Complex Data: Mining using Patterns

Arno Siebes[1] and Zbyszek Struzik[2]

[1] Utrecht University
Utrecht, The Netherlands
arno@cs.uu.nl
[2] CWI
Amsterdam, The Netherlands
zbyszek@cwi.nl

Abstract. There is a growing need to analyse sets of complex data, i.e., data in which the individual data items are (semi-) structured collections of data themselves, such as sets of time-series. To perform such analysis, one has to redefine familiar notions such as similarity on such complex data types. One can do that either on the data items directly, or indirectly, based on features or patterns computed from the individual data items. In this paper, we argue that wavelet decomposition is a general tool for the latter approach.

1 Introduction

One of the many variants of Moore's law [10] is the exponential growth of hard-disk capacity per euro. This growth has enabled the rise of a fast increasing number of fast growing *complex data* sets. That is, data collections in which the individual data items are no longer "simple" (atomic in database terminology) values but are (semi-)structured collections of data themselves. For example, large text corpera, multi-media data collections, databases with milions of time-series, and large collections of DNA and/or protein data.

With the advent of sets of complex data grows the need to analyse such collections. This is, e.g., illustrated by the existence of satelite workshops on topics such as multimedia data mining, text mining, and spatio-temporal data mining around all the major KDD conferences.

To apply standard or new data-analysis techniques, fundamental notions that or obvious for numbers have to be redefined. One example of such a notion is *similarity*, which is essential, e.g., for clustering and classification [5]. There are two ways in which one can define similarity for complex data types. Firstly, one can define it directly on the complex data items. Secondly, one derive features or patterns from complex data items and define the similarity on these features or patterns.

Both approaches have their respective advantages. The former allows the definition of, e.g., a similarity measure that is based on all aspects of the data items rather than only on those that are represented by features. Measures that find their roots (ultimately) in *Kolmogorov complexity* [8] are an example of this

class. The latter, in contrast, allow one to take only relevant "local structure" into account for the similarity measure, such as measures based on *wavelet decomposition* [12]. Which of the two is best, thus depends primarily on the notion of similarity one wants to use.

For both approaches, it is important that the way one defines the similarity measure is sufficiently general. That is, it is applicable across specific problems and across various complex data types. Otherwise the analysis of complex data would degenerate into a ragbag of ad-hoc techniques. For the first approach, Kolmogorov complexity seems a good candidate for such a general technique. We briefly discuss this in Section 2.

The main argument of this paper is that wavelets provide such a general technique for similarity (and other notions) based on local structure in the complex data items, this is discussed in the rest of the paper. In Section 3, we discuss how we have used wavelets on two problems on sets of time series. Since wavelets are not yet a standard tool in the data miners toolbox, this section also contains a brief introduction into wavelet decomposition. In Section 4, we argue that the wavelet decomposition is also applicable for other types of complex data. In the final section we formulate our conclusions.

Perhaps the simplest form of complex data is a collection of texts. For example, a set of articles or books. A recurrent problem in history has been: can we assign an author to a text. Either because the author has stayed anonymous or because it is suspected that the author has used a pseudonym. A controversial example of the latter is the question whether all of the works attributed to Shakespeare were actually written by Shakespeare.

A less controversial and perhaps more famous example in the statistical literature is that of the *The Federalist* papers [11]. These papers were written by Alexander Hamilton, John Jay, and James Madison to persuade the citizens of the State of New York to ratify the Constitution [11]. The author of most of these papers is known, but the authorship of 12 of them is in dispute between Hamilton and Madison. In their study [11] Mosteller and Wallace search for words whose usage-frequency discriminates between Hamilton and Madison.

Word-frequencies are probably the most simple features one can derive from text. But they play an important role in information retrieval and in text mining. For classification, however, word-frequencies are perhaps too simple features since other stylistic aspects are not taken into account at all. Hence, it is interesting to see what similarity measures one can define on the complete texts directly.

