Hierarchical Clustering of Massive, High Dimensional Data Sets by Exploiting Ultrametric Embedding

Fionn Murtagh (1), Geoff Downs (2), and Pedro Contreras (3)

(1) Department of Computer Science, Royal Holloway, University of London
Egham TW20 0EX, UK fmurtagh@acm.org

(2) Digital Chemistry Ltd., The Iron Shed, Harewood House Estate
Harewood, Leeds LS17 9LF, UK geoff.downs@digitalchemistry.co.uk

(3) Department of Computer Science, Royal Holloway, University of London
Egham TW20 0EX, UK pedro@cs.rhul.ac.uk

June 10, 2007

Running title: Massive, High Dimensional Data Clustering

Abstract

Coding of data, usually upstream of data analysis, has crucial implications for the data analysis results. By modifying the data coding – through use of less than full precision in data values – we can aid appreciably the effectiveness and efficiency of the hierarchical clustering. In our first application, this is used to lessen the quantity of data to be hierarchically clustered. The approach is a hybrid one, based on hashing and on the Ward minimum variance agglomerative criterion. In our second application, we derive a hierarchical clustering from relationships between sets of observations, rather than the traditional use of relationships between the observations themselves. This second application uses embedding in a Baire space, or longest common prefix ultrametric space. We compare this second approach, which is of $O(n \log n)$ complexity, to k-means.

Key words. Hierararchical clustering, ultrametric, tree distance, partitioning, hashing

AMS subject classifiers. 98.52.Cf, 89.75.Hc, 89.75.Fb

1 Introduction

In section 1 we describe the problem and data properties.

In section 2, we show how what amounts to data truncation can be exploited for clustering data. We relate data precision to the topology of the space within which the data is embedded.

In section 3, we put this perspective to use, in a way that we characterize as "data condensation". In fact, for the chemical data used here, we show how the clustering is exact.

In section 4, and in particular from subsection 4.3 onwards, another approach is studied, which we call the Baire, or longest common prefix, ultrametric embedding.

1.1 Chemical Clustering and Matching

Clustering is a requirement for dataset matching, and to support fast proximity searching. For the processing of chemical structure databases, see [3, 6, 7, 30, 9, 14]. Both locally (e.g., for neighborhood search) and globally (for data summarization), hierarchical clustering is beneficial.

In the 1990s, the Ward minimum variance hierarchical clustering method became the method of choice in the chemoinformatics community due to its hierarchical nature and the quality of the clusters produced. Unfortunately the method reached its limits once the pharmaceutical companies tried processing datasets of more than 500,000 compounds due to: the $O(n^2)$ processing requirements of the reciprocal nearest neighbor algorithm; the requirement to hold all chemical structure "fingerprints" in memory to enable random access; and the requirement that parallel implementation use a shared-memory architecture. In this article we develop and study alternative hierarchical agglomerative clustering algorithms that bypass these difficulties. Our first innovative algorithm is a hybrid one, incorporating the Ward agglomerative criterion; and our second algorithm adopts a different target in regard to the ultrametric (or tree distance based) output.

The structure of this article is as follows. After having noted how crucially important data coding is, for facilitating the finding of local ultrametric relationships in one's data, we then explore data coding that is based on (real-valued, or floating point) data precision. We find that this can make our data ultrametric to a greater or lesser extent, thereby bringing us closer or more remote from our objective.

In this article, our focus is strongly on ultrametric topology. The reader could validly replace this term with "hierarchy": an ultrametric topology expresses a hierarchical or tree-like structuring of the data. In the case of a metric, distance such as the Euclidean requires the triangular inequality. This will be looked at in section 2.1 and is the defining mathematical property of any distance or metric. (We ignore for convenience the properties of positivity, symmetry, and how 0 distance implies identity, because these also hold for the ultrametric case.)

An ultrametric has a stronger triangular inequality (again to be looked at in section 2.1).

An ultrametric is a distance that is defined strictly on a tree. It is the closest common ancestor distance. To check either the (metric) triangular inequality or the ultrametric or strong triangular inequalty we use a triplet of points. Any and every triplet of points must satisfy the respective inequality for us to be be able to state that the data is metric, resp. ultrametric. A range of remarkable properties of ultrametric spaces (i.e. the space we are considering *is* the tree or the hierarchy) ensue from this fairly inauspicious beginning. Of great importance to us will be the following: all triangles must be either isosceles with small base, or equilateral, in an ultrametric space. If we "ultrametrize" our data, or alternatively expressed if we induce a hierarchical or tree structure on it, we are thereby embedding it in an ultrametric topology.

We explore two different directions for this topologically inspired perspective on the data analysis. In section 3 we use it to decrease the quantity of data that we will handle using a traditional hierarchical clustering approach. In section 4.3, we explore the benefits of inducing an ultrametric through embedding our data in a Baire space.

1.2 Background to Algorithms

Massive and high dimensional data spaces often have hidden regularity. An important, practical form of regularity is hierarchical regularity. If we can exploit regularity in massive, high dimensional data sets, then we have one approach to addressing the computational and other performance problems raised by best match (or nearest neighbor) search, and related matching problems. In this work we discuss such a solution for data sets of large numbers of objects (tens of thousands), in high dimensional spaces.

The next two subsections consider, firstly, the mapping of metric (or other) data into an ultrametric, or embedding a metric in an ultrametric, which leads to the study of the distortion involved in this. Secondly, we consider the traditional multivariate data analysis approach of fitting a hierarchy to data, i.e., fitting a model, which leads to questions of optimization. As opposed to such work, we focus on recoding the data, principally through modifying the data precision. By a focus, in this way, on the data measurement process, we can find a new, general approach to the discovery of (hierarchical) structure in data.

1.3 Ultrametric Embedding: Mapping a Metric into Ultrametric

In [1], it is discussed how "large systems necessarily contain large, highly structured subsystems". Among solutions investigated in such work (see [2, 12]), there is the mapping of a point set, in a high dimensional space, into a hierarchy or tree, with known bound on the distortion of the output (ultrametric) relative to input (metric) data, and with a known bound also on the computational requirement. Our work, presented in this article, does not use these solutions because (i) the bounds are often not tight and they hold globally but not locally, (ii) classes of metric data are studied, which are not sufficiently relevant to our specific inputs, and (iii) algorithms are not well enough elucidated for our purposes. Our biggest objection to these distortion-characterizing approaches is that data coding or recoding is not addressed, yet such data recoding is crucial in practice in data analysis and handling.

We keep the same objective as that which is studied in the foregoing references, i.e., we seek the "highly structured subsystems" that are contained in larger, complex data collections. We take "hierarchical substructures" as defined by being endowed with an ultrametric. We look at the precise characterization of the data in terms of ultrametricity. Then we study efficient and effective algorithms for finding an ultrametric embedding of the data.

1.4 Fitting a Hierarchical Structure

Our objective is not the same as fitting a hierarchical structure, as is traditionally used in multivariate data analysis.

A mainstream approach over at least 4 decades of data analysis has been to fit a tree structure well to a data set, with quality of fit presupposing a clustering (mostly agglomerative, but possibly divisive). Instead we seek *inherent* ultrametricity in a data set, and so the more ultrametricity we find in our data the better: we open the door to very much easier (and possibly more computationally efficient) processing.

Both traditional agglomerative (and some divisive) algorithms and our new approach can be considered as mapping our data into an ultrametric space. In the traditional approach, for example, we can consider an agglomerative algorithm as a mapping of all pairwise distances (or indeed dissimilarities) into ultrametric distances [29]. As an example, the single link hierarchical agglomerative clustering criterion furnishes the subdominant ultrametric: for any pair of points, the ultrametric distance will always be less than or equal to an input distance. We too, in our new approach, seek an ultrametric embedding, or ultrametrization [31], in that we look for subsets of the data that ab initio are ultrametric. There are two major practical implications of this. Firstly, we bypass the need for a clustering criterion because natural or inherent ultrametricity leads to an identical result with most commonly used agglomerative criteria (for example, as specified by the Lance-Williams formula: see [26]). Secondly, how we code our data becomes very central: coding our data in the most propitious way can help greatly with how inherently ultrametric our data is.

1.5 Notation Used, and Data Normalization

We will use the notation x for the data matrix to be analyzed, and x_i denotes any particular row. A chemical structure (or chemical) i is represented by a row, and the set of chemical structures, or rows, is denoted I. We work with just over 1.2 million chemicals, $i \in I$. Similarly the column codes or attributes, 1052 in number, are denoted by set J. Needless to say, our chemicals × codes view of the data, used here for convenience of exposition, is fully compatible with a more appropriate form of storage.

We will take the notation a little further (as in [29]) by writing x_{IJ} for the given data, and a row of this matrix is denoted x_{iJ} (so we are indicating row i and the column set, J). The sum of the columns gives the vector (marginal) x_J. We normalize the data by dividing each matrix value by its column sum, and the resulting normalized matrix is denoted x_{IJ}^J. Here we are saying: the presence of a code j in chemical i must take into account whether that code is rare, implying importance of the presence property; or common, implying a lower value of presence. Given our notation, a tensor product allows us to reconstruct our original data: $x_{IJ}^J \circ x_J = x_{IJ}$. Normalization can be very important, to homogenize the effects of the coding identifiers (set J) that are used: see Figure 1.

1.6 Distributional Properties of the Data Used

We use a set of 1,219,553 chemical structures coded through 1052 presence/absence values, using the Digital Chemistry bci1052 dictionary of fragments [36]. Our experimental work is based on a matrix of binary-valued vectors: in some instances it would be more efficient to work directly on the small set of code offsets rather than a 1052-vector. The binary-valued matrix is sparse: occupancy is 8.6347%.

