# Hierarchical Distance-based Conceptual Clustering*

A. Funes[1,2], C. Ferri[2], J. Hernández-Orallo[2], M. J. Ramírez-Quintana[2]

[1]Universidad Nacional de San Luis, Ejército de los Andes 950, 5700 San Luis, Argentina
afunes@unsl.edu.ar
[2]DSIC, Universidad Politécnica de Valencia, Camino de Vera s/n, 46022 Valencia, España
{cferri, jorallo, mramirez}@dsic.upv.es

**Abstract.** In this work we analyse the relation between hierarchical distance-based clustering and the concepts that can be obtained from the hierarchy by generalisation. Many inconsistencies may arise, because the distance and the conceptual generalisation operator are usually incompatible. To overcome this, we propose an algorithm which integrates distance-based and conceptual clustering. The new dendrograms can show when an element has been integrated to the cluster because it is near in the metric space or because it is covered by the concept. In this way, the new clustering can differ from the original one but the metric traceability is clear. We introduce three different levels of agreement between the clustering hierarchy obtained from the linkage distance and the new hierarchy, and we define properties these generalisation operators should satisfy in order to produce distance-consistent dendrograms.

**Keywords:** conceptual clustering, hierarchical clustering, generalisation, distances.

## 1 Introduction

Distances and generalisations are the underlying concepts to two different approaches for machine learning. Similarity, which is a broader concept than distance, is the basis for many inductive inference techniques, since similar elements are expected to behave similarly. Distances do not only formalise the notion of similarity between cases or individuals, but provide the additional properties of metric spaces, which are advantageously exploited by many techniques, known as distance-based.

Generalisation is also another key concept in machine learning. Any inductive learning involves some kind of generalisation. Unlike distance-based methods, some approaches are based on the idea that a generalisation or pattern discovered from old data can be used to describe new data covered by this pattern. These techniques are known as model-based.

* This work has been partially supported by the EU (FEDER) and the Spanish MEC/MICINN under grant TIN2007-68093-C02 and the Spanish project "Agreement Technologies" (Consolider Ingenio CSD2007-00022). A. Funes was supported by a grant from the Alfa Lernet project and the UNSL.

Distance-based techniques are quite intuitive and flexible, in the sense that we only need to define a distance function for the data we are working with. However, distance-based methods do not provide a pattern or explanation which justifies the decision made for a given individual. In particular, distance-based clustering systems arrange elements into groups based on a numerical measure of similarity between elements. Therefore, the resulting clusters lack conceptual descriptions making them difficult to interpret. For instance, it is helpful to know that a given molecule belongs to a cluster because it is similar to the elements of the cluster according to a certain distance measure, but it would even be more interesting to know what chemical properties are shared by all the molecules in the cluster.

A well-known approach for distance-based clustering is hierarchical clustering [1, 2]. In hierarchical distance-based clustering, data are split into clusters during several partition steps forming a hierarchy of clusters from a single cluster containing all the elements to *n* clusters containing just one element. Depending on how the hierarchy is built, hierarchical clustering can be classified as agglomerative (bottom-up) or divisive (top-down).

A different approach to clustering is conceptual clustering defined by Michalski [3, 4]. Conceptual clustering overcomes the cluster interpretation problem by forming clusters that can be described by properties involving relations on a selected set of attributes. A conceptual clustering system accepts a set of object descriptions and produces a partition over the observations. These descriptions can be viewed as cluster generalisations, which are expressed as patterns common to all the elements of the cluster.

In this work we present a general approach for clustering in such a way that we use a distance to construct the cluster hierarchy while also producing patterns. The core of the approach is an algorithm for Hierarchical Distance-based Conceptual Clustering (HDCC). The key issue here, which has been neglected by other conceptual clustering methods that use distances, is whether the hierarchy induced by a distance and the discovered patterns are consistent, i.e. are all the elements covered by a pattern close with respect to the underlying distance? To answer the question, first we need to clearly show when this happens. This has led to a new graphical representation of the resulting dendrogram (that we have named conceptual dendrogram). We also need to analyse a priori whether the inconsistencies will appear or not. This has given rise to the development of three levels of consistency between distances and generalisations and the corresponding properties which ensure (in a higher or lower degree) that the conceptual clustering also reflects the distribution of examples in the metric space. This means that if for a given problem we are able to prove these properties, we will know beforehand that the resulting hierarchy of patterns is at the same time consistent with the distance and the concepts expressed by each pattern in the hierarchy.

The main contribution of this work is a practical and general way to integrate hierarchical distance-based and conceptual clustering smoothly. Additionally, the algorithm is also a general way to construct an *n*-ary generalisation operator from binary generalisation operators in a metric space. Our approach is general in the sense that it can be applied to any distance, pattern language and generalisation operator. Consequently, this idea is directly applicable to structured data. One possible instantiation would provide us with the descriptions or generalisations for clusters of first order atoms obtained by the application of Plotkin´s least general generalisation

operator (*lgg*) at the same time that the process of clustering uses a distance for atoms, e.g. the distance defined in [5]. Another direct instantiation would be for example the clustering of lists using regular patterns and the edit distance.

The work is organised as follows. In Section 2 some necessary previous concepts are summarized. Our proposal (HDCC) is presented in Section 3. In Section 4 we show theoretical results about some generalisation operator properties. In section 5 we present some experiments which compare our method to traditional conceptual clustering. Finally, Section 6 closes the paper with the conclusions and future work.

## 2 Preliminaries

Intuitively, the generalisation of a finite set of elements $E$ in a metric space $(X, d)$ could be extensionally defined as a set that contains $E$. However, this kind of extensional definition gives no insight on the concept or pattern that the elements in the generalisation share. We say that a pattern $p \in \mathcal{L}$, where $\mathcal{L}$ is the pattern language, is an intensional way of representing a set of elements of $X$, which are denoted by $Set(p)$.

First we introduce the definition of binary generalisation operators over a metric space and then we extend this concept to patterns in Definition 2.

**Definition 1.** *Let* $(X, d)$ *be a metric space and* $\mathcal{L}$ *a pattern language. A* binary generalisation operator *is a function* $\Delta: X \times X \rightarrow \mathcal{L}$ *such that given* $x_1 \in X, x_2 \in X, \Delta(x_1, x_2) = p$, *where* $p \in \mathcal{L}$ *and* $x_1 \in Set(p)$ *and* $x_2 \in Set(p)$.

Fig. 1 (Left) shows five possible generalisations of two points in the metric space $(\Re^2, d)$, where $d$ is the Euclidean distance.

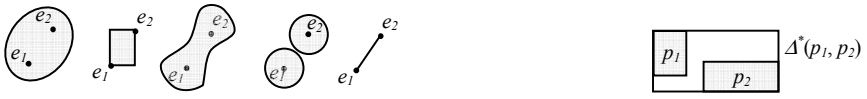


**Fig. 1. (Left)** Five possible generalisations of two points in $\Re^2$. **(Right)** A generalisation of two patterns $p_1$ and $p_2$ in $\Re^2$.