In a recent paper [1], the authors propose to use WinZip for the task of assigning papers to authors. The motivation is that compression algorithms such as Lempel-Ziv encoding exploit the structure in a text. Hence, for texts A and B, they define:

- L_A = the length of Lempel-Ziv encoding of A in bits,
- let b be a (small) introductory part of B:

$$\Delta_{Ab} = L_{A+b} - L_A$$

To test how well Δ_{Ab} can be used to assign texts to authors, they use a corpus of 90 texts by 11 authors and perform 90 experiments. In each experiment, they take one of the texts as produced by an unknown author X. To discover the author, the search for the (remaining) text A_i that minimizes $\Delta_{A_i x}$. The author of A_i is then predicted to be the author of X. In 84 out of the 90 experiments this prediction was correct.

On request of a Dutch newspaper, NRC Handelsblad, Benedetto et al have used this technique to solve an open case in contemporary Dutch literature: are the authors Arnon Grunberg and Marek van der Jagt one and the same person? Based on style-similarities and the elusiveness of the second author (only contact via e-mail), this is what critics suspected. Despite this controversy, van der Jagt got a prize for the best debut a couple of years after Grunberg picked it up.

Using a collection of other contemporary writers, the results pointed to Grunberg. Clearly, in itself this proves nothing. However, the author conceded. Fortunately, right after this show-down a new unknown writer, Aristide von Bienefelt, debuted. Again it is rumored that this is actually Grunberg.

To motivate the suitability of their approach, Benedetto et al refer to Kolomogorov complexity [8], but they do not actually explore this avenue. In that book, Li and Vitányi already introduced an *information distance* between two binary strings. In subsequent work, together with co-authors, they refined this notion, see e.g., [2] and showed that it applicable in many cases.

The Kolmogorov complexity of string x, denoted by $K(x)$ is the length of the shortest program of a universal computer that outputs x. Up to an additive constant, $K(x)$ is independent of the particular machine chosen. Similarly, $K(x|y^*)$ is defined as the length of the shortest program that computes x given y^* as input; see [8] for more details.

In [7] the normalised information distance between two strings x and y is defined by:

$$d(x,y) = \frac{max\{K(x|y^*), K(y|x^*)\}}{max\{K(x), K(y)\}}$$

Moreover, it is proved that this measure is universal in the sense that it minorizes every remotely computable type of dissimilarity measure.

The problem in applying such distance measures is that $K(x)$ is not computable. Hence, the trick in the applications is to use a suitable approximation of $K(x)$, see the cited literature for more details.

2 Complex Data: Time Series

Kolmogorov complexity is universal. However, there are problems and complex data types for which it is not obviously the best solution. For example, in cases where it is not the complete complex data item that is important, but "only" local patterns in that data. In this section, we discuss such an example and illustrate how we have tackled it using wavelets.

2.1 The Problem

Banks follow large collections (over a million items) of financial and economic time series, on stock-prices, exchange rates, interest rates, unemployment rates et cetera. This information is, e.g., used to compute the combined risk of the banks portfolio or to determine opportunities for stock-exchange transactions.

Unfortunately, the time-series are not without error. Since the data are used almost instantaneously and potentially large amounts of money are involved, it is of utmost importance that such errors are recognized as early as possible. The sheer number of time series implies that most of the work has to be done automatically, involving people only if one is fairly certain of a mistake. Both the number of false positives and the number of false negatives should be as small as possible. Large mistakes are easily recognisable, hence we focus on relatively small errors.

One way to attack this problem is by modelling each individual time-series and signal whenever the new reported value differs from the predicted value. The problem here lies obviously in the accuracy of the prediction. Although there are numerous studies in which the authors attempt to model stock prices, we have chosen another approach. If only because the number of data miners we know that live on an estate in the country and are driven around in Rolls-Royces is fairly low[3].

Our approach is based on the *behaviour* of time series.

Outliers: are signalled if an individual time-series suddenly behaves completely differently from its (recent) past.

