An Instantiation for Sequences of Hierarchical Distance-based Conceptual Clustering

Ana Funes[1], María José Ramírez-Quintana[2], Jose Hernández-Orallo[2], Cèsar Ferri[2]

[1] Universidad Nacional de San Luis, Ejército de los Andes 950, 5700 San Luis, Argentina
afunes@unsl.edu.ar
[2] Universidad Politécnica de Valencia, Camino de Vera s/n, 46022 Valencia, España

Abstract. In this work, we present an instantiation of our framework for Hierarchical Distance-based Conceptual Clustering (HDCC) using sequences, a particular kind of structured data. We analyse the relationship between distances and generalisation operators for sequences in the context of HDCC. HDCC is a general approach to conceptual clustering that extends the traditional algorithm for hierarchical clustering by producing conceptual generalisations of the discovered clusters. Since the approach is general, it allows to combine the flexibility of changing distances for different data types at the same time that we take advantage of the interpretability offered by the obtained concepts, which is central for descriptive data mining tasks. We propose here different generalisation operators for sequences and analyse how they work together with the edit and linkage distances in HDCC. This analysis is carried out based on three different properties for generalisation operators and three different levels of agreement between the clustering hierarchy obtained from the linkage distance and the hierarchy obtained by using generalisation operators.

Keywords: conceptual clustering, distance-based clustering, hierarchical clustering, generalisation, structured data, lists, sequences, edit distance.

1 Introduction

Distance-based methods in machine learning made decisions based on the similarity between cases. Some examples of popular methods based on similarity are the k-nearest neighbours [8] for classification, the k-means clustering algorithm [9], Fisher discriminant [10] and the hierarchical clustering algorithms [11,14,15] in which HDCC [4] is based. Distance-based techniques, although flexible and intuitive, have associated a lack of comprehensibility, i.e. they cannot give an explanation to their answers. For example, a classification by the k-nearest neighbours could recommend certain book as appropriate for a given customer because the k-nearest neighbours of the book were appropriate for the same customer. However, this technique does not provide a pattern or a common description to all these books, which give a better idea of the characteristics they share.

On the other hand, some techniques are based on the idea that a discovered pattern or generalisation from old data can be used to describe new data covered by this pattern. These techniques are known as symbolic. Some well-known symbolic techniques are association rules, decision trees and Michalski's conceptual clustering [11, 12].

An important issue related to the integration of distance-based with symbolic techniques is the existence of a possible inconsistency between the underlying distance and the discovered generalisations. HDCC is an algorithm that integrates both techniques. Based on the agglomerative hierarchical clustering, it constructs a cluster hierarchy by using a distance at the same time that produces a hierarchy of patterns, which results in an extended dendrogram referred as conceptual dendrogram.

A key aspect considered in HDCC is the possibility of determining a priori whether the hierarchy of clusters induced by the underlying distance is consistent with the discovered patterns, i.e. how much the cluster elements covered by a given pattern reproduce the distribution of the elements in the metric space. Accordingly, we have defined in [4] three different levels of consistency between distance and generalisations based on the divergences between the clustering hierarchy obtained from the linkage distance and the hierarchy obtained by using generalisation operators. In [4] we have also given the properties the generalisation operators used by HDCC must meet in order to reach a given consistency level. Therefore, this general framework allows the instantiation of the algorithm for different distances, generalisation operators and data types and to determine a priori how consistent are these distances with the employed generalisation operators.

In the present work, we propose a particular instantiation of the general framework [4] for sequences of elements. In this instantiation we make use of the edit distance between sequences and we propose and analyze two different pairs of generalisation operators under different linkage distances accordingly with the consistency levels given in [4, 6]. We prove that HDCC when instantiated with the edit distance and one of the here proposed pairs of generalisation operators is highly consistent under complete linkage distance, producing conceptual dendrograms equivalent to the dendrograms obtained by the only use of the linkage distance. We also show that when used under single linkage we obtain acceptable conceptual dendrograms according to the consistency levels defined in [4]. These results expand the set of consistent instantiations already found for HDCC (see [7]).

The paper is organised as follows. Due to space limitations, all necessary preliminary concepts about the HDCC approach can be found in [4] and in [6]. In Section 2.1 and Section 2.2, respectively, we recall some necessary concepts, which are used in the instantiation of our framework, and propose a first pair of generalisation operators for HDCC. In Section 2.3, we analyse the consistency between the operators proposed in Section 2.2 and the edit and linkage distances, according to the three levels of consistency between distances and generalisations presented in [4]. In Section 2.4, we give a more suitable pair of operators and show that they satisfy the property of strong boundedness under complete linkage distance as well as the acceptability property under single linkage. Finally, Section 3 closes the paper with the conclusions and future work.

2 An Instantiation for Sequences

In Propositional Learning, evidence is described by means of tuples of numerical and nominal values. However, sometimes it is necessary to use more expressive but complex representations such as sets, sequences, graphs, etc. These representations are referred as structured data.

In the section, we analyse our framework applied to lists, a structured data type whose elements are sequences of data where order matters and repetitions are allowed. We analyse the properties proposed in [4] for different generalisation operators of lists and list patterns.

In the presented instantiation, lists are sequences of symbols from a set of symbols $\Sigma = \{a, b, c, \ldots\}$ called the *alphabet*. The distinguished symbol λ denotes the empty list. We denote by Σ^* the space of lists formed from Σ included the empty list λ. Examples of lists on Σ are *aa, babab, c, acba,* λ.

2.1 The Distance *d*

We can found several distance functions for lists in the literature. Among them, we have *Hamming distance* [1], which can be applied only for equal-length lists. Here the distance between two lists is given by the number of positions where the corresponding symbols are different. For instance, $d(bbab, abab) = 1$.

