Symbolic Dynamics in Text: Application to Automated Construction of Concept Hierarchies

Fionn Murtagh
Department of Computer Science
Royal Holloway, University of London
Egham, Surrey TW20 0EX, England
fmurtagh@acm.org

August 25, 2006

Abstract

Following a symbolic encoding of selected terms used in text, we determine symmetries that are furnished by local hierarchical structure. We develop this study so that hierarchical fragments can be used in a concept hierarchy, or ontology. By "letting the data speak" in this way, we avoid the arbitrariness of approximately fitting a model to the data.

1 Symmetry Group and Alternating Permutation Ordinal Encodings in Symbolic Dynamics

In symbolic dynamics, we seek to extract symmetries in the data based on topology alone, before considering metric properties. For example, instead of listing a sequence of iterates, $\{x_i\}$, we may symbolically encode the sequence in terms of up or down, or north, south, east and west moves. This provides a sequence of symbols, and their patterns in a phase space, where the interest of the data analyst lies in a partition of the phase space. Patterns or templates are sought in this topology. Sequence analysis is tantamount to a sort of topological time series analysis.

Thus, in symbolic dynamics, the data values in a stream or sequence are replaced by symbols to facilitate pattern-finding, in the first instance, through topology of the symbol sequence. This can be very helpful for analysis of a range of dyamical systems, including chaotic, stochastic, and deterministic-regular time series. Through measure-theoretic or Kolmogorov-Sinai entropy of the dynmical system, it can be shown that the maximum entropy conditional on past values is consistent with the requirement that the symbol sequence retains as much of the original data information as possible. Alternative approaches to

quantifying complexity of the data, expressing the dynamical system, is through Lyapanov exponents and fractal dimensions, and there are close relationships between all of these approaches [16].

Later in this work, we will use a "change versus no change" encoding, firstly using univariate time series, and then using a multivariate time series based on the sequence of terms used in a document.

From the viewpoint of practical and real-world data analysis, however, many problems and open issues remain. Firstly [1], noise in the data stream means that reproducibility of results can break down. Secondly, the symbol sequence, and derived partitions that are the basis for the study of the symbolic dynamic topology, are not easy to determine. Hence [1] enunciate a pragmatic principle, whereby the symbol sequence should come as naturally as possible from the data, with as little as possible by way of further model assumptions. Their approach is to define the symbol sequence through (i) comparison of neighboring data values, and (ii) up-down or down-up movements in the data stream.

Taking into account all up-down and down-up movements in a signal allows a permutation representation.

Examples of such symbol sequences from [1] follow. They consider the data stream $(x_1, x_2, \ldots, x_7) = (4, 7, 9, 10, 6, 11, 3)$. Take the order as 3, i.e. consider the up-down and down-up properties of successive triplets. $(4, 7, 9) \longrightarrow 012; (7, 9, 10) \longrightarrow 012; (9, 10, 6) \longrightarrow 201; (6, 11, 3) \longrightarrow 201; (10, 6, 11) \longrightarrow 102$. (In the last, for instance, we have $x_{t+1} < x_t < x_{t+2}$, yielding the symbolic sequence 102.) In addition to the order, here 3, we may also consider the delay, here 1. In general, for delay τ, the neighborhood consists of data values indexed by $t, t-\tau, t-2\tau, t-3\tau, \ldots, t-d\tau$ where d is the order. Thus, in the example used here, we have the symbolic representation 012012201201102. The symbol sequence (or "itinerary") defines a partition – a separation of phase space into disjoint regions (here, with three equivalence classes, 012, 201, and 102), which facilitates finding an "organizing template" or set of topological relationships [26]. The problem is described in [11] as one of studying the qualitative behavior of the dynamical system, through use of a "very coarse-grained" description, that divides the state space (or phase space) into a small number of regions, and codes each by a different symbol.