**Definition 2.** *Let* $(X, d)$ *be a metric space and* $\mathcal{L}$ *a pattern language*. *A* pattern binary generalisation operator *is a function* $\Delta^*: \mathcal{L} \times \mathcal{L} \rightarrow \mathcal{L}$ *such that given* $p_1 \in \mathcal{L}$ *and* $p_2 \in \mathcal{L}$, $\Delta^*(p_1, p_2) = p$, *where* $p \in \mathcal{L}$ *and* $Set(p_i) \subseteq Set(p)$ $(i \in \{1,2\})$.

Definition 2 establishes that a generalisation of two patterns must describe at least all the elements described by both patterns. In Fig. 1 (Right) we show a possible generalisation for two patterns $p_1$ and $p_2$ in $\mathcal{L}$, where $\mathcal{L}$ is the set of all axis-parallel rectangles.

Note that when $\mathcal{L} = X$, as it happens, e.g. with *lgg* for atoms, the operator $\Delta^*$ and $\Delta$ can be the same.

## 3 Hierarchical Distance-based Conceptual Clustering Algorithm

The approach to clustering we propose is based on one of the most known and simple bottom-up distance-based algorithms, the agglomerative hierarchical clustering. It builds a hierarchy of clusters from individual elements by progressively merging clusters. Clusters are joined based on the distance between them, referred as the linkage distance. Usually, the linkage distance is determined by the maximum distance between elements of each cluster (i.e. complete linkage distance, $d^c_L$), by the minimum distance between elements of each cluster (i.e. single linkage distance, $d^s_L$), by the mean distance between elements of each cluster (i.e. average linkage distance, $d^a_L$), or by the minimum distance between the cluster prototypes (i.e. prototype linkage distance, $d^p_L$), among others. In the rest of the paper we will only consider these four functions, $d^c_L$, $d^s_L$, $d^a_L$, and $d^p_L$. We use $d_L$ to refer any of them.

In traditional agglomerative hierarchical clustering, the process of clustering starts at the leaves of the tree where each leaf corresponds to a one-element cluster. Then it joins the two closest clusters into a new cluster that becomes the parent of the formers into the hierarchy. Now the new cluster and the rest minus the two closest ones compose the new set of clusters. This process is repeated until eventually the set of clusters is formed by only one cluster containing all the elements.

A problem appears if we want a pattern or description for each cluster. Since the clustering process is driven by the underlying distance, a discovered pattern obtained by generalisation may describe the elements of a cluster but it might describe other elements of the metric space that are not included into the cluster. This can lead to an inconsistency between the clusters described by the patterns and those resulting from the hierarchical algorithm. To illustrate the problem let us consider the example for lists of symbols given in Fig. 2 (Left). The elements belong to the metric space $(X, d)$ where $X$ is the set of all the finite list of symbols on the alphabet $\Sigma = \{a, b\}$ and $d$ is the edit distance or Levenshtein distance [6] considering the cost of a replacement as the cost of a supression plus an insertion. The figure shows four elements (*aa*, *aab*, *abb*, *aabbbbbb*) and the distances between them.

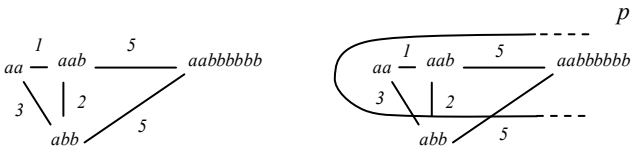


**Fig. 2. (Left)** Four examples of lists in $(\Sigma^*, d)$. **(Right)** The coverage of pattern $p = aa*$.

According to the hierarchical clustering algorithm with single linkage and taking into account the distances between the examples, the resulting clusters are those shown in Fig. 3 (Left). Let us suppose that the chosen generalisation operator produces the pattern $aa*$ for the cluster $\{aa, aab\}$. Clearly, there is a metric inconsistency between the elements described by $aa*$ (*aa*, *aab*, *aaa*, *aaba*, *aabb*,…) and the clusters induced by the distance, since $aa*$ covers *aabbbbbb* but it does not cover *abb*, which is closer (see Fig. 2 (Right)).

With this idea, the proposed approach to hierarchical distance-based conceptual clustering (HDCC) makes a generalisation operator and a distance work together by achieving a simple adaptation to the hierarchical base algorithm. This adaptation consists in merging to each new cluster all those clusters covered by its generalisation. In this way, the final patterns provide a description common to all the elements that are close according to the underlying distance but also of those that although not close enough to be part of the cluster are covered by the pattern. To represent the resulting clustering we use an extended dendrogram that we have named *conceptual dendrogram*. A conceptual dendrogram provides not only with the traditional information about what elements are in each cluster but it also gives a description of the common properties of their elements in the form of a pattern. A solid line links the clusters merged by the distance, while a dashed line links those merged by a pattern. Fig. 3 (Right) shows the conceptual dendrogram for the current example. The pattern $p = aa*$ covers the cluster $\{aa, aab, aabbbbbb\}$, which has been formed considering in first place the distance between the clusters and in second place the coverage of the resulting pattern $aa*$.

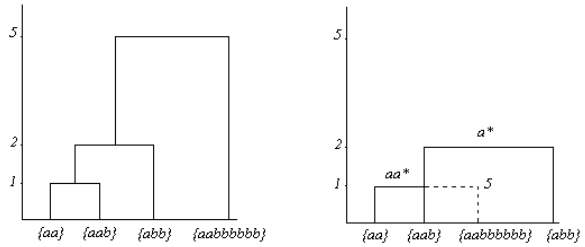


**Fig. 3. (Left)** Traditional dendrogram. **(Right)** Conceptual dendrogram.

To overcome the inconsistency problem between the distance and the generalisation operator mentioned above, HDCC performs a coverage-reorganisation process that consists in merging to the new cluster $C$ with pattern $p$ all those clusters in the hierarchy that are included in $Set(p)$. Hence these conceptually-added clusters can play a very different role in the construction of the hierarchy. Note that this process is performed during the construction of the hierarchy, and not as a post-process. A post-processing over the original dendrogram would not yield a distance-consistent explanation of the hierarchy and it would imply a much more complex, costly and thorough reorganization of the hierarchy.

Table 1 shows a pseudo code for HDCC. The output is a tree $T$ where each node is a cluster with its corresponding pattern and linkage distance (shown on the Y-axis). The HDCC is in fact a $n$-ary generalisation operator.

**Table 1.** Hierarchical distance-based conceptual clustering

**Input**: $E=\{e_1, e_2, \ldots, e_n\} \subseteq X$ and a distance $d$, with $(X, d)$ a metric space; $\Delta^*$: $\mathcal{L} \times \mathcal{L} \rightarrow \mathcal{L}$ a pattern binary generalisation operator; $\Delta: X \times X \rightarrow \mathcal{L}$ a binary generalisation operator; $d_L: 2^X \times 2^X \times (X \times X \rightarrow \Re) \rightarrow \Re$ a linkage distance.