The most commonly used distance function for sequences is the *edit distance*, also known as *Levenshtein distance* [2], which can be used for variable-length sequences. Here the distance between two sequences is given by the number of operations (insertion, deletion and substitution) required to transform one sequence into another. Different costs can be assigned to each of these operations.

Example 1. Let us assume that the cost of one substitution is 1, equal to the cost of one insertion and to the cost of one deletion. Consequently, the edit distance between sequences $s_1 = aabb$ and $s_2 = cbba$ is 3. That is, s_1 is transformed into s_2 by deleting the first *a*, by replacing the second *a* by a *c* and finally adding one *a* at the end.

There are variations for this distance, e.g. the metric obtained by allowing only additions and deletions but not substitutions. That is, it considers the cost of one substitution as the cost of one deletion plus one insertion. In this case, the resulting distance between the sequences s_1 and s_2 of Example 1 is 4. In this work, we adopt the first assumption, i.e. the cost of one substitution is 1, equal to the cost of one insertion or the cost of one deletion.

2.2 The Language of Patterns and the Generalisation Operators

In this section, we analyse the application of the edit distance *d* together with two pairs of generalisation operators, which are defined on a same pattern language $\mathcal{L}_0$. Patterns in the considered language $\mathcal{L}_0$ are sequences built from the extended alphabet $\Sigma' = \Sigma \cup V \cup \{\lambda\}$ where Σ is the set of symbols from which sequences are defined and $V = \{V_1, V_2, V_3, \ldots\}$ is a set of variables. A same variable cannot appear twice in a pattern. Each variable in a pattern represents a symbol from $\Sigma \cup \{\lambda\}$. Examples of patterns on $\mathcal{L}_0$ are aaV_1b, V_1V_2a. [1]

The first binary generalisation operator[2] Δ we analyse is based on the concept of pattern associated to an alignment, given in [5]. We propose here a pattern binary

[1] For the sake of simplicity, sometimes we will use V^n to denote the sequence of variables $V_1V_{2\ldots}V_n$. For instance, V^2a in place of V_1V_2a.

[2] For the definition of *binary generalisation operator*, see [4].

generalisation operator[3] Δ^* that is a natural extension for patterns of Δ. We also analyse their consistency with respect to the edit and linkage distances.

In order to propose formally the first pair of operators Δ and Δ^*, we present the previous concepts of optimal alignment and pattern associated to an alignment. Firstly, we formalize in Definition 1 the concept of *alignment*.

Definition 1. Given two sequences s_1 and s_2 in Σ^*, an *alignment on* $\Sigma^*$of s_1 and s_2 is given by the mapping M: $\Sigma^* \times \Sigma^* \rightarrow \mathbb{N}^* \times \mathbb{N}^*$ defined by $M(s_1, s_2) = ((i_{11},i_{12},...,i_{1n}), (i_{21},i_{22},...,i_{2n}))$ such that

i) $s_1(i_{1j}) = s_2(i_{2j})$, for all $j = 1.. n$.

ii) $i_{1j} < i_{1j+1}$ and $i_{2j} < i_{2j+1}$, for all $j = 1.. n\text{-}1$

Remark 1. Note in Definition 1 that $0 \leq n \leq min\{|s_1|, |s_2|\}$. $n = 0$ corresponds to the empty alignment $M = ((),())$ that is obtained when there is no matching between both sequences. $n = |s_1| = |s_2|$ corresponds to the alignment M when $s_1 = s_2$.

Example 2. Let us suppose we want to generalise the sequences s_1=*aabaaa* and s_2=*ababaa.* One possible alignment between them is

```
a   a b a a a
a b a b a a
```

that we denote as $M(s_1, s_2) = ((1, 2, 3, 4, 5), (1, 3, 4, 5, 6))$.

Other valid alignments for s_2 and s_2 are, among others, $M'(s_1, s_2) = ((2, 3, 4, 5, 6), (1, 2, 3, 5, 6))$ and $M'' = ((1, 2), (3, 5))$

```
           M'                                     M''
    a a b a   a a                           a   a   b a a a
      a b a b a a                       a b a b a a
```

Although these three alignments are valid, we are interested in optimal alignments. In the example, only M and M' are optimal. An optimal alignments is one alignment where the sequence formed by the symbols of s_1 (or s_2) pointed by its respective indexes in the alignment constitute a *longest common subsequence* (*lcs*)[4].

Example 3. Let s_1, s_2, M, M', M'' be the sequences and alignments given in Example 2. According to M we have[5] $s_1(1).s_1(2).s_1(3).s_1(4).s_1(5) = s_2(1).s_2(3).s_2(4).s_2(5).s_2(6) =$ *aabaa* and according to M' $s_1(2).s_1(3).s_1(4).s_1(5).s_1(6) = s_2(1).s_2(2).s_2(3).s_2(5).s_2(6) =$ *abaaa*. Since *aabaa* and *abaaa* are *lcs* for s_1 and s_2, M and M' are optimal alignments. Although M'' is a valid alignment it is not an optimal alignment since $s_1(1).s_1(2) = s_2(3).s_2(5) =$ *aa* and *aa* is not a *lcs* of s_1 and s_2.

The concept of optimal alignment is formalized by Definition 2.

Definition 2. Let s_1 and s_2 be two elements in Σ^* and $M(s_1, s_2)$=$((i_{11}, i_{12},..., i_{1n}), (i_{21},i_{22},...,i_{2n}))$ an alignment of s_1 with s_2. M is an *optimal alignment on* Σ^* iff $s_1(i_{11}).s_1(i_{12}).\ldots.s_1(i_{1n})$ is a *lcs* of s_1 and s_2.

