The Haar Wavelet Transform of a Dendrogram – I

Fionn Murtagh*

June 26, 2005

Abstract

While there is a very long tradition of approximating a data array by projecting row or column vectors into a lower dimensional subspace the direct approximation of a data matrix through smoothing is less common. Applications of data array smoothing include visualization; filtering of less relevant, and thus harder to interpret, values; and as a means towards compression. Wavelet smoothing or regression is a term applied to data filtering in wavelet space, followed by data reconstruction. Due to boundaries, and invariance of rows and columns, applying a wavelet transform to a data array is very problematic, unlike applying a wavelet transform to a two-dimensional pixelated image. We develop a new wavelet transform for application to data arrays. This is based on prior hierarchical clustering, which takes internal data structure and interrelationships into account. We motivate and describe the integrated clustering and wavelet transform in this work, and discuss its use for data array smoothing, and for approximating a dendrogram by "collapsing" clusters. In a companion paper (Paper II) we explore background theory and address the question as to how this new post-processing analysis of hierarchical clustering is indeed a wavelet transform.

Keywords: multivariate data analysis, hierarchical clustering, data summarization, data approximation, compression, wavelet transform.

1 Introduction

In this paper, the new data analysis approach to be described can be understood as a transform which maps a hierarchical clustering into a transformed

*F. Murtagh is with the Department of Computer Science, Royal Holloway, University of London, Egham, Surrey, TW20 0EX, England. Email fmurtagh@acm.org

set of data; and this transform is invertible, meaning that the original data can be exactly reconstructed. Such transforms are very often used in data analysis and signal processing because processing of the data may be facilitated by carrying out such processing in transform space, followed by reconstruction of the data in some "good approximation" sense. We will now take smoothing as a case in point of such processing.

Smoothing of data is important for exploratory visualization, for data understanding and interpretation, and as an aid in model fitting (e.g., in time series analysis or more generally in regression modeling). The wavelet transform is often used for signal (and image) smoothing in view of its "energy compaction" properties, i.e., large values tend to become larger, and small values smaller, when the wavelet transform is applied. Thus a very effective approach to signal smoothing is to selectively modify wavelet coefficients (for example, put small wavelet coefficients to zero) before reconstructing an approximate version of the data. See Härdle (2000), Starck and Murtagh (2002).

The wavelet transform, developed for signal and image processing, has been extended for use on relational data tables and multidimensional data sets (Vitter and Wang, 1999; Joe, Whang and Kim, 2001) for data summarization (micro-aggregation) with the goal of anonymization (or statistical disclosure limitation) and macrodata generation; and data summarization with the goal of computational efficiency, especially in query optimization. A survey of data mining applications (including applications to image and signal content-based information retrieval) can be found in Tao Li, Qi Li, Shenghuo Zhu and Ogihara (2002). In the next section, we will briefly review these applications, and we will point to the novelty of the work that we present in this article.

A hierarchical representation is used by us, as a first phase of the processing, (i) in order to cater for the lack of any inherent row/column order in the given data table and to get around this obstacle to freely using a wavelet transform; and (ii) to take into account structure and interrelationships in the data. For the latter, a hierarchical clustering furnishes an embedded set of clusters, and obviates any need for a priori fixing of number of clusters.

Once this is done, the hierarchy is wavelet transformed. The approach is a natural and integral one.

A hierarchy may be constructed through use of any constructive, hierarchical clustering algorithm (Benzécri, 1979; Johnson, 1967; Murtagh, 1985). In this work we will assume that some agglomerative criterion is satisfactory from the perspective of the type of data, and the nature of the data analysis or processing. In a wide range of practical scenarios, the minimum variance

(or Ward) agglomerative criterion can be strongly recommended due to its data summarizing properties (Murtagh, 1985).

The remainder of this article is organized as follows. In section 2 a short review is presented of the inherent limitations to date of the use of wavelet transforms of data tables, in the data mining context. Section 3 presents our new wavelet transform which uses as input both the original data array, and the hierarchical clustering of this data. This description of our new wavelet transform is sufficiently detailed to allow it to be easily programmed. In section 4, some practical examples are described for clarity of exposition of the new wavelet transform.

Then in section 5 a range of assessments and evaluations are carried out, to show how well the integrated hierarchical clustering and wavelet transform can be applied in practice.

2 Previous Work on Wavelet Transforms of Data Tables

In this section we will review recent work using wavelet transforms on data tables, and show how our work represents a radically new approach to tackling similar objectives.

Approximate query processing arises when data must be kept confidential so that only aggregate or macro-level data can be divulged. Approximate query processing also provides a solution to access of information from massive data tables.

One approach to approximate database querying through aggregates is sampling. However a join operation applied to two uniform random samples results in a non-uniform result, which furthermore is sparse (Chakrabarti, Garofalakis, Rastogi and Shim, 2001). A second approach is to keep histograms on the coordinates. For a multidimensional feature space, one is faced with a "curse of dimensionality" as the dimensionality grows. A third approach is wavelet-based, and is of interest to us in this article.

A form of progressive access to the data is sought, such that aggregated data can be obtained first, followed by greater refinement of the data. The Haar wavelet transform is a favored transform for such purposes, given that reconstructed data at a given resolution level is simply a recursively defined mean of data values. Vitter and Wang (1999) consider the combinatorial aspects of data access using a Haar wavelet transform, and based on a multiway data hypercube. Such data, containing scores or frequencies, is often found in the commercial data mining context of OLAP, On-Line Analytical

Processing.

As pointed out in Chakrabarti et al. (2001), one can treat multidimensional feature hypercubes as a type of high dimensional image, taking the given order of feature dimensions as fixed. As an alternative a uniform "shift and distribute" randomization can be used (Chakrabarti et al., 2001).

There are problems, however, in directly applying a wavelet transform to a data table. Essentially, a relational table (to use database terminology; or matrix) is treated in the same way as a 2-dimensional pixelated image, although the former case is invariant under row and column permutation, whereas the latter case is not (Murtagh, Starck and Berry, 2000). Therefore there are immediate problems related to non-uniqueness, and data order dependence.

