Ultrametric and Generalized Ultrametric in Logic and in Data Analysis

Fionn Murtagh
Science Foundation Ireland
Wilton Park House, Wilton Place, Dublin 2, Ireland
and
Department of Computer Science
Royal Holloway, University of London
Egham TW20 0EX, UK
fmurtagh@acm.org

August 24, 2010

Abstract

Following a review of metric, ultrametric and generalized ultrametric, we review their application in data analysis. We show how they allow us to explore both geometry and topology of information, starting with measured data. Some themes are then developed based on the use of metric, ultrametric and generalized ultrametric in logic. In particular we study approximation chains in an ultrametric or generalized ultrametric context. Our aim in this work is to extend the scope of data analysis by facilitating reasoning based on the data analysis; and to show how quantitative and qualitative data analysis can be incorporated into logic programming.

1 Introduction

The applicability of metric spaces to applications related to logic has long been known. For example Lawvere [19, 20] starts with the observation of the analogy of the triangular inequality and a categorical composition law. A comprehensive survey of this area can be found in [36].

Hierarchies as used in data analysis are presented in terms of finding various forms of symmetry in data in [29]. We could describe hierarchy built from pairwise dissimilarities as a "precision tool" for data mining; and hierarchies built from the generalized ultrametric (see section 4) as leading to a "power tool" for data mining. The former is (without special algorithmic speedups) typically quadratic or $O(n^2)$ in its computational requirement. The latter can be linear or $O(n)$ in its computation. Here n relates to number of observations.

We begin in section 2 with data analysis. We motivate the hierarchical structuring of data, describing at a general level how the geometry and the topology of information come into play, related respectively to metric and ultrametric embedding of data.

In section 3 we show how hierarchy, induced from data, can be made use of for approximating data. The latter, approximating data, is applicable and important for computational purposes.

In logic, chains of implications or conditionals have to be analyzed. When we consider a partial order of conditionals, then the framework of spherical (ultrametric) completeness or inductive limit (sections 4.1 and especially 3.1) become very useful indeed.

In section 4.1, we will look at how, [5], a "*computable real number* is ... the lub [least upper bound] of a shrinking sequence of rational intervals which is generated by a master program", and therefore how a real number is computable "in the interval approach to computability on the real line".

The convergence to fixed points that are based on a generalized ultrametric system is precisely the study of spherically complete systems and expansive automorphisms discussed in section 3.1. As expansive automorphisms we see here again an example of data and information symmetry at work.

2 From Metric to Ultrametric Topology

We will discuss how an ultrametric topology – a tree structuring of the data – is induced from data, using pairwise dissimilarities.

2.1 Pairwise Dissimilarities

Given an observation set, X, we define dissimilarities as the mapping $d : X \times X \longrightarrow \mathbb{R}^+$, where $\mathbb{R}^+$ are the positive reals. A dissimilarity is a positive, definite, symmetric measure (i.e., $d(x, y) \geq 0; d(x, y) = 0$ if $x = y; d(x, y) = d(y, x)$). If in addition the triangular inequality is satisfied (i.e., $d(x, y) \leq d(x, z) + d(z, y), \forall x, y, z \in X$) then the dissimilarity is a distance.

2.1.1 From Dissimilarities to an Ultrametric

If X is endowed with a metric, then we now describe how this metric is mapped onto an ultrametric. In practice, there is no need for X to be endowed with a metric. Instead a dissimilarity is satisfactory.

A hierarchy, H, is defined as a binary, rooted, node-ranked tree, also termed a dendrogram [3, 16, 21, 24]. A hierarchy defines a set of embedded subsets of a given set of objects X, indexed by the set I. These subsets are totally ordered by an index function ν, which is a stronger condition than the partial order required by the subset relation. A bijection exists between a hierarchy and an ultrametric space.

Let us show these equivalences between embedded subsets, hierarchy, and binary tree, through the constructive approach of inducing H on a set I.

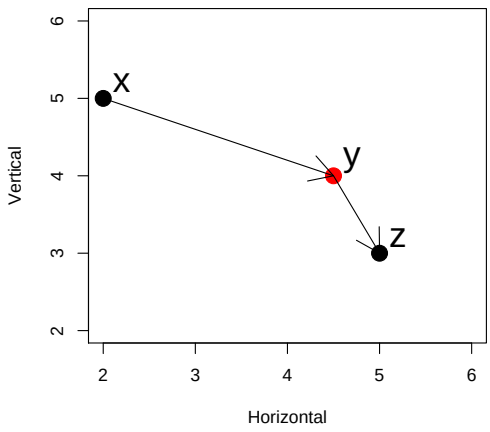


Figure 1: The triangular inequality defines a metric: every triplet of points satisfies the relationship: $d(x,z) \leq d(x,y) + d(y,z)$ for distance d.

Hierarchical agglomeration on n observation vectors with indices $i \in I$ involves a series of $1, 2, \ldots, n-1$ pairwise agglomerations of observations or clusters, with the following properties. A hierarchy $H = \{q | q \in 2^I\}$ such that (i) $I \in H$, (ii) $i \in H \; \forall i$, and (iii) for each $q \in H, q' \in H : q \cap q' \neq \emptyset \implies q \subset q'$ or $q' \subset q$. Here we have denoted the power set of set I by 2^I. An indexed hierarchy is the pair (H, ν) where the positive function defined on H, i.e., $\nu : H \to \mathbb{R}^+$, satisfies: $\nu(i) = 0$ if $i \in H$ is a singleton; and $q \subset q' \implies \nu(q) < \nu(q')$. Here we have denoted the positive reals, including 0, by $\mathbb{R}^+$. Function ν is the agglomeration level. Take $q \subset q'$, let $q \subset q''$ and $q' \subset q''$, and let q'' be the lowest level cluster for which this is true. Then if we define $D(q, q') = \nu(q'')$, D is an ultrametric. In practice, we start with a Euclidean or alternative dissimilarity, use some criterion such as minimizing the change in variance resulting from the agglomerations, and then define $\nu(q)$ as the dissimilarity associated with the agglomeration carried out.