**Output**: A tree $T$ of clusters and generalisations.

1. $S \leftarrow \{\{e_1\}, \{e_2\}, \ldots, \{e_n\}\}$.
2. Insert tuple $(\{e_i\}, \Delta(e_i, e_i), 0)$ as a leaf of $T$, for all $\{e_i\}$ in $S$.
3. While $S \neq \{E\}$ do
   3.1. Compute $d_L(C_i, C_j, d)$ between each pair of clusters $C_i, C_j \in S$ with $i<j$, using the distance $d$.
   3.2. Compute the pattern $p_{Cxy}$ of cluster $C_{xy}$ as $\Delta^*(p_{Cx}, p_{Cy})$, where $C_{xy} = C_x \cup C_y$, $p_{Cx}, p_{Cy}$ are the patterns of $C_x$ and $C_y$, respectively, and $C_x$ and $C_y$ are the closest clusters in $S$ according to $d_L$.
   3.3. $S \leftarrow S \cup \{C_{xyz}\}$ and $C_{xyz} = C_{xy} \cup C_z$ and $C_z = \{ e \mid e \in C_i \wedge C_i \in S \wedge C_i \subseteq Set(p_{Cxy}) \}$
   3.4. Insert $(C_{xyz}, p_{Cxy}, d_{LCxy})$ in $T$ as the parent node of $(C_x, p_{Cx}, d_{LCx})$, $(C_y, p_{Cy}, d_{LCy})$ and of nodes $(C_i, p_{Ci}, d_{LCi})$ where $C_i \in S$ and $C_i \subseteq Set(p_{Cxy})$.
   3.5. $S \leftarrow S - \{C_i\}$ for all $C_i$ s.t. $C_i \subseteq Set(p_{Cxy})$.
4. Return $T$.

The following simple example illustrates how the HDCC algorithm works under single linkage. Let us suppose the evidence is the set of points in $\Re^2$ shown in Fig. 4 (Left) while the generalisation operator and the pattern language are the same used in the example shown in Fig. 1 (Right).

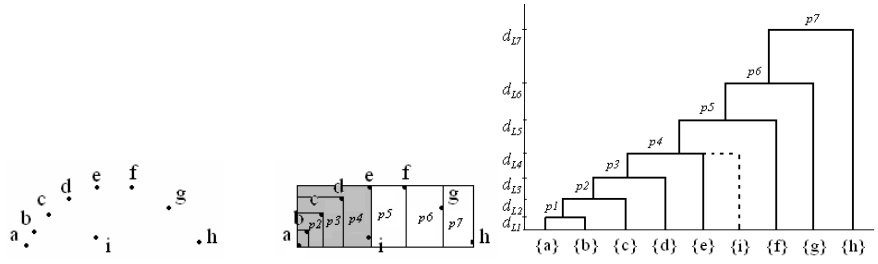


**Fig. 4. (Left)** A set of points in $\Re^2$. **(Center)** The discovered patterns $p1, \ldots, p7$. The shadowed area shows the evidence covered by $p4$. **(Right)** Conceptual dendrogram.

Fig. 4 (Right) shows the resulting conceptual dendrogram. The clusters $\{a, b\}$, $\{a, b, c\}$, $\{a, b, c, d\}$ and $\{a, b, c, d, e\}$ have been formed driven by the distance. However, as we can see in Fig. 4 (Center) the cluster $\{i\}$ has been merged to $\{a, b, c, d, e\}$ by the pattern $p_4$ that covers both. Note that $\{i\}$ would have been the last merged cluster by $d^s_L$ in the traditional dendrogram.

## 4 Consistency between Distances and Generalisation Operators

The exact shape of the conceptual dendrogram and whether it has dashed links depends not only on the distance $d$ and the generalisation operators used but also on the linkage distance $d_L$. We can talk of several degrees of consistency between distances and generalisations on the basis of the similarity between a conceptual dendrogram and the traditional one. The more similar the dendrograms are the more

consistent the distance is wrt. the generalisation operator. Next we present three different conditions to ensure that the generalisation operator produces distance-consistent dendrograms.

### 4.1 Equivalent Dendrograms

In some cases, the conceptual dendrogram is isomorphic to the traditional dendrogram. This happens when the discovered patterns do not cover any other cluster besides those linked by the distance, i.e. each new cluster is formed only by merging the closest clusters or it is composed of only one element (i.e. it is a leaf cluster). Therefore, we say that a conceptual dendrogram is *equivalent to a traditional dendrogram* if for each cluster $C$ which is not a leaf all its children are linked at the same distance $l$. This is formalised in Definition 3.

**Definition 3.** *Let T be the tree resulting from* HDCC. *T is* equivalent to a traditional dendrogram *iff* $\forall\ (C, p, l) \in T : (|C| = 1 \vee (\forall\ (C_i, p_i, l_i)$ *child of* $(C, p, l), \exists\ (C_j, p_j, l_j)$ *child of* $(C, p, l)$ *in T such that* $d_L(C_i, C_j, d) = l))$.

If we want equivalent dendrograms, each time HDCC determines the two closest clusters $C_1$ and $C_2$ with linkage distance $l$, the corresponding pattern $p$ should not cover any other cluster $C$ whose distances $l_1$ and $l_2$ to $C_1$ and $C_2$ respectively are greater than $l$. Note that $l_1$ and $l_2$ can not be lower than $l$ since in this case HDCC would have merged this cluster to $C_1$ or $C_2$ before. We say that generalisation operators that generate patterns whose coverage satisfies this condition are *strongly bounded by* $d_L$. Intuitively, a pattern binary generalisation operator is strongly bounded by $d_L$ when for any pair of patterns $p_1$, $p_2$, and any pair of sets $C_1$ and $C_2$ covered by each, the linkage distances from the new elements covered by the generalisation of $p_1$ and $p_2$ to $C_1$ and $C_2$ are equal or lower than the linkage distance between $C_1$ and $C_2$, i.e. the new elements covered by the generalisation of $p_1$ and $p_2$ fall into the balls of radius $d_L(C_1, C_2, d)$ and centre in the linkage points of $C_1$ and $C_2$. The linkage points are, in the case of $d^s_L$, the two closest elements in $C_1$ and $C_2$; the two most distant elements in $d^c_L$; the prototypes in the case $d^p_L$ and the centroids in the case of $d^a_L$ (assuming the metric space is continuous). This concept is formalised in Definition 4.

**Definition 4.** *Let* $(X, d)$ *be a metric space,* $\mathcal{L}$ *a pattern language and* $d_L$ *a linkage distance. A pattern binary generalisation operator* $\Delta^*$ *is* strongly bounded by $d_L$ *iff* $\forall\ p_1, p_2 \in \mathcal{L}, C_1 \subseteq Set(p_1), C_2 \subseteq Set(p_2), C \subseteq Set(\Delta^*(p_1, p_2)) - (Set(p_1) \cup Set(p_2)) : d_L(C, C_1, d) \leq d_L(C_1, C_2, d) \vee d_L(C, C_2, d) \leq d_L(C_1, C_2, d)$.