What if, however, one organizes the data such that adjacency has a meaning? This implies that similarly-valued objects, and/or similarly-valued features, are close together. This is what we do, using any hierarchical clustering algorithm (e.g., the Ward or minimum variance one). Examples, to be discussed below, of hierarchical clustering results can be seen in Figures 1 and 2.

Without loss of generality, as seen in these figures, we assume that a hierarchy is a binary, rooted tree; and equivalently that the series of agglomerations involve precisely two clusters (possibly singleton clusters) at each of the $n - 1$ agglomerations where there are n observations. These n observations are usually represented by n row vectors in our data table.

A significant advantage in regard to hierarchical clustering is that partitions of the data can be read off at a succession of levels, and this obviates the need for fixing the number of clusters in advance. All possible clustering outcomes are considered. (Remark: of course, relative to any one of the commonly used cluster homogeneity criteria, each partition is guaranteed to be sub-optimal at best.)

3 Bases of the Hierarchic Haar Wavelet Transform

Linkages between the classical wavelet transform, as used in signal processing, and multivariate data analysis, were investigated in Murtagh (1998). The wavelet transform to be described now is fundamentally new, and works on a hierarchy.

The Haar wavelet transform can be simply described in terms of the following algorithm: recursively carry out averaging and differencing of adjacent pairs of data values (pixels, voxels, time steps, etc.) at a sequence

of geometrically (factor 2) increasing resolution levels. Our innovation is to apply the Haar wavelet transform to a binary rooted tree (viz., the clustering hierarchy) in terms of the following algorithm: recursively carry out pairwise averaging and differencing at the sequence of levels in the tree.

Consider any hierarchical clustering, H, represented as a binary rooted tree. For each cluster q'' with offspring nodes q and q', we define $s(q'')$ through application of the low-pass filter $\begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix}$ which can be implemented as a scalar product:

$$s(q'') = \frac{1}{2}\left(s(q) + s(q')\right) = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}^t \begin{pmatrix} s(q) \\ s(q') \end{pmatrix} \tag{1}$$

The application of the low-pass filter is carried out in order of increasing node number (i.e., from the smallest non-terminal node, through to the root node). For a terminal node, $s(i)$ is just the given vector, and this aspect is addressed further below, in subsection 3.1.

Next for each cluster q'' with offspring nodes q and q', we define detail coefficients $d(q'')$ through application of the band-pass filter $\begin{pmatrix} \frac{1}{2} \\ -\frac{1}{2} \end{pmatrix}$:

$$d(q'') = \frac{1}{2}(s(q) - s(q')) = \begin{pmatrix} 0.5 \\ -0.5 \end{pmatrix}^t \begin{pmatrix} s(q) \\ s(q') \end{pmatrix} \tag{2}$$

Again, increasing order of node number is used for application of this filter.

The scheme followed is illustrated in Figure 2, which shows the hierarchy (constructed by the median agglomerative method, although this plays no role here), using for display convenience just the first 8 observation vectors in Fisher's iris data (Fisher, 1936).

We call our algorithm a Haar wavelet transform because, traditionally, this wavelet transform is defined by a similar set of averages and differences. A more detailed study of why it can with justice be called a wavelet transform can be found in the companion paper, Paper II.

3.1 Two Cases Corresponding to Two Types of Input Data

We now return to the issue of how we start this scheme, i.e. how we define $s(i)$, or the "smooth" of a terminal node, representing a singleton cluster.

We consider two cases:

1. $s(i)$ is a vector in $\mathbb{R}^m$, and the ith row of a data table.

2. $s(i)$ is an n-dimensional indicator vector. So the third, in sequence, out of a population of $n = 8$ observations has indicator vector $\{00100000\}$. We can of course take a data table of all indicator vectors: it is clear that the data table is symmetric, and is none other than the identity matrix.

The first of these two cases implies that the hierarchy, H, represents a hierarchically-structured set of relationships, in the form of a taxonomy. From the point of view of practical application, this case is the more important one.

The second of these two cases may be of help in more directly representing an embedded set of sets.

Our hierarchical Haar wavelet transform can easily handle either case, depending on the input data table used.

3.2 The Inverse Transform

Constructing the hierarchical Haar wavelet transformed data is referred to as the forward transform. Reconstructing the input data is referred to as the inverse transform.

The inverse transform allows exact reconstruction of the input data. We begin with s_{n-1}. If this root node has subnodes q and q', we use $d(q)$ and $d(q')$ to form $s(q)$ and $s(q')$.

We continue, step by step, until we have reconstructed all vectors associated with terminal nodes.

4 Hierarchical Haar Wavelet Transform: Two Case Studies

In a practical way, using small data sets, we will describe our new hierarchical Haar wavelet transform in this section.

In the second input data case of subsection 3.1, consider the indicator vector of cluster q and of observation x. Thus in Figure 1, $x_1 = \{100000000\}$, and $q_1 = \{11000000\}$. The indicator vectors will be taken below as column vectors. This form of coding was used by Nabben and Varga (1994).

Now we use equations 1 and 2.

4.1 Case Study 1

In Figure 1, we have:

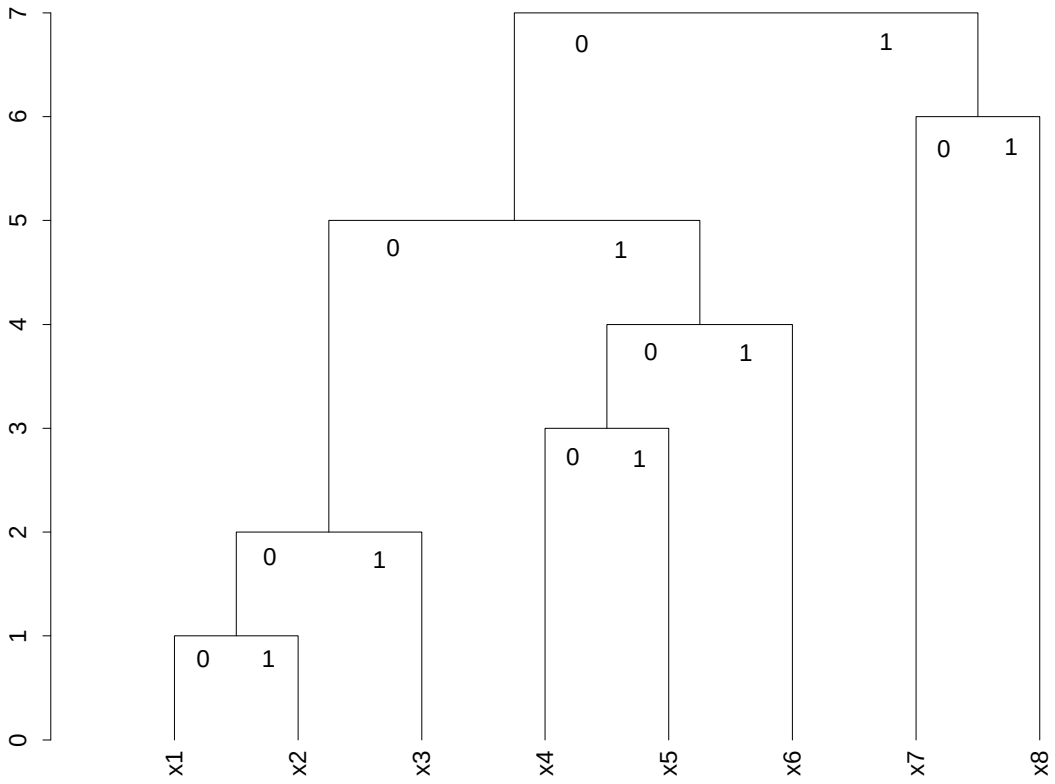

Figure 1: Labeled, ranked dendrogram on 8 terminal nodes. Branches labeled 0 and 1.

$s(q_1) = \frac{1}{2}(x_1 + x_2) = (\frac{1}{2}\ \frac{1}{2}\ 0\ 0\ 0\ 0\ 0\ 0)$
Also: $s(q_1) = \frac{1}{2}q_1$

$s(q_2) = \frac{1}{2}(s(q_1) + x_3) = (\frac{1}{4}\ \frac{1}{4}\ \frac{1}{2}\ 0\ 0\ 0\ 0\ 0)$
Also: $s(q_2) = \frac{1}{2}(\frac{1}{2}q_1 + x_3) = \frac{1}{4}q_1 + \frac{1}{2}x_3$

$s(q_3) = \frac{1}{2}(x_4 + x_5) = (0\ 0\ 0\ \frac{1}{2}\ \frac{1}{2}\ 0\ 0\ 0)$
Also: $s(q_3) = \frac{1}{2}q_3$

$s(q_4) = \frac{1}{2}(s(q_3) + x_6) = (0\ 0\ 0\ \frac{1}{4}\ \frac{1}{4}\ \frac{1}{2}\ 0\ 0)$

$s(q_5) = \frac{1}{2}(s(q_2) + s(q_4)) = (\frac{1}{8}\ \frac{1}{8}\ \frac{1}{4}\ \frac{1}{8}\ \frac{1}{8}\frac{1}{4}\ 0\ 0)$

$s(q_6) = \frac{1}{2}(x_7 + x_8) = (0\ 0\ 0\ 0\ 0\ 0\ \frac{1}{2}\ \frac{1}{2})$

$s(q_7) = \frac{1}{2}(q_5 + q_6) = (\frac{1}{16}\ \frac{1}{16}\ \frac{1}{8}\ \frac{1}{16}\ \frac{1}{16}\ \frac{1}{8}\ \frac{1}{4}\ \frac{1}{4})$

Next we turn attention to the detail coefficients.

$d(q_1) = \frac{1}{2}(x_1 - x_2) = (\frac{1}{2}\ -\frac{1}{2}\ 0\ 0\ 0\ 0\ 0\ 0)$
Alternatively, for $q'' = q \cup q'$, the detail coefficients are defined as: $d(q'') = s(q'') - s(q') = -(s(q'') - s(q))$.

Thus $d(q_1) = s(q_1) - x_2 = (\frac{1}{2}\ \frac{1}{2}\ 0\ 0\ 0\ 0\ 0\ 0) - (0\ 1\ 0\ 0\ 0\ 0\ 0\ 0) = (\frac{1}{2}\ -\frac{1}{2}\ 0\ 0\ 0\ 0\ 0\ 0)$

For any $d(q_j)$ we have: $\sum_k d(q_j)_k = 0$, i.e. the detail coefficient vectors are each of zero mean.

Let us redo in vector and matrix terms this description of the hierarchical Haar wavelet transform algorithm.

We take our initial or input data as follows.

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{3}$$

The hierarchical Haar wavelet transform of this input data is then as follows.

$$\begin{pmatrix} d(q_1) \\ d(q_2) \\ d(q_3) \\ d(q_4) \\ d(q_5) \\ d(q_6) \\ d(q_7) \\ s_7 \end{pmatrix} =$$

$$\begin{pmatrix} \frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} & -\frac{1}{2} & 0 & 0 \\ \frac{1}{8} & \frac{1}{8} & \frac{1}{4} & -\frac{1}{8} & -\frac{1}{8} & -\frac{1}{4} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{16} & \frac{1}{16} & \frac{1}{8} & \frac{1}{16} & \frac{1}{16} & \frac{1}{8} & -\frac{1}{4} & -\frac{1}{4} \\ \frac{1}{16} & \frac{1}{16} & \frac{1}{8} & \frac{1}{16} & \frac{1}{16} & \frac{1}{8} & \frac{1}{4} & \frac{1}{4} \end{pmatrix} \tag{4}$$

As already noted in this subsection, the succession of $n-1$ wavelet coefficient vectors are of zero mean. Therefore, due to the input data used (relation (3)), each row of the right hand matrix in equation 4 is of zero mean.