Group-outliers: are time-series that suddenly behave differently from their *peer group*. These peer-groups are constructed by clustering time-series on their recent behaviour; note that the idea of peer-groups has also been used for fraud-detection in [3].

Note that in both cases, the similarity is based on the recent past. It is not a priori clear what time-frame defines the recent past. In other words, applying Kolomogorov complexity based techniques is far from straightforward. We have opted to characterise the recent past using wavelet decomposition. The advantage of such a multi-scale method is that one discovers what constitutes the recent past while analysing.

2.2 Introducing Wavelets

Since wavelets are not yet a common tool in data mining, we first give a brief (and simplified) introduction to wavelets. It is well-known that a function f supported on $[-s, s]$ can be represented by a *Fourier series*:

$$f(x) = \sum_{n=0}^{\infty} c_n e^{2\pi i n(x/2s)}$$

[3] Of course, it is entirely possible that these successful data miners do not publish their methods *because* they make so much money out of their ideas.

in which the Fourier transforms are computed by the convolutions with $e^{-2\pi it(n/2s)}$, i.e.,

$$c_n = \int_{-s}^{s} f(t)e^{-2\pi it(n/2s)}dt$$

Moreover, for reasonable functions f (integrable in the L^2 sense) this can be extended to the whole domain of f.

For our purposes, the disadvantage of global transformations such as the Fourier transform is that they do not support the local analysis of functions. That is, it is hard to see patterns The wavelet transforms provides such locality because of the limited (effective) support of wavelets. In addition, they poses the often very desirable ability of filtering the polynomial behaviour to some predefined degree. For time-series data this means that we can easily handle non-stationarities like global or local trends or biases. Last described, but certainly not least for our purposes, one of the main aspects of wavelet transforms is the ability to reveal the *hierarchy* of (singular) features including the scaling behaviour - the so-called *scale-free* behaviour.

Wavelet transforms are, like Fourier transforms, computed by convolutions. The difference is that we don't use a function of infinite support like e^x, but a function with a localised (effective) support: the wavelet.

Usually, one starts with a smoothing kernel θ. The wavelet is then a derivative ψ of the kernel θ. With Fourier transforms, we only have a scaling parameter s, but since wavelets have a limited domain, we also have a *translation parameter* b. That is, we compute transforms by convolutions with ψ. Denoting the convolution by $< f, \psi >$ we have:

$$Wf(s,b) =< f, \psi > (s,b) = \frac{1}{s}\int_{\Omega} f(x)\psi(\frac{x-b}{s})dx$$

In the continuous case, the the Gaussian smoothing kernel $\theta(x) = exp(-x^2/2)$ has optimal localisation in both frequency and position. An often used wavelet derived from this kernel is its second derivative which is called the *Mexican hat*[4].

If we compute the continuous wavelet transform (CWT) of fractional Brownian motion using the Mexican hat wavelet, we get figure 1.

The front axis is the position, the scale axis pointing "in depth" is (as traditional) in logarithmic scale, while the vertical axis denotes the magnitude of the transform. This 3D plot shows how the wavelet transform reveals more and more detail while going towards smaller scales, i.e. towards smaller $\log(s)$ values. This is why the wavelet transform is sometimes referred to as the 'mathematical microscope'.

As well as continuous wavelet transforms, there exist discrete transforms (DWT). In this case the simplest kernel is the block function:

$$\theta(x) = \begin{cases} 1 & \text{for } 0 \leq x \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

[4] If you wonder why, draw its graph.

