, `PIL` and `scipy`. The `PyWavelets` package contains all features required to edge detection algorithm, i.e. 2D Forward and Inverse Discrete Wavelet Transform, build-in many wavelet functions and thresholding functionality (used to denoise coefficients).
Lets go deeper into algorithm, using a simple example - white square on a black background.

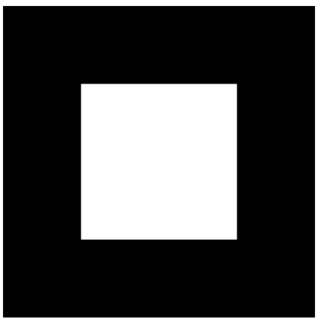


Figure 2.1: An initial image - a white square.

Initially, a colour image must be simplified by conversion to grey scale. Edges are recognised as changes in brightness, so a single pixel should contains only information about a colour (black) intensity. Our initial image is already black and white so we do not need to convert it. Now, we can apply the 2D DWT function. As a result, according to the description in section 1.4, we obtain four components of wavelet coefficients: LL, LH, HL and HH show on a graph 2.2. It is clearly visible that the LL part reflects the initial image and the rest components contains information about rapid brightness changes in particular directions.

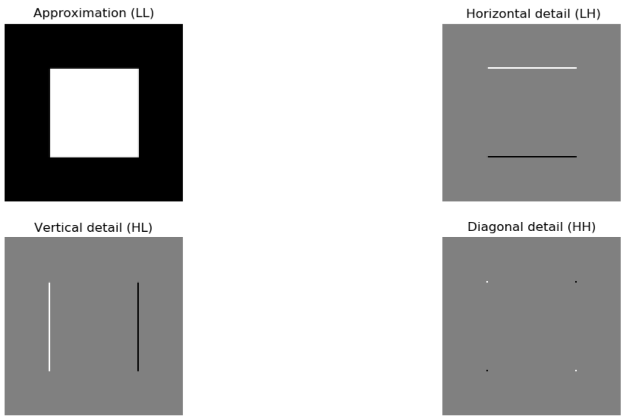


Figure 2.2: 2-D DWT coefficients.

Thus, we are interested only in components which gives information about edges, so we remove the approximation part - figure 2.3.

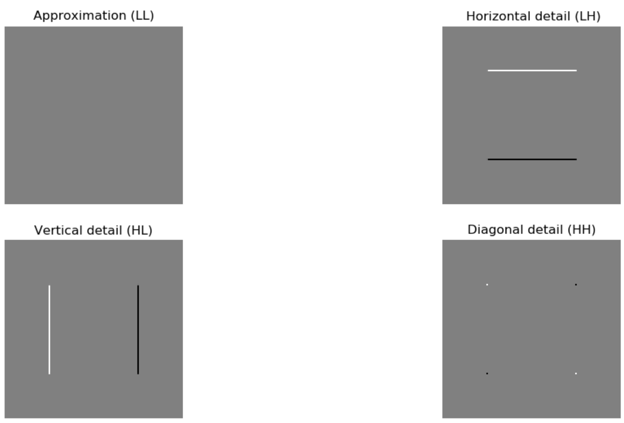


Figure 2.3: 2-D DWT coefficients with removed the LL part.

Subsequently, remaining components can be denoised. It means, we can /pozbyc sie/ small, insignificant coefficients by thresholding. More about

setting the threshold is described in section 2.2. This simple example do not require any denoising, so lets go to further.
The last main step is reconstruction of the initial image, i.e. application an inverse 2-D DWT on the denoised coefficients. In the result, we obtain the edges of the initial image presented on a graph 2.4.

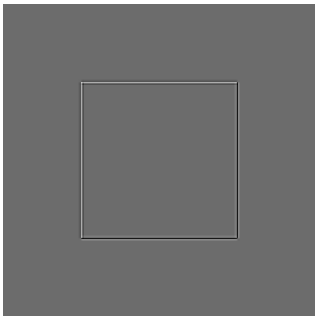


Figure 2.4: A reconstructed image showing the edges.