Note that this transform is a function of the hierarchy, H. Here we are using the hierarchy of Figure 1. H is needed to define the structure of the right hand matrix in equation 4.

4.2 Case Study 2

In Tables 1 and 2 we directly transform a small data set consisting of the first 8 observations in Fisher's iris data.

Note that in Table 2 it is entirely appropriate that at more smooth levels (i.e., as we proceed through levels d1, d2, ..., d6, d7) the values become more "fractionated" (i.e., there are more values after the decimal point).

The minimum variance agglomeration criterion, with Euclidean distance, is used to induce the hierarchy on the given data. Each detail signal is of dimension $m = 4$ where m is the dimensionality of the given data. The smooth signal is of dimensionality m also. The number of detail or wavelet signal levels is given by the number of levels in the labeled, ranked hierarchy, i.e. $n-1$.

	Sepal.L	Sepal.W	Petal.L	Petal.W
1	5.1	3.5	1.4	0.2
2	4.9	3.0	1.4	0.2
3	4.7	3.2	1.3	0.2
4	4.6	3.1	1.5	0.2
5	5.0	3.6	1.4	0.2
6	5.4	3.9	1.7	0.4
7	4.6	3.4	1.4	0.3
8	5.0	3.4	1.5	0.2

Table 1: First 8 observations of Fisher's iris data. L and W refer to length and width.

	s7	d7	d6	d5	d4	d3	d2	d1
Sepal.L	5.146875	0.253125	0.13125	0.1375	−0.025	0.05	−0.025	0.05
Sepal.W	3.603125	0.296875	0.16875	−0.1375	0.125	0.05	−0.075	−0.05
Petal.L	1.562500	0.137500	0.02500	0.0000	0.000	−0.10	0.050	0.00
Petal.W	0.306250	0.093750	−0.01250	−0.0250	0.050	0.00	0.000	0.00

Table 2: The hierarchical Haar wavelet transform resulting from use of the first 8 observations of Fisher's iris data shown in Table 1. Wavelet coefficient levels are denoted d1 through d7, and the continuum or smooth component is denoted s7.

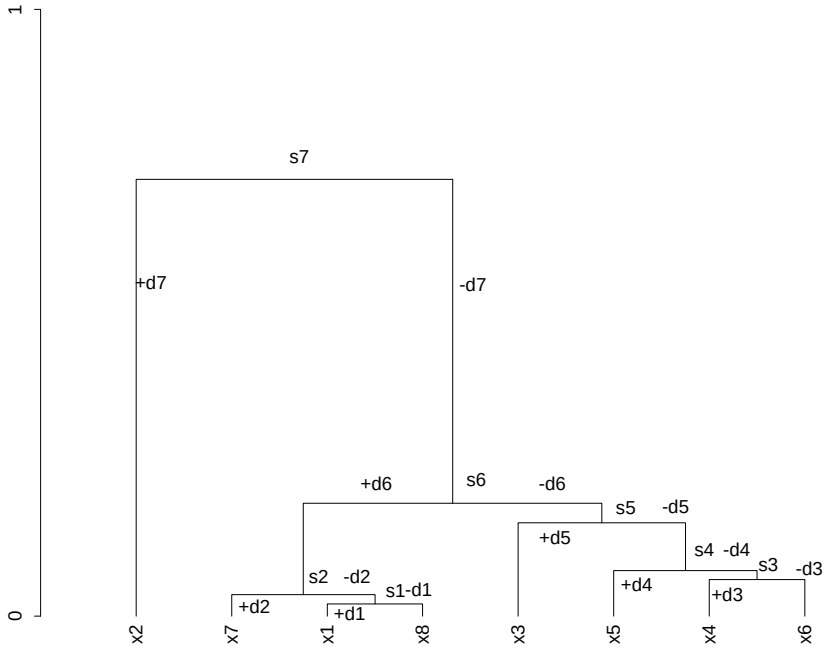

Figure 2: Dendrogram on 8 terminal nodes constructed from first 8 values of Fisher iris data. (Median agglomerative method used in this case.) Detail or wavelet coefficients are denoted by d, and data smooths are denoted by s.

5 Examples and Assessments of Hierarchical Wavelet Filtering

5.1 The Smoothing Algorithm

We use the following generic data analysis processing path, which is applicable to any input tabular data array of numerical values. We assume only that there are no missing values in the data array.

1. Given a dissimilarity, induce a hierarchy on the set of observations. (We generally use the Euclidean distance, and the minimum variance agglomerative hierarchical clustering criterion, in view of the synoptic properties (Murtagh, 1985). Additionally the vectors used in the clustering can be weighted: we use identical weights in this work.)

2. Carry out a Haar wavelet transform on this hierarchy. This gives a tree-based compaction of energy (Starck and Murtagh, 2002: large values tend to become larger, and small values tend to become smaller) in our data. Filter the wavelet coefficients (i.e., carry out wavelet regression or smoothing, here using hard thresholding by setting small wavelet coefficients to zero).

3. Determine the inverse of the wavelet transform, in order to reconstruct an approximation to the original multidimensional data values.

5.2 The Fisher Iris Data

In this first filtering study we use Fisher's iris data (Fisher, 1936), an array of dimensions 150×4, in view of its well known characteristics. If x_{ij} is a typical data value, then the energy of this data is $1/(nm) \sum_{ij} x_{ij}^2 = 15.8988$. If we set wavelet coefficients to zero based on a hard threshold, then a very large number of coefficients may be set to zero with minor implications for approximation of the input data by the filtered output. Table 3 shows this. The minimum variance hierarchical clustering method was used as the first phase of the processing, followed by the second, wavelet transform, phase. Then followed wavelet coefficient truncation, and reconstruction or the inverse transform. We see that a mean square error between input and output of value 0.1040 is the relatively good approximation quality, when nearly 98% of wavelet coefficients are zero-valued.