Different encodings are feasible and [13, 14] use the following. Again consider the data stream $(x_1, x_2, \ldots, x_7) = (4, 7, 9, 10, 6, 11, 3)$. Now given a delay, $\tau = 1$, we can represent the above by $(x_{6\tau}, x_{5\tau}, x_{4\tau}, x_{3\tau}, x_{2\tau}, x_\tau, x_0)$. Now look at rank order and note that: $x_\tau > x_{3\tau} > x_{4\tau} > x_{5\tau} > x_{2\tau} > x_{6\tau} > x_0$. We read off the final permutation representation as (1345260). There are many ways of defining such a permutation, none of them best, as [13] acknowledge. We see too that our m-valued input stream is a point in $\mathbb{R}^m$, and our output is a permutation $\pi \in S_m$, i.e. a member of the permutation group.

[13] explore invariance properties of the permutations expressing the ordinal, symbolic coding. Resolution scale is introduced through the delay, τ. (An alternative approach to incorporating resolution scale is used in [7]: consecutive, sliding-window based, binned or averaged versions of the time series are used. This is not entirely satisfactory: it is not robust and is very dependent on data

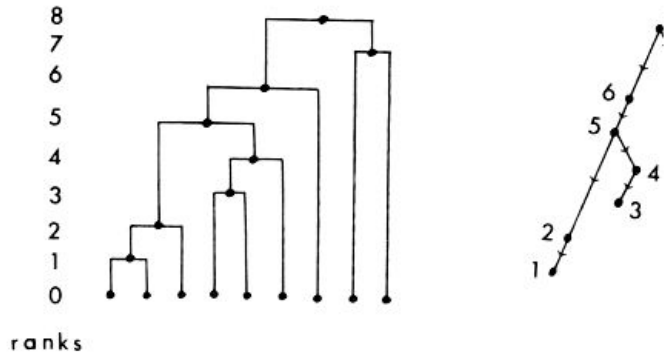

Figure 1: Left: dendrogram with lower ranked subtree always to the left. Right: oriented binary tree associated with the non-terminal nodes.

properties such as dynamic range.) Application is to EEG (univariate) signals (with some discussion of MRI data) [15]. Statistical properties of the ordinal transformed data are studied in [2], in particular through the S_3 symmetry group. We have noted the symbolic dynamics motivation for this work; in [3] and other work, motivation is provided in terms of rank order time series analysis, in turn motivated by the need for robustness in time series data analysis.

We note in passing that there is an isomorphism between a class of hierarchic structures, termed unlabeled, ranked, binary, rooted trees, and the class of permutations used in symbolic dynamics. Each non-terminal node in the tree shown in Fig. 1 has one or two child nodes. This is a dendrogram, representing a set of $n-1$ agglomerations based on n initial data vectors. A packed representation [24] or permutation representation of a dendrogram is derived as follows. Put lower ranked subtree always to the left; and read off oriented binary tree on non-terminal nodes (see Fig. 1). Then for any terminal node indexed by i, with the exception of the rightmost which will always be n, define $p(i)$ as the rank at which the terminal node is first united with some terminal node to its right. For the dendrogram shown, the packed representation is: (125346879). This is also an inorder traversal of the oriented binary tree. The packed representation is a uniquely defined permutation of $1 \ldots n$. Dendrograms (on n terminals) of the sort shown in Fig. 1, referred to as non-labeled, ranked (NL-R) in [17], are isomorphic to either down-up permutations, or up-down permutations (both on $n-1$ elements).

1.1 Motivation for an Alternative Ordinal Symbolic Dynamics Encoding

In some respects we follow the work of Keller, Bandt, and their colleagues in using an ordinal coding to provide for an encoding of the data sequence. However

in the following areas we need to adopt a different approach.

- We need to handle multivariate time series.
- We need to bypass the two analyses that the ordinal symbolic encoding necessarily leads to, viz. either up-down or down-up.
- Biological verisimilitude is not strong with the ordinal encoding as discussed so far.

We look at each of these in turn.

To handle multivariate time series, [11, 12] find the best composite time series, using projections on the first factor furnished by correspondence analysis. Correspondence analysis uses a weighted Euclidean distance between profiles (or, using the input data, the χ^2 distance) and for time-varying signals such as EEG signals, it is a superior choice compared to, say, principal components analysis.

In [4], the need for multivariate analysis is established. Among tentative steps towards this are window-based averages of distances.