Fig. 5 (Left) shows clusters $\{a, b, c\}$ and $\{d, e, f\}$ formed under single linkage. The patterns used are unions of axis-parallel rectangles. $A$ is the generalisation of $\{d, e\}$, $A \cup C$ of $\{d, e, f\}$, $B$ of $\{a, b\}$; $B \cup D$ of $\{a, b, c\}$ and $A \cup C \cup B \cup D \cup E$ of $\{a, b, c, d, e, f\}$. The union of the circles determines the area where the new elements in the generalisation of $A \cup C$ and $B \cup D$ should be if $\Delta^*$ is strongly bounded by $d^s_L$.

Definition 5 gives the same property for a binary generalisation operator. A binary generalisation operator is strongly bounded by $d_L$ when for any pair of elements $e_1$ and $e_2$, the linkage distances from $\{e_1\}$ and $\{e_2\}$ to any cluster $\{e\}$ covered by the

generalisation of $e_1$ and $e_2$ is lower than the linkage distance between $\{e_1\}$ and $\{e_2\}$, i.e. $e$ must fall into the balls of radius $d_L(\{e_1\}, \{e_2\}, d)$ and centre in $e_1$ and $e_2$.


**Fig. 5. (Left)** The union of the circles shows the region where the new elements in the generalisation of $A \cup C$ and $B \cup D$ should be. **(Right)** Maximum coverage of a binary generalisation operator $\Delta$ strongly bounded by $d_L$.

**Definition 5.** *Let* $(X, d)$ *be a metric space,* $\mathcal{L}$ *a pattern language and* $d_L$ *a linkage distance. A binary generalisation operator* $\Delta$ *is* strongly bounded by $d_L$ *iff* $\forall\ e, e_1, e_2 \in X$*: if* $e \in Set(\Delta(e_1, e_2))$ *then* $d_L(\{e\}, \{e_1\}, d) \leq d_L(\{e_1\}, \{e_2\}, d) \vee d_L(\{e\}, \{e_2\}, d) \leq d_L(\{e_1\}, \{e_2\}, d)$.

Fig. 5 (Right) shows the area in $\Re^2$ that a pattern $p$ is allowed to cover when $\Delta$ is strongly bounded by $d_L$. Note that when we generalise only two elements the linkage distance $d_L$ is equal to the distance $d$ between the elements for any $d_L$.

The linkage distance $d_L$ used by HDCC affects the boundedness property of generalisation operators. Given a distance $d$, a generalisation operator could be strongly bounded under a given $d_L$ but not under a different one. For example, the pattern generalisation operator $\Delta^*$ shown in Fig. 4 (Center) is not strongly bounded by $d^s_L$ but it is strongly bounded by $d^c_L$. We can easily see that it is not strongly bounded by $d^s_L$ because, for instance, the point $i$ covered by the pattern $p_4$ is outside the balls with centre in $d$ and $e$ and radius $d^s_L(\{a, b, c, d\}, \{e\}, d)$. Note that $d^s_L(\{a, b, c, d\}, \{e\}, d) = d(d, e)$. However, $\Delta^*$ is strongly bounded by $d^c_L$ since each pattern covers a rectangle that is determined by the two most distant points $e_1$ and $e_2$ in $C1$ and $C2$, and this rectangle is always in the intersection of the two balls $B(e_1, l)$ and $B(e_2, l)$, where $l = d^c_L(C1, C2, d) = d(e_1, e_2)$ and $e_1, e_2$ are the linkage points in $C1$ and $C2$.

**Proposition 1.** *Let* $(X, d)$ *be a metric space,* $\mathcal{L}$ *a pattern language for* $X$*,* $\Delta$ *a binary generalisation operator,* $\Delta^*$ *a pattern binary generalisation operator and* $d_L$ *a linkage distance. For any evidence* $E \subseteq X$*, the conceptual dendrogram* $T$ *resulting from* HDCC($E$, $X$, $d$, $\Delta^*$, $\Delta$, $d_L$) *is equivalent to the traditional dendrogram if the generalisation operators* $\Delta$ *and* $\Delta^*$ *are strongly bounded by* $d_L$.

**Proof**. There are two different cases to consider in *T:* (*a*) the leaves and (*b*) the internal nodes.

*Case* (*a*): In the first step HDCC builds $n$ clusters $\{e\}$ and their corresponding generalisations as $\Delta(e, e)$. Since $\Delta$ is strongly bounded by $d_L$ we have by Definition 5 $\forall\ e', e \in E$*:* if $e' \in Set(\Delta(e, e))$ then $d_L(\{e'\}, \{e\}, d) \leq d_L(\{e\}, \{e\}, d)$. Since $d_L(\{e\}, \{e\}, d) = 0$ and $d_L$ is positive, then $d_L(\{e'\}, \{e\}, d) = 0$. The only element $e'$ that satisfies this is $e' = e$. Therefore, after a pattern is computed no other element can be

added to the cluster by HDCC. Therefore, in this case, $T$ is equivalent to the traditional dendrogram by Definition 3.

*Case* (*b*): In the following steps, each new node $(C, p, l)$ in $T$ is formed by merging in first place the two clusters $(C_1, p_1, l_1)$, $(C_2, p_2, l_2)$ whose linkage distance $l$ is the lowest and $p$ is computed as $\Delta^*(p_1, p_2)$. Since $\Delta^*$ and $\Delta$ are generalisation operators we have that $C \subseteq Set(p)$ and $C_1 \subseteq Set(p_1)$ and $C_2 \subseteq Set(p_2)$.

- If $\Delta^*(p_1, p_2)$ does not cover any other cluster in addition to $C_1$ and $C_2$, we have $C = C_1 \cup C_2$ and $d_L(C_1, C_2, d) = l$.
- Let us suppose there is another child $(C_3, p_3, l_3)$ of $(C, p, l)$ such that $C_3 \subseteq Set(\Delta^*(p_1, p_2)) - (Set(p_1) \cup Set(p_2))$. Since $\Delta^*$ is strongly bounded, by Definition 4 we have $d_L(C_3, C_1, d) \leq d_L(C_1, C_2, d) \vee d_L(C_3, C_2, d) \leq d_L(C_1, C_2, d)$. However, $d_L(C_3, C_1, d)$ must be equal to $d_L(C_1, C_2, d)$ and $d_L(C_3, C_2, d)$ must be equal to $d_L(C_1, C_2, d)$ otherwise HDCC should have merged before $C_1$ and $C_3$ or $C_2$ and $C_3$ than $C_1$ and $C_2$.