To further illustrate what is happening in this approximation by the wavelet filtered data, Table 4 shows the last 10 iris observations, as given

Filt. threshold	% coeffs. set to zero	mean square error
0	16.95	0
0.1	70.13	0.0098
0.2	91.95	0.0487
0.3	97.15	0.0837
0.4	97.82	0.1040

Table 3: Hierarchical Haar smoothing results for Fisher's 150×4 iris data.

for input, and as filtered. The numerical precisions shown are as generated in the reconstruction, which explains why we show some values to 4 decimal places, and some to 6.

To show that wavelet filtering is effective, we will next compare wavelet filtering with direct filtering of the given data. By "directly filtered" we mean that we processed the original data without recourse to a hierarchical clustering. This "straw man" processing is intended as a simple, default baseline with which we can compare our results.

Taking the *original* Fisher data, we find the median value to be 3.2. Putting values less than this median value to 0, we find the MSE to be 2.154567, i.e., implying a far less satisfactory fit to the data. (Thresholding by using $<$ versus $\leq$ median had no effect.)

5.3 Uniform Realization of Same Dimensions as Fisher Data

We next generated an array of dimensions 150×4 of uniformly distributed random values on $[0, 7.9]$, where 7.9 was the maximum value in the Fisher iris data. The energy of this data set was 21.2097. Results of filtering are shown in Table 5. The minimum variance hierarchical clustering method was used. Again good approximation properties are seen, even if the compression is not as impressive as for the Fisher data.

Uniformly distributed data coordinate values are a taxing case, since such data are very unlike data with clear cluster structures (as is the case for the Fisher iris data).

5.4 Inherent Clustering Structure in a Data Array

When a data set is inherently clustered (and possibly inherently hierarchically clustered) then the energy compaction properties of the wavelet transformed data ought to be correspondingly stronger. We will show this

	Sepal.L	Sepal.W	Petal.L	Petal.W
140	6.9	3.1	5.4	2.1
141	6.7	3.1	5.6	2.4
142	6.9	3.1	5.1	2.3
143	5.8	2.7	5.1	1.9
144	6.8	3.2	5.9	2.3
145	6.7	3.3	5.7	2.5
146	6.7	3.0	5.2	2.3
147	6.3	2.5	5.0	1.9
148	6.5	3.0	5.2	2.0
149	6.2	3.4	5.4	2.3
150	5.9	3.0	5.1	1.8
140	6.739063	3.119824	5.4125	2.239258
141	6.782813	3.307324	5.7250	2.564258
142	6.839063	3.119824	5.1125	2.239258
143	5.737500	2.808496	5.0000	2.039258
144	6.782813	3.307324	5.8250	2.314258
145	6.782813	3.307324	5.7250	2.564258
146	6.639063	3.119824	5.1125	2.239258
147	6.196875	2.480371	5.0000	1.964258
148	6.364063	3.019824	5.2625	2.089258
149	6.320313	3.307324	5.4750	2.439258
150	5.937500	3.008496	5.1375	1.864258

Table 4: Last 10 values of input data, and of the approximation to these based on the hierarchical Haar wavelet transform filtering with a hard threshold of 0.1, implying 70.13% of the wavelet coefficients equaling 0.

Filt. threshold	% coeffs. set to zero	mean square error
0	0	0
0.1	14.77	0.0022
0.2	31.54	0.0249
0.3	42.79	0.0622
0.4	53.52	0.1261

Table 5: Hierarchical Haar filtering results for uniformly distributed 150×4 data.

through the processing of data sets containing cluster structure relative to the processing of data sets containing uniformly distributed values (and hence providing a baseline for no cluster structure).

Firstly we verified that data set size is relatively unimportant in terms of wavelet-based smoothing. We took artificially generated, uniformly distributed in [0, 1], random data matrices of dimensions: 500×40, 1000×40, 1500×40, and 2000×40. For each we applied a fixed threshold of 0.1 to the wavelet coefficients, setting values less than or equal to this threshold to 0, and retaining wavelet coefficient values above this threshold, before reconstructing the data. Then we checked mean square error between reconstructed data and the original data. For the four different data matrix dimensions, we found: 0.463, 0.461, 0.465, 0.466.

From the clustering point of view, the foregoing data matrices are simply clouds of 500, 1000, 1500 and 2000 points in 40-dimensional real space, or $\mathbb{R}^{40}$. To check if space dimensionality could matter we checked the mean square error for a data matrix of uniformly distributed values with dimensions 2000×400, with the additional necessary adjustment for the total number of data values (viz., 800,000 for the matrix of dimensions 2000×400, as opposed to 80,000 data values for the matrix of dimensions 2000×40). With this relative adjustment, the mean square error was found to be 0.458. (Compare this to the mean square error of 0.466 for the 2000×40 data array, discussed in the previous paragraph.)

We conclude that neither embedding spatial dimensionality, i.e., number of columns in the data matrix, nor also data set size as given by the number of rows, are inherent determinants of the smoothing properties of our new method.

So what is important? Clearly if the hierarchical clustering is pulling large clusters together, and facilitating the "energy compaction" properties of the wavelet transform, then what is important is clustering structure in our data.

We generated structure by placing Gaussians centered at the following row, column locations in a 1200×400 data array: 300,100; 800,300; 1000,200; 500,150; 900,150. These bivariate Gaussians were of total 10 units in each case. A full width at half maximum (equal to 2.35482 times the standard deviation of a Gaussian), was used in each case, respectively: 20, 50, 10, 100, 125. We will call this the data array containing structure. Figure 3 shows a schematic view of it. This data array was motivated by an analogous image; but, unlike in the image case, our approach to processing a data array has absolutely no boundary effects or considerations.

As a benchmark, a second data array of the same dimensions 1200×400

Threshold	With structure	Random
0.05	1.28	1.35
0.10	4.56	4.61
0.15	8.95	8.99
0.20	13.73	13.76
0.25	18.49	18.52
0.30	23.03	23.07
0.40	30.27	30.31
0.50	33.24	33.29

Table 6: Mean square errors for the noisy data array with structure, A_s, in column 2, and the noisy random data array, A_r, in column 3. The fixed thresholds applied are in column 1. We see that the fit of filtered to input data is always best (i.e., lowest mean square error) for the structured data, column 2.

was used, containing uniformly distributed values in [0,1]. We will call this the random data array, A_r.