Given that more than one optimal alignment can be obtained from two sequences s_1 and s_2, and we are interested in obtaining only one optimal alignment, a total order $\prec$ over the set of optimal alignments in Σ^* is defined in order to specify the optimal alignment we want.

Definition 3. Given two sequences s_1 and s_2 and the optimal alignments $M =$

[3] For the definition of *pattern binary generalisation operator,* see [4].

[4] A *longest common subsequence* (*lcs*) is given by the longest (not necessarily contiguous) subsequence of s_1 and s_2. Note that the *lcs* is not unique.

[5] "**.**" denotes the concatenation operator between sequences.

$((a_{11},a_{12},...,a_{1n}),\ (a_{21},a_{22},...,a_{2n}))$ and $N = ((b_{11},b_{12},...,b_{1n}),\ (b_{21},b_{22},...,b_{2n}))$ of s_1 and s_2, we say that $M \prec N$ iff $(a_{11},a_{12},...,a_{1n},a_{21},a_{22},...,a_{2n}) <_{\text{Lex}} (b_{11},b_{12},...,b_{1n},b_{21},b_{22},...,b_{2n})$ where $<_{\text{Lex}}$ is the lexicographical order between sequences.

Remark 2. Note that Definition 3 applies not only for sequences in Σ^* but also for sequences in $\mathcal{L}_0$ (patterns).

Example 4. Following with the previous example, we have that $M \prec M'$ given that (1, 2, 3, 4, 5, 1, 3, 4, 5, 6) $<_{\text{Lex}}$ (2, 3, 4, 5, 6, 1, 2, 3, 5, 6).

Every alignment between two sequences s_1 and s_2 induces a pattern p in $\mathcal{L}_0$, which covers[6] both s_1 and s_2. This pattern is unique and it is called the *pattern associated to an alignment on* Σ^*.

Definition 4. Let s_1 and s_2 be two sequences in Σ^*, $M = ((i_{11},i_{12},...,i_{1n}),\ (i_{21},i_{22},...,i_{2n}))$ an alignment on Σ^* of s_1 and s_2, and s the sequence of symbols $s_1(i_{11}).s_1(i_{12}).\ \ldots\ .s_1(i_{1n})$.
$p \in \mathcal{L}_0$ is the *pattern associated to the alignment on* Σ^* M iff

i) The concatenation of the ground symbols in p is equal to s.

ii) The variable symbols in p are distributed as follows:

- The number of variables in the pattern p before the first ground symbol is equal to $(i_{11} - 1) + (i_{21} - 1)$.
- The number of variables in p between any pair of adjacent ground symbols $s(j)$ and $s\ (j+1)$, with $j=1..n-1$, is equal to $(i_{1(j+1)} - i_{1j} - 1) + (i_{2(j+1)} - i_{2j} - 1)$.
- The number of variables after the last ground symbol in p is equal to $|s_1| - i_{1n} + |s_2| - i_{2n}$.

Remark **3.** If M is the empty alignment then $p = V_1V_2\ldots\ V_{|s_1|+|s_2|}$

Example 5. The following table illustrates the concept of pattern associated to an alignment on Σ^*.

s_1	s_2	M	p
aabaaa	*ababaa*	((1,2,3,4,5),(1,3,4,5,6))	aV_1abaaV_2
aabaaa	*ababaa*	((2,3,4,5,6),(1,2,3,5,6))	V_1abaV_2aa
aaab	*baa*	((4),(1))	$V_1V_2V_3bV_4V_5$
aaa	*bb*	((),())	$V_1V_2V_3V_4V_5$

We use the concept of *pattern associated to an alignment on* Σ^* to define the generalisation operator Δ for sequences in Σ^* that it is given in Proposition 1.

Proposition 1. Let Σ^* the set of all sequences of ground symbols in Σ, and $\mathcal{L}_0$ the pattern language defined over $\Sigma \cup V \cup \{\lambda\}$. The function $\Delta\colon \Sigma^* \times \Sigma^* \rightarrow \mathcal{L}_0$ defined by $\Delta(s_1,\ s_2) = p_M$, where p_M is the pattern associated to the minimum ($\prec$) optimal alignment M between s_1 and s_2 is a binary generalisation operator for sequences.

Proof. By definition 1 in [4], $\Delta\colon \Sigma^* \times \Sigma^* \rightarrow \mathcal{L}_0$ is a binary generalisation operator iff for all $s_1 \in \Sigma^*$, $s_2 \in \Sigma^*$, $s_1 \in Set(p)$ and $s_2 \in Set(p)$, with $p = \Delta(s_1, s_2)$.

By definition of coverage, $Set(p) = \{s \in \Sigma^* |\ \exists\ \sigma : p\sigma = s\}$ and given that p is the

[6] We say that a sequence s is covered by a pattern p in $\mathcal{L}_0$ if exists a substitution σ such that $s = p\sigma$.
A substitution σ is a set of pairs V_i/e_i, with $V_i \in V$ ($V_i \neq V_j$, $i \neq j$) and $e_i \in \Sigma \cup \{\lambda\}$, that applied to a pattern p returns a new pattern p' obtained from p by simultaneously replacing each occurrence of the variable V_i in p by e_i (i=1, …,n).
Example. Sequences s_1=*aabaa* and s_2=*ababaa* are covered by pattern $p = aV_1abaaV_2$ given that exists substitutions $\sigma_1 = \{V_1/\lambda,\ V_2/\lambda\}$ and $\sigma_2 = \{V_1/b,\ V_2/\lambda\}$ such that $p\sigma_1 = s_1$ and $p\sigma_2 = s_2$.

pattern associated to the minimum ($\prec$) optimal alignment M between s_1 and s_2 we can build a substitution σ_1 from the alignment M, that revert the process of building the pattern associated to the alignment, such that $p\sigma_1 = s_1$ as follows: For each variable V in p, if V is the generalisation of a ground symbols e_1 in s_1 that do not much any symbol in s_2, then the pair V/e_1 must be added to σ_1, otherwise V/λ. The same process can be done for s_2.□

We define the pattern binary generalisation operator Δ^* by analogy with Δ. We take into account when defining the alignment on $\mathcal{L}_0$ that symbols in pattern p_1 match those in pattern p_2 both when they are equal ground symbols or when one of them is a variable. We illustrate this concept in Example 6 and formalize it in Definition 5.