A power law (see [25]) is a distribution (e.g. of frequency of occurrence) containing the general form $x^{-\alpha}$ where constant $\alpha > 0$; and an exponential law is of the form e^{-x}. For a power law, $P(x > x_0) \sim cx^{-\alpha}$, $c, \alpha > 0$. A power law has heavier tails than an exponential distribution. In practice $0 \leq \alpha \leq 3$. For such values, x has infinite (i.e. arbitrarily large) variance; and if $\alpha \leq 1$ then the mean of x is infinite. The density function of a power law is $f(x) = \alpha c x^{-\alpha-1}$, and so $\ln f(x) = -\alpha \ln x + C$, where C is a constant offset. Hence a log-log plot shows a power law as linear. Power laws have been of great importance for modeling networks and other complex data sets.

Figure 2 shows a log-log plot based on the 1052 presence/absence attributes, using all 1.2 million chemicals. In a very similar way to the power law properties of large networks (or file sizes, etc.) we find an approximately linear regime, ending (at the lower right) in a large fan-out region. The slope of the linear region characterizes the power law. For this data, we find that the probability of having more than n chemicals per attribute to be approximately $c/n^{1.23}$ for large n.

The histogram of attributes per chemical, on the other hand, does not show a pronounced power law: see Figure 3. In fact, it it close to a Gaussian.

Our motivation for checking these distributional properties of our data is to show that the lessons drawn from this work will also be relevant for data sharing the same properties, from many other application domains.

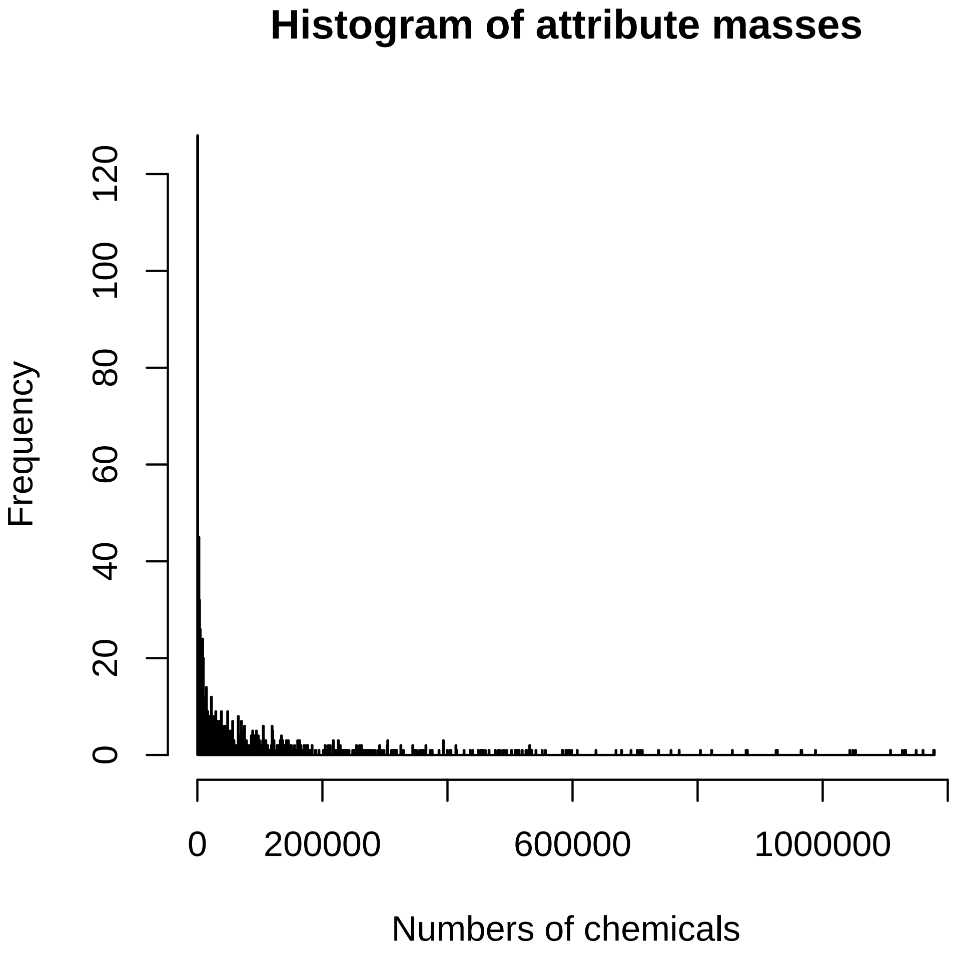


Figure 1: Histogram of column sums, denoted x_J in the text.

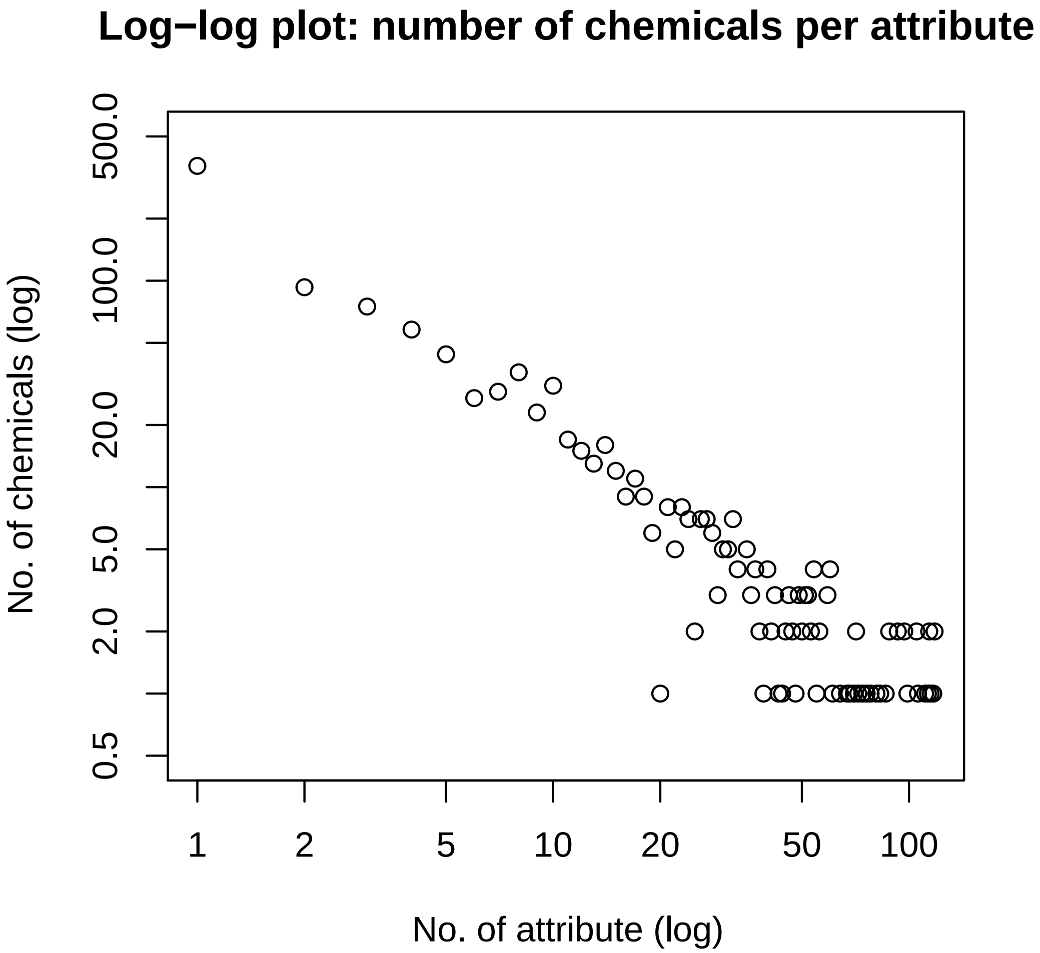


Figure 2: Log-log plot of numbers of chemicals per attribute, based on the whole data set of 1.2M chemicals.