It is immediate in any inequality, $x_t > x_{t-1}$, that reversing the inequality (e.g. through considering an axial symmetry in the time axis) can lead to a new and different outcome. When we have multivariate data streams, enforcing symmetry is very restrictive. We bypass this difficulty very simply by instead using a change/no change symbolic representation. Financial verisimilitude is lost in doing this (if up = gain, down = loss); but biological verisimilitude, and that of other areas, is aided greatly.

Based on their EEG analysis, [13] ask: "Does there exist a basic (individual) repertoire of 'ordinal' states of brain activity?". As opposed to this, we target the hierarchy or branching fragment as the pattern that is sought, which suits the dendritic structures of the brain. While rank order alone is a useful property of data, we seek to embed our data (globally or locally) in an ultrametric topology, which also offers scope for p-adic algebraic processing. We move from real data, we take account of ordinal properties, and we end up with a topological and/or algebraic framework. This implies a data analysis perspective which is highly integrated and comprehensive. Furthermore, as an analysis pipeline, it is very powerful in bridging observed data with theoretically-supported interpretation.

2 The Topological View: Ultrametric Embedding

1. We seek uncontestable local hierarchical structure in the data. The traditional alternative is to impose hierarchical structure on the data (e.g. through hiearchical clustering, or otherwise inducing a classification tree).

2. We seek to avoid having any notion of hierarchical direction. In practice this would imply that hierarchical "up" (e.g. agglomerative or bottom-up) and hierarchical "down" (e.g. divisive or top-down) should each be considered independently.

3. We may wish to accommodate (i.e., include in our analysis) outliers and random exceptional values in the data. More particularly: we want to handle power law distributions, characterized by independent but not identically distributed values. An example is Zipf's law for text.

4. Therefore, for text we will use the property of linearity of text: words are linearly ordered from start to finish.

The approach to finding local hierarchical structure is described for time series data in [20]. We use the same approach here. The algorithm is as follows. The data used is the sequence of frequencies of occurrence of the terms of interest – nouns, noun-substantives – in their text-based order. These terms are found using TreeTagger [23, 25].

3 Inherent Hierarchical Structure

We now turn attention to the automated construction of concept hierarchies, based on local hierarchic structure in one or more documents. "Ontologies are often equated with taxonomic hierarchies of classes ... but ontologies need not be limited to such a form" [10]. In this work, we seek fragments of hierarchical relationships between terms in one or more documents, and we seek to position these hierarchy fragments in one global hierarchy.

The first problems to be addressed, therefore, are whether or not the document has any hierarchical structure to begin with; and what the hierarchical structure is derived or induced from. Maybe we have a fully tagged document (based, e.g., on part-of-speech tagging, [23, 25]). However in this work, we start with free text, because it is the most generally available and applicable framework. Our point of departure is the set of words or stemmed terms comprising the document. Additional information provided by part-of-speech can be of use to us, as we will show later.

Next we consider the issue of whether or not a document has sufficient inherent hierarchical structure to warrant further investigation. We could approach this problem by fitting a hierarchy, and there are many algorithms for doing so (such as any hierarchical clustering algorithm; de Soete [8] describes a least squares optimal fitting approach). However departure from inherent hierarchical structure is not easily pinpointed. After all, we have an output induced structure, and we are told, let's say, that the fit is 80% (defined as $\sum(\delta - d)^2 / \sum d^2$ where d is input dissimilarity, δ is tree or ultrametric distance read off the output, and the sums are over all pairs), which is not very revealing nor useful.

An altnernative "bottom-up" approach is pursued here, which allows easy assessment of inherent structure, and also pinpointing where this occurs or does not occur.

3.1 Local Ultrametricity and Quantifying Extent of Ultrametricity

A formal definition of hierarchical structure is provided by ultrametric topology (in turn, related closely to p-adic number theory). The triangular inequality holds for a metric space: $d(x, z) \leq d(x, y) + d(y, z)$ for any triplet of points x, y, z. In addition the properties of symmetry and positive definiteness are respected. The "strong triangular inequality" or ultrametric inequality is: $d(x, z) \leq \max\{d(x, y), d(y, z)\}$ for any triplet x, y, z. An ultrametric space implies respect for a range of stringent properties. For example, the triangle formed by any triplet is necessarily isosceles, with the two large sides equal; or is equilateral. In an ultrametric space (i.e., a space endowed with an ultrametric, or an ultrametric topology), one "lives", so to speak, in a tree. All "moves" between one location and another are as if one descended the tree to a common tree node, and then reclimbed to the target point. Topologically, an ultrametric goes a lot further: all points in a circle or sphere are centers, for example; or the radius of a sphere is identical to its diameter.