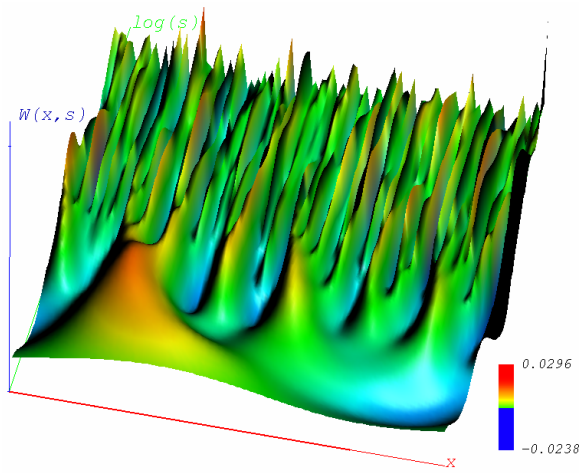

Fig. 1. CWT of Brownian motion

Which yields the *Haar* wavelet:

$$\psi(x) = \begin{cases} 1 & \text{if } 0 \leq x \leq 0.5 \\ -1 & \text{if } 0.5 \leq x \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

For a particular choice of scaling and translation parameters, we get the Haar system:

$$\psi_{s,b}(x) = 2^{-s}\psi(2^{-s}x - b) \quad s > 0, b = 0, \ldots, 2^s$$

For an arbitrary time series $f = \{f_i\}_{i \in \{1,\ldots,2^N\}}$ on normalised support $[0, 1]$ we have in analogy with the Fourier series:

$$f = f^0 + \sum_{m=0}^{N} \sum_{l=0}^{2^m} c_{m,l}\psi_{m,l}$$

In which $f^0 =< f, \theta >$ and $c_{m,l} =< f, \psi_{m,l} >$ Moreover, the approximations f^j of the time series f with the smoothing kernel $\theta_{j,k}$ form a 'ladder' of multi-resolution approximations:

$$f^{j-1} = f^j + \sum_{k=0}^{2^j} < f, \psi_{j,k} > \psi_{j,k},$$

where $f^j =< f, \theta_{j,k} >$ and $\theta_{j,k} = 2^{-j}\theta(2^{-j}x - k)$.

It is thus possible to 'move' from one approximation level $j - 1$ to another level j by simply adding (subtracting for j to $j-1$ direction), the detail contained in the corresponding wavelet coefficients $c_{j,k}, k = 0 \ldots 2^j$.

The CWT is an extremely redundant representation of the data. A more usful representation is the Wavelet Transform Modulus Maxima (WTMM) representation, introduced by Mallat [9]. A maxima line in the 3D plot of the CWT is a line where the wavelet transform reaches local maximum (with respect to the position coordinate). Connecting such local maxima within the continuous wavelet transform 'landscape' gives rise to the entire tree of maxima lines; the WTTM. For our Brownian motion, we get figure 2

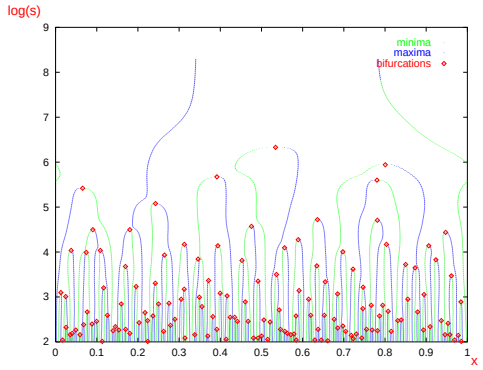

Fig. 2. WTTM plot of Brownian motion. The light lines are the minima lines, i.e., they connect the local minima, the dark lines the maxima lines, and the dots are the bifurcations

Restricting oneself to the collection of such maxima lines provides a particularly useful representation of the entire CWT. In particular (under suitable conditions), we have the following power law proportionality for the wavelet transform of the *isolated* cusp singularity in $f(x_0)$:

$$W^{(n)}f(s, x_0) \sim |s|^{h(x_0)} \ .$$

The exponents $h(x_0)$ are the local *Hölder exponents*, which describe the local *roughnes* of the time-series The Hölder exponents provide a usful characterisation of the time-series as we will see in the next section.