Therefore, $d_L(C_i, C_j, d) = l$ for any child $(C_i, p_i, l_i)$, $(C_j, p_j, l_j)$ of $(C, p, l)$ and consequently, in both cases, $T$ is equivalent to the traditional dendrogram by Definition 3. □

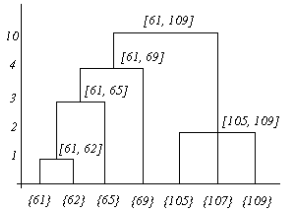


**Fig. 6.** An equivalent conceptual dendrogram.

Fig. 6 shows a simple example of a conceptual dendrogram that is equivalent to the traditional dendrogram under single linkage. $\mathcal{L}$ is the set of the finite closed intervals in $\Re$, and $d$ the absolute difference. $\Delta^*(p_1, p_2)$ is the interval $[min, max]$, where $min$ is the minimum value of the lower bounds of $p_1$ and $p_2$, and $max$ is the maximum of the upper bounds. $\Delta(e_1, e_2)$ is $[min(e_1, e_2), max(e_1, e_2)]$. It is easy to see that (a) $\Delta^*$and (b) $\Delta$ are strongly bounded by $d^s_L$:

- (a) Each generalisation of two patterns $p_1$ and $p_2$ is a new interval $p$ that only covers the elements covered by $p_1$ and $p_2$ and those that are in between of them. If $e_1$ and $e_2$ are the linkage points in $C_1$ and $C_2$ that have determined the single linkage distance $l = d^s_L(C_1, C_2, d)$, the new elements in the interval $p$ must be included into the two balls $B(e_1, l)$ or $B(e_2, l)$, i.e. the intervals $[e_1 - l, e_1 + l]$ or $[e_2 - l, e_2 + l]$. Since $e_1$ and $e_2$ are the closest elements in $C_1$ and $C_2$, we have that $p_1 = [a, e_1]$ and $p_2 = [e_2, b]$ and $p = [a, b]$. Clearly the new elements in $p$, i.e. the elements in the interval $]e_1, e_2[$, are included in $[e_1 - l, e_1 + l]$ and also in $[e_2 - l, e_2 + l]$ because $l = | e_2 - e_1 |$.
- (b) $\Delta$ is strongly bounded by $d^s_L$ too because any element in $Set(\Delta(e_1, e_2))$ will be always in between of $e_1, e_2$. Note that (a) and (b) holds for any $d_L$ here considered.

The condition for having equivalent dendrograms, however, is too strong for many datatypes and generalisation operators given that it forces to minimal generalisations. In fact a pattern generalisation operator $\Delta^*(p_1, p_2)$ whose coverage $Set(\Delta^*(p_1, p_2))$ is equal to $Set(p_1) \cup Set(p_2)$ is strongly bounded because there is no new elements in $\Delta^*(p_1, p_2)$, so the only set $C$ which must satisfy $(d_L(C, C_1, d) \leq d_L(C_1, C_2, d) \vee d_L(C, C_2, d) \leq d_L(C_1, C_2, d))$ is $C = \emptyset$ and since the $d_L$ from a cluster to $\emptyset$ is zero, the condition holds for any $p_1, p_2 \in \mathcal{L}$, $C_1 \subseteq Set(p_1)$, $C_2 \subseteq Set(p_2)$. The same happens with $\Delta$ when $\Delta(e_1, e_2)$ is defined as $\{e_1, e_2\}$.

### 4.2 Order-preserving Dendrograms

Sometimes for a given pair of generalisation operators $\Delta$ and $\Delta^*$, a distance $d$ and a linkage distance $d_L$, the conceptual dendrogram –although not equivalent to the traditional one– can just preserve the order in which clusters are merged by $d_L$, i.e. a discovered pattern will never cover a farther cluster leaving out a closer one. In that case, we say that the conceptual dendrogram is *order-preserving*.

More specifically, an order-preserving conceptual dendrogram is one where for any node $(C, p, l)$ in the hierarchy, its children are linked at the same distance $l$ or they are linked by the pattern at a linkage distance lower than the linkage distance from any other cluster in the hierarchy not covered by the pattern. This concept is formalised by Definition 6.

**Definition 6.** *Let* $(X, d)$ *be a metric space and T the tree resulting from* HDCC. *T is* order-preserving *iff* $\forall (C, p, l), (C_i, p_i, l_i) \in T, \exists (C_j, p_j, l_j) \in T$ *with* $(C_i, p_i, l_i)$ *and* $(C_j, p_j, l_j)$ *children of* $(C, p, l)$ *such that* $d_L(C_i, C_j, d) = l \vee (d_L(C_i, C_j, d) < d_L(C', C_i, d) \wedge d_L(C_i, C_j, d) < d_L(C', C_j, d))$, *for all* $(C', p', l') \in T$, $C' \not\subset Set(p)$.

To obtain an order-preserving conceptual dendrogram, any time HDCC merges two clusters $C_1$ and $C_2$ with patterns $p_1$ and $p_2$, any other cluster $C$ covered by the generalisation of $p_1$ and $p_2$ that has not been linked by the distance $d_L$ must have lower linkage distances to $C_1$ and $C_2$ than the linkage distances to $C_1$ and $C_2$ from any other cluster $C'$ not covered by the pattern. This is formalized by the property we call *weak boundedness* and that is given by Definition 7. Analogously, Definition 8 establishes the same property for binary generalisation operators.

**Definition 7.** *Let* $(X, d)$ *be a metric space*, $\mathcal{L}$ *a pattern language and* $d_L$ *a linkage distance. A pattern binary generalisation operator* $\Delta^*$ *is* weakly bounded by $d_L$ *iff* $\forall p_1, p_2 \in \mathcal{L}$, $C_1 \subseteq Set(p_1)$, $C_2 \subseteq Set(p_2)$, $C \subseteq Set(\Delta^*(p_1, p_2)) - (Set(p_1) \cup Set(p_2))$, $C' \not\subset Set(\Delta^*(p_1, p_2)) : (d_L(C, C_1, d) \leq d_L(C_1, C_2, d) \vee d_L(C, C_2, d) \leq d_L(C_1, C_2, d)) \vee (d_L(C, C_1, d) < d_L(C', C_1, d) \wedge d_L(C, C_2, d) < d_L(C', C_2, d))$.

**Definition 8.** *Let* $(X, d)$ *be a metric space*, $\mathcal{L}$ *a pattern language and* $d_L$ *a linkage distance. A binary generalisation operator* $\Delta$ *is* weakly bounded by $d_L$ *iff* $\forall e, e', e_1, e_2 \in X$: if $e \in Set(\Delta(e_1, e_2))$ and $e' \notin Set(\Delta(e_1, e_2))$ *then* $d_L(\{e\}, \{e_1\}, d) \leq d_L(\{e_1\},\{e_2\}, d) \vee d_L(\{e\}, \{e_2\}, d) \leq d_L(\{e_1\},\{e_2\}, d) \vee ((d_L(\{e\}, \{e_1\}, d) < d_L(\{e'\},\{e_1\}, d) \wedge d_L(\{e\}, \{e_2\}, d) < d_L(\{e'\},\{e_2\}, d))$.

**Proposition 2.** *Let* $(X, d)$ *be a metric space*, $\mathcal{L}$ *a pattern language*, $d_L$ *a linkage distance*, $\Delta$ *a binary generalisation operator*, *and* $\Delta^*$ *a pattern binary generalisation operator.*

*(a) If* $\Delta^*$ *is strongly bounded by* $d_L$ *then* $\Delta^*$ *is weakly bounded by* $d_L$.

*(b) If* $\Delta$ *is strongly bounded by* $d_L$ *then* $\Delta$ *is weakly bounded by* $d_L$.