2.2 Metric and Ultrametric for Geometry and Topology of Information

The *geometry of information* is a term and viewpoint used by [37]. The triangular inequality holds for metrics. An example of a metric is the Euclidean distance, exemplified in Figure 1, where each and every triplet of points satisfies the relationship: $d(x,z) \leq d(x,y) + d(y,z)$ for distance d. Two other relationships also must hold. These are symmetry and positive definiteness, respectively: $d(x,y) = d(y,x)$, and $d(x,y) > 0$ if $x \neq y$, $d(x,y) = 0$ if $x = y$.

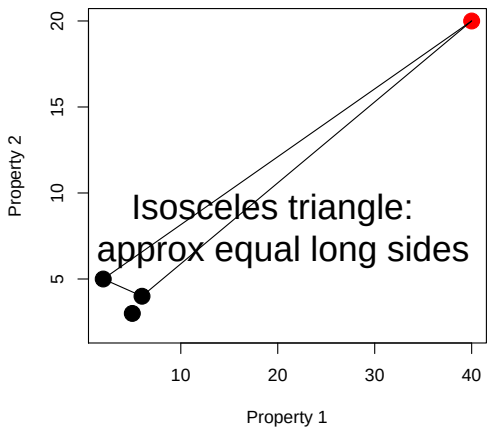


Figure 2: The query is on the far right. While we can easily determine the closest target (among the three objects represented by the dots on the left), is the closest really that much different from the alternatives?

We come now to a different principle: that of the *topology of information*. The particular topology used is that of hierarchy. Euclidean embedding provides a very good starting point to look at hierarchical relationships. An innovation in our work is as follows: the hierarchy takes sequence, e.g. timeline, into account. This captures, in a more easily understood way, the notions of novelty, anomaly or change.

Let us take an informal case study to see how this works. Consider the situation of seeking documents based on titles. If the target population has at least one document that is close to the query, then this is (let us assume) clearcut. However if all documents in the target population are very unlike the query, does it make any sense to choose the closest? Whatever the answer here we are focusing on the inherent ambiguity, which we will note or record in an appropriate way. Figure 2 illustrates this situation, where the query is the point to the right.

By using approximate similarity this situation can be modeled as an isosceles triangle with small base, as illustrated in Figure 2. An ultrametric space has properties that are very unlike a metric space, and one such property is that the only triangles allowed are either (i) equilateral, or (ii) isosceles with small base. So Figure 2 can be taken as representing a case of ultrametricity. What this means is that the query can be viewed as having a particular sort of dominance or hierarchical relationship vis-à-vis any pair of target documents. Hence any triplet of points here, one of which is the query (defining the apex of the isosceles, with small base, triangle), defines local hierarchical or ultrametric structure.

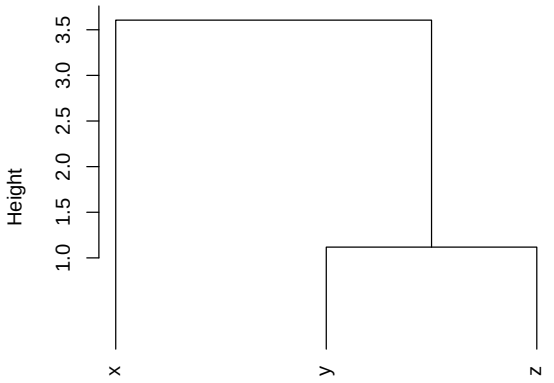


Figure 3: The strong triangular inequality defines an ultrametric: every triplet of points satisfies the relationship: $d(x,z) \leq \max\{d(x,y), d(y,z)\}$ for distance d. Cf. by reading off the hierarchy, how this is verified for all x, y, z: $d(x,z) = 3.5; d(x,y) = 3.5; d(y,z) = 1.0$. In addition the symmetry and positive definiteness conditions hold for any pair of points.

(See [26] for case studies.)

It is clear from Figure 2 that we should use approximate equality of the long sides of the triangle. The further away the query is from the other data then the better is this approximation [26].

What sort of explanation does this provide for our conundrum? It means that the query is a novel, or anomalous, or unusual "document". It is up to us to decide how to treat such new, innovative cases. It raises though the interesting perspective that here we have a way to model and subsequently handle the semantics of anomaly or innocuousness.

The strong triangular inequality, or ultrametric inequality, holds for tree distances: see Figure 3. The closest common ancestor distance is such an ultrametric.

2.3 Hierarchical Agglomerative Clustering

Since pairwise dissimilarities are used in constructing the hierarchy, the computation complexity of hierarchical clustering is at least $O(n^2)$. As the closest clusters (including singletons) are agglomerated at each of $n-1$ agglomerations (card X = card $I = n$), the newly created cluster must be related to others.

This is part and parcel of the agglomeration criterion, and can be viewed either as the cluster update rule, or the agglomerative criterion (e.g., based on compactness, or connectivity).