The triangle property respected by any triplet of points in an ultrametric space affords a useful way to quantify extent of hierarchical structure. We will describe our "extent of hierarchical structure", on a scale of 0 (no respect for ultrametricity) to 1 (everywhere, respect for the ultrametric or tree distance) algorithmically. We examine triplets of points (exhaustively if possible, or otherwise through sampling), and determine the three angles formed by the associated triangle. We select the smallest angle formed by the triplet points. Then we check if the other two remaining angles are approximately equal. If they are equal then our triangle is isosceles with small base, or equilateral (when all triangles are equal). The approximation to equality is given by 2 degrees (0.0349 radians). Our motivation for the approximate ("fuzzy") equality is that it makes our approach robust and independent of measurement precision.

This approach works very well in practice [20, 21].

3.2 Structure of a Text

In seeking to use free text, we will also take into consideration the strongest "given" in regard to any classical text: its linearity (or total) order. A text is read from start to finish, and consequently is linearly ordered.

A text endowed with this linear order is analogous to a time series. (This opens up the possibility to generalize the work described here to (i) speech signals, or (ii) music. We will pursue these generalizations in the future.)

4 Hierarchical Structure in a Linear or Total Ordered Set

4.1 Extent of Ultrametricity, or Inherent Hierarchical Structure, in a Time Series

In Murtagh [20] we use the following coding to show that chaotic time series are less ultrametric than, say, financial (futures, FTSE – Financial Times Stock Exchange index, stock price index), biomedical (EEG for normal and epileptic subjects, eyegaze trace), telecoms (web traffic) or meteorological (Mississippi water level) time series; random generated (uniformly distributed) time series data are remarkably similar in their ultrametric properties; and ultrametricity can be used to distinguish various types of biomedical (EEG) signals.

A time series can be easily embedded in a space of dimensionality m, by taking successive intervals of length m, or a delay embedding of order m. Thus we define points

$$\mathbf{x}_r = (x_{r-m+1}, x_{r-m+2}, \ldots, x_{r-1}, x_r)^t \in \mathbb{R}^m$$

where t denotes vector transpose.

Given any $\mathbf{x}_r = (x_{r-m+1}, x_{r-m+2}, \ldots, x_{r-1}, x_r)^t \in \mathbb{R}^m$, let us consider the set of s such contiguous intervals determined from the time series of overall size n. For convenience we will take $s = \lfloor n/m \rfloor$ where $\lfloor . \rfloor$ is integer truncation. The contiguous intervals could be overlapping but for exhaustive or near-exhaustive coverage it is acceptable that they be non-overlapping. In our work, the intervals were non-overlapping. The quantification of the ultrametricity of the overall time series is provided by the aggregate over s time intervals of the ultrametricity of each $\mathbf{x}_r$, $1 \leq r \leq s$.

We seek to directly quantify the extent of ultrametricity in time series data. Increase in ambient spatial dimensionality leads to greater ultrametricity. However it is not satisfactory from a practical point of view to simply increase the embedding dimensionality m insofar as short memory relationships are of greater practical relevance (especially for prediction). The greatest possible value of $m > 1$ is the total length of the time series, n. Instead we will look for an ultrametricity measurement approach for given and limited sized dimensionalities m. Our experimental results for real and for random data sets are for "window" lengths $m = 5, 10, \ldots, 105, 110$.

We seek local ultrametricity, i.e. hierarchical structure, by studying the following: Euclidean distance squared, $d_{jj'} = (x_{rj} - x_{rj'})^2$ for all $1 \leq j, j' \leq m$ in each time window, $\mathbf{x}_r$. It will be noted below in this section how this assumption of Euclidean distance squared has worked well but is not in itself important: in principle any dissimilarity can be used.