The most direct representation of the time series with the Haar decomposition scheme would be encoding a certain predefined, highest, i.e. most coarse, resolution level s_{max}, say one year resolution, and the details at the lower scales: half (a year), quarter (of a year) etc., down to the minimal (finest) resolution of interest s_{min}, which would often be defined by the lowest sampling rate of the signals. The coefficients of the Haar decomposition between scales $s_{max}..s_{min}$ will be used for the representation:

$$Haar(f) = \{c_{i,j} \ : i = s_{max}..s_{min}, \ j = 1..2^i \} \ .$$

The Haar representation is directly suitable to serve for comparison purposes when the absolute (i.e. not relative) values of the time series (and the local slope) are relevant. In many applications one would, however, rather work with value independent, scale invariant representations. For that purpose, we have used a number of different, special representations derived from the Haar decomposition WT. To begin with, we will use the sign based representation. It uses only the sign of the wavelet coefficient:

$$s_{i,j} = sgn(c_{i,j})$$

Another possibility to arrive at a scale invariant representation is to use the difference of the logarithms (DOL) of values of the wavelet coefficient at the highest scale and at the working scale:

$$v_{i,j}^{DOL} = log(|c_{i,j}|) - log(|c_{1,1}|) \ ,$$

where i, j are working scale and position respectively, and $c_{1,1}$ is the first coefficient of the corresponding Haar representation. Note that the sign representation $s_{i,j}$ of the time series is complementary/orthogonal to the DOL representation.

The DOL representation can be conveniently normalised to give the rate of increase of v^{DOL} with scale:

$$h_{i,j} = v_{i,j}^{DOL} / \log(2^{(i)}) \qquad \text{for } i > 0 \ .$$

This representation resembles the Hölder exponent approximation introduced above at the particular scale of resolution

2.3 Mining with Wavelets

To illustrate the usefulness of wavelets for our time-series problems, we recall some of our published results. We start by looking for outliers in the single time-series of figure 3.

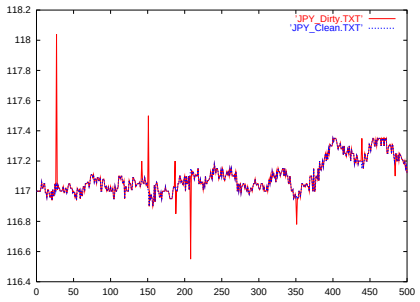

Fig. 3. The original time-series including spikes

The red spikes are obvious and are removed before we start our analysis. The local Hölder exponents are plotted in figure 4. By thresholding on h we separate the outliers from the rest. The reader is referred to [15] for more details.

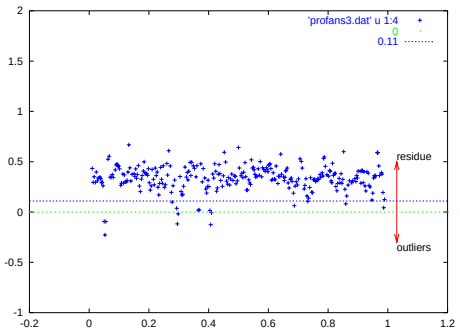

Fig. 4. The local Hölder exponents of our time-series

For the similarity, we use a set of exchange rates against the dollar depicted in fig 5

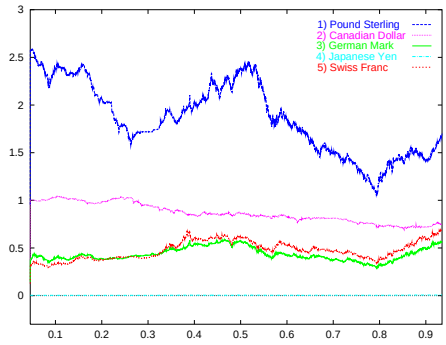

Fig. 5. The exchange rate of 1: Pound Sterling, 2: Canadian Dollar, 3: German Mark, 4: Japanese Yen, and 5: Swiss Frank against the US Dollar

We plot the values of the correlation products for each of the pairs compared, obtained with the Haar representation, the sign representation and the Hölder representation in figure 6. The reader is urged to see how well the representations match his/her own visual estimate of similarity [16].