The most efficient algorithms are based on nearest neighbor chains, which by definition end in a pair of agglomerable reciprocal nearest neighbors. $O(n^2)$ computation time is guaranteed. The uniqueness and acceptability of on-the-fly agglomeration based on reciprocal nearest neighbors can be proven (respectively, disproven) for the given agglomerative criterion. The reciprocal nearest neighbor algorithm was first proposed in two articles in the journal *Les Cahiers de l'Analyse des Données* in 1980 and 1982, and are now used in software packages such as Clustan and R. Further information can be found in [22, 23, 24, 25].

2.4 Hierarchy as the Wreath Product Group expressing Symmetries

A dendrogram like that shown in Figure 1 is invariant relative to rotation (alternatively, here: permutation) of left and right child nodes. These rotation (or permutation) symmetries are defined by the wreath product group (see [8, 9, 7] for an introduction and applications in signal and image processing), and can be used with any m-ary tree, although we will treat the binary case here.

For the group actions, with respect to which we will seek invariance, we consider independent cyclic shifts of the subnodes of a given node (hence, at each level). Equivalently these actions are adjacency preserving permutations of subnodes of a given node (i.e., for given q, with $q = q' \cup q''$, the permutations of $\{q', q''\}$). We have therefore cyclic group actions at each node, where the cyclic group is of order 2.

The symmetries of H are given by structured permutations of the terminals. The terminals will be denoted here by Term H. The full group of symmetries is summarized by the following generative algorithm:

1. For level $l = n - 1$ down to 1 do:
2. Selected node, $\nu \longleftarrow$ node at level l.
3. And permute subnodes of ν.

Subnode ν is the root of subtree H_ν. We denote H_{n-1} simply by H. For a subnode ν' undergoing a relocation action in step 3, the internal structure of subtree $H_{\nu'}$ is not altered.

The algorithm described defines the automorphism group which is a wreath product of the symmetric group. Denote the permutation at level ν by P_ν. Then the automorphism group is given by:

$$G = P_{n-1} \text{ wr } P_{n-2} \text{ wr } \ldots \text{ wr } P_2 \text{ wr } P_1$$

where wr denotes the wreath product.

Call Term H_ν the terminals that descend from the node at level ν. So these are the terminals of the subtree H_ν with its root node at level ν. We can alternatively call Term H_ν the cluster associated with level ν.

We will now look at shift invariance under the group action. This amounts to the requirement for a constant function defined on Term $H_\nu, \forall \nu$. A convenient way to do this is to define such a function on the set Term H_ν via the root node alone, ν. By definition then we have a constant function on the set Term H_ν.

Let us call V_ν a space of functions that are constant on Term H_ν. Possible bases of V_ν that were considered in [27] are:

1. Basis vector with $|\text{Term}H_{n-1}|$ components, with 0 values except for value 1 for component i.
2. Set (of cardinality $n = |\text{Term}H_{n-1}|$) of m-dimensional observation vectors.

The constant function for each node or level ν is:

$$L : \text{Term}H_\nu \longrightarrow V_\nu$$

Consider the resolution scheme arising from moving from $\{\text{Term}H_{\nu'}, \text{Term}H_{\nu''}\}$ to $\text{Term}H_\nu$. From the hierarchical clustering point of view it is clear what this represents, simply, an agglomeration of two clusters called Term $H_{\nu'}$ and Term $H_{\nu''}$, replacing them with a new cluster, Term H_ν.

Let the spaces of constant functions corresponding to the two cluster agglomerands be denoted $V_{\nu'}$ and $V_{\nu''}$. These two clusters are disjoint initially, which motivates us taking the two spaces as a couple: $(V_{\nu'}, V_{\nu''})$. In the same way, let the space of constant functions corresponding to node ν be denoted V_ν.

Let us exemplify a case that satisfies all that has been defined in the context of the wreath product invariance that we are targeting. It is the algorithm discussed in depth in [27]. Take the constant function on $V_{\nu'}$ to be $f_{\nu'}$. Take the constant function on $V_{\nu''}$ to be $f_{\nu''}$. Then define the constant function, the *scaling function*, on V_ν to be $(f_{\nu'} + f_{\nu''})/2$. Next define the zero mean function, $(w_{\nu'} + w_{\nu''})/2 = 0$, the *wavelet function*, as follows:

$$w_{\nu'} = (f_{\nu'} + f_{\nu''})/2 - f_{\nu'}$$

in the support interval of $V_{\nu'}$, i.e. Term $H_{\nu'}$, and

$$w_{\nu''} = (f_{\nu'} + f_{\nu''})/2 - f_{\nu''}$$

in the support interval of $V_{\nu''}$, i.e. Term $H_{\nu''}$.

Since $w_{\nu'} = -w_{\nu''}$ we have the zero mean requirement.

3 Approximation in an Ultrametric Topology

We now seek to use a hierarchical clustering for successively approximating an object. In [28] we have examples of application to facial recognition and textual analysis.

Following a general view of hierarchical approximation in subsection 3.1, we then proceed to an algorithm, and a data analysis framework, to support hierarchical approximation.

3.1 Approximation from a Hierarchy: Dilation Operation as p-Adic Multiplication by $1/p$

Scale-related symmetry is very important in practice. In this subsection we introduce an operator that provides this symmetry. We also term it a dilation operator, because of its role in the wavelet transform on trees (see [27] for discussion and examples).