We enforce sparseness on our given distance values, $\{d_{jj'}\}$. We do this by linearly approximating each value $d_{jj'}$, in the range $\max_{jj'} d_{jj'} - \min_{jj'} d_{jj'}$, by an integer in $1, 2, \ldots p$. Note that the range is chosen with reference to the currently considered time series window, $1 \leq j, j' \leq m$. Note too that the value

of p must be specified. In our work we set $p = 2$. Thus far, the recoded value, $d'_{jj'}$ is not necessarily a distance. With the extra requirement that $d'_{jj'} \longrightarrow 0$ whenever $j = j'$ it can be shown that $d'_{jj'}$ is a metric [20]:

Theorem: *The recoded pairwise measure, d', defined as described above from any dissimilarity, is a distance, satisfying the properties of: symmetry, positive definiteness, and triangular inequality.*

To summarize, in our coding, a small pairwise transition is mapped onto a value of 1; and a large pairwise transition is mapped onto a value of 2. A pairwise transition is defined not just for data values that are successive in time but for any pair of data values in the window considered.

This coding can be considered as (i) taking a local region, defined by the sliding window, and (ii) coding pairwise "change" = 2, versus "no change" = 1, relationships. Then, based on these new distances, we use the ultrametric triangle properties to assess conformity to ultrametricity. The average overall ultrametricity in the time series, quantified in this way, allows us to fingerprint our time series.

A wide range of window sizes (i.e., lengths), m, was investigated. Window size is not important: in relative terms the results found remain the same. Taking part of a time series and comparing the results to the full time series gave similar outcomes, thus indicating that the fingerprinting was an integral property of the data.

Our "change/no change" metric is crucial here, and not the input dissimilarity which is mapped onto it. Generalization to multivariate time series is straightforward, which is what we will now do.

4.2 Extent of Ultrametricity in Multivariate Time Series

The data recoding described above was based on Euclidean distance squared, $d_{jj'} = (x_{rj} - x_{rj'})^2$ for all $1 \leq j, j' \leq m$, and where r indicates a time window. Generalization to multivaraite time series is immediate: we consider vector-valued x_{rj} and $x_{rj'}$.

Our data recoding was carried out by linearly approximating each value $d_{jj'}$, in the range $\max_{jj'} d_{jj'} - \min_{jj'} d_{jj'}$, with a value in $\{1, 2\}$ (representing respectively "no change" and "change"), and this mapping was carried out locally, in each sliding window. For possibly non-stationary time series, this makes a lot of sense. However for a given document, we will make the assumption that it is stationary, in order to achieve consistency in recoded values.

Let the threshold used in the linear mapping onto $\{1, 2\}$ be given by $1/|\{j, j'\}| \ \sum_{j,j'} d^2_{j,j'}$, i.e., the average distance squared over all pairs. A squared distance less than this threshold is mapped onto a value of 1; and a squared distance greater than or equal to this threshold is mapped onto a value of 2. We assume distinct j, j'. The following consistency result ensues:

Theorem: *For any pair j, j', the "no change/change" distance, d', defined as above with a global threshold will be identical.*

5 Quantifying Hierarchical Structure in a Linear Ordered Set: Implementation

5.1 Application-Specific Choice of Algorithm Parameters

Just as the stationarity property motivated the local threshold for work on meteorological or biomedical time series, and was replaced by a global threshold for analysis of a given text, so also the window length was specified by the aplication.

For numerical temporal signals, the issue of noise is posed. Taking a window of some length allows local properties to be, to some degree, resistant to the noise properties. For a document, we assume that noise (i.e., a semantically irrelevant or a semantically random word) is not likely. (It could be, of course, through a typographic error; or a mistake on the part of the document's author.) The reasonable decision not to consider semantic noise leads to a window length which is strictly 3, i.e., the minimum necessary window length.

Hence we consider successive triplets in the linearly ordered text.