Example 6. Given the patterns $p1 = aV_1V_2V_3aa$ and $p2 = baa$, some valid alignments on $\mathcal{L}_0$ of p_1 and p_2 are $M_1 = ((2, 3, 4), (1, 2, 3))$; $M_2 = ((2, 5, 6), (1, 2, 3))$; $M_3 = ((1, 5),(2, 3))$ and $M_4 = ((4, 5, 6),(1, 2, 3))$.

$M_1 = ((2, 3, 4), (1, 2, 3))$	$M_2 = ((2, 5, 6), (1, 2, 3))$	$M_3 = ((1, 5),(2, 3))$	$M_4 = ((4, 5, 6),(1, 2, 3))$
$aV_1V_2V_3aa$	$aV_1V_2V_3aa$	$aV_1V_2V_3aa$	$aV_1V_2V_3aa$
$\ \ b\ \ a\ \ a$	$\ \ b\qquad aa$	$b\ a\qquad a$	$\qquad\quad b\ aa$

Definition 5 Given two patterns p_1 and p_2 in $\mathcal{L}_0$, an *alignment on* $\mathcal{L}_0$ of p_1 and p_2 is given by the mapping M: $\mathcal{L}_0 \times \mathcal{L}_0 \rightarrow \mathbb{N}^* \times \mathbb{N}^*$ defined by $M(p_1, p_2) = ((i_{11}, i_{12},..., i_{1n}), (i_{21}, i_{22},..., i_{2n}))$ such that

i) $p_1(i_{1j})$ and $p_2(i_{2j})$ are equal ground symbols or at least one of them is a variable, with $j=1.. n$.

ii) $i_{1j} < i_{1j+1}$ and $i_{2j} < i_{2j+1}$, with $j = 1.. n-1$.

Once again, we are interested in minimum ($\prec$) optimal alignments, but this time on $\mathcal{L}_0$. Therefore, we need formally define the concepts of optimal alignment on $\mathcal{L}_0$ and pattern associated to an alignment on $\mathcal{L}_0$, which is based on the concept of *binary generalisation operator* $\Delta_{\Sigma'}$ of symbols in Σ' given in Proposition 2.

Proposition 2. Let Σ' the set of symbols in $\Sigma \cup V \cup \{\lambda\}$.
The function $\Delta_{\Sigma'}$:$\Sigma' \times \Sigma' \rightarrow \Sigma'$ defined by

$$\Delta_{\Sigma'}(s_1 , s_2) = \begin{cases} s_{1,} & when\ s_1, s_2 \in \Sigma\, and\ s_1 = s_2 \\ V_1, & otherwise \end{cases}$$

is a binary generalisation operator for symbols in Σ'.

Proof. $\Delta_{\Sigma'}$ is a binary generalisation operator given that for any pair of symbols s_1 and s_2 in Σ'

(a) If $s_1, s_2 \in \Sigma$ and $s_1 = s_2$ then, by definition of $\Delta_{\Sigma'}$ we have that $\Delta_{\Sigma'}(s_1, s_2) = p = s_1$ and trivially $p \in Set(p)$.

(b) Otherwise, by definition of $\Delta_{\Sigma'}$ we have that $\Delta_{\Sigma'}(s_1, s_2) = p=V_1$ and given that $Set(V) = \Sigma'$ for any variable V and $s_1, s_2 \in \Sigma'$, then $s_1 \in Set(V_1)$ and $s_2 \in Set(V_1)$.□

Definition 6. Let p_1 and p_2 two patterns in $\mathcal{L}_0$ and $M(p_1, p_2)=((a_{11},a_{12},...,a_{1n}), (a_{21},a_{22},...,a_{2n}))$ an alignment on $\mathcal{L}_0$ of p_1 and p_2. M is an *optimal alignment on* $\mathcal{L}_0$ iff it does not exist an alignment $N(p_1, p_2) = ((b_{11},b_{12},...,b_{1m}), (b_{21},b_{22},...,b_{2m}))$ such that $m>n$.

Example 7. Let $p1 = aV_1V_2V_3aa$ and $p2 = baa$ and M_1, M_2, M_3 and M_4 the alignments of Example 6. All are optimal alignments, with the exception of M_3 whose length is 2.

Definition 7. Let p_1 and p_2 be two patterns in $\mathcal{L}_0$, $M = ((i_{11},i_{12},...,i_{1n}), (i_{21},i_{22},...,i_{2n}))$ an alignment on $\mathcal{L}_0$ of p_1 and p_2, and $p' = \Delta_{\Sigma'}(p_1(i_{11}), p_2(i_{21}))$**.** $\Delta_{\Sigma'}(p_1(i_{12}), p_2(i_{22}))$**.** … **.**$\Delta_{\Sigma'}(p_1(i_{1n}), p_2(i_{2n}))$, $p \in \mathcal{L}_0$ is the *pattern associated to the alignment on* $\mathcal{L}_0$ M iff the symbols in p are distributed as follows:

- The number of variables in p before the first symbol $p'(1)$ is equal to $(i_{11}-1)+(i_{21}-1)$.
- The number of variables in p between any pair of adjacent symbols $p'(j)$ and $p'(j+1)$, with $j=1..n-1$, is equal to $(i_{1(j+1)} - i_{1j} - 1) + (i_{2(j+1)} - i_{2j} - 1)$.
- The number of variables in p after the last symbol $p'(n)$ is equal to $|p_1|-i_{1n} + |p_2|- i_{2n}$.