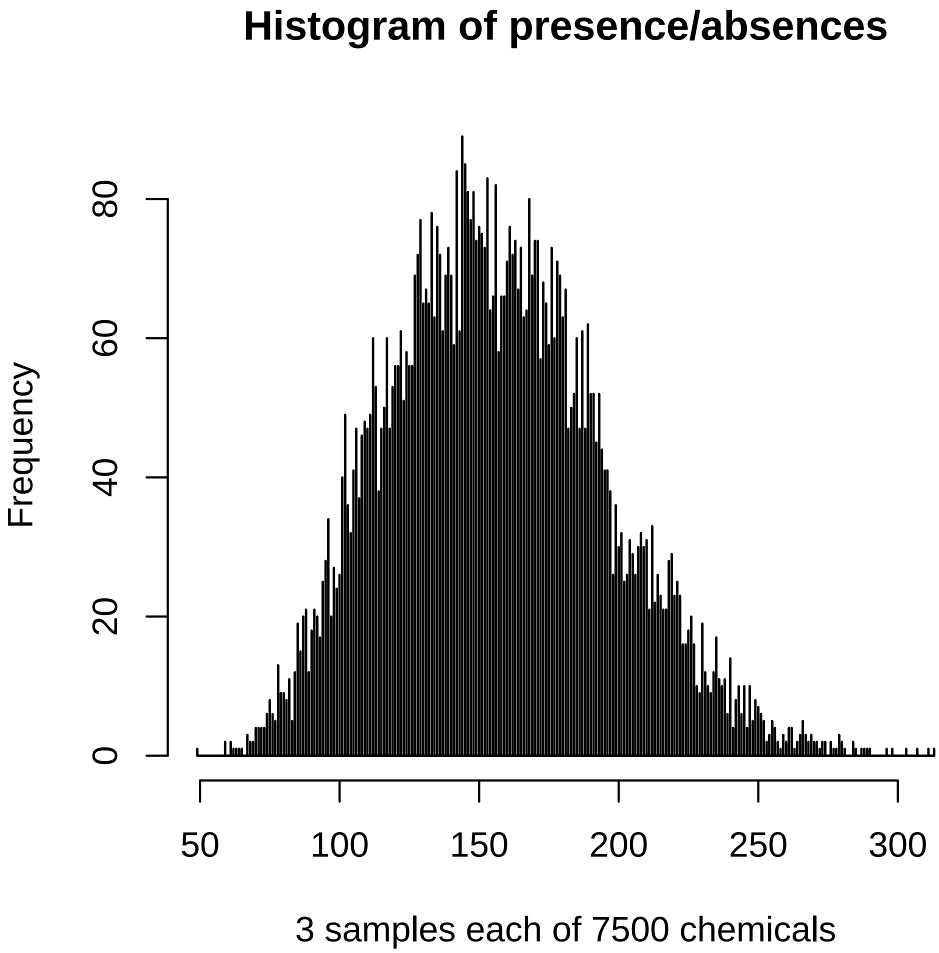


Figure 3: Histogram of presence/absence attributes for 3 different subsets of the chemicals.

2 Measuring Hierarchical Substructure

2.1 Quantifying Degree of Ultrametricity

The essential defining properties of metric versus ultrametric are the triangular versus the ultrametric inequalities. The extent to which all triplets in a data set respect one or both of these (since the ultrametric is a stronger condition, compared to the triangular inequality) leads to a measure of inherent metricity or ultrametricity in our data.

Among various properties of an ultrametric topology (see e.g. [21]) we focus on the triangular inequality, a relationship holding for all triplets of points, in order to quantify extent of ultrametricity, or inherent hierarchical structure. 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.

Our measure of extent of ultrametricity, introduced in [27], can be described algorithmically. We assume a Euclidean metric. We examine triplets of points (exhaustively if possible, but in practice with large data sets 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 angles 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.

A note on the use of Euclidean distances follows. A complete, normed vector space, which additionally has a scalar product associated with the norm, defines a Hilbert space. Scalar product gives us vector projection, and subject to normalization of the vectors it furnishes the angle between the two vectors. Hence our assumption that, by working in a Hilbert space, we have a convenient and practical environment. With finiteness, we have the Euclidean environment. In other non-Hilbert spaces, such as the L_1 Minkowski space, also referred to as the space endowed with the cityblock, Manhattan, or taxicab metric, then the notion of triangle and angle must be defined. For the L_1 space, a discussion and proposal can be found in [33]. In our experience there is little, if anything, to be gained in departing from the Euclidean or (allowing infinite dimensionality) Hilbert space context.

Commonly used dissimilarities are often not metric. By data recoding, we may well embed our data in a metric space, and this will be discussed in the next section, 2.2. Furthermore – a major motivation for us – data recoding may well enhance how well the data is embedded not just in a metric space but in an ultrametric space.

2.2 Increasing Ultrametricity Through Data Recoding

Data recoding plays an important role in the correspondence analysis tradition [29]. Data may be *doubled* meaning that each data value, and its difference from a maximum value (for an attribute) are used. The row and column marginal distributions, which furnish row and column, respectively, mass distributions, are of course altered if we recode an input data array in this way.

More generally, booleanizing, or making qualitative, data in this way, for a varying (value-dependent) number of target value categories (or modalities) leads to the form of coding known as *complete disjunctive form*.

Such coding increases the embedding dimension of the data to be analyzed, and data sparseness, and thus may encourage extent of ultrametricity. That it can do more we will now show.

The iris botanical data has been very widely used as a toy data set since Fisher used it in 1936 ([13], taking the data from a 1935 article by Anderson) to exemplify discriminant analysis. It consists of 150 iris flowers, each characterized by 4 petal and sepal, width and breadth, measurements. On the one hand, therefore, we have the 150 irises in $\mathbb{R}^4$. Next, each variable value was recoded to be a rank (all ranks of a given variable considered) and the rank was boolean-coded (viz., for the top rank variable value, $1000\ldots$, for the second rank variable value, $0100\ldots$, etc.). As a result of equal ranks, the second data set embedded the 150 irises in $\mathbb{R}^{123}$ (rather than $\mathbb{R}^{150}$). Actually, this definition of the 150 irises is more accurately in the space $\{0,1\}^{123}$ rather than in $\mathbb{R}^{123}$.

Our triangle-based measure of the degree of ultrametricity in a data set (here the set of irises), with 0 = no ultrametricity, and 1 = every triangle an ultrametric-respecting one, gave the following: for irises in $\mathbb{R}^4$, 0.017; and for irises in $\{0,1\}^{123}$: 0.948.

This provides a nice illustration of how recoding can dramatically change the picture provided by one's data. Furthermore it provides justification for data recoding if the ultrametricity can be instrumentalized by us in some way.

2.3 Recoding Distances between Chemicals Increases Ultrametricity

Given the m-dimensional encoding of chemical structures, it is known that ties in proximities necessarily result [24]. We have therefore that $d(i, i') = 0$ does not imply that $x_i = x_{i'}$. (This runs counter to most axiomatizations of distance. We can simply say that within the confines of the encoding or characterization scheme used, chemical structures i and i' are identical.) Hence a preliminary stage of finding clusters furnished by 0-valued distances is sensible. Next, we proceed somewhat further in this direction, by forcing small-valued distances to be 0 in a particular way.

Our procedure is as follows: (i) normalize the chemicals data by dividing each 0/1 chemical/attribute value by the column (or attribute) total; (ii) determine the Euclidean distance (remark: not the distance squared) between the vectors of any two chemicals, normalized in this way; and (iii) recode this distance to a limited precision. The last step is implemented by multiplying by a fixed constant, e.g. 1000, and then taking the integer part of the value. The latter, recoded, distances were assessed for their ultrametricity.

Typical sets of results are shown in Table 1. The first set of results is based on 1000 chemical structures at the start of the dataset, and this is followed by 20,000 chemical structures also from the start of the dataset used. The third set of results are on a set of 20,000 chemical structures from the 10,000th chemical structure onwards. In these results, we used 2000 samplings of triplets, but we looked at other numbers of samplings also (e.g., 1000, 4000). These results are also based on one random number seed used for the random samplings but various other seeds were used, and gave very similar results.

For the first batch of (1000 chemical structures) assessments, clearly the ultrametricity degrades with precision. For the second (20,000) chemical structures assessments, we find that the precision = 1 case is very trivial. The values in the input matrix (viz., x_{IJ}^J) are smaller because the column sums are so much smaller compared to the 1000 chemical structures case. What we have in the precision = 1 case is that the digit 0, only, is nearly always used for the distances. By taking a bigger precision, this triviality disappears.

Our conclusions are as follows. For the chemical data, we find that ultrametricity is small, – at full precision. But if we allow for distances between chemical structures (Euclidean distances between "profiles", i.e. where value of the ith chemical structure, on the jth code identifier, is x_{ij}/x_j, x_j being the column sum) to be of less than full precision, then we can find far higher ultrametricity. We have confirmed the consistency of this result for the various system parameters used (such as the sampling of triplets, or the sampling of chemical structures, on which this conclusion was based).

For searching, both isosceles with small base, or (the very rare) equilateral, cases of ultrametricity imply that data points are pulled apart. This can be of help in expediting search.

This finding can be interpreted as a hashing of the distances into fixed bin sizes. But hashing usually is applied to the original data, not the distances. So, this finding justifies the interpretation of this procedure as a new data coding procedure. Replacing each original distance, we map a pair of points onto a precision-limited dissimilarity.

2.4 Remarks on Our Recoding Procedure

Firstly, a note on metric properties of this recoding scheme follow. Whether or not the precision-based recoding of distances remains Euclidean is important for the clustering application below. For, if the recoded distances remain distances, then by virtue of the triangular inequality, having recoded distance $d'(x, y) = 0$ and $d'(x, z) = 0$ implies that $d'(y, z) = 0$. Unfortunately we cannot guarantee this. Consider a situation where we have input distances $d(x, y) = 1.0, d(x, z) = 0.9, d(y, z) = 0.9$. Then we have the triangular inequality for any triplet, e.g., $d(x, y) \leq d(x, z) + d(y, z)$ or $1.0 \leq 0.9 + 0.9$. With our recoding we now ask if $d'(x, y) \leq d(x, z) + d(y, z)$? For one digit precision re-

No. points	Dim. (Eff.)	Precision	Non-degen.	Equil.	Ultrametricity
1000	1052 (886)	1	1776	176	0.653716
1000	1052 (886)	2	1990	2	0.162312
1000	1052 (886)	3	1991	1	0.102963
1000	1052 (886)	4	1991	1	0.097438
1000	1052 (886)	5	1991	1	0.097941
1000	1052 (886)	6	1991	1	0.097941
20000	1052 (1000)	1	1	0	1.0
20000	1052 (1000)	2	1109	235	0.710550
20000	1052 (1000)	3	2000	14	0.270000
20000	1052 (1000)	4	2000	1	0.097500
20000	1052 (1000)	5	2000	1	0.099000
20000	1052 (1000)	6	2000	1	0.099500
20000	1052 (1007)	1	2	0	1.0
20000	1052 (1007)	2	1062	238	0.734463
20000	1052 (1007)	3	1999	6	0.277639
20000	1052 (1007)	4	2000	2	0.087000

Table 1: “No. of points” are the numbers of chemical structures taken on each occasion. “Dim.” is the dimensionality, or number of descriptive codes (as used in all of this work). Since a limited number of chemical structures was used on each occasion, the “Eff.” or effective dimensionality was smaller, i.e., some columns were all 0-valued. “Precision” is the number of significant digits retained. “Non-degen.” is the number of non-degenerate (i.e., degeneracy is when two or three points in the triplet overlap), out of the 2000 samplings of triplets used in each case; “Equil.” is the number of equilateral triangles found; “Ultrametricity” is 1 for complete ultrametricity.

coding we have, respectively, 1, 0 and 0, and the triangular inequality no longer holds. In practice the situation may be very favorable for us.