**Proof.** Part (a) of Proposition 2 follows immediately from definitions of strongly and weakly bounded operators. Any pattern generalisation operator that is strongly bounded by the linkage distance is also weakly bounded given that Definition 7 relaxes the condition in Definition 4. The same holds for part (b) since Definition 8 relaxes the condition in Definition 5. □

As before, we want to show that the weakly bounded property is a sufficient condition to preserve the order.

**Proposition 3.** Let $(X, d)$ be a metric space, $\mathcal{L}$ a pattern language, $\Delta$ a binary generalisation operator, $\Delta^*$ a pattern binary generalisation operator and $d_L$ a linkage distance. For any evidence $E \subseteq X$, the conceptual dendrogram $T$ resulting from HDCC($E, X, d, \Delta^*, \Delta, d_L$) is order-preserving if the generalisation operators $\Delta$ and $\Delta^*$ are weakly bounded by $d_L$.

**Proof.** There are two different cases to consider in *T:* (*a*) the leaves and (*b*) the internal nodes.

*Case* (*a*)*:* In the first step HDCC builds $n$ nodes $(\{e\}, \Delta(e, e), l)$ with $l = 0$. If $\Delta(e, e)$ covers any other element this is merged to $\{e\}$.

Since $\Delta$ is weakly bounded by $d_L$ we have by Definition 8 $\forall\ e, e', e_1 \in E$: if $e \in Set(\Delta(e_1, e_1))$ and $e' \notin Set(\Delta(e_1, e_1))$ then $d_L(\{e\}, \{e_1\}, d) \leq d_L(\{e_1\},\{e_1\}, d) \vee (d_L(\{e\}, \{e_1\}, d) < d_L(\{e'\},\{e_1\}, d)$. Since $d_L(\{e_1\}, \{e_1\}, d) = 0$ and $d_L$ is positive we have $\forall\ e, e', e_1 \in E$: if $e \in Set(\Delta(e_1, e_1))$ and $e' \notin Set(\Delta(e_1, e_1))$ then $d_L(\{e\},\{e_1\}, d) = 0 \vee d_L(\{e\},\{e_1\}, d) < d_L(\{e'\},\{e_1\}, d)$. Therefore, $T$ is order-preserving by Definition 6.

*Case* (*b*)*:* In the following steps, each node $(C, p, l)$ in $T$ is formed by merging (in first place) the two clusters $(C_1, p_1, l_1)$, $(C_2, p_2, l_2)$ whose linkage distance $l$ is the lowest and $p$ is computed as $\Delta^*(p_1, p_2)$. Since $\Delta^*$ and $\Delta$ are generalisation operators we have that $C \subseteq Set(p)$ and $C_1 \subseteq Set(p_1)$ and $C_2 \subseteq Set(p_2)$.

- If $\Delta^*(p_1, p_2)$ does not cover any other cluster different to $C_1$ and $C_2$, we have $C = C_1 \cup C_2$ and $d_L(C_1, C_2, d) = l$.
- Let us suppose there is another child $(C_3, p_3, l_3)$ of $(C, p, l)$ such that $C_3 \subseteq Set(\Delta^*(p_1, p_2)) - (Set(p_1) \cup Set(p_2))$. Since $\Delta^*$ is weakly bounded, by Definition 7 we have $d_L(C_3, C_1, d) \leq d_L(C_1, C_2, d) \vee d_L(C_3, C_2, d) \leq d_L(C_1, C_2, d) \vee (d_L(C_3, C_1, d) < d_L(C', C_1, d) \wedge d_L(C_3, C_2, d) < d_L(C', C_2, d))$ for all $C' \not\subset Set(\Delta^*(p_1, p_2))$. By reasoning as in Proposition 1 we have, $d_L(C_3, C_1, d) = d_L(C_3, C_2, d) = d_L(C_1, C_2, d) = l \vee (d_L(C_3, C_1, d) < d_L(C', C_1, d) \wedge d_L(C_3, C_2, d) < d_L(C', C_2, d))$.

Therefore, in both cases, $T$ is order-preserving by Definition 6. □

The conceptual dendrogram of Fig. 3 (Right) is not order-preserving under the single linkage. $\Delta^*$ is not weakly bounded by $d^s_L$ since the pattern $aa*$ has linked first the cluster $\{aabbbbbb\}$, which is farther from $\{aa\}$ and $\{aab\}$ than $\{abb\}$.

Fig. 7 shows an example of an order-preserving dendrogram for nominal data using $d^s_L$. We have used a distance similar to the one defined in [14], a distance induced by a relationship $R$, where $R$ is a partial order. $R$ is defined as $xRy$ if $x$ is a $y$. Fig. 7 (Left) shows part of a relationship $R$ as a tree hierarchy. The distance between two elements is the sum of costs associated to each edge of the shortest path connecting them. The cost of an edge of level $i$ is $w_i = 1/2^i$. $\Delta(e_1,e_2)$ is defined as the minimun ancestor of $e_1$ and $e_2$ if $e_1 \neq e_2$ otherwise is equal to $e_1$, and $\Delta^*(p_1, p_2)$ is defined anlogously. In Fig. 7 (Top right) we can see the traditional dendrogram, and the corresponding conceptual dendrogram in Fig. 7 (Bottom right). The evidence is formed only by elements in the leaves of $R$. The internal nodes are generalisations.

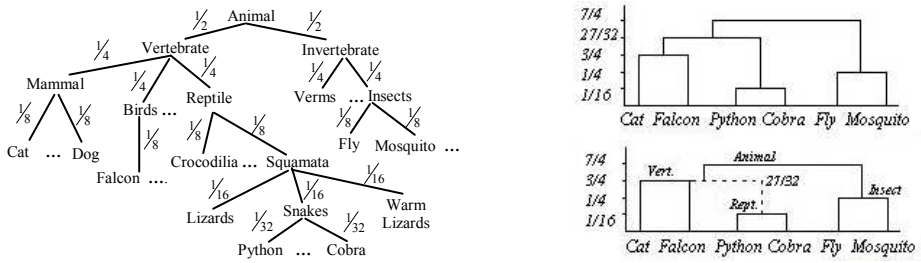


**Fig. 7. (Left)** Relationship $R$ as a tree**. (Top right)** Traditional dendrogram. **(Bottom right)** Not equivalent but order-preserving conceptual dendrogram.

Note that a pattern generalisation operator that covers all the space (the maximal operator $\Delta^*(p_1, p_2) = p$ where $\text{Set}(p) = X$) is trivially weakly bounded because we cannot find a cluster $C$ such that $C \not\subset Set(\Delta^*(p_1, p_2))$.

### 4.3 Acceptable generalisation operators

There are some generalisation operators that although not (weakly) bounded lead to dendrograms which are consistent with the distance in a broader sense. The idea is that a pattern should not cover new elements whose distance to the old elements is greater than the greatest distance between the old elements. We refer to the operators that produce this kind of patterns as *acceptable*. In this case, the dendrograms can differ significantly.