The random data array was added (element-wise) to the data array containing structure (so as to avoid lots of zero data values), yielding the noisy structured data array, A_s. The maximum value in the data array containing structure, A_s, now became 1.02, while the maximum value in the random data array, A_r, was 1. The total of all values in the data array containing structure, A_s, was 240553.3. The total of all values in the random data array, A_r, was 240504.3. Note how alike the "With structure" and "Random" data sets were; and consequently how close the results can be expected to be.

Fixed thresholds were applied in wavelet space and the data reconstructed. Table 6 shows the mean square error between input data and reconstructed data. What is noticeable about the better results seen here for column 2, A_s, compared to column 3, A_r, is that the values are somewhat bigger for A_s but nonetheless the fit to the input of wavelet filtered and reconstructed output is better for all threshold values.

5.5 An Example from Clustering of Texts

In a range of text data mining case studies (Murtagh, 2005) 910 short texts were used, consisting of chapters from three Jane Austen books (61 chapters from Pride and Prejudice, 24 chapters from Persuasion, 50 chapters from

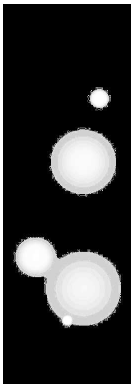

Figure 3: Visualization of the artificial structure defined from 5 different Gaussian distributions, before noise was added. Data array dimensions: 1200×400.

Sense and Sensibility, and 131 subchapters from Sense and Sensibility), 209 Grimm Brothers' fairy tales, 50 aviation accident reports from the National Transport Safety Board, and 385 dream reports from the online DreamBank repository. These 910 texts were characterized by the most frequent 500 words appearing in the entire collection. Further details can be found in Murtagh (2005). A correspondence analysis was carried out, and the first two factors – see Figure 4 – were used to characterize the easily distinguished texts. The cloud of points to the lower right (denoted "a") are the technical aviation accident reports. The well demarcated, lower part of the upper left clusters comprises all of the Jane Austen material (denoted "1", "2", "3" and "4"). The clump in the middle of the upper left (denoted "g") are the Grimm Brothers' fairy tales. The upper clump (denoted "d") are the various dream reports.

We used the two coordinate, principal factor, representation, as shown in Figure 4, to characterize the 910 texts, A_s. As a baseline we generated uniformly distributed random two-dimensional coordinates, A_r, with the same minimum and maximum values as in the case of A_s.

Applying a filtering threshold of 0.5 to the wavelet coefficients gave a mean square error between A_s and reconstructed A_s of 0.467. Applying the same threshold led to the error between A_r and the reconstructed A_r as the somewhat worse value of 0.535.

Again data with cluster structure is better wavelet-smoothed compared to data with no cluster structure.

While all examples shown here underscore the superiority of the combined clustering and wavelet transform approach described, we will now mention some limitations. Firstly, a baseline of random data giving a better fit to the input can be found if the random realization is based on the empirical distribution function of the given data. Hence our approach is not to be considered as a means of testing clustering structure, nor even of departure from randomness. Secondly, as exemplified by the case studies involving Figures 3 and 4, the cluster structure has to be quite pronounced. Again, we note that our combined clustering/wavelet approach makes use of clear structure in the data, and is not to be considered as providing a means for testing clustering structure.

5.6 Approximating a Hierarchy by Collapsing Clusters

A binary rooted tree, H, on n observations has precisely $n-1$ levels; or H contains precisely $n-1$ subsets of the set of n observations. The interpretation of a hierarchical clustering often is carried out by cutting the tree to

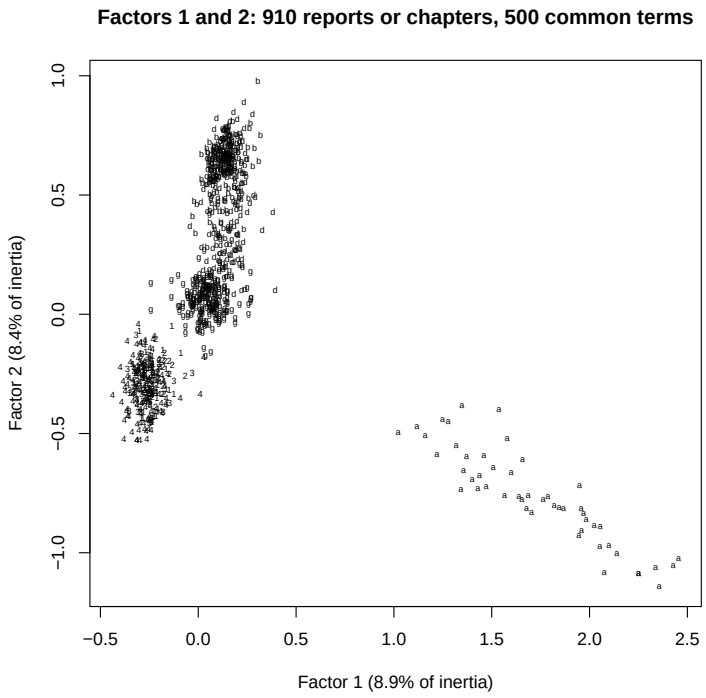

Figure 4: Positions of 910 texts, crossed by their 500 most common words, shown in the correspondence analysis principal factor plane.

yield any one of the $n-1$ possible partitions. Our hierarchical Haar wavelet transform affords us a neat way to approximate H using a smaller number of possible partitions.

Consider the detail vector at any given level: cf. the examples exemplified in relation 4 or Table 2. Any such detail vector is associated with (i) a node of the binary tree; (ii) the level or height index of that node; and (iii) a cluster, or subset of the observation set. In the data approximation or filtering, so far, we have set coordinate values of the each detail vector to zero. Now, with the goal of "collapsing" clusters, i.e. removing clusters that are not unduly valuable for interpretation, we will impose a hard threshold on each detail vector. Although other rules could be considered, we will assess here the use of the following threshold:
If the norm of the detail vector is less than a user-specified threshold, then set all values of the detail vector to zero.