First we introduce a p-adic encoding of a hierarchy, using Figure 4 as an example. By means of terminal-to-root traversals, we define the following p-adic encoding of terminal nodes, and hence objects, in Figure 4.

$$
\begin{aligned}
x_1: &\quad +1\cdot p^1 + 1\cdot p^2 + 1\cdot p^5 + 1\cdot p^7 \\
x_2: &\quad -1\cdot p^1 + 1\cdot p^2 + 1\cdot p^5 + 1\cdot p^7 \\
x_3: &\quad -1\cdot p^2 + 1\cdot p^5 + 1\cdot p^7 \\
x_4: &\quad +1\cdot p^3 + 1\cdot p^4 - 1\cdot p^5 + 1\cdot p^7 \\
x_5: &\quad -1\cdot p^3 + 1\cdot p^4 - 1\cdot p^5 + 1\cdot p^7 \\
x_6: &\quad -1\cdot p^4 - 1\cdot p^5 + 1\cdot p^7 \\
x_7: &\quad +1\cdot p^6 - 1\cdot p^7 \\
x_8: &\quad -1\cdot p^6 - 1\cdot p^7
\end{aligned}
\tag{1}
$$

If we choose $p = 2$ the resulting decimal equivalents could be the same: cf. contributions based on $+1 \cdot p^1$ and $-1 \cdot p^1 + 1 \cdot p^2$. Given that the coefficients of the p^j terms ($1 \leq j \leq 7$) are in the set $\{-1, 0, +1\}$ (implying for x_1 the additional terms: $+0 \cdot p^3 + 0 \cdot p^4 + 0 \cdot p^6$), the coding based on $p = 3$ is required to avoid ambiguity among decimal equivalents.

Consider the set of objects $\{x_i | i \in I\}$ with its p-adic coding considered above. Take $p = 2$. (Non-uniqueness of corresponding decimal codes is not of concern to us now, and taking this value for p is without any loss of generality.) Multiplication of $x_1 = +1 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^5 + 1 \cdot 2^7$ by $1/p = 1/2$ gives: $+1 \cdot 2^1 + 1 \cdot 2^4 + 1 \cdot 2^6$. Each level has decreased by one, and the lowest level has been lost. Subject to the lowest level of the tree being lost, the form of the tree remains the same. By carrying out the multiplication-by-$1/p$ operation on all objects, it is seen that the effect is to rise in the hierarchy by one level.

Let us call product with $1/p$ the operator A. The effect of losing the bottom level of the dendrogram means that either (i) each cluster (possibly singleton) remains the same; or (ii) two clusters are merged. Therefore the application of A to all q implies a subset relationship between the set of clusters $\{q\}$ and the result of applying A, $\{Aq\}$.

Repeated application of the operator A gives Aq, A^2q, A^3q, ... Starting with any singleton, $i \in I$, this gives a path from the terminal to the root node

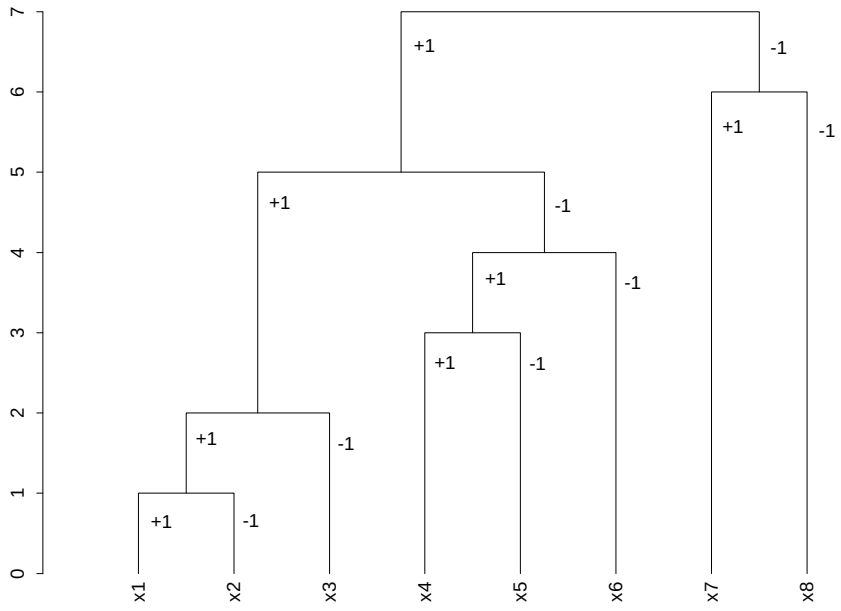


Figure 4: Labeled, ranked dendrogram on 8 terminal nodes, $x_1, x_2, \ldots, x_8$. Branches are labeled $+1$ and -1. Clusters are: $q_1 = \{x_1, x_2\}, q_2 = \{x_1, x_2, x_3\}, q_3 = \{x_4, x_5\}, q_4 = \{x_4, x_5, x_6\}, q_5 = \{x_1, x_2, x_3, x_4, x_5, x_6\}, q_6 = \{x_7, x_8\}, q_7 = \{x_1, x_2, \ldots, x_7, x_8\}$.

	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.

in the tree. Each such path ends with the null element, and therefore the intersection of the paths equals the null element.

Benedetto and Benedetto [1, 2] discuss A as an expansive automorphism of I, i.e. form-preserving, and locally expansive. Some implications [1] of the expansive automorphism follow. For any q, let us take $q, Aq, A^2q, \ldots$ as a sequence of open subgroups of I, with $q \subset Aq \subset A^2q \subset \ldots$, and $I = \bigcup\{q, Aq, A^2q, \ldots\}$. This is termed an inductive sequence of I, and I itself is the inductive limit ([32], p. 131).

Each path defined by application of the expansive automorphism defines a spherically complete system [34, 10, 33], which is a formalization of well-defined subset embeddedness.