5.2 Summary of the Local Ultrametricity Finding Algorithm

1. Take each triplet of terms in turn.
2. Define the squared Euclidean distance between each successive pair of terms. (We return below to a discussion of the vector used for each term.)
3. Use the pairwise average of these squared distances as a threshold.
4. If the pair of terms is of squared distance less than the threshold, then define their relationshop as "no change".
5. If the pair of terms is of squared distance greater than or equal to the threshold, then define their relationshop as "change (either up or down)".
6. With "no change" coded as 1, and "change" coded as 2, and self-distances coded as 0, Murtagh [20] shows that the resulting mapping of the Cartesian product of terms $\times$ terms onto the set $d' \in \{0, 1, 2\}$ defines a metric. For terms i, j, k, we therefore have $d'_{ij} \leq d'_{ik} + d'_{kj}$.
7. For the given triplet we check if this metric is an ultrametric: For terms i, j, k, we therefore seek whether $d'_{ij} \leq \max\{d'_{ik}, d'_{kj}\}$.
8. If the triplet i, j, k respects the ultrametric relation, then there are two possible cases. Firstly, the triangle formed by these terms is equilateral, which is implied whenever $d'_{ij} = d'_{ik} = d'_{kj}$. Secondly, the triangle is isosceles with small base, which is implied by two d's being equal, and greater in value to the third.

9. No other triangle configurations are consistent with the ultrametric relationship.

10. Over all triplets considered, the ultrametricity index of the document is the proportion of ultrametricity-respectiving triplets.

5.3 Euclidean Representation of Text

Our discussion up to now has taken input pairwise dissimilarity as squared Euclidean distance. It could be any dissimilarity: the thresholding involved in recoding into the new "no change/change" metric is not related to any metric properties in the input. However (squared) Euclidean distance offers us a well-behaved framework. The ordinary Euclidean distance uses constant and identical weights, which is convenient also. In this section, we discuss how we can easily map presence/absence, or frequencies of occurrence (including zero occurrence) data into a Euclidean space. Thereafter, once this is done, we can work on the Euclidean coordinates.

A commonly used methodology for studying a set of texts, or a set of parts of a text (which is what we will describe below), is to characterize each text with numbers of terms appearing in the text, for a set of terms. The χ^2 distance is an appropriate weighted Euclidean distance for use with such data [5, 19] Consider texts i and i' crossed by words j. Let k_{ij} be the number of occurrences of word j in text i. Then, omitting a constant, the χ^2 distance between texts i and i' is given by $\sum_j 1/k_j(k_{ij}/k_i - k_{i'j}/k_{i'})^2$. The weighting term is $1/k_j$. The weighted Euclidean distance is between the *profile* of text i, viz. k_{ij}/k_i for all j, and the analogous *profile* of text i'. (Our discussion is to within a constant because we actually work on *frequencies* defined from the numbers of occurences.)

Correspondence analysis allows us to project the space of documents (we could equally well explore the terms in the *same* projected space) into a Euclidean space. It maps the all-pairs χ^2 distance into the corresponding Euclidean distance.

The algorithm described in section 5.2 can now be completed: for a term, we use the (full rank) projections on factors resulting from correspondence analysis. As noted, this factor space is endowed with the (constant and identical weighted) Euclidean distance.

6 Application

We proceed now to particular engineering aspects of this work. We require a frequency of occurrence matrix which crosses the terms of interest with parts of a free text document. For the latter we could well take documentary segments like paragraphs.

O'Neill [22] is a 660-word discussion of ubiquitous computing from the perspective of human computing interaction. With this short document we used individual lines (as proxies for the sequence of sentences) as the component parts of the document. There were 65 lines.

Based on a list of nouns and substantives furnished by the part-of-speech tagger [23, 25], we focused on the following 30 terms:

`support` = { "agents", "algorithms", "aspects", "attempts", "behaviours", "concepts", "criteria", "disciplines", "engineers", "factors", "goals", "interactions", "kinds", "meanings", "methods", "models", "notions", "others", "parts", "people", "perceptions", "perspectives", "principles", "systems", "techniques", "terms", "theories", "tools", "trusts", "users" }.

This set of 30 terms was used to characterize through presence/absence the 65 successive lines of text, leaading to correspondence analysis of the 65×30 presence/absence matrix. This yielded then the definition of the 30 terms in a factor space. In principle the rank of this space (taking account of the trivial first factor in correspondence analysis, relating to the centering of the cloud of points) is $\min(\ 65-1, 30-1)$. However through all zero-valued rows and/or columns, the actual rank was 25. Therefore the full rank projection of the terms into the factor space gave rise to 25-dimensional vectors for each term, and these vectors are endowed with the Euclidean metric.