Remark 4. If M is the empty alignment then $p= V_1V_2\ldots\ V_{|p_1|+|p_2|}$

Example 8. Following with Example 5, we have that the patterns p_{M_1}, $p_{M_2}$and p_{M_4} in $\mathcal{L}_0$ associated to the alignments M_1, M_2 and M_4 are equal to V^4aa, while $p_{M_3} = VaV^3aV$.

Proposition 3. Let $\mathcal{L}_0$ the pattern language defined over $\Sigma \cup V \cup \{\lambda\}$. The function $\Delta^*: \mathcal{L}_0 \times \mathcal{L}_0 \rightarrow \mathcal{L}_0$ defined by $\Delta^*(p_1, p_2) = p_M$, where p_M is the pattern associated to the minimum ($\prec$) optimal alignment M on $\mathcal{L}_0$ between p_1 and p_2 is a pattern binary generalisation operator.

Proof. By definition 2 in [4], $\Delta^*: \mathcal{L}_0 \times \mathcal{L}_0 \rightarrow$ is a pattern binary generalisation operator iff for all $p_1 \in \mathcal{L}_0, p_2 \in \mathcal{L}_0$: $Set(p_1) \subseteq Set(p)$ and $Set(p_2) \subseteq Set(p)$, with $p=\Delta^*(p_1, p_2)$.

The coverage of a pattern is given by the set of sequences that are covered by the pattern. Note that a pattern in $\mathcal{L}_0$ not only generalize sequences of ground symbols but also other less general patterns, i.e. $Set(p) = \{p' \in \mathcal{L}_0 \mid \exists\ \sigma' : p\sigma'=p'\}$. Given that p is the pattern associated to the minimum ($\prec$) optimal alignment M on $\mathcal{L}_0$ between p_1 and p_2 we can build a substitutions σ'_1 such that $p\sigma'_1=p_1$ as follows:

For each variable V in p,

- If V is the generalisation of a variable in p_1 and any other symbol in p_2 then V/V.
- If V is the generalisation of two different grounds symbols e_1 and e_2 in p_1 and p_2 respectively, then V/e_1 .

By following the same reasoning we can prove that $p_2\in\ Set(p)$□

2.3 Analysis of Consistency between Distances and Generalisations

Figure 1(b) shows a simple example of an application of HDCC for lists using the edit distance, the single linkage distance d^s_L and the generalisation operators given in Proposition 1 and Proposition 3. The evidence in the example is given by the set of lists E = {*baaa*, *aaaa*, *ba*, *bbb*, *caccc*}. In Figure 1(a), we can also see the corresponding traditional dendrogram for the same evidence.

From the example, it follows that this first pair of generalisation operators do not satisfy any of the consistency levels proposed in [4] when applied under single linkage distance d^s_L and the edit distance, namely:

(a) The conceptual and the traditional dendrograms are not equivalent and therefore, we can affirm by Proposition 1 in [6] that either Δ^* or Δ is not strongly bounded by the single linkage distance d^s_L. In fact, as we show below in point (c), Δ is not strongly bounded by the single linkage distance d^s_L and Δ^* either given that, for instance, d^s_L({*baaa*, *aaaa*}, {*ba*}) = 2 and its pattern VaV^3 covers cluster {*caccc*} whose single linkage distance to {*baaa*, *aaaa*, *ba*} is 4.

(b) Δ^* is not weakly bounded by d^s_L given that {*caccc*} is linked to {*baaa*, *aaaa*, *ba*} by its pattern VaV^3 before than {*bbb*} that is not covered by VaV^3 and whose linkage distance to the cluster {*baaa*, *aaaa*, *ba*} is 2 < d^s_L({*baaa*, *aaaa*, *ba*},{*caccc*}, d) = 4.

(c) Δ^* is not acceptable either given that the greatest distance between any pair of elements in cluster {*baaa*, *aaaa*, *ba*} is 3 and its pattern VaV^3 covers {*caccc*} whose minimum distance to {*baaa*, *aaaa*, *ba*} is 4.

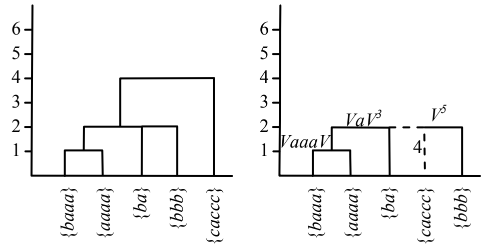

Figure 1. (a) Traditional dendrogram using d^s_L. **(b)** Conceptual dendrogram using d^s_L.

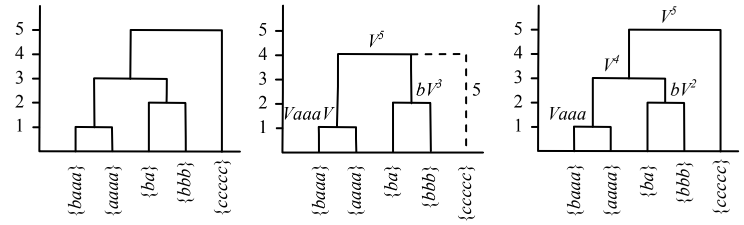

Figure 2. (a) Traditional dendrogram using complete linkage distance d^c_L. **(b)** Conceptual dendrogram using d^c_L. **(c)** Conceptual dendrogram using d^c_L and less general patterns.