3.2 Haar Wavelet Transform of a Dendrogram

Determining successive approximations of data, based on the data itself, leads us to the Haar wavelet transform of a hierarchy, or on a dendrogram.

The discrete wavelet transform is a decomposition of data into spatial and frequency components. In terms of a dendrogram these components are with respect to, respectively, within and between clusters of successive partitions. We show how this works taking the data of Table 1.

The hierarchy built on the 8 observations of Table 1 is shown in Figure 5.

Something more is shown in Figure 5, namely the detail signals (denoted $\pm d$) and overall smooth (denoted s), which are determined in carrying out the wavelet transform, the so-called forward transform.

The inverse transform is then determined from Figure 5 in the following way. Consider the observation vector x_2. Then this vector is reconstructed exactly by reading the tree from the root: $s_7 + d_7 = x_2$. Similarly a path from root to terminal is used to reconstruct any other observation. If x_2 is a vector of dimensionality m, then so also are s_7 and d_7, as well as all other detail signals.

This procedure is the same as the Haar wavelet transform, only applied to the dendrogram and using the input data.

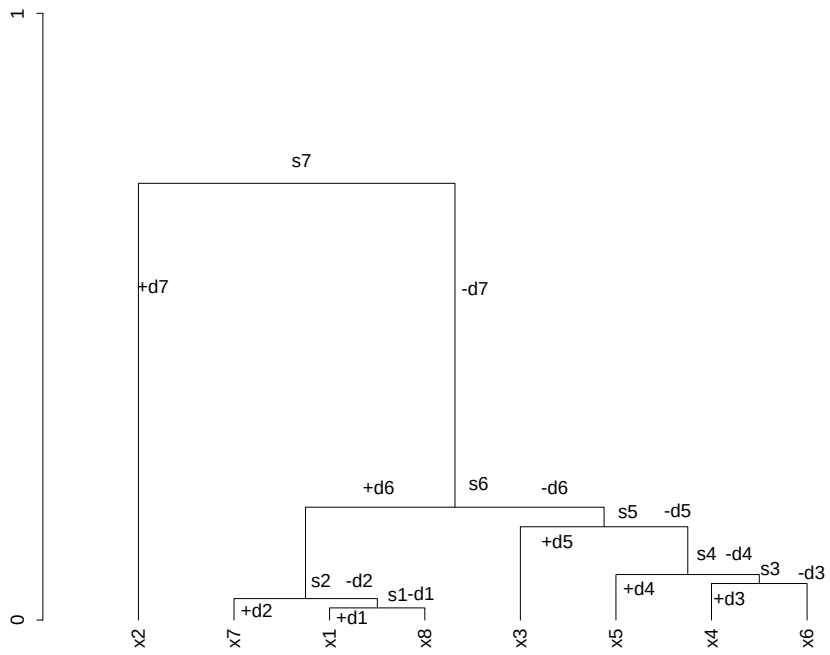


Figure 5: Dendrogram on 8 terminal nodes constructed from first 8 values of Fisher iris data. (Median agglomerative method used in this case.) Detail or wavelet coefficients are denoted by d, and data smooths are denoted by s. The observation vectors are denoted by x and are associated with the terminal nodes. Each *signal smooth*, s, is a vector. The (positive or negative) *detail signals*, d, are also vectors. All these vectors are of the same dimensionality.

	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 as shown in Table 1. Wavelet coefficient levels are denoted d1 through d7, and the continuum or smooth component is denoted s7.

The data required to define this wavelet transform, for the data in Table 1, is shown in Table 2.

The principle of "folding" the hierarchy onto an external signal is as follows. The wavelet transform codifies the hierarchy. Having that, we apply the "codification" of the hierarchy with the new, external signal as input.

Wavelet regression entails setting small and hence unimportant detail coefficients to 0 before applying the inverse wavelet transform.

More discussion can be found in [27].

3.3 Representation of an Object as a Chain of Successively Finer Approximations

From the wavelet transformed hierarchy we can read off that, say, $x_1 = d_2 + d_5 + d_7 + s_7$: cf. Figure 5. Or $x_8 = d_6 - d_7 + s_7$. These relationships use the appropriate vectors shown in Table 2. Such relationships furnish the definitions used by the inverse wavelet transform, i.e. the recreation of the input data from the transformed data.

Thus, the Haar dendrogram wavelet transform gives us an additive decomposition of a given observation (say, x_1) in terms of a degrading approximation, with a variable number of terms in the decomposition. The objects, or observations, are those things which we are analyzing and on which we have (i) induced a hierarchical clustering, and (ii) further processed the hierarchical clustering in such a way that we can derive the Haar decomposition. In this section we will look at how this allows us to consider each object as a limit point. Our interest lies in our object set, characterized by a set of data, as a set of limit or fixed points.

Using notation from domain theory (see, e.g., [5]) we write:

$$s_7 \sqsubseteq s_7 + d_7 \sqsubseteq s_7 + d_7 + d_5 \sqsubseteq s_7 + d_7 + d_5 + d_2 \tag{2}$$

The relation $a \sqsubseteq b$ is read: a is an approximation to b, or b gives more information than a. (Edalat [6] discusses examples.) Just rewriting the very last, or rightmost, term in relation (2) gives:

$$s_7 \sqsubseteq s_7 + d_7 \sqsubseteq s_7 + d_7 + d_5 \sqsubseteq x_1 \tag{3}$$

Every one of our observation vectors (here, e.g., x_1) can be increasingly well approximated by a *chain* of the sort shown in relations (2) or (3), starting with a least element (s_7; more generally, for n observation vectors, s_{n-1}). The observation vector itself (e.g., x_1) is a least upper bound (lub) or supremum (sup), denoted $\sqcup$ in domain theory, of this chain. Since every observation vector has an associated chain, every chain has a lub. The elements of the "rolled down" tree, s_7, $s_7 + d_7$ and $s_7 - d_7$, $s_7 + d_7 + d_5$ and $s_7 + d_7 - d_5$, and so on, are clearly representable as a binary rooted tree, and the elements themselves comprise a partially ordered set (or poset). A *complete partial order* or *cpo* or *domain* is a poset with least element, and such that every chain has a lub.