Define this set of 30 terms as the support of the document. Based on their occurrences in the document, we generated the following *reduced* version of the document, defined on this support, which consists of the following ordered set of 52 terms:

`Reduced document` = "goals" "techniques" "goals" "disciplines" "meanings" "terms" "others" "systems" "attempts" "parts" "trusts" "trusts" "people" "concepts" "agents" "notions" "systems" "people" "kinds" "behaviours" "people" "factors" "behaviours" "perspectives" "goals" "perspectives" "principles" "aspects" "engineers" "tools" "goals" "perspectives" "methods" "techniques" "criteria" "criteria" "perspectives" "methods" "techniques" "principles" "concepts" "models" "theories" "goals" "tools" "techniques" "systems" "interactions" "interactions" "users" "perceptions" "algorithms"

This reduced document is now analyzed using the algorithm described earlier. Each term in the sequence of 52 terms is represented by its 25-dimensional factor space vector.

For successive triples, if the triple is to be compatible with the ultrametric inequality, we require the recoded distances to be one of the following patterns: 1,1,1 or 2,2,2 for an equilateral triangle; and 1,2,2 in any order for an isosceles triangle with small base.

The only other pattern is 1,1,2 (in any order) which is not compatible with the ultrametric inequality. (It is seen to represent the case of an isosceles triangle with large base.)

Out of 43 unique triplets, with self-distances removed, we found 31 to respect the ultrametric inequality, i.e. 72%. The ultrametricity of this document, based on the support used, was thus 0.72.

For a concept hierarchy we need an overall fit to the data. Using the Euclidean space perspective on the data, furnished by correspondence analysis, we can easily define a terms $\times$ terms distance matrix; and then hierarchically cluster that. Consistent with our analysis we recode all these distances, using the mapping onto $\{1, 2\}$ for unique pairs of terms.

Note that this is tantamount to having a window, r (in the notation used in section 4.1), encompassing all of the reduced document. It is also interesting to check the ultrametricity coefficient here. This means therefore the ultrametricity coefficient in the window length n case, versus the ultrametricity coefficient in the window length 3 case. The latter was seen to be (from exhaustive calculation) above, 0.72. For the window length n case, we sampled 2000 triplets, and found the ultrametricity coefficient to be 0.56. Since the linear order is of greater ultrametric (hence, hierarchical) structure, an evident question arises as to whether it should be used as the basis for a retrieved overall or global hierarchy. We do not do this, however, because the greater hierarchical structure comes as the cost of being overly fragmentary.

Now approximating a global ultrametric from below, achieved by the single linkage agglomerative hierarchical clustering method (and this best fit from below is optimal), and an approximation from above, achieved by the complete linkage agglomerative hierarchical clustering method (and this best fit from above is non-unique and hence is one of a number of best fits from above), will be identical if the data is fully ultrametric-embeddable. If we had an ultrametricity coefficient equal to 1 – we found it to be 0.72 for this data – then it would not matter what agglomerative hierarchical clustering algorithm (among the usual Lance-Williams methods) that we select.

In fact, we found, with an ultrametricity coefficient equal to 0.72, that the single and complete linkage methods gave an identical result. This result is shown in Figure 2.

7 Conclusion

References

[1] C. Bandt and B. Pompe, "Permutation entropy: a natural complexity measure for time series", Physical Review Letters, 88, 174102(4), 2002.

[2] C. Bandt and F. Shiha, "Order patterns in time series", preprint 3/2005, Institute of Mathematics, Greifswald, http://www.math-inf.uni-greifswald.de/∼bandt/pub.html

[3] C. Bandt, "Ordinal time series analysis", Ecological Modelling, 182, 229–238, 2005.

[4] C. Bandt and A. Groth, "Ordinal time series analysis", Poster Freiburg, 2005. www.math-inf.uni-greifswald.de/∼groth

[5] J.P. Benzécri, L'Analyse des Données, Tome I Taxinomie, Tome II Correspondances, 2nd ed., Dunod, Paris, 1979.

[6] L. Comtet, Advanced Combinatorics, Reidel, Dordrecht, 1974.