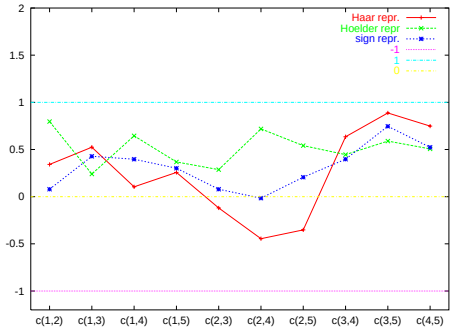

Fig. 6. The correlations for each of the pairs for the Haar, sign, and Hölder representation. Note, c(2, 5) stands for the correlation of the pair Canadian Dollar - Swiss Frank.

3 Other Types of Complex Data

The fact that wavelets work well for specific problems on sets of time-series does not prove their generality. To argue that indeed they do form a widely applicable tool, we briefly discuss two other complex data types in this section. Firstly we discuss multimedia data, secondly we focus on DNA data. Since wavelets originate in signal analysis, it should not come as a surprise that they are useful for multimedia data, which is after all a collection of signals. That DNA data is also a possible application area is perhaps more surprising.

3.1 Multimedia Data

Consider for example a database with tens or hundreds of thousands or even millions of pictures. Ideally, each of this pictures would be amply annotated such that one could find a picture of Marilyn Monroe in the sunset near the Golden Gate bridge by simply typing in a few keywords. However, annotating such large collections is prohibitively expensive. In fact, it is wishful thinking that one could get coherent annotation schemes for pictures in a loosely coupled distributed database as the world wide web.

Therefore, a large part of research in multi-media databases is devoted to exploitation of patterns in the data items to enable searching for similar items in such databases. This would allow, e.g., searching for pictures that are similar to a given picture and, more difficult, to search for pictures using a sketch.

The difficulty of the problem is perhaps best illustrated by the wide variety of of techniques persued for image databases. They range from global colour histograms in some appropriate colour-space via collections of local colour histograms and textures to wavelet transforms [13]. With the possible exception of global colour histograms, all these techniques focus on (local) structures in the images to define similarity measures. The success of wavelet measures of

similarity for retrieval point to the possible usefulness of such measures for data mining; see [14, 17] for such and other appraoches.

3.2 DNA Data

The difference between species and between individual of one species is visible at the molecular level as smaller or larger differences in their DNA. Because of the mechanisms of evolution, the DNA of related species is highly similar or *homologuous*. *Alignment* of two DNA strings is (in-exact) string matching while maximising homology. The string matching for DNA differs from exact matching in two ways [4].

1. Two different characters may be matched for a cost that reflects the probability that the two characters derive from a common ancestor.
2. While matching, one may introduce a *gap* in one of the two strings; i.e., part of the other string is matched to nothing. Again, there are costs associated with starting and extending a gap.

With the cost information, the "cheapest" match is the most likely alignment under the assumption that both strings derive from a common ancestor.

Rather than aligning just two strings one can also align multiple strings using the same costs as for alignment of two strings. This results in an alignment that is most likely if all strings derive from a common ancestor [4]. Aligning two or more strings can, e,g, be used to cluster genomes to recover the path of evolution in phylogenetic analysis [4].

For many problems, using the whole, or large parts of the, genome is best. However, there are problems where one is only interested in local homology. One such example is caused by the fact that bacteria are able to swap genes across species. In such cases, local patterns can be useful.

Given the discrete nature of DNA data, it is perhaps surprising that wavelet transforms can be exploited here. However, in [6] the authors show that one can built an index on DNA strings using wavelets that facilitates range and k-nearest neighbour searches.