Clearly, by construction, the distances remain Euclidean if the precision is high enough. In the final batch of results in Table 1, this is the case for the precision with 4 decimal digits (i.e., last row of the table). With 3 decimal digits precision, we find 3998 triplets used in a 4000-samples set to be Euclidean, viz., practically all. With 2 decimal digits precision, we find 2102 triplets used in a 4000-samples set to be all Euclidean, viz., about 50%. Finally, with 1 decimal digit precision, which from Table 1 gives very degenerate results (most distances become equal to 0), we find just 3 triplets out of a 4000-samples set to be Euclidean. What this indicates is that in the 2 decimal digits case, about 50% of the triplets change (actually, our tests indicate that at least one of the distances in the triplet of distances is so affected) to dissimilarities and no longer respect the triangular inequality. This is not a problem for us: among the distances, and among triplets of distances, we find a sufficiently many, nonetheless, to be ultrametric.

Zero distances are of greater interest to us, and we again checked the cases considered in Table 1. For the 2000-sampling cases, and for the precision 1 cases, we found 98% and 99.95% of triplets respecting the triangle inequality. We can claim that we can approximate the triangular inequality for recoded distances very well (without guaranteeing this in all cases).

A note on possible use of this recoding follows. We checked the 2 decimal digits result towards the end of Table 1 (3rd last row) using 4000 triplets, and hence 12000 distances, and found just 31 unique dissimilarities or distances. For the 4 decimal digits result towards the end of Table 1 (last row) using 4000 triplets, and hence 12000 distances, we found 4666 unique distances. The far more discrete set of dissimilarities or distances, in the former case, may allow greater ease when establishing bounds on nearest neighbor proximities. What we have just sketched out will now be further studied for data clustering.

3 Data Condensation through Recoding

3.1 Implications of Increasing Ultrametricity

Data clustering is facilitated greatly through recoding, and this will be discussed in this section.

Non-uniqueness of distances such as the Euclidean distance, or other dissimilarities, in the sense of chemicals i and i' having identical dissimilarity as do chemicals i and i'', is a known issue with the data being studied [8]. We will now go further and use an "aggressive" distance recoding as described in the last section (section 2.4) to further map chemicals into clusters.

A few principles, some demonstrable and some heuristic, underly this procedure.

In agglomerative hierarchical clustering, the agglomerative criterion must be chosen. Consider the relative extremes of the single and complete link methods, where the minimal, resp. maximal dissimilarity from a cluster member to an outsider is used. If the data are ultrametric then the minimal and the maximal dissimilarities will be the same. So if the extent of ultrametricity in our data is high, we will approximate an "all agglomerative algorithms lead to the same outcome" situation.

The results of Table 1 in regard to ultrametricity indicate that for crude precision-based recoding, we are approximating global ultrametricity, and the approximation that we have the situation of "all agglomerative algorithms the same" is approximately tenable. For the 0-distance case, respect for the triangular inequality would suffice. As discussed in section 2.4, we have an excellent basis for assuming that this is generally the case in practice, although it cannot be guaranteed.

By retaining distance values with limited, small precision, we are essentially mapping distances onto clusters. Furthermore it may be of interest to pursue a phase of data condensation, by finding clusters of chemicals that are mapped onto the same point through our distance recoding. It is clear therefore that many 0-valued distances leads to an exploitable situation, for data condensation, and not a trivial one.

While there is pronounced ultrametricity in the data shown in Table 2, Figure 4 shows the implications. Perfect ultrametricity implies identity in agglomerative clustering based on the three agglomerative criteria used in Figure 4. What we find is closeness but not identity of results, due to lack of full ultrametricity. So, chemicals 20 and 28 are agglomerated late; 30 and 19, and 21 and 29, slightly earlier; and for all remaining chemicals, there is just the subclustering at a very small criterion value that is evident in the Ward (or minimum variance) and complete link agglomerative criteria.

3.2 Fast Processing of Near Neighbors

For each of n chemicals to seek a nearest neighbor among all other chemicals implies $O(n^2)$ computational cost, and we seek a way to avoid this, even if we have to accept an approximate solution. We have already noted that the data we are handling has many tied distances, and we have also seen how recoding our data – more specifically, recoding our distances – leads to many additional 0 distances. Therefore we will give particular attention to the fast finding of 0 distances. Because of the recoding, this particular interest in 0 distances is quite general, and widely applicable. Note too that 0 distances are trivially ultrametric; and all commonly used agglomerative clustering criteria will behave the same: pairwise agglomerations will be carried out at the *same* criterion value.

A spanning path [20] has been used to facilitate finding near neighbors (and also locally dense regions of space). It incorporates all points (hence is "spanning"). A minimal total distance spanning path is a Hamiltonian path, and a heuristic solution to the traveling salesperson problem, which is well-known to be NP-complete. We experimented with many ways to define a spanning path which help subsequently with finding near neighbor, and in particular 0-valued

	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	3	1	0	0	0	0	0	0	2	1	1
0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	3	1	0	0	0	0	0	0	2	1	1
0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	3	1	0	0	0	0	0	0	2	1	1
0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	3	1	0	0	0	0	0	0	2	1	1
0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	1	3	1	0	0	0	0	0	0	2	1	1
0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	1	3	1	0	0	0	0	0	0	2	1	1
0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	1	3	1	0	0	0	0	0	0	2	1	1
0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	1	3	1	0	0	0	0	0	0	2	1	1
0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	1	3	1	0	0	0	0	0	0	2	1	1
0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	1	3	1	0	0	0	0	0	0	2	1	1
0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	1	3	1	0	0	0	0	0	0	2	1	1
0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	1	3	1	0	0	0	0	0	0	2	1	1
0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	1	3	1	0	0	0	0	0	0	2	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	1	3	1	0	0	0	0	0	0	2	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	1	3	1	0	0	0	0	0	0	2	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		1	0	1	3	1	1	0	1	1	1	1	2	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1		0	1	3	1	0	0	0	0	0	0	2	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		1	3	1	0	0	0	0	0	0	2	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		3	2	1	1	1	1	1	1	2	2	1
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3		3	3	3	3	3	3	3	3	3	3
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	3		1	1	1	1	1	1	2	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	3	1		0	0	0	0	0	2	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	3	1	0		0	0	0	0	2	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	3	1	0	0		0	0	0	2	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	3	1	0	0	0		0	0	2	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	3	1	0	0	0	0		0	2	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	3	1	0	0	0	0	0		2	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3	2	2	2	2	2	2	2		2	2
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	3	1	1	1	1	1	1	1	2		1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3	1	1	1	1	1	1	1	2	1	

Table 2: Sample of 30 chemicals: recoded distances shown. Recoding used: integer part of 100 * distance. A blank is used in the principal diagonal.