In Figure 2(b) we show the conceptual dendrogram for the evidence *E*={*aaaa*, *baaa*, *bbb*, *ba*, *ccccc*} using the same generalisation operators but under complete linkage distance d^c_L. Figure 2(a) depicts the corresponding traditional dendrogram. As we can see from this example, these operators are not consistent with the edit and d^c_L either. One problem with these generalisation operators comes from the number of variables in the resulting patterns. These patterns have a number of variables grater or equal to the maximum edit distance between the generalized elements and we need operators that return patterns whose number of variables is equal to the maximum edit distance to guarantee the strongest level of the consistency: the strong boundedness. Figure 2 (c) shows the conceptual dendrogram we get when applying this idea to the resulting patterns. Next section presents a new pair of generalisation operators based on this notion.

2.4 A Pair of Strongly Bounded Generalisation Operators

In this section, we propose a new pair of generalisation operators Δ^* and Δ that satisfy the property of strong boundedness given in [4] and [6]. These operators produce patterns having a number of variables equal to the maximum edit distance between the generalized elements, guarantying under complete linkage distance d^c_L the strong boundedness property. Accordingly, we need to redefine the concept of pattern associated to an alignment in Σ^* and in $\mathcal{L}_0$ in order to make the number of variables equal to the edit distance.

Definition 8. Let s_1 and s_2 be two sequences in Σ^*, $M = ((i_{11},i_{12},\ldots,i_{1n}), (i_{21},i_{22},\ldots,i_{2n}))$ an alignment on Σ^* of s_1 and s_2, and s the sequence of symbols $s_1(i_{11}).s_1(i_{12}). \ldots .s_1(i_{1n})$.

$p \in £_0$ is the *pattern associated to the alignment on* Σ^* M iff
i) The concatenation of the ground symbols in p is equal to s.
ii) The variable symbols in p are distributed as follows:

- The number of variables in the pattern p before the first ground symbol is equal to $max\{(i_{11} - 1); (i_{21} - 1)\}$.
- The number of variables in p between any pair of adjacent ground symbols $s(j)$ and $s\ (j+1)$, with $j=1..n-1$, is equal to $max\{(i_{1(j+1)} - i_{1j} - 1); (i_{2(j+1)} - i_{2j} - 1)\}$.
- The number of variables after the last ground symbol in p is equal to $max\{|s_1| - i_{1n}; |s_2| - i_{2n}\}$.

Informally, the generalisation of two sequences s_1, s_2 in Σ^* is the pattern associated to the minimum optimal alignment on Σ^* of s_1 and s_2 according to a total order on optimal alignments that considers the number of variables the patterns associated to the alignment have and then their lexicographical order. In this total order, the minimum is given by the optimal alignment that has the less number of variables or if there is more than one then the minimum is given by the lexicographical order. We formalize this concept in Definition 9 and, in Proposition 4, we formally propose the corresponding binary generalisation operator Δ.

Definition 9. Let s_1 and s_2 be two sequences, $M = ((a_{11},a_{12},...,a_{1n}), (a_{21},a_{22},...,a_{2n}))$ and $N = ((b_{11},b_{12},...,b_{1n}), (b_{21},b_{22},...,b_{2n}))$ two optimal alignments of s_1 and s_2, and p_M and p_N the patterns associated to the alignments M and N, respectively.
$M \prec_V N$ iff $\#\text{var}(p_M) < \#\text{var}(p_N)$ or ($\#\text{var}(p_M) = \#\text{var}(p_N)$ and $(a_{11}, a_{12},..., a_{1n}, a_{21}, a_{22},..., a_{2n}) <_{\text{Lex}} (b_{11}, b_{12},..., b_{1n}, b_{21}, b_{22},..., b_{2n})$), where $\#\text{var}(p)$ is the number of variables in pattern p and $<_{\text{Lex}}$ is the lexicographical order between sequences.

Remark 5. Definition 9 applies for sequences in Σ^* and in $£_0$ (patterns).

Example 9. Given the sequences $s_1= baaa$ and $s_2 = aaaa$ and the optimal alignments M=((2,3,4),(1,2,3)) and M'= ((2,3,4),(2,3,4)) and their associated patterns $p_M = VaaaV$ and $p_{M'} = Vaaa$. According to Definition 9, we have that $M' \prec_V M$.

Proposition 4. Let Σ^* the set of all sequences of ground symbols in Σ, and $£_0$ the pattern language defined over $\Sigma \cup V \cup \{\lambda\}$. The function $\Delta: \Sigma^* \times \Sigma^* \rightarrow £_0$ defined by $\Delta(s_1, s_2) = p_M$, where p_M is the pattern associated to the minimum ($\prec_V$) optimal alignment M between s_1 and s_2 is a binary generalisation operator for sequences.

Proof. By following the same reasoning that in Proposition 1.□

Next, we propose the new pattern binary generalisation operator Δ^* defined by analogy with Δ. Consequently, we also need to redefine the concept of pattern associated to an alignment on $£_0$. This is done in Definition 10.

Definition 10. Let p_1 and p_2 be two patterns in $£_0$, $M = ((i_{11},i_{12},...,i_{1n}), (i_{21},i_{22},...,i_{2n}))$ an alignment on $£_0$ of p_1 and p_2, and $p' = \Delta_{\Sigma'}(p_1(i_{11}), p_2(i_{21}))\text{.}\ \Delta_{\Sigma'}(p_1(i_{12}), p_2(i_{22}))\text{.}\ \ldots\ \text{.}\Delta_{\Sigma'}(p_1(i_{1n}), p_2(i_{2n}))$, $p \in £_0$ is the *pattern associated to the alignment on* $£_0$ M iff the symbols in p are distributed as follows:

- The number of variables in p before the first symbol $p'(1)$ is equal to $max\{(i_{11}-1);(i_{21}-1)\}$.
- The number of variables in p between any pair of adjacent symbols $p'(j)$ and $p'(j+1)$, with $j=1..n-1$, is equal to $max\{\ (i_{1(j+1)} - i_{1j} - 1); (i_{2(j+1)} - i_{2j} - 1)\}$.
- The number of variables in p after the last symbol $p'(n)$ is equal to $max\{|p_1|-i_{1n}\ ; |p_2|- i_{2n}\ \}$.