**Definition 9.** *Let* $(X, d)$ *be a metric space,* $\mathcal{L}$ *a pattern language and* $d^c_L(\ ,\ ,\ ,\ )$ *the complete linkage distance. A pattern binary generalisation operator* $\Delta^*$ *is* acceptable *iff* $\forall\ p_1, p_2 \in \mathcal{L}, e \in Set(\Delta^*(_1, p_2)), \exists\ e' \in Set(p_1) \cup Set(p_2) : d(e, e') \leq d^c_L(Set(p_1),Set(p_2), d)$.

In Fig. 8 (Left) the union of the circles whose centres are in $Set(p_1) \cup Set(p_2)$ and radius equal to $d^c_L(Set(p_1), Set(p_2), d)$ determines the maximum coverage for a pattern produced by an acceptable generalisation operator for the evidence $\{a, b, c, d\}$ in $\Re^2$.

Note that a pattern generalisation operator is acceptable independently of the linkage distance used. It only depends on the distance $d$ between the two most distant

elements in the clusters. We use $d^c_L$ in the definition to simplify the notation. Definition 10 gives the same concept applied to binary generalisation operators.

**Definition 10.** *Let* $(X, d)$ *be a metric space*, $\mathcal{L}$ *a pattern language*. *A binary generalisation operator* $\Delta$ *is* acceptable *iff* $\forall\ e, e_1, e_2 \in X$ : *if* $e \in Set(\Delta(e_1, e_2))$ *then* $d(e, e_1) \leq d(e_1, e_2)$ *or* $d(e, e_2) \leq d(e_1, e_2)$.

We can see from Definition 10 and Definition 5 that any binary generalisation operator $\Delta$ is acceptable if and only if it is strongly bounded by the linkage distance since the linkage distance $d_L$ is equal to $d$ when applied to single sets for any of the linkage distances here considered.

The pattern binary generalisation operator $\Delta^*$ used in Fig. 6 is acceptable, since all the elements in $p$ will always fall between the bounds of $p_1$ and $p_2$ and consequently at a distance lower than the two most distant elements in $p_1$ and $p_2$. $\Delta$ is also acceptable because, as we showed for the example of Fig. 6, it is strongly bounded by the linkage distance.

The good thing is that $\Delta(e, e) = \{e\}$ is strongly bounded, and hence acceptable. This operator can usually be expressed in most $\mathcal{L}$. So, only $\Delta^*$ must be analysed in most cases and, additionally, it is independent from $d_L$. Results obtained for acceptability will be then extensible to whatever linkage function.

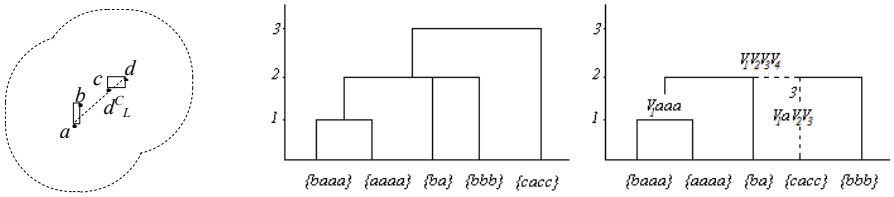


**Fig. 8. (Left)** The coverage of an acceptable $\Delta^*$ for the evidence $\{a, b, c, d\}$ in $\Re^2$. **(Center)** Traditional dendrogram. **(Right)** Conceptual dendrogram.

Fig. 8 (Center) shows the traditional dendrogram for the evidence {*baaa*, *aaaa*, *ba*, *bbb*, *cacc*} and Fig. 8 (Right) the corresponding conceptual dendrogram, both using $d^s_L$. The metric space is $(\Sigma^*, d)$, where $\Sigma^*$ is the space of states of lists formed from $\Sigma$ included the empty list $\lambda$, $\Sigma = \{a, b, c\}$ and $d$ is the edit distance. The pattern language $\mathcal{L}$ is given by all the finite lists from the alphabet $\Sigma' = \Sigma \cup V \cup \{\lambda\}$ where $V = \{V_1, V_2, \ldots, V_n\}$ is a set of variables. The variables are used to generalise symbols in $\Sigma \cup \{\lambda\}$. The generalisation of two lists is given by $\Delta(l_1, l_2) = p$ where $p$ is formed by the patterns given by the optimal alignments of $l_1$ and $l_2$ and whose length is given by the length of the longest pattern $l_1$ or $l_2$. Variables represent the symbols that do not match. For instance, $\Delta(aabaaa, ababaa) = aV_1V_2V_3aa$. $\Delta^*(p_1, p_2)$ is computed analogously, e.g. $\Delta^*(aV_1V_2V_3aa, baa) = V_1V_2V_3V_4aa$. Although $\Delta$ and $\Delta^*$ are not bounded under the single linkage distance, they are acceptable given that each pattern covers elements whose distances are at most the number of variables in the pattern and this is precisely the maximum distance possible between two elements in $Set(p)$, in particular to the elements in $Set(p_1) \cup Set(p_2)$.

## 5 Experimental Results

One question that arises from the previous proposal is whether the new conceptual clustering, coming from the on-line re-arrangement of the dendrogram might undermine cluster quality (in the cases where the dendrograms are not equivalent, naturally). In order to bring some light on this, the experiments below compare HDCC against the traditional version of the hierarchical clustering algorithm. We constructed 100 artificial datasets by drawing points from a finite mixture of $k$ Gaussian distributions in $\Re^2$ whose means are randomly located in $[0, 100]^2$ with a standard deviation of 1. Although $k$ represents the actual number of *gaussians* in a dataset, note that there might be overlapping between *gaussians*, so having fewer clusters. We set $k = 3$, and each dataset was formed by 600 points (200 points were drawn from each of the 3 Gaussian distributions). The experiments were conducted under single and complete linkage and using two different language patterns $\mathcal{L}_1$ and $\mathcal{L}_2$. $\mathcal{L}_1$ is the language of axis-parallel rectangles and $\mathcal{L}_2$ is the language of circles.

Fig. 9 shows the discovered patterns in $\mathcal{L}_1$ and $\mathcal{L}_2$ for one dataset with 600 points drawn from three Gaussian distributions, one using $d^s_L$ (Left) and the other using $d^c_L$ (Right). Note that the rectangles obtained incrementally by HDCC fit the points as well as an $n$-ary operator. This is not the case in $\mathcal{L}_2$ where the discovered patterns are more general than using an $n$-ary operator. However, as we can see in Table 2, it does not affect the clustering quality because they are built incrementally and HDCC in each step only merges those clusters that are completely covered by the pattern.

To assess the quality of the clustering we employed a measure, $S$, that reflects the mean scattering over the $k$ clusters (see eq. (1)). The lower $S$ is the better the clustering is. Table 2 shows $S$ averaged over $n$ different experiments. Note that $n$ can take values less than 100 in HDCC since the resulting hierarchies do not always have a clustering of $k$ clusters (several clusters may be joined by a discovered cluster pattern in one step).

$$S = \frac{1}{k}\sum_{i=1}^{k}\sqrt{\sum_{j=1 \wedge l=j+1}^{m}\sum^{m} d(\vec{x_j}, \vec{x_l})^2}\,. \qquad \textbf{(1)}$$

The experiments show that not only quality is not degraded, but for $\mathcal{L}_2$ HDCC sometimes outperforms the traditional algorithm under single linkage. Similar results were obtained with points in $[0, 10]^2$. Logically, different results might be obtained using non-convex or complex-shaped patterns.