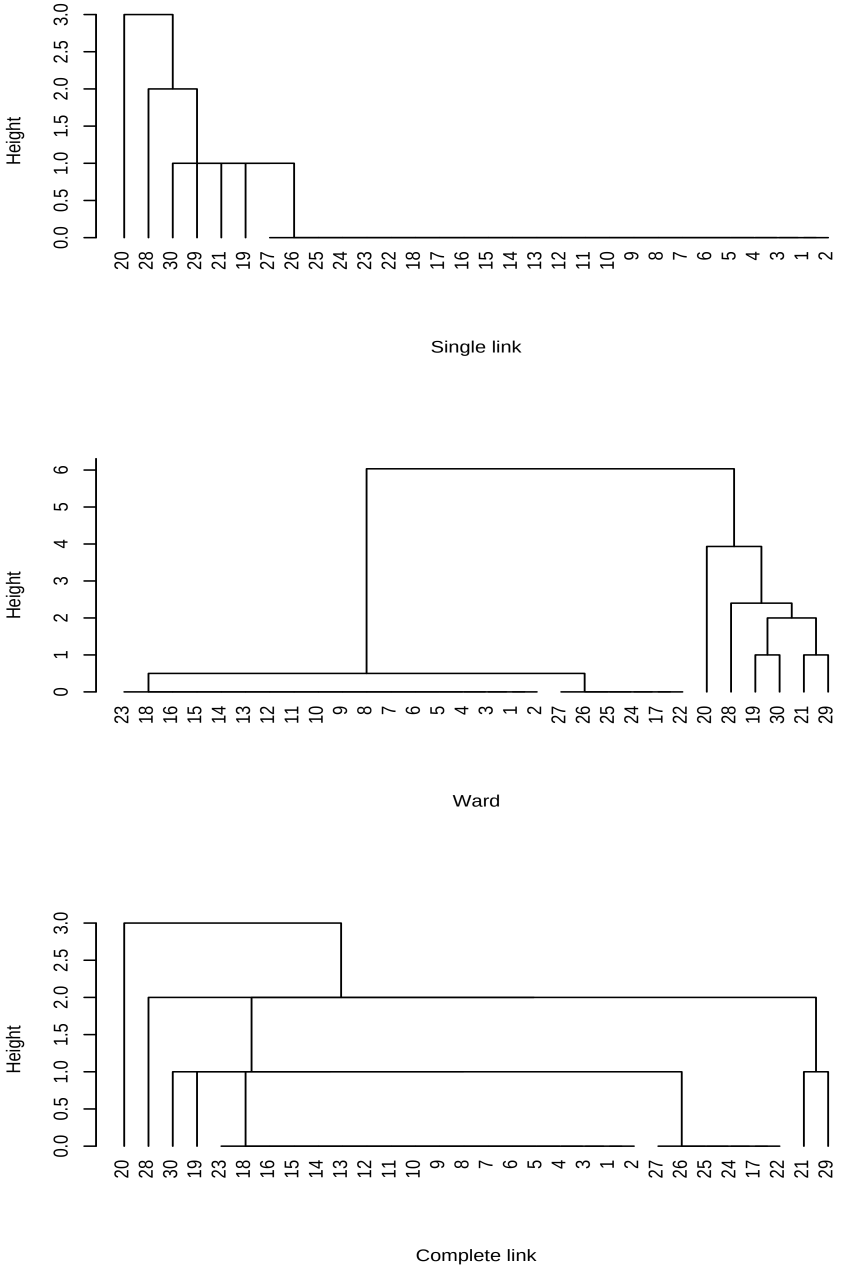


Figure 4: Hierarchical clustering, Ward's minimum variance method, with clustering cardinality weighting.

near neighbor, relationships, including random projection onto one or more axes [35], and random attribute selection. We retained from this study the use of row sums, i.e., x_I in the notation introduced earlier, or the row marginal density distribution, in order to define the spanning path. The row density itself can be found in $O(n)$ time for n rows. Finding the order of n row density values requires a sort, and therefore $O(n \log n)$ time.

Consider the set of all 0-distance objects, S_0; and the set of 0-distance neighbors in the spanning path, S_p. Then $S_p \subseteq S_0$. Clearly the reverse subset relationship does not often hold, and so there is no guarantee of having each 0-distance as a neighbor, nor necessarily close, in the spanning path. Let us see how this works in practice, since we have already noted that 0-distances are very robust relative to agglomerative criterion.

Seeking 0-distance neighbors in the spanning path, and agglomerating them, leads to a first pass over the spanning path. The spanning path itself provides candidates. By repeating (i) the defining of a new spanning path, and (ii) use of this new spanning path for locating 0-distance neighbors, we carry out a new pass over the data. Typically we quickly exhaust the possibilities of this approach to find further agglomerable 0-distance neighbors. However this exhausting of possibilities is not guaranteed. Some 0 distances may elude us, but our crude clustering based on 0 distances is exact. We could say that "precision" is 100% but "recall" is not guaranteed to be 100%. Given that the Ward minimum variance hierarchical agglomerative clustering method can use weights associated with object vectors to be clustered, we conclude that this property of our condensation algorithm (viz., that "recall" of all 0-valued distances is not 100% successful) has no effect whatsoever on the Ward hierarchy. We are simply taking as given some of the initial, 0 criterion valued, agglomerations. The only effect is in not performing as good a job as possible on condensing the input data, with the ensuing computational disadvantage for the Ward clustering.

3.3 Scaling Up

Table 3 shows results for four data sets. Most 0-distances are found on the first pass. Subsequent passes add to this, and since the processing is carried out on much shorter spanning paths, in each successive path, the computational time required is shorter. Figure 5 illustrates the cluster properties, with truncation of very frequent cluster sizes. Processing 20,000 chemicals, with 1052 attributes, takes a few minutes on a 1.8 GHz PowerPC G5 Apple processor, with 256 MB memory, using the high level, interpreted (and hence inefficient) R environment.

As noted previously: (i) the clusters that are based on the 0-valued recoded distances are exact and not approximated in any way; (ii) however we may well miss out on 0-valued recoded distances that should be included; (iii) furthermore we cannot exclude the possibility that clusters found ought to be merged, through having 0-valued recoded distances between at least some of their members.

Our preprocessing provides a good basis for subsequently hierarchically clustering the "condensed" data. From Table 3, between 30% and 40%, roughly, of the data remains to be processed. Typically from the 20,000 chemical set, about 8000 chemicals, available as an approximate 8000×1052 data array, are finally hierarchically clustered using, e.g. the Ward minimum variance agglomerative criterion.

A typical timing test for the $O(n^2)$ Ward minimum variance agglomerative criterion, on a 5-year old 2.4MHz Pentium 4 with 1 GB RAM running Windows XP SP2 using non-optimized compilations of the clustering programs, was as follows. For 15,465 structures, represented by the 1052-bit fingerprints, the time taken was 42.5 minutes.

3.4 Conclusions on Data Condensation through Recoding

The approach described in this article represents a new perspective relative to mainstream approaches using indexing (e.g. [4], using datasets of $n = 1$ million, and $m = 64$) or hashing (e.g. [10]). A simple statement of our approach is: use partial single linkage agglomerative clustering as a first phase, followed by the superior [30, 6, 7, 3] minimal variance (or Ward) agglomerative clustering algorithm. Furthermore, this first single linkage phase is for one level only, where by design the agglomerative criterion value is 0. What we show empirically in this article, and theoretically using the utrametric topology viewpoint, is that the single linkage criterion used in the first phase is not important: essentially any other commonly used criterion, including the minimum variance one, would do equally well.

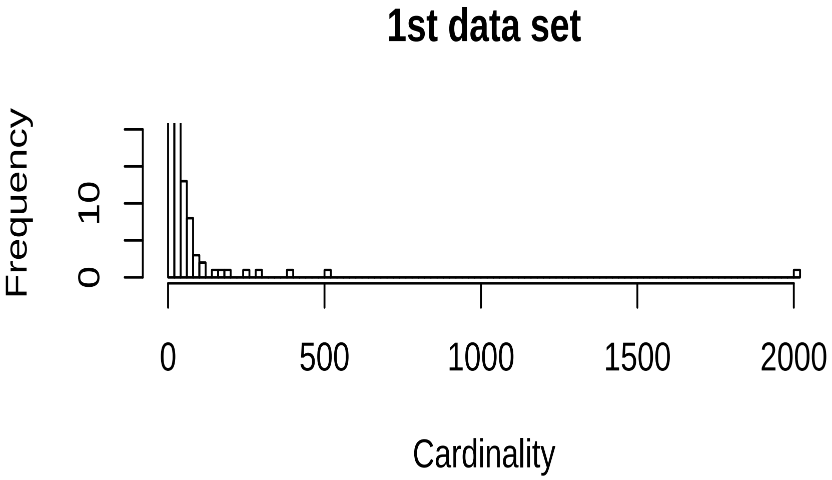
Our approach is practical and generally applicable. The data characteristics do play a role, and it was for this reason that our work began with a study of chemical/attribute frequency of occurrence density, and marginal densities. The general principles of our procedure are clear. Firstly, we discretize the precision with which we measure distance, in order to force a good deal of the data into clusters, where each cluster has its members (i.e., observation vectors) superimposed. This principle is a filtering one, widely used for denoising in signal and image processing, or regression in statistical data analysis. This condenses the data considerably. Then, secondly, on the condensed data we apply an agglomerative hierarchical clustering criterion of choice. For the data under consideration, and for the software environments used, we recommend preprocessing of about 20,000 observations at a time, and the exact hierarchical clustering on up to about 8500 observations, which can be carried out in reasonable compute time, and which lends itself to the processing in parallel of very large data sets on a computing cluster.

3.5 Case of Large Number of Observations, Limited Number of Attributes

In [24] it is noted that for a fixed length string of $|J|$ bits the number of possible, unique Euclidean distances is $|J| + 1$. Not surprisingly we have many exact ties in the data under investigation. We refer to [24] for the chemistry matching issues that ensue from this.

Data set	Pass	No. rows
1	–	20000
1	1	8487
1	2	8393
1	3	8372
1	4	8364
1	5	8360
2	–	20000
2	1	6969
2	2	6825
2	3	6776
2	4	6757
2	5	6747
3	–	20000
3	1	7266
3	2	7142
3	3	7111
3	4	7102
3	5	7097
4	–	20000
4	1	5365
4	2	5269
4	3	5248
4	4	5238
4	5	5235

Table 3: Results of 5 passes of 0-finding on four data sets, each of 20,000 chemicals (in sequence from, respectively, the 20 thousandth, 30 thousandth, 90 thousandth and 100 thousandth).