Other rules could be chosen, in particular rules related directly to the agglomerative clustering criterion used. Our norm-based rule is not directly related to the agglomerative criterion for the following reasons: (i) we seek a generic interpretative aid, rather than an optimal but criterion-specific rule; (ii) an optimal, criterion-specific rule would in any case be best addressed by studying the overall optimality measure rather than availing of the stepwise suboptimal hierarchical clustering; and (iii) from naturally occurring hierarchies, as occur in very high dimensional spaces, the issue of an agglomerative criterion is not important.

Following use of the norm-based cluster collapsing rule, the representation of the reconstructed hierarchy is straightforward: the hierarchy's level index is adjusted so that the *previous* level index additionally takes the place of the given level index. The following example will exemplify this.

We took Aristotle's *Categories* (see Murtagh, 2005) containing 14,483 individual words. We broke up the text into 24 files, in order to study the sequential properties of the argument developed in this short philosophical work. In these 24 files, there were 1269 unique words. We selected 66 nouns of particular interest. With frequencies of occurrence in parentheses we had (sample only): man (104), contrary (72), same (71), subject (60), substance (58), species (54), knowledge (50), qualities (47), etc. A correspondence analysis was carried out on the 66×24 table of frequencies with the aim of taking the set of 66 nouns endowed with the χ^2 metric (i.e., a weighted Euclidean distance between *profiles* into a factor space endowed with the (unweighted) Euclidean metric. A hierarchical clustering (minimum variance method) was carried out on the factor coordinates of the 66 nouns.

The norms of detail vectors had minimum, median and maximum values

as follows: 0.0866, 0.5408 and 3.042, and these influenced the choice of threshold. Applying thresholds of 0, 1.3, 1.9 and 2.3 gave rise to the following numbers of "collapsed" clusters (with in brackets the mean squared error between approximated data and original input data): 0 (0.0), 43 (0.1558), 58 (0.1878), and 62 (0.2099). Figure 5 shows the corresponding reconstructed and approximated hierarchies.

In the case of the threshold 1.9 (lower left in Figure 5) we have noted that 58 clusters were collapsed, leaving just 8 partitions. As stated the objective here is precisely to approximate the dendrogram output data structure in order to facilitate further study and interpretation of these partitions.

5.7 Traditional versus Hierarchical Haar Wavelet Transforms

Consider a set of 8 input data objects, each of which is scalar:
$(64, 48, 16, 32, 56, 56, 48, 24)$. A traditional Haar wavelet transform of this data can be quickly done, and gives: $(43, -3, 16, 10, 8, -8, 0, 12)$. Here, the first value is the final smooth, and the remaining values are the wavelet coefficients read off by a traversal from final smooth towards the input data values. Showing the output in the same way, the hierarchical Haar wavelet transform of the same data gives: $(40, 14, 6, -6, -4, 4, 0, 0)$.

A little reflection shows that the greater number of zeros in the hierarchical Haar wavelet transform is no accident. In fact, with the following conditions: 10 different digits in the input data; processing of an n-length string of digits; use of an unweighted average agglomerative criterion; $n - 10 = 2^k$ for some integer k; then the number of zero wavelet coefficients will be $n - 10$. This remarkable result points to the powerful data compression potential of the hierarchical Haar wavelet transform. We must note though that this rests on the dendrogram, and the computational requirements of the latter are not in any way bypassed.

5.8 Computational Complexity Properties

The computational complexity of our algorithms are as follows. The hierarchical clustering is $O(n^2)$, coded in C (and earlier in Fortran and Java). All other programs, as follows, were coded initially in R and then, for efficiency, in C++. The forward hierarchical Haar wavelet transform is $O(n)$. The filtering is $O(n)$. Finally, the inverse wavelet transform is $O(n^2)$. All programs are run from an R harness. On Macintosh G4 or G5 machines, all phases of the processing took 4–5 minutes for the 12000×400 array. The data sets used in subsection 5.4 were generated using the MR Multiresolution

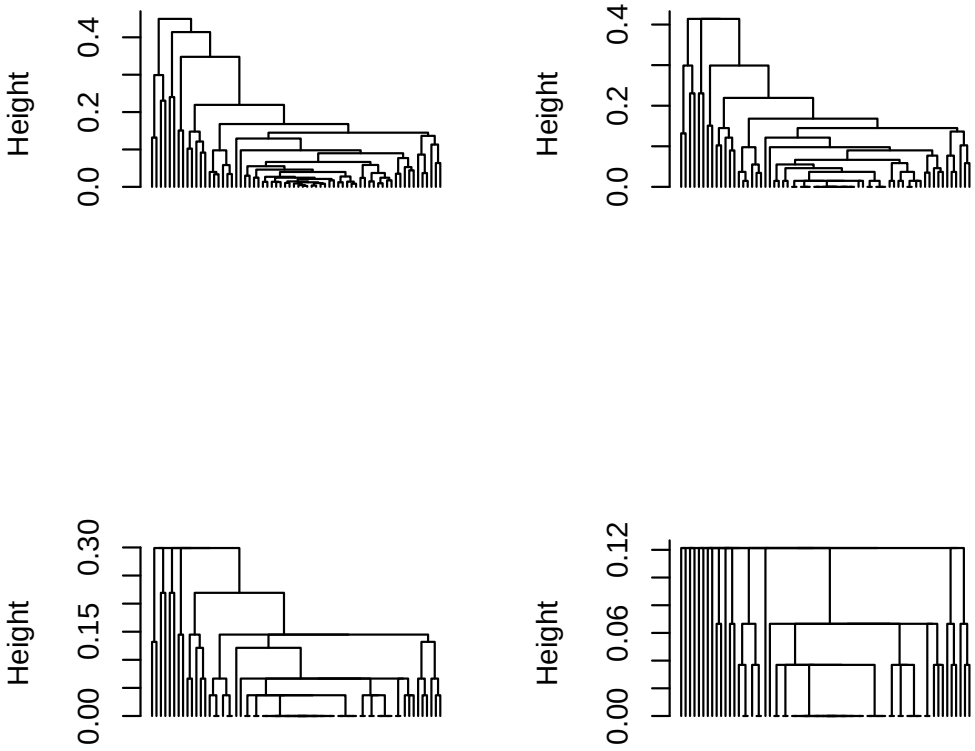

Figure 5: Upper left: original hierarchy. Upper right, lower left, and lower right show increasing approximations to the original hierarchy based on the "cluster collapsing" approach described.