Proposition 5. Let $\mathcal{L}_0$ the pattern language defined over $\Sigma \cup V \cup \{\lambda\}$. The function $\Delta^*: \mathcal{L}_0 \times \mathcal{L}_0 \to \mathcal{L}_0$ defined by $\Delta^*(p_1, p_2) = p_M$, with p_M the pattern associated to the minimum ($\prec_V$) optimal alignment M between p_1 and p_2 is a pattern binary generalisation operator.

Proof. By following the same reasoning that in Proposition 3.□

Proposition 6 shows that Δ is a strongly (and weakly) bounded binary generalisation operator.□

Proposition 6. Let $\mathcal{L}_0$ the pattern language defined over $\Sigma \cup V \cup \{\lambda\}$, Δ: $\Sigma^* \times \Sigma^* \to \mathcal{L}_0$ the binary generalisation operator given in Proposition 4, and d the edit distance.

(a) Δ is strongly bounded by distance d.

(b) Δ is weakly bounded by distance d.

Proof. (a) Given two sequences s_1 and s_2 with edit distance $d(s_1, s_2)$, we want to show that any sequence s_3 covered by the pattern $p = \Delta(s_1, s_2)$ is at most at distance $d(s_1, s_2)$ from s_1 and s_2.

We know that (i) By definition of Δ, the edit distance $d(s_1, s_2)$ determines the number of variables v in a pattern $p = \Delta(s_1, s_2)$, so $d(s_1, s_2) = v$. (ii) Any element covered by p can differ from other in at most v symbols, i.e their edit distance must be less or equal to v. Given that s_1, s_2 and s_3 are covered by p we have $d(s_3, s_1) \leq v$ and $d(s_3, s_2) \leq v$.

From (i) and (ii), $d(s_3, s_1) \leq d(s_1, s_2)$ and $d(s_3, s_2) \leq d(s_1, s_2)$.

(b) Given that by (a) Δ is strongly bounded by d, by part (ii) of Proposition 2 in [6] we have that Δ is weakly bounded by d.□

Proposition 7. Let $\mathcal{L}_0$ be a language of patterns defined over $\Sigma \cup V \cup \{\lambda\}$, d the edit distance, $d^c_L(., ., .)$ the complete linkage distance and Δ^*: $\mathcal{L}_0 \times \mathcal{L}_0 \to \mathcal{L}_0$ the pattern binary generalisation operator given in Proposition 5.

(a) Δ^* is strongly bounded by the complete linkage distance d^c_L.

(b) Δ^* is weakly bounded by the complete linkage distance d^c_L.

(c) Δ^* is an acceptable pattern binary generalisation operator.

Proof. (a) We want to show $\forall\, p_1, p_2 \in \mathcal{L}_0, C_1 \subseteq Set(p_1), C_2 \subseteq Set(p_2), C \subseteq Set(\Delta^*(p_1, p_2)) - (Set(p_1) \cup Set(p_2)) : d^c_L(C,C_1,d) \leq d^c_L(C_1,C_2,d) \vee d^c_L(C,C_2,d) \leq d^c_L(C_1,C_2,d)$.

Let us suppose that exists a cluster $C \subseteq Set(\Delta^*(p_1, p_2)) - (Set(p_1) \cup Set(p_2)) : d^c_L(C, C_1, d) > d^c_L(C_1, C_2, d) \wedge d^c_L(C, C_2, d) > d^c_L(C_1, C_2, d)$ with s_1 and s_2 the linkage points between C_1 and C_2. It means that exists $x \in C$ and $y \in C_1$ such that $d(x, y) > d(s_1, s_2)$. Given that s_1 and s_2 are the complete linkage points, $d(s_1, s_2)$ is the grater distance between any pair of points in C_1 and C_2, and we know also that $d(s_1, s_2)$ = number of variables in $\Delta^*(p_1, p_2)$. Since x and y are covered by $\Delta^*(p_1, p_2)$, the distance d between them is bounded by the number of variables in $\Delta^*(p_1, p_2)$, that is $d(x, y) \leq d(s_1, s_2)$.

(b) Given that by (a) Δ^* is strongly bounded by d^c_L, by part (i) of Proposition 2 in [6] Δ^* is weakly bounded by d^c_L.

(c) We want to show that for any pair of patterns p_1 and p_2 in $\mathcal{L}_0$ and for any sequence s in $Set(\Delta^*(p_1, p_2))$ exists a sequence s' in $Set(p_1) \cup Set(p_2)$ such that $d(s, s') \leq d^c_L(Set(p_1), Set(p_2), d)$.

The edit distance d between any element covered by a pattern p=$\Delta^*(p_1, p_2)$ is less or equal to the number v of variables in p. Since s and s' are covered by p then $d(s, s') \leq v$. By definition of Δ^*, the maximum distance between the elements in $Set(p_1)$ and

$Set(p_2)$ determines the number of variables in p, then $d^c_L(Set(p_1), Set(p_2), d) = v$. Consequently, $d(s, s') \leq d^c_L(Set(p_1), Set(p_2), d)$.□

Although Δ^* is strongly bounded by the complete linkage distance d^c_L, it is not strongly bounded by the single linkage distance d^s_L. We illustrate this in Figure 3 by showing that the traditional (Left) and conceptual (Right) dendrograms for the evidence {*baaa*, *aaaa*, *ba*, *bbb*, *cacc*} are not equivalent. Given that Δ is strongly bounded by d then, by Proposition 1 in [6], Δ^* cannot be strongly bounded by d^s_L.