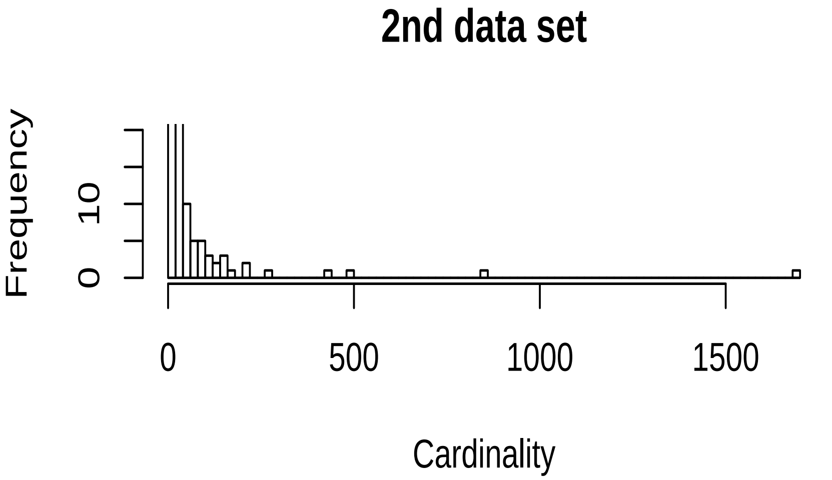


Figure 5: Histogram of cardinalities resulting from the 0-distance processing of two different data sets.

Our data condensation approach is exact in this context, as we will now show. Consider the case where we use the algorithm described in foregoing subsections to find 0-valued distances only. We do not need to find all: finding some and noting the numbers of identical cases found is fully sufficient to guarantee an exact hierarchical clustering outcome.

As a proof of concept we took the 4-dimensional Fisher iris data, [13], and created three data sets as follows. Note the close relationship between the three data sets.

- Iris flowers 1, 20, 20, 52, 80, 80, 105, 130, 130. (Note the identical cases.) Weights of 1/9 were used for each flower.
- Iris flowers 1, 20, 52, 80, 105, 130. Here we used as weights: 1/9, 2/9, 1/9, 2/9, 1/9, 2/9.
- Iris flowers 1, 20, 20, 52, 80, 105, 130. Here we consider a mixture of uniques and identicals because we take the corresponding weights as: 1/9, 1/9, 1/9, 1/9, 2/9, 1/9, 2/9.

So our three cases correspond to: presence of identicals; identicals "condensed" and weights taking this into account; and a mixture of the two cases, where only some of the identicals have been detected.

With the 4-dimensional vectors, and the weights, the Ward minimum variance clustering gave the following agglomeration criterion values, respectively:

- 0.000000000, 0.000000000, 0.000000000, 0.008148148, 0.062962963, 0.155555556, 0.896111111, 3.370061728
- 0.000000000, 0.008148148, 0.062962963, 0.155555556, 0.896111111, 3.370061728
- 0.008148148, 0.062962963, 0.155555556, 0.896111111, 3.370061728

For n vectors, we have $n-1$ levels, listed for the case of each data set above. The exact same agglomerative criterion values that we see here, account taken of the different inputs, is a very convincing demonstration of the exactness of results when either all or just some of the 0-distances can be determined.

4 Ultrametric from Longest Common Prefixes

4.1 Hierarchical Clustering and Formal Concept Analysis

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 (but we will ignore this here, since our data encoding may well have non-identical objects being identically represented). That is the traditional approach. Now we consider a different

	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

Table 4: Example dataset: 5 objects, boolean 3 attributes.

definition of distance, such that it 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.

Now, consider, say, $n = 5$ objects characterized by 3 boolean (presence/absence) attributes, shown in Table 4.

Define dissimilarity between a pair of objects in Table 4 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 [18]. 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:

$d(a, b) = 1, 1, 0$

$d(a, c) = 0, 1, 0$

$d(a, e) = 0, 1, 1$

$d(a, f) = 1, 1, 0$

$d(b, c) = 1, 1, 0$

$d(b, e) = 1, 1, 1$

$d(b, f) = 1, 1, 0$

$d(c, e) = 0, 1, 1$

$d(c, f) = 1, 1, 0$

$d(e, f) = 1, 1, 1$

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 subset $d1, d2$), and finally such that the order is defined on subset cardinality, then we see that the scheme shown in Figure 6 suffices to illustrate all salient relationships.

In Formal Concept Analysis, it is the lattice itself which is of primary interest. In [18] there is discussion of the close relationship between the traditional hierarchical cluster analysis based on $d : I \times I \rightarrow \mathbb{R}^+$, and hierarchical cluster analysis "based on abstract posets" (a poset is a partially ordered set), based on $d : I \times I \rightarrow 2^J$.

```
Potential lattice vertices      Lattice vertices found       Level

       d1,d2,d3                        d1,d2,d3                3
                                         /  \
                                        /    \
  d1,d2   d2,d3   d1,d3            d1,d2     d2,d3             2
                                        \    /
                                         \  /
   d1      d2      d3                     d2                   1
```

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 4 with the simple matching coefficient used. (See text for discussion.)

4.2 From Boolean Data to Normalized, Real-Valued Data

Consider now our need to normalize the data. We divide each boolean (presence/absence) value by its corresponding column sum. We can consider the hierarchical cluster analysis from abstract posets as based on $d : I \times I \rightarrow \mathbb{R}^{|J|}$ (section 4.1). In [18], the median of the $|J|$ distance values is used, as input to a traditional hierarchical clustering, with alternative schemes discussed. See also [17] for an early elaboration of this approach.

4.3 Ultrametrization through Baire Space Embedding: Notation

A Baire space [22] consists of countably infinite sequences with a metric defined in terms of the longest common prefix: the longer the common prefix, the closer a pair of sequences. What is of interest to us here is this longest common prefix metric, which additionally is easily seen to be an ultrametric. The longest common prefixes at issue here are those of precision of any value (i.e., x_{ij}, for chemical compound i, and chemical structure code j). Consider two such values, x_{ij} and y_{ij}, which, when the context easily allows it, we will call x and y. Each are of some precision, and we take the integer $|K|$ to be the maximum precision. We pad a value with 0s if necessary, so that all values are of the same precision. Finally, we will assume for convenience that each value $\in [0, 1)$ and this can be arranged by normalization.

4.4 The Case of One Attribute

Thus we consider ordered sets x_k and y_k for $k \in K$. In line with our notation, we can write x_K and y_K for these numbers, with the set K now ordered. (So, $k = 1$ is the first decimal place of precision; $k = 2$ is the second decimal place; $\ldots$; $k = |K|$ is the $|K|$th decimal place.) The cardinality of the set K is the precision with which a number, x_K, is measured. Without loss of generality, through normalization, we will take all $x_K, y_K \leq 1$. We will also consider decimal numbers, only, in this article (hence $x_k \in \{0, 1, 2, \ldots, 9\}$ for all numbers x, and for all digits k), again with no loss of generality to non-decimal number representations.

Consider as examples $x_K = 0.478$; and $y_K = 0.472$. In these cases, $|K| = 3$. For $k = 1$, we find $x_k = y_k = 4$. For $k = 2, x_k = y_k$. But for $k = 3, x_k \neq y_k$.

We now introduce the following distance:

$$d_B(x_K, y_K) = \begin{cases} 1 & \text{if } x_1 \neq y_1 \\ \inf 2^{-n} & x_n = y_n \quad 1 \leq n \leq |K| \end{cases} \tag{1}$$

The Baire distance is used in denotational semantics where one considers x_K and y_K as words (of equal length, in the finite case), and then this distance is defined from a common n-length prefix, or left substring, in the two words. For a set of words, a prefix tree can be built to expedite word matching, and the Baire distance derived from this tree.

We have $1 \geq d_B(x_K, y_K) \geq 2^{-|K|}$. Identical x_K and y_K have Baire distance equal to $2^{-|K|}$. The Baire distance is a 1-bounded ultrametric.

The Baire ultrametric defines a hierarchy, which can be expressed as a multiway tree, on a set of numbers, x_{IK}. So the number x_{iK}, indexed by i, $i \in I$, is of precision $|K|$. It is actually simple to determine this hierarchy. The partition at level $k = 1$ has clusters defined as all those numbers indexed by i that share the same kth or 1st digit. The partition at level $k = 2$ has clusters defined as all those numbers indexed by i that share the same kth or 2nd digit; and so on, until we reach $k = |K|$. A strictly finer, or identical, partition is to be found at each successive level (since once a pair of numbers becomes dissimilar, $d_B > 0$, this non-zero distance cannot be reversed). Identical numbers at level $k = 1$ have distance $\leq 2^0 = 1$. Identical numbers at level $k = 2$ have distance $\leq 2^{-1} = 0.5$. Identical numbers at level $k = 3$ have distance $\leq 2^{-2} = 0.25$; and so on, to level $k = |K|$, when distance $= 2^{-|K|}$.

4.5 Analysis: Baire Ultrametrization from Numerical Precision

In this section we use (i) a random projection of vectors into a 1-dimensional space (so each chemical structure is mapped onto a scalar value, by design ≥ 0 and ≤ 1) followed by (ii) implicit use of a prefix tree constructed on the digits of the set of scalar values. First we will look at this procedure. Then we will return to discuss its properties.

We seek all i, i' such that

1. for all $j \in J$,
2. $x_{ijK} = x_{i'jK}$
3. to fixed precision K

Recall that K is an ordered set. We impose a user specified upper limit on precision, $|K|$.

Now rather than $|J|$ separate tests for equality (point 1 above), a *sufficient condition* is that $\sum_j w_j x_{ijK} = \sum_j w_j x_{i'jK}$ for a set of weights w_j. What helps in making this sufficient condition for equality work well in practice is that many of the x_{iJK} values are 0: cf. the approximate 8% matrix occupancy rate that holds here. We experimented with such possibilities as $w_j = j$ (i.e., $\{1, 2, \ldots, |J|\}$ and $w_j = |J|+1-j$ (i.e., $\{|J|, |J|-1, \ldots, 3, 2, 1\}$. A first principal component would allow for the definition of the least squares optimal linear fit of the projections. The best choice of w_j values we found for uniformly distributed values in $(0, 1)$: for each j, $w_j \sim U(0, 1)$.