At the end, we can do some post-processing to emphasize obtained lines. Currently, a background has grey colour and the edges are white or black. Therefore, using simple mathematical calculations we can modify image to have black background and white edges. It is enough to get an absolute value, subtract 128 and then scale by multiplying 2 times. *** Add some more about values in an image array, here or in the previous paragraph *** Finally, we obtain the edges of the square shown on a figure 2.5.

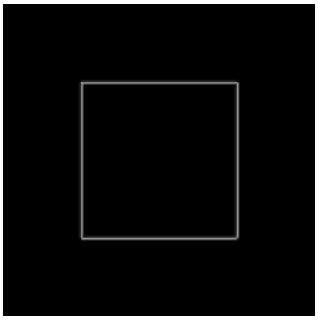


Figure 2.5: The edges of the square image after post-processing.

2.2. Thresholding

There are two types of thresholding hard and soft. Lets denote λ as threshold value and d as wavelet coefficient. The hard one /nadaje/ zero value for coefficients below the set threshold. It is defined as follow

$$D^H(d|\lambda) = \begin{cases} 0, & \text{for } |d| \leq \lambda, \\ d, & \text{for } |d| > \lambda. \end{cases} \tag{2.1}$$

While, the soft thresholding works in the same way on the coefficients smaller than λ, but additionally the coefficient bigger than the set threshold are "shrinked" towards zero /o jego wartosc/, as it is defined below.

$$D^S(d|\lambda) = \begin{cases} 0, & \text{for } |d| \leq \lambda, \\ d-\lambda, & \text{for } d > \lambda, \\ d+\lambda, & \text{for } d < -\lambda. \end{cases} \tag{2.2}$$

2.2.1. Hard or soft thresholding

To see the differences between both types of thresholding lets consider a noisy image (fig. 2.6).

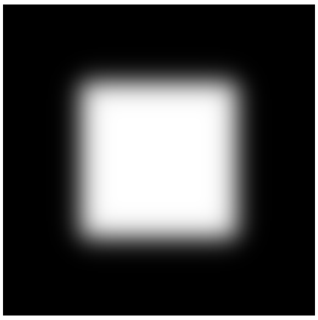


Figure 2.6: A noisy square image.

The coefficients of 2-D DWT are shown on the graph 2.7.

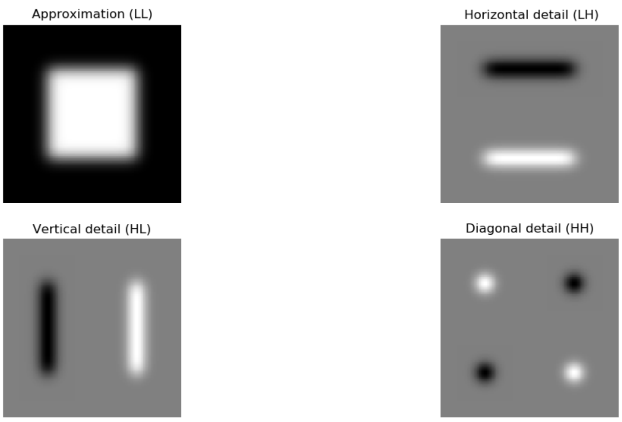


Figure 2.7: 2-D DWT coefficients.