They define the frequency vector $f(s)$ of a string s as the vector $[n_A, n_C, n_T, n_G]$, in which n_X denotes the frequency of X in s. One vector for each DNA string in the database is clearly insufficient. Hence, they built a a hierarchy of wavelet transforms of the string, where each transform is computed from the function f on substrings of s. Of course, on smaller scales smaller substrings are used. In other words, again the wavelet transforms focus on smaller and smaller details of the data item.

The paper shows that this index works well if the edit distance is used. That is, the *costs* we discussed earlier do not play a role. But the authors plan to extend their work in this direction.

4 Conclusions

There are more and more complex datasets that need to be analysed. Such an analysis requires that familiar notions such as similarity are generalised to complex data types. There are two ways in which one can approach this problem. In the first approach one defines the similarity directly on the complex data types. In the second approach, one first computes local features or patterns on the individual data items and defines the similarity on these local patterns.

In this paper we have argued that wavelets are a general tool for this second approach. More in particular, we have discussed how we have used wavelets to detect errors in collections of time-series. Moreover, we have discussed how wavelets are used for multimedia data and DNA data.

Clearly, this paper is neither the first nor the last word on this topic. We plan more applications of wavelets on complex data types to convince others and ourselves of their use in the data miners toolbox.

References

1. Dario Benedetto, Emanuele Caglioti, and Victor Loreto. Language trees and zipping. *Physical Review Letters*, 88(4), 2002.
2. C.H. Bennet, P. Gács, M. Li, P.M.B. Vitányi, and W. Zurek. Information distance. *IEEE Trans. on Information Theory*, 44(4):1407 – 1423, 1998.
3. R.J. Bolton and D.J. Hand. Unsupervised profiling methods for fraud detection. *Credit Scoring and Control VII, Edinburgh*, 2001.
4. R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis - Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.
5. David Hand, Heikki Mannila, and Padhraic Smyth. *Principles of Data Mining*. MIT Press, 2001.
6. Tamer Kahveci and Ambuj K. Singh. An efficient index structure for string databases. In *Proceedings of the 27th VLDB*, pages 351 – 360. Morgan Kaufmann, 2001.
7. Ming Li, Xin Li, Bin Ma, and Paul Vitányi. *Normalized Information Distance and Whole Mitochondrial Genome Phylogeny Analysis*. arXiv:cs.CC/0111054v1, 2001.
8. Ming Li and Paul Vitányi. *An Introduction to Kolmogorov Complexity and its Applications*. Springer Verlag, 1993.
9. S.G. Mallat and W.I. Wang. Singularity detection and processing with wavelets. *IEEE Trans. on Information Theory*, 38, 1992.
10. Gordon E. Moore. Cramming more components onto integrated circuits. *Electronics*, 38(8), 1965.
11. Frederick Mosteller and David L. Wallace. *Applied Bayesian and Classical Inference - The Case of The Federalist Papers*. Springer Verlag, 1984.
12. R. Todd Ogden. *Essential Wavelets for Statistical Applications and Data Analysis*. Birkhäuser, 1997.
13. Simone Santini. *Exploratory Image Databases - Content-Based Retrieval*. Academic Press, 2001.
14. Simeon J. Simoff and Osmar R. Zaïane, editors. *Proceedings of the First International Workshop on Multimedia Data Mining,MDM/KDD2000*. http://www.cs.ualberta.ca/ zaiane/mdm_kdd2000/, 2000.

15. Zbigniew R. Struzik and Arno Siebes. Wavelet transform based multifractal formalism in outlier detection and localisation for financial time series. *Physica A: Statistical Mechanics and its Applications*, 309(3-4):388 – 402, 2002.
16. Z.R. Struzik and A.P.J.M. Siebes. The haar wavelet in the time series similarity paradigm. In *Proceedings of PKDD99, LNAI 1704*, pages 12 – 22. Springer Verlag, 1999.
17. Osmar R. Zaïane and Simeon J. Simoff, editors. *Proceedings of the Second International Workshop on Multimedia Data Mining, MDM/KDD2001.* http://www.acm.org/sigkdd/proceedings/mdmkdd01, 2001.