Cpos generalize complete lattices: see [4] for lattices, domains, and their use in fixpoint applications.

3.4 Approximation Chain using a Hierarchy

An alternative, although closely related, structure with which domains are endowed is that of spherically complete ultrametric spaces. The motivation comes from logic programming, where non-monotonicity may well be relevant (this arises, for example, with the negation operator). Trees can easily represent positive and negative assertions. The general notion of convergence, now, is related to *spherical completeness* ([34, 12]; see also [17], Theorem 4.1). If we have any set of embedded clusters, or any chain, q_k, then the condition that such a chain be non-empty, $\bigcap_k q_k \neq \emptyset$, means that this ultrametric space is non-empty. This gives us both a concept of completeness, and also a fixed point which is associated with the "best approximation" of the chain.

Consider our space of observations, $X = \{x_i | i \in I\}$. The hierarchy, H, or binary rooted tree, defines an ultrametric space. For each observation x_i, by considering the chain from root cluster to the observation, we see that H is a spherically complete ultrametric space.

3.5 Mapping of Spherically Complete Space into Dendrogram Wavelet Transform Space

Consider analysis of the set of observations, $\{x_i \in X \subset \mathbb{R}^m\}$. Through use of any hierarchical clustering (subject to being binary, a sufficient condition for which is that a pairwise agglomerative algorithm was used to construct the hierarchy), followed by the Haar wavelet transform of the dendrogram, we have an approximation chain for each $x_i \in X$. This approximation chain is defined in terms of embedded sets. Let $n = \text{card } X$, the cardinality of the set X. Our Haar dendrogram wavelet transform allows us to associate the set $\{\nu_j | 1 \leq j \leq n-1\} \subset \mathbb{R}^m$ with the chains, as seen in section 3.3.

We have two associated vantage points on the generation of observation $i, \forall i$: the set of embedded sets in the approximation chain starting always with the entire observation set, I, and ending with the singleton observation; or the global smooth in the Haar transform, that we will call ν_{n-1}, running through all details ν_j on the path, such that an additive combination of path members increasingly approximates the vector x_i that corresponds to observation i. Our two associated views are, respectively, a set of sets; or a set of vectors in $\mathbb{R}^m$. We recall that m is the dimensionality of the embedding space of our observations. Our two associated views of the (re)generation of an observation both rest on the hierarchical or tree structuring of our data.

4 Generalized Ultrametric

4.1 Applications of Generalized Ultrametrics

As noted in the previous subsection, the usual ultrametric is an ultrametric distance, i.e. for a set I, $d : I \times I \longrightarrow \mathbb{R}$ (so the ultrametric distance is a real value). The generalized ultrametric is: $d : I \times I \longrightarrow \Gamma$, where Γ is a partially ordered set. In other words, the *generalized* ultrametric distance is a set. With this set one can have a value, so the usual and the generalized ultrametrics can amount to more or less the same in practice (by ignoring the set and concentrating on its associated value). After all, in a dendrogram one does have a set associated with each ultrametric distance value (and this is most conveniently the terminals dominated by a given node; but we could have other designs, like some representative subset or other, of these terminals). Remember that the set, Γ, is defined from the original attributes (which we denote by the set J); whereas the sets of observations read off a dendrogram are subsets of the observation set (which we label with the index set I). So $\Gamma = 2^J$ (and not 2^I).

In the theory of reasoning, a monotonic operator is rigorous application of a succession of conditionals (sometimes called consequence relations). However: "In order to deal with programs of a more general kind (the so-called disjunctive programs) it became necessary to consider multi-valued mappings", supporting non-monotonic reasoning in the way now to be described ([30], pp. 10, 13). The novelty in the work of [30, 31] is that these authors use the generalized ultrametric as a multivalued mapping.

(A more critical view of the usefulness of the generalized ultrametric perspective is presented by [18].)

The generalized ultrametric approach has been motived [35] as follows. "Situations arise ... in computational logic in the presence of negations which force non-monotonicity of the operators involved". To address non-monotonicity of operators, one approach has been to employ metrics in studying some problematic logic programs. These ideas were taken further in examining quasi-metrics, and generalized ultrametrics i.e. ultrametrics which take values in an arbitrary partially ordered set (not just in the non-negative reals). Seda and Hitzler [35] "consider a natural way of endowing Scott domains [see [4]] with generalized ultrametrics. This step provides a technical tool [for finding fixpoints – hence for analysis] of non-monotonic operators arising out of logic programs and deductive databases and hence to finding models for these."

A further, similar, viewpoint is [12]: "Once one introduces negation, which is certainly implied by the term *enhanced syntax* ... then certain of the important operators are not monotonic (and therefore not continuous), and in consequence the Knaster-Tarski theorem [i.e. for fixed points; again see [4]] is no longer applicable to them. Various ways have been proposed to overcome this problem. One such [approach is to use] syntactic conditions on programs ... Another is to consider different operators ... The third main solution is to introduce techniques from topology and analysis to augment arguments based on order ... [latter include:] methods based on metrics ... on quasi-metrics ... and finally ...