Now, according to the edge detection algorithm, we remove the LL component and we can denoise the others. Results of hard and soft thresholding are presented respectively on a figures 2.8 and 2.9.
Especially on the LH and HL components we can see the distinction between both thresholding. After soft one the lines are smoother and slightly narrower the in the other case. Therefore, the soft thresholding is better for edge detection (smoother edges are better, /wybacza bledy bardziej niz hard/). To confirm this conclusion lets compare the final results, after inverse DWT (fig. 2.10)

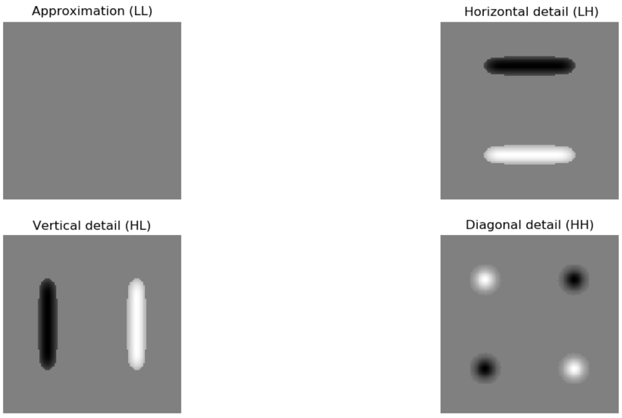


Figure 2.8: 2-D DWT coefficients after hard thresholding.

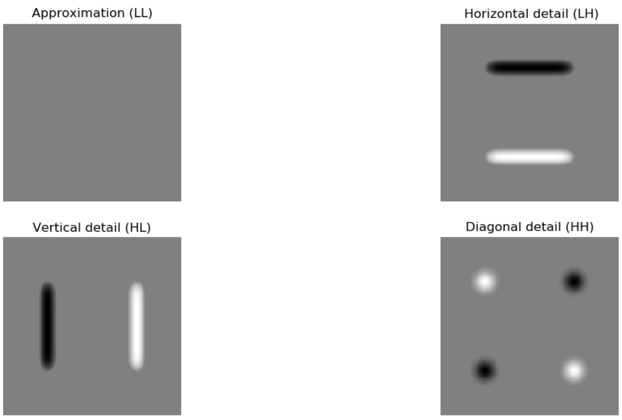


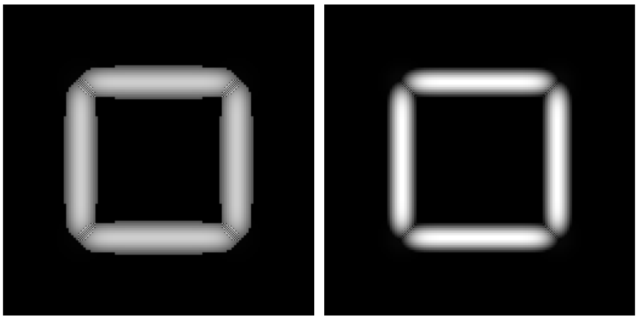
Figure 2.9: 2-D DWT coefficients after soft thresholding.

2.2.2. Selection of the threshold

The key is how to set the λ to denoise coefficients and do not loose any significant information. *** Some more about it and how I choose the threshold ***

2.3. Wavelet types

*** Show how algorithm work on a simple square ***



(a) (b)

Figure 2.10: Results of edge detection with hard (a) and soft (b) thresholding.

Chapter 3

*** Results ***

List of Figures

List of Tables

Bibliography

[1] V. R. Chaganti. Edge detection of noisy images using 2-d discrete wavelet transform. Master's thesis, The Florida State University, 2005.

[2] B. Kessler, G. Payne, W. Polyzou. Wavelet notes. The University of Iowa, Feb. 2008.

[3] S. Lahmiri, M. Boukadoum. Hybrid discrete wavelet transform and gabor filter banks processing for features extraction from biomedical images. *Journal of Medical Engineering*, 2013.

[4] T. Li, S. Ma, M. Ogihara. Wavelet methods in data mining. *Data Mining and Knowledge Discovery Handbook*, pages 603–626, 2005.

[5] L. Zhang, W. Zhou, L. Jiao. Wavelet support vector machine. *IEEE Transactions on Systems, Man, and Cybernetics*, Part B (Cybernetics)(34.1):34–39, 2004.