In Figure 3 we can also see that Δ^* is not weakly bounded by d^s_L either given that pattern VaV^2 covers {*cacc*} whose single linkage distance to {*baaa*, *aaaa*, *ba*} = 3 is grater than 2, the single linkage distance from {*baaa*, *aaaa*, *ba*} to cluster {*bbb*}, which is not covered by the pattern VaV^2.

We want to remark that although these operators are not bounded by the single linkage distance d^s_L, we always get conceptual dendrograms that satisfy the property of acceptability (see page 54 in [6]). In fact, as it has been proved in part (c) of Proposition 7 and in part (a) of Proposition 6, Δ^* is an acceptable generalisation operator for any linkage distance with the edit distance d and Δ is strongly bounded by d satisfying in this way the sufficient conditions for getting acceptable conceptual dendrograms in HDCC.

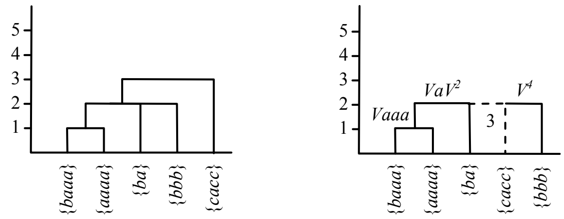

Figure 3. Traditional (Left) and Conceptual dendrogram (Right) using d^s_L.

3 Conclusions and Future Work

It can be easily shown that when integrating traditional hierarchical distance-based clustering with conceptual clustering, the conceptual dendrograms obtained by applying generalisation operators can differ significantly from the hierarchy induced only by the distance. Having in mind this problem, the notion of conceptual dendrogram and three different levels of consistency have been defined on the basis of the similarity between a conceptual dendrogram and its corresponding traditional dendrogram. At the same time the sufficient conditions the used generalisation operators Δ^* and Δ must satisfy to obtain a given level of consistency have been also defined. This has given place to a general framework that allows the analysis of different pairs of generalisation operators, which can result compatible with the distances at some degree while some other pairs cannot, showing, therefore, that some distances and generalisation operators should not be used together.

In this sense, we have found and presented here a positive result for a particular kind of structured data type: sequences of elements. We have proposed a pair of generalisation operators Δ and Δ^* that, when using the most common distance for sequences, i.e. the edit distance, and under complete linkage, they meet the higher level of consistency with respect to the underlying distances, giving place in HDCC to conceptual dendrograms equivalent to the traditional ones. It is also important to note that, although this result does not hold under single linkage, we have shown that the proposed generalisation operators produce acceptable dendrograms under single linkage.

From these results, we can affirm that the integration of hierarchical distance-based clustering and conceptual clustering for sequences of any kind of elements is feasible, congruent and relatively straightforward. At the same time, we have increased our set of consistent generalisation operators for several datatypes, namely numeric and nominal data and tuples of numeric and nominal data that have been proposed in [7].

In this regard, we plan to find new operative pairs of distances and generalisation operators for other data types used in data mining applications, such as sets, graphs and multimedia objects. Part of our immediate future work is also directed to do some experiments to determine the quality of the resulting clustering under single linkage and see if the new conceptual clustering, coming from the on-line re-arrangement of the dendrogram, although not equivalent to the traditional dendrogram does not undermine cluster quality when applied under single linkage.

References

1. R. W. Hamming. Error detecting and error correcting codes. Bell System Technical Journal., 26(2):147-160, (1950)
2. Levenshtein, V. I. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*., 10:707–710. (1966)
4. Funes, A., Ferri, C., Hernández-Orallo, J., Ramirez-Quintana, M.J.: Hierarchical Distance-based Conceptual Clustering. LNAI 5212, pp. 349–364. ©Springer (2008)
5. Estruch, V.: Bridging the gap between distance and generalisation: Symbolic learning in metric spaces. PhD thesis, DSIC-UPV (2008) http://www.dsic.upv.es/~vestruch/thesis.pdf
6. Funes, A.: Agrupamiento Conceptual Jerárquico Basado en Distancias, Definición e Instanciación para el Caso Proposicional. Master Thesis, DSIC-UPV (2008).
7. Funes, A., Ferri, C. , Hernández-Orallo, J., Ramírez-Quintana, M. J.: An Instantiation of Hierarchical Distance-based Conceptual Clustering for Propositional Learning. LNAI 5476, pp. 637–646, 2009. Springer-Verlag Berlin Heidelberg (2009)
8. Cover, T. and Hart, P.: Nearest neighbour pattern classification, in IEEE Transactions on Information Theory, pp. 13–27. (1967)
9. MacQueen, J. B.: Some methods for classification and analysis of multivariate observations, Proc. of the 5th Berkeley Symposium on Math. Statistics and Probability, pp. 281–297, Univ. of California Press. (1967)
10. Fisher, R.: The use of multiple measurements in taxonomic problems, in Ann. Eurgenics, Vol. 7, Part II, pp. 179–188. (1936)
11 Johnson, S. C.: Hierarchical clustering schemes, Psychometrika, Vol. 2, pp. 241–254. (1987)
12 Michalski, R. S.: Knowledge acquisition through conceptual clustering, in Policy Analysis and Information Systems, Vol. 4, pp. 219–244. (1980)
13 Michalski, R. S. and Stepp, R. E.: Machine Learning: An Artificial Intelligence Approach, Learning from Observation: Conc. Clustering, pp. 331–363, TIOGA Publishing Co. (1983)
14. 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).
15. Berkhin, P. “A Survey of Clustering Data Mining Techniques”, *Grouping Multidimensional Data*, pp. 25-71, Springer (2006).