Table 3: Example dataset: 5 objects, 3 boolean attributes.

	v_1	v_2	v_3
a	1	0	1
b	0	1	1
c	1	0	1
e	1	0	0
f	0	0	1

on ultrametric spaces."

The convergence to fixed points that are based on a generalized ultrametric system is precisely the study of spherically complete systems and expansive automorphisms discussed in section 3.1. As expansive automorphisms we see here again an example of symmetry at work.

4.2 Link with Formal Concept Analysis

In this subsection, we consider an ultrametric defined on the powerset or join semilattice. Comprehensive background on ordered sets and lattices can be found in [4].

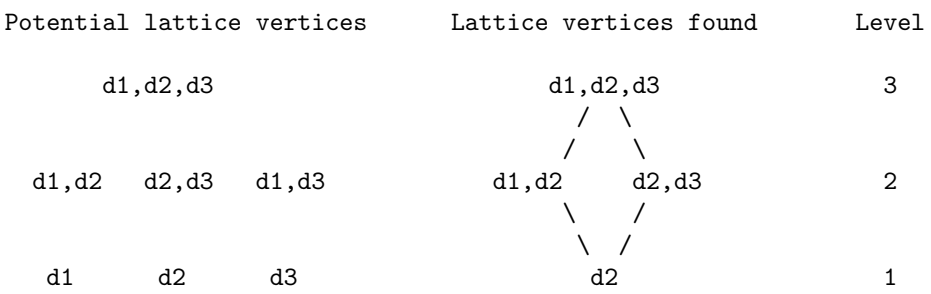
As noted in section 2, typically hierarchical clustering is based on a distance (which can be relaxed often to a dissimilarity, not respecting the triangular inequality, and *mutatis mutandis* to a similarity), defined on all pairs of the object set: $d : I \times I \to \mathbb{R}^+$. I.e., a distance is a positive real value. Usually we require that a distance cannot be 0-valued unless the objects are identical. That is the traditional approach.

A different form of ultrametrization is achieved from a dissimilarity defined on the power set of attributes characterizing the observations (objects, individuals, etc.) X. Here we have: $d : X \times X \longrightarrow 2^J$, where J indexes the attribute (variables, characteristics, properties, etc.) set.

We consider a different notion of distance, that maps pairs of objects onto elements of a join semilattice. The latter can represent all subsets of the attribute set, J. That is to say, it can represent the power set, commonly denoted 2^J, of J.

As an example, consider, say, $n = 5$ objects characterized by 3 boolean (presence/absence) attributes, shown in Table 3.

Define dissimilarity between a pair of objects in Table 3 as a *set* of 3 components, corresponding to the 3 attributes, such that if both components are 0, we have 1; if either component is 1 and the other 0, we have 1; and if both components are 1 we get 0. This is the simple matching coefficient [14]. We could use, e.g., Euclidean distance for each of the values sought; but we prefer to treat 0 values in both components as signaling a 0 contribution. We get then: $d(a, b) = 1, 1, 0$



The set d1,d2,d3 corresponds to: $d(b, e)$ and $d(e, f)$
The subset d1,d2 corresponds to: $d(a, b), d(a, f), d(b, c), d(b, f)$, and $d(c, f)$
The subset d2,d3 corresponds to: $d(a, e)$ and $d(c, e)$
The subset d2 corresponds to: $d(a, c)$

Clusters defined by all pairwise linkage at level ≤ 2:
a, b, c, f
a, e
c, e

Clusters defined by all pairwise linkage at level ≤ 3:
a, b, c, e, f

Figure 6: Lattice and its interpretation, corresponding to the data shown in Table 3 with the simple matching coefficient used. (See text for details.)

$d(a, c) = 0, 1, 0$
$d(a, e) = 0, 1, 1$
$d(a, f) = 1, 1, 0$
$d(b, c) = 1, 1, 0$
$d(b, e) = 1, 1, 1$
$d(b, f) = 1, 1, 0$
$d(c, e) = 0, 1, 1$
$d(c, f) = 1, 1, 0$
$d(e, f) = 1, 1, 1$

If we take the three components in this distance as $d1, d2, d3$, and considering a lattice representation with linkages between all ordered subsets where the subsets are to be found in our results above (e.g., $d(c, f) = 1, 1, 0$ implies that we have a lattice node associated with the subset $d1, d2$), and finally such that the order is defined on subset cardinality, then we see that the representation shown in Figure 6 suffices.

In Formal Concept Analysis [4, 11, 15], it is the lattice itself which is of primary interest. In [14] there is discussion of, and a range of examples on, the close relationship between the traditional hierarchical cluster analysis based on

$d : I \times I \to \mathbb{R}^+$, and hierarchical cluster analysis "based on abstract posets" (a poset is a partially ordered set), based on $d : I \times I \to 2^J$. The latter, leading to clustering based on dissimilarities, was developed initially in [13].

5 Conclusion

Data analysis allows us to go from measured data to a computational path or a set of approximations used to represent the objects of analysis. We have noted that examples of application to face recognition and to documents can be seen in [28].

Computational logic in an analogous way used metric and ultrametric embeddings. Within such topologies, computation is carried out. We have focused in this article on ultrametric embedding, i.e. given as a hierarchy or tree.

It is interesting, and without question exciting, to envisage further cross-linkage between data analysis and computational logic.