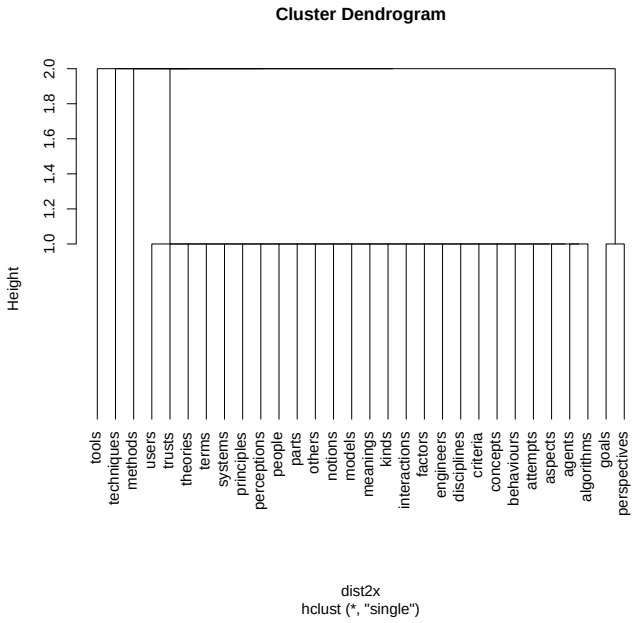

Figure 2: Single (or identially, complete) linkage hierarchy of 30 terms, comprising the support of the document, based on (i) "no change/change" metric recoded (ii) 25-dimensional Euclidean representation.

[7] M. Costa, A.L. Goldberger and C.-K. Peng, "Multiscale entropy analysis of biological signals", Physical Review E, 71, 021906(18), 2005.

[8] G. de Soete, A least squares algorithm for fitting an ultrametric tree to a dissimilarity matrix, Pattern Recognition Letters, 2, 133–137, 1986.

[9] R. Donaghey, "Alternating permutations and binary increasing trees", J. Combin. Theory (A) 18, 141–148, 1975.

[10] T. Gruber, "What is an ontology?", http://www-ksl.stanford.edu/kst/what-is-an-ontology.html, Sept. 2001.

[11] K. Keller, H. Lauffer, Symbolic analysis of high-dimensional time series, Int. J. Bifurcation Chaos, 13, 2657–2668, 2003.

[12] K. Keller and K. Wittfeld, "Distances of time series components by means of symbolic dynamics", Int. J. Bifurcation Chaos, 693–704, 2004.

[13] K. Keller and M. Sinn, "Ordinal symbolic dynamics", Tech. Rep. A-05-14, www.math.mu-luebeck.de/publikationen/pub2005.shtml

[14] K. Keller and M. Sinn, "Ordinal analysis of time series", Physica A 356, 114-120, 2005.

[15] K. Keller, H. Lauffer and M. Sinn, "Ordinal analysis of EEG time series", to appear in Chaos and Complexity Letters, 2005

[16] V. Latora and M. Baranger, "Kolmogorov-Sinai entropy rate versus physical entropy", Physical Review Letters, 82, 520(4), 1999.

[17] F. Murtagh, "Counting dendrograms: a survey", Discrete Applied Math. 7, 191–199, 1984.

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

[19] F. Murtagh, Correspondence Analysis and Data Coding with R and Java, Chapman and Hall/CRC Press, 2005.

[20] F. Murtagh, "Identifying the ultrametricity of time series", European Physical Journal B, 43, 573–579, 2005.

[21] F. Murtagh, "Ultrametricity in data: identifying and exploiting local and global hierarchical structure", Pattern Recognition Letters, submitted, 2006. arXiv:math.ST/0605555v1 19 May 2006.

[22] E. O'Neill, "Understanding ubiquitous computing: a view from HCI", in Discussion following R. Milner, "Ubiquitous computing: how will we understand it?", Computer Journal, in press, 2006.

[23] H. Schmid, "Probabilistic part-of-speech tagging using decision trees", Proc. Intl. Conf. New Methods in Language Processing, 1994 (see TreeTagger site, [25]).

[24] R. Sibson, "SLINK: an optimally efficient algorithm for the single-link cluster method", Computer Journal, 16, 30–34, 1980

[25] TreeTagger, www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/ DecisionTreeTagger.html

[26] W. Weckesser, "Symbolic dynamics in mathematics, physics, and engineering", based on a talk by N. Tuffilaro, http://www.ima.umn.edu/∼weck/nbt/nbt.ps