Table 5 shows, in immediate succession, results for the same set of three data sets used previously. The normalizing column sums were calculated and applied independently to each of the three data sets. Insofar as x_J is directly proportional, whether calculated on 7500 chemical structures or 1.2 million, leads to a

Sig. dig. c	No. clusters
4	6591
4	6507
4	5735
3	6481
3	6402
3	5360
2	2519
2	2576
2	2135
1	138
1	148
1	167

Table 5: Results for the three different data sets, each consisting of 7500 chemicals, are shown in immediate succession. The number of significant decimal digits is 4 (more precise, and hence more different clusters found), 3, 2, and 1 (lowest precision in terms of significant digits).

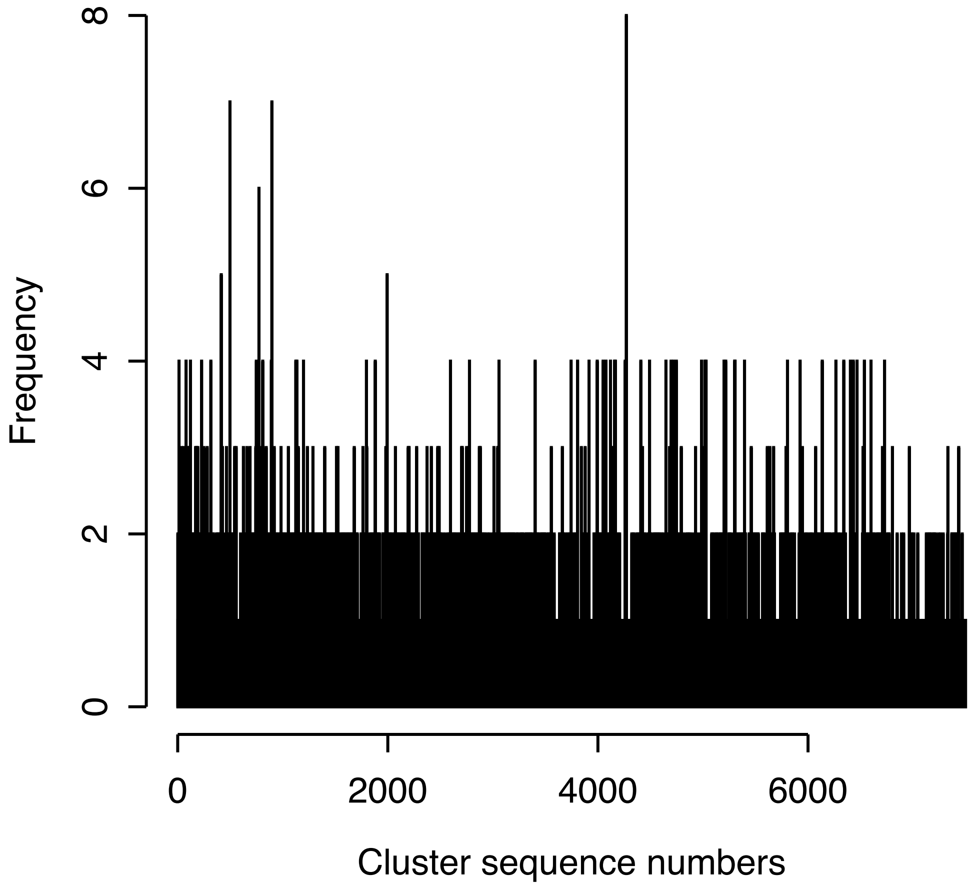


Figure 7: Histogram of cluster sizes.

constant of proportionality, only, between the two cases. A random projection was used. Finally, identical projected values were read off, to determine clusters.

Let us look closer at one outcome here, the 4-digit precision set of 6591 clusters found for the first of the three data sets used. We may ask whether these clusters are "balanced" or if, in fact, one massive cluster accounts for most of the chemical structures. Figure 7 shows a histogram, indicating clearly the "balance" in cluster cardinalities.

For a smaller precision, however, such as 1-digit, we find that one very large cluster dominates in terms of cardinality (cf. discussion of k-means results in section 4.9 below).

4.6 Discussion: Random Projection and Hashing

Random projection is the finding of a low dimensional embedding of a point set – dimension equals 1, or a line or axis, in this work – such that the distortion of any pair of points is bounded by a function of the lower dimensionality [35]. There is a burgeoning literature in this area, e.g. [10]. While random projection *per se* will not guarantee a bijection of best match in original and in lower dimensional spaces, our use of projection here is effectively a hashing method ([23] uses MD5 for nearest neighbor search), in order to deliberately find hash collisions – thereby providing a sufficient condition for the mapped vectors to be identical.

Collision of identically valued vectors is guaranteed, but what of collision of non-identically valued vectors, which we want to avoid?

To prove such a result may require an assumption of what distribution our original data follow. A general class is referred to as a stable distribution [19]: this is a distribution such that a limited number of weighted sums of the variables is also itself of the same distribution. Examples include both Gaussian and long-tailed or power law distributions.

Interestingly, however, very high dimensional (or equivalently, very low sample size or low n) data sets, by virtue of high relative dimensionality alone, have points mostly lying at the vertices of a regular simplex or polygon [27, 16]. This intriguing aspect is one reason, perhaps, why we have found random projection to work well. Another reason is the following: if we work on normalized data, then the values on any two attributes j will be small. Hence x_j and x'_j are small. Now if the random weight for this attribute is w_j, then the random projections are, respectively, $\sum_j w_j x_j$ and $\sum_j w_j x'_j$. But these terms are dominated by the random weights. We can expect near equal x_j and x'_j terms, for all j, to be mapped onto fairly close resultant scalar values.

Further work is required to confirm these hypotheses, viz., that high dimensional data may be highly "regular" or "structured" in such a way; and that, as a consequence, hashing is particularly well-behaved in the sense of non-identical vectors being nearly always collision-free.

4.7 Discussion: Prefix Trees or Tries

A prefix tree, or trie, is well-known in the searching and sorting literature [15], and is used to expedite the finding of longest common prefixes. At level one, nodes are associated with the first digit. At level two, nodes are associated with the second digit, and so on through deeper levels of the tree.

Berkeley DB (Berkeley Database, www.oracle.com/database/berkeley-db.html) provides for great scalability in dataset size, and furthermore supports trie storage. In future work we will investigate its use for efficiently and effectively supporting ultrametrization through Baire space embedding.

4.8 Simple Clustering Hierarchy from the Baire Space Embedding

The Baire ultrametrization induces a (fairly flat) multiway tree on the given data set.

Consider a partition yielded by identity (over all the attribute set) at a given precision level. Then for precision levels $k_1, k_2, k_3, \ldots$ we have, at each, a partition, such that all member clusters are ordered by reverse embedding (or set inclusion): $q_{(1)} \supseteq q_{(2)} \supseteq q_{(3)} \supseteq \ldots$. Call each such sequence of embeddings a chain. The entire data set is covered by a set of such chains. This sequence of partitions is ordered by set inclusion.

The computational time complexity is as follows. As usual, let the number of chemicals be denoted $n = |I|$; the number of attributes is $|J|$; and the total number of digits precision is $|K|$. Consider a particular number of digits precision, k_0, where $1 \leq k_0 \leq |K|$. Then the random projection takes $n \cdot k_0 \cdot |J|$ operations. A sort follows, requiring $O(n \log n)$ operations. Then clusters are read off with $O(n)$ operations. Overall, the computational effort is bounded by $c_1 \cdot |I| \cdot |J| \cdot |K| + c_2 \cdot |I| \cdot \log |I| + c_3 |I|$ (where c_1, c_2, c_3 are constants), which is equal to $O(|I| \log |I|)$ or $O(n \log n)$.

4.9 Comparison with Other Clustering Algorithms

Consider the choice of a given ("significant") digit, $|K|$, and consider distance 0 between two values. Then for digit $|K| + 1$, a maximum value of 0 and a minimum value of 1 is possible. So any two values will be strictly less than digit $|K|$ in value: the tolerance on the 0-value of the pair is $10^{-|K|}$.

Example: 0.9124 and 0.9127. Let $|K| = 3$, implying that we focus on the same 3 significant digits, here, viz. 0.912. The greatest discrepancy (e.g., using L_1 or L_2 distance) between two 0 values is the case, for example, were we to have 0.9120 and 0.9129. The tolerance on such a 0-value is therefore 10^{-3}.

Now we are considering, in all, the set J of such values. For a Euclidean distance, the overall tolerance on a 0-distance is therefore $(|J|(10^{-|K|})^2)^{1/2}$. For a city-block L_1 distance, the tolerance on a 0-distance is $|J|(10^{-|K|})$; and for a Chebyshev L_∞ distance, the tolerance on a 0-distance is $10^{-|K|}$. Whatever distance we choose we have a tolerance.

It results that our clusters of approximately, mutually, all 0-distance pairs, can also be characterized as being within a diameter given by one of the foregoing tolerances (which one depends on the distance chosen). Our clustering therefore is a fixed diameter one, or – using more usual terminology – a fixed radius clustering method. The distinction between radius and diameter holds for non-ultrametric frameworks, which is likely to be our point of departure. In an ultrametric space, the radius is equal to the diameter (just as every point in a sphere is its center).