References

[1] J.J. Benedetto and R.L. Benedetto. A wavelet theory for local fields and related groups. *The Journal of Geometric Analysis*, 14:423–456, 2004.

[2] R.L. Benedetto. Examples of wavelets for local fields. In *Wavelets, Frames, and Operator Theory, Contemporary Mathematics Vol. 345*, pages 27–47. 2004.

[3] J.-P. Benzécri. *La Taxinomie*. Dunod, Paris, 2nd edition, 1979.

[4] B.A. Davey and H.A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, 2nd edition, 2002.

[5] A. Edalat. Domains for computation in mathematics, physics and exact real arithmetic. *Bulletin of Symbolic Logic*, 3:401–452, 1997.

[6] A. Edalat. Domain theory and continuous data types, lecture notes, 2003. www.doc.ic.ac.uk/∼ae/teaching.html.

[7] R. Foote. An algebraic approach to multiresolution analysis. *Transactions of the American Mathematical Society*, 357:5031–5050, 2005.

[8] R. Foote, G. Mirchandani, D. Rockmore, D. Healy, and T. Olson. A wreath product group approach to signal and image processing: Part I – Multiresolution analysis. *IEEE Transactions on Signal Processing*, 48:102–132, 2000.

[9] R. Foote, G. Mirchandani, D. Rockmore, D. Healy, and T. Olson. A wreath product group approach to signal and image processing: Part II – Convolution, correlations and applications. *IEEE Transactions on Signal Processing*, 48:749–767, 2000.

[10] L. Gajić. On ultrametric space. *Novi Sad Journal of Mathematics*, 31:69–71, 2001.

[11] B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer, 1999. *Formale Begriffsanalyse. Mathematische Grundlagen*, Springer, 1996.

[12] P. Hitzler and A.K. Seda. The fixed-point theorems of Priess-Crampe and Ribenboim in logic programming. *Fields Institute Communications*, 32:219–235, 2002.

[13] M.F. Janowitz. An order theoretic model for cluster analysis. *SIAM Journal on Applied Mathematics*, 34:55–72, 1978.

[14] M.F. Janowitz. Cluster analysis based on abstract posets. Technical report, 2005–2006. http://dimax.rutgers.edu/∼melj.

[15] M.F. Janowitz. *Ordinal and Relational Clustering*. World Scientific, 2010.

[16] S.C. Johnson. Hierarchical clustering schemes. *Psychometrika*, 32:241–254, 1967.

[17] A.Yu. Khrennikov. *Information Dynamics in Cognitive, Psychological, Social and Anomalous Phenomena*. Kluwer, 2004.

[18] M. Krötzsch. Generalized ultrametric spaces in quantitative domain theory. *Theoretical Computer Science*, 368:30–49, 2006.

[19] F.W. Lawvere. Metric spaces, generalized logic, and closed categories. *Rendiconti del seminario matématico e fisico di Milano*, XLIII:135–166, 1973.

[20] F.W. Lawvere. Metric spaces, generalized logic, and closed categories. *Reprints in Theory and Applications of Categories*, 1:1–37, 2002.

[21] I.C. Lerman. *Classification et Analyse Ordinale des Données*. Dunod, Paris, 1981.

[22] F. Murtagh. A survey of recent advances in hierarchical clustering algorithms. *Computer Journal*, 26:354–359, 1983.

[23] F. Murtagh. Complexities of hierarchic clustering algorithms: state of the art. *Computational Statistics Quarterly*, 1:101–113, 1984.

[24] F. Murtagh. *Multidimensional Clustering Algorithms*. Physica-Verlag, Heidelberg and Vienna, 1985.

[25] F. Murtagh. Comments on: Parallel algorithms for hierarchical clustering and cluster validity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:1056–1057, 1992.

[26] F. Murtagh. On ultrametricity, data coding, and computation. *Journal of Classification*, 21:167–184, 2004.

[27] F. Murtagh. The Haar wavelet transform of a dendrogram. *Journal of Classification*, 24:3–32, 2007.

[28] F. Murtagh. On ultrametric algorithmic information. *Computer Journal*, 2007. In press. Advance Access online, 9 Oct. 2007.

[29] F. Murtagh. Symmetry in data mining and analysis: a unifying view based on hierarchy. *Proceedings of Steklov Institute of Mathematics*, 265:177–198, 2009.

[30] S. Priess-Crampe and P. Ribenboim. Logic programming and ultrametric spaces. *Rendiconti de Matematica, Serie VII*, 19:155–176, 1999.

[31] S. Priess-Crampe and P. Ribenboim. Ultrametric spaces and logic programming. *Journal of Logic Programming*, 42:59–70, 2000.

[32] H. Reiter and J.D. Stegeman. *Classical Harmonic Analysis and Locally Compact Groups*. Oxford University Press, Oxford, 2nd edition, 2000.

[33] A.C.M. Van Rooij. *Non-Archimedean Functional Analysis*. Dekker, 1978.

[34] W.H. Schikhof. *Ultrametric Calculus*. Cambridge University Press, Cambridge, 1984. (Chapters 18, 19, 20, 21).

[35] A.K. Seda and P. Hitzler. Generalized ultrametrics, domains and an application to computational logic. *Irish Mathematical Society Bulletin*, 41:31–43, 1998.

[36] A.K. Seda and P. Hitzler. Generalized distance functions in the theory of computation. *Computer Journal*, 2008. In press, Advance Access online 17 January 2008.

[37] C.J. van Rijsbergen. *The Geometry of Information Retrieval*. Cambridge University Press, 2004.