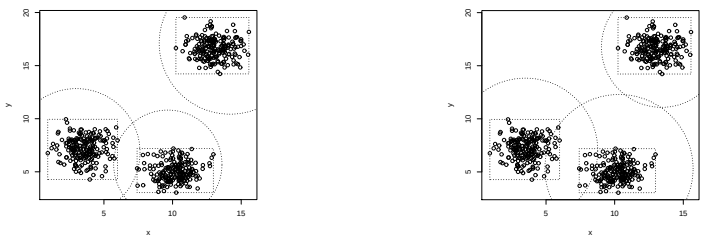


**Fig. 9.** Discovered patterns in $\mathcal{L}_1$ and $\mathcal{L}_2$ using $d^s_L$ **(Left)** and using $d^c_L$ **(Right)**.

**Table 2.** Values of $S$ for traditional and conceptual dendrograms for $k$ =3.

| | | Traditional $S$ $n$ | Conceptual $S$ $n$ | Dendrogram Relation |
|---|---|---|---|---|
| $\mathcal{L}_1$ | Single | 292,293 (100) | 292,295 (100) | Not equivalent |
| | Complete | 281,393 (100) | 281,393 (100) | Equivalent |
| $\mathcal{L}_2$ | Single | 292,293 (100) | 281,549 (98) | Not equivalent |
| | Complete | 281,393 (100) | 281,727 (100) | Not equivalent |

## 6 Conclusions

We have presented a general approach for hierarchical conceptual clustering based on distances and generalisation operators. It puts together the flexibility of hierarchical distance-based clustering and the interpretability of conceptual clustering. For instance, a user can choose any part of a dendrogram, get a description also learning whether all the covered elements are close wrt. the underlying metric.

Several clustering algorithms that generate concept descriptions can be found in the literature. On the one hand we have those coming from traditional conceptual clustering such as CLUSTER/2 [4], COBWEB [7] and GCF [8]. On the other hand we have those that, using a subset of first-order logic as representation language, apply traditional distance-based clustering algorithms. In this second group we can find KBG [9], C 0.5 [10], COLA-2 [11], and TIC [12, 13] among others. Our proposal is different to all the conceptual clustering methods which also use a distance in the way that it is general to any datatype (any generalisation operator and distance can be used). Moreover, we present graphical extensions to see the divergence between the distance and the generalisation operator a posteriori, but also conditions that can be checked a priori to ensure that the resulting conceptual dendrograms are consistent with the underlying distance. Our work is related to [14] where the author analyses the relationship between distances and generalisations and proposes a framework where these two paradigms can be integrated in a consistent way. In [14] the analysis is achieved on generalisation operators defined on a metric space and not over a language of patterns as it is done here.

Additionally, as we have said, HDCC can be seen as an $n$-ary operator constructed over binary operators by only applying the binary operators at most $n$ times, where $n$ is the number of examples. This is an interesting property for machine learning areas which have well-established binary generalisations operators, such as ILP.

The instantiation of HDCC to propositional clustering is direct, when datatypes are nominal or numerical. We have shown in [15] that the common generalisation operators for nominal data (extensional set) and numerical data (intervals) are strongly bounded in the metric spaces defined by the distance functions commonly used for these datatypes (discrete distance and difference distance). Hence in this case the distance-based conceptual dendrograms are equivalent to classical distance-based dendrograms, independently of the linkage distance. The problem is also analysed when the tuple is composed of both nominal and numerical data, and the generalisation operators are extended accordingly. Examples of this have been shown in the experiments section in this paper.

Things are more diverse (and interesting) when applying the proposal to structured datatypes. We have seen several examples in this paper when the conditions hold for the complete linkage but not for the single linkage, or only one of the degrees (the weakest one, acceptability) is met. We are currently working on the establishment of operative combinations of distances and generalisation operators for lists and sets.

## References


1. Jain, A. K., Murty, M. N. and Flynn, P. J., "Data clustering: a review", *ACM Comput. Survey*, Vol. 31, Nº 3, pp. 264-323, (1999).
2. Berkhin, P. "A Survey of Clustering Data Mining Techniques", *Grouping Multidimensional Data*, pp. 25-71, Springer (2006).
3. Michalski, R. S. "Knowledge Acquisition Through Conceptual Clustering: A Theoretical Framework and an Algorithm for Partitioning Data into Conjunctive Concepts", *Policy Analysis and Information Systems*, Vol. 4, Nº 3, pp. 219-244. (1980)
4. Michalski, R. S. and Stepp, R. E. "Learning from Observation: Conceptual Clustering", in Michalski et al (eds.) *Machine Learning: An Artificial Intelligence Approach*, pp. 331-363. TIOGA Publishing Co. (1983)
5. Ramon, J., Bruynooghe, M. and Van Laer, W. Distance measures between atoms. *CompulogNet Meeting ComputingLogic & Machine Learning*, pp. 35–41. (1998).
6. Levenshtein, V. I. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*., 10:707–710. (1966).
7. Fisher, D. "Knowledge Acquisition Via Incremental Conceptual Clustering", *Machine Learning* 2: 139-172, Kluwer Acadenfic Publishers. (1987).
8. Talavera, L. and Béjar, J. Generality-Based Conceptual Clustering with Probabilistic Concepts. *IEEE Transactions on Pattern Analysis & Machine Inteligence* 23, No. 2. (2001).
9. Bisson, G. Conceptual Clustering in a First Order Logic Representation. *European Conference on Artificial Intelligence.* pp. 458-462. (1992).
10. De Raedt, L. and Blockeel, H. Using Logical Decision Trees for Clustering. *Proc. 7th Intl Workshop on ILP*, LNCS 1297, pp 133-140, Springer. (1997).
11. Emde, W. Inductive learning of characteristic concept descriptions. *Proc. 4th Intl Workshop on Inductive Logic Programming (ILP-94)*. (1994).
12. Blockeel, H. and De Raedt, L. Top-down induction of first order logical decision trees. *Artificial Intelligence.* 101: 285-297. (1998).
13. Blockeel, H., De Raedt, L., Ramon, J. Top-down induction of clustering trees. *Proceedings of the 15th Intl. Conference on Machine Learning*, pp. 55–63. (1998).
14. Estruch, V. Bridging the gap between distance and generalisation:Symbolic learning in metric spaces. PhD thesis, DSIC-UPV, http://www.dsic.upv.es/~vestruch/thesis.pdf, (2008).
15. Funes, A., Ferri, C., Hernández-Orallo, J., Ramírez-Quintana, M.J. Technical Report, DSIC, http://www.dsic.upv.es/~flip/#Papers. (2008).