A fixed radius clustering can be used as input to hierarchical clustering in the following way. Using Euclidean or some other distance, determine all neighbors of a given point that are within distance ρ. Then from Bruynooghe's reducibil-

Sig. dig.	No. clusters	Largest cluster	No. discrep.	No. discrep. cl.
1	138	7037	3	3
1	148	7034	1	1
1	167	6923	9	7

Table 6: Results of k-means for the same three data sets used heretofore, each relating to 7500 chemical structures, with 1052 descriptors. "Sig. dig.": number of significant digits used. "No. clusters": number of clusters in the data set of 7500 chemical structures, associated with the number of significant digits used in the Baire scheme. "Largest cluster": cardinality. "No. discrep.": number of discrepancies found in k-means clustering outcome. "No. discrep. cl.": number of clusters containing these discrepant assignments.

ity property ([5]; respected by a clustering criterion) any pair of reciprocal (or mutual) nearest neighbors can be agglomerated to form a cluster, with sufficient guaranteed separation from all other later agglomerations. ("Sufficient" means of course that influence leading to a so-called reversal in the succession of increasing agglomeration criterion values is ruled out.)

In Table 6 we look at k-means, using as input the cluster centers provided by the 1-significant digit Baire approach. Relatively very few changes were found. We note that the partitions in each case are dominated by a very large cluster, which is a direct consequence of the data used. In cases that do not give rise to such "imbalanced" cluster cardinalities, our Baire-related approach should perform even better, in that it will give rise to more equal cardinality clusters. Far more "balanced" cluster cardinalities result from clustering of the Sloan Digital Sky Survey (SDSS) archive. We are pursuing this work, using both (high quality, more costly to collect) spectroscopic and (lower quality, more readily available) photometric redshifts, that will be reported on in due course. Typically in this case we are dealing with millions of objects in a low dimensional attribute space.

5 Conclusions

We have developed and carried out assessments on (i) an approach to "condensing" a data set by finding very close neighbors, followed by a traditional hierarchical clustering of the condensed data; and (ii) an approach to "threading" all values that are identical up to successively finer digits of precision. We linked the latter to Formal Concept Analysis. A hierarchical clustering, in the second case a flat, multiway tree, is the overall objective of both approaches.

In both cases, data precision plays a central role. We linked data precision to data coding. In turn we strongly motivated our interest in such data coding by the benefits it brings vis-à-vis the hierarchical structure property, or ultrametricity, that is our ultimate goal. We have shown both approaches to be eminently practical, for our purposes of a hierarchical summarization of data.

A range of avenues for further work have been touched on in this article. Principal among them are the following. Firstly, both our algorithms implicitly use hashing. Hence further comparisons with hashing approaches pursued by Indyk (cf. [19]) could be of importance. So too would further study of the partitioning that is implicit in the flat hierarchies that we have obtained. In [37] and related work by Karypis, the benefits of partitioning over hierarchical structure are pointed to, and [32] proposes a "bisecting" or divisive (hence hybrid partioning/hierarchical) k-means algorithm. In subsection 4.9 we have started a comparative evaluation with k-means. Our preliminary results show that a similar outcome is obtained. Further work is needed, in particular given the $O(n \log n)$ computational requirement of our approach and of the "bisecting" k-means algorithm.

References

[1] Y. Bartal, N. Linial, M. Mendel and A. Naor, "On metric Ramsey-type phenomena", *Proc. of the 35th Ann. ACM Symp. on Theory of Computing*, June 2003.

[2] Y. Bartal, N. Linial, M. Mendel and A. Naor, "On metric Ramsey-type phenomena", *Annals of Mathematics*, 2006, in press.

[3] R.D. Brown and Y.C. Martin, "Use of structure-activity data to compare structure-based clustering methods and descriptors for use in compound selection, *J. Chem. Inf. Comput. Sci.*, 36 (3), 572-584, 1996.

[4] Bin Cui, Beng Chin Ooi, Jianwen Su and Kian-Lee Tan, "Contorting high dimensional data for efficient main memory KNN processing", *Proc. 2003 ACM SIGMOD Int'l. Conf. on Management of Data*, pp. 479–490, 2003.

[5] M. Bruynooghe, "Classification ascendante hiérarchique des grands ensembles de données: un algorithme rapide fondé sur la construction des voisinages réductibles", *Les Cahiers de l'Analyse de Données*, III, 7–33, 1978.

[6] G.M. Downs, P.T. Walsh and A.M. Booth, "Similarity and clustering of chemical structures for property prediction", Presented at the *2nd International Workshop on Computer Chemistry: Structure-Property Relations*, at the Technical University, Merseburg, Germany, 2-4 October 1990. (HSE Report IR/L/FT/90/6)

[7] G.M. Downs and P. Willett, "The use of similarity and clustering techniques for the prediction of molecular properties", In *Applied Multivariate Analysis in SAR and Environmental Studies*, Devillers, J.; Karcher, W. (Eds.). Kluwer Academic Publishers, Dordrecht, pp. 247-280, 1991.

[8] G.M. Downs, P. Willet, and W. Fisanick, "Similarity searching and clustering of chemical structure databases using molecular property data", *Journal of Chemical Information and Computer Science*, 34, 1094–1102, 1994.

[9] G.M. Downs and P. Willett, "Similarity searching in databases of chemical structures", *Reviews in Computational Chemistry*, 7, 1–66, 1995.

[10] D. Dutta, R. Guha, P. Jurs and T. Chen; "Scalable partitioning and exploration of chemical spaces using geometric hashing", *J. Chem. Inf. Model.*, 46(1), 321-333, 2006.

[11] N. Eiron and K.S. McCurley, "Link structure of hierarchical information networks", *Proc. Third Workshop on Algorithms and Models for the Web-Graph (WAW 2004), Lecture Notes in Computer Science*, pp. 143–155, 2004.

[12] J Fakcharoenphol, S Rao and K Talwar, "A tight bound on approximating arbitrary metrics by tree metrics", *Proc. of the 35th Ann. ACM Symp. on Theory of Computing*, June 2003.

[13] R.A. Fisher, "The use of multiple measurements in taxonomic problems", *The Annals of Eugenics*, **7**, 179–188, 1936.

[14] V.J. Gillet, D.J. Wild, P. Willett and J. Bradshaw, "Similarity and dissimilarity methods for processing chemical structure databases", *Computer Journal*, 41, 547–558, 1998.

[15] D. Gusfield, *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*, Cambridge University Press, 1997.

[16] P. Hall, J.S. Marron and A. Neeman, "Geometric representation of high dimension, low sample size data", *Journal of the Royal Statistical Society B*, 67, 427–444, 2005.

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

[18] M.F. Janowitz, "Cluster analysis based on abstract posets", preprint. Also: M.F. Janowitz, "Cluster analysis based on posets", presentations, 2005-2006. http://dimax.rutgers.edu/∼melj

[19] P. Indyk, A. Andoni, M. Datar, N. Immorlica and V. Mirrokni, "Locally-sensitive hashing using stable distributions", in T. Darrell and P. Indyk and G. Shakhnarovich (eds.), *Nearest Neighbor Methods in Learning and Vision: Theory and Practice*, MIT Press, 2006.

[20] R.C.T. Lee, "Clustering analysis and its applications", in J.T. Toum, Ed., *Advances in Information Systems Science*, Plenum Press, New York, 1981.

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

[22] A. Levy, *Basic Set Theory*, Dover, 1979 (reprinted 2002).

[23] M.L. Miller, M.A. Rodriguez and I.J. Cox, "Audio fingerprinting: nearest neighbor search in high dimensional binary spaces", *Journal of VLSI Signal Processing*, 41, 285-291, 2005.

[24] J. MacCuish, C. Nicolaou and N.E. MacCuish, "Ties in proximity and clustering compounds", *J. Chem. Inf. Comput. Sci.*, 41, 134–146, 2001.

[25] M. Mitzenmacher, "A brief history of generative models for power law and lognormal distributions", *Internet Mathematics*, 1, 226–251, 2004.

[26] F. Murtagh, *Multidimensional Clustering Algorithms*, Physica-Verlag, 1985.

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

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

[29] F. Murtagh, *Correspondence Analysis and Data Coding with R and Java*, Chapman & Hall/CRC, 2005.

[30] E.M. Rasmussen, G.M. Downs and P. Willett, "Automatic classification of chemical structure databases using a highly parallel array processor", *Journal of Computational Chemistry*, 9, 378-386, 1988.

[31] A.C.M. van Rooij, *Non-Archimedean Functional Analysis*, Marcel Dekker, 1978.

[32] M. Steinbach, G. Karypis and V. Kumar, "A comparison of document clustering techniques", Computer Science and Engineering, University of Minnesota, technical report 00-034, 2000.

[33] K. Thompson and T. Dray, "Taxicab angles and trigonometry", *Pi Mu Epsilon Journal*, 11, 87–96, 2000.

[34] C.J. van Rijsbergen, *Information Retrieval*, 2nd ed. Butterworths, 1979.

[35] S.S. Vempala, *The Random Projection Method*, DIMACS: Series in Discrete Mathematics and Theoretical Computer Science, Rutgers University, Vol. 65, American Mathemtical Society, 2004.

[36] M. Wright, "Fingerprinting and dictionary generation",
http://www.digitalchemistry.co.uk/prod_fingerprint.html, 2006.

[37] Ying Zhao and G. Karypis, "Hierarchical clustering algorithms for document datasets", *Data Mining and Knowledge Discovery*, 10, 141–168, 2005.