Analysis software (described in Starck and Murtagh, 2002).

A pipeline of C and R code used in this work (which can be used with, for example, the Fisher iris data) is available at the following address: http://astro.u-strasbg.fr/∼fmurtagh/mda-sw

6 Conclusion

It is interesting to compare some global properties of our approach relative to the Fourier transform approach applied to decision trees in Kargupta and Park (2004). The Fourier transform lends itself well to a frequency spectrum analysis of binary decision vectors, and the latter can be of importance for supervised classification. On the other hand, our work has made use of binary trees but in the framework of unsupervised classification. The wavelet transform shares with the Fourier transform the property that frequency spectral information is determined from the data; and the wavelet transform additionally determines spatial or resolution scale information from the data. We have found the wavelet transform, as described in this article, to be appropriate for the type of input data that we have considered. In general terms, both we in this article, and Kargupta and Park (2004), have as objectives the filtering and compression of data.

We have described a novel hierarchical wavelet transform, developed to allow wavelet filtering of data tables. It relies on an available hierarchic clustering of the data table, and for that we have generally (but not exclusively) used Ward's minimum variance agglomerative hierarchical clustering in this work.

We have described this new method through a number of small examples. We then used the well known Fisher iris data to show how wavelet filtering performs in practice. Furthermore we took a range of randomly (uniformly) generated data tables, to investigate the filtering and the scaling properties.

We used two cases studies involving synthetic data with cluster structure, and data on 910 short texts, to show that the cluster structure is beneficial for this approach to data smoothing.

Finally we again underline the innovation in this approach: for handling data arrays, it is not acceptable to employ approaches developed for very different types of data (e.g., 2-dimensional images, time series, etc.).

A further motivation for the development of wavelet transforms based on hierarchical data structures is to cater for "naturally" (in some sense) hierarchically structured data. Recent work has shown that very high dimensional, spatially sparse, data can be considered as naturally hierarchi-

cally structured. Examples of such high-dimensional data include speech analysis, genomics and proteomics, and other fields (Rammal, Toulouse and Virasoro, 1986; Murtagh, 2004). This points to very exciting possibilities in data mining of very high dimensional data sets.

Acknowledgements

Dimitri Zervas converted the hierarchical clustering and new Haar wavelet transform into C and C++ from the author's R and Java codes.

References

[1] Benzécri, J.P. (1979). *La Taxinomie*, 2nd ed., Paris: Dunod.

[2] Chakrabarti, K., Garofalakis, M., Rastogi, R. and Shim, K. (2001). "Approximate Query Processing using Wavelets", *VLDB Journal, International Journal on Very Large Databases*, 10, 199–223.

[3] Fisher, R.A. (1936). "The Use of Multiple Measurements in Taxonomic Problems", *The Annals of Eugenics*, 7, 179–188.

[4] Härdle, W. (2000). *Wavelets, Approximation, and Statistical Applications*, Berlin: Springer.

[5] Joe, M.J., Whang, K.-Y. and Kim, S.-W. (2001). "Wavelet Transformation-Based Management of Integrated Summary Data for Distributed Query Processing", *Data and Knowledge Engineering*, 39, 293–312.

[6] Johnson, S.C. (1967). "Hierarchical Clustering Schemes", Psychometrika, 32, 241–254.

[7] Kargupta, H. and Park, B.-H. (2004). "A Fourier Spectrum-Based Approach to Represent Decision Trees for Mining Data Streams in Mobile Environments", *IEEE Transactions on Knowledge and Data Engineering*, 16, 216–229.

[8] Murtagh, F. (1985). *Multidimensional Clustering Algorithms*, Würzburg: Physica-Verlag.

[9] Murtagh, F. (1998). "Wedding the Wavelet Transform and Multivariate Data Analysis", *Journal of Classification*, 15, 161–183.

[10] Murtagh, F., Starck, J.-L. and Berry, M. (2000). "Overcoming the Curse of Dimensionality in Clustering by Means of the Wavelet Transform", *The Computer Journal*, 43, 107–120.

[11] Murtagh, F. (2004). "On Ultrametricity, Data Coding, and Computation", *Journal of Classification*, 21, 167–184.

[12] Murtagh, F. (2005). *Correspondence Analysis and Data Coding with Java and R*, Chapman and Hall.

[13] Nabben R. and Varga, R.S. (1994). "A Linear Algebra Proof that the Inverse of a Strictly Ultrametric Matrix is a Strictly Diagonal Dominant Stieltjes Matrix", *SIAM Journal on Matrix Analysis and Applications*, 15, 107–113.

[14] Paper II (2005). "The Haar Wavelet Transform of a Dendrogram – II", companion paper.

[15] Rammal, R., Toulouse, G. and Virasoro, M.A. (1986). "Ultrametricity for Physicists", *Reviews of Modern Physics*, 58, 765–788.

[16] Starck J.-L. and Murtagh, F. (2002). *Astronomical Image and Data Analysis*, Heidelberg: Springer. Chapter 9: "Multiple Resolution in Data Storage and Retrieval".

[17] Tao Li, Qi Li, Shenghuo Zhu, and Mitsunori Ogihara (2002). "A Survey on Wavelet Applications in Data Mining", SIGKDD Explorations, 4, 49–68.

[18] Vitter, J.S. and Wang, M. (1999). "Approximate Computation of Multidimensional Aggregates of Sparse Data using Wavelets", in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 193–204.