Pattern Recognition 43 (2010) 2439–2455

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

An incremental nested partition method for data clustering

Jyrko Correa-Morris [a,*], Dustin L. Espinosa-Isidrón [b], Denis R. Álvarez-Nadiozhin [a]

[a] Mathematic Department, Faculty of Mathematic and Computer Sciences, Havana University, Cuba
[b] Pattern Recognition Department, Advanced Technologies Application Center, Havana, Cuba

ARTICLE INFO

Article history:
Received 2 October 2008
Received in revised form
11 December 2009
Accepted 27 January 2010

Keywords:
Nested partition
Data clustering
Incremental clustering

ABSTRACT

Clustering methods are a powerful tool for discovering patterns in a given data set through an organization of data into subsets of objects that share common features. Motivated by the independent use of some different partitions criteria and the theoretical and empirical analysis of some of its properties, in this paper, we introduce an incremental nested partition method which combines these partitions criteria for finding the inner structure of static and dynamic datasets. For this, we proved that there are relationships of nesting between partitions obtained, respectively, from these partition criteria, and besides that the sensitivity when a new object arrives to the dataset is rigorously studied. Our algorithm exploits all of these mathematical properties for obtaining the hierarchy of clusterings. Moreover, we realize a theoretical and experimental comparative study of our method with classical hierarchical clustering methods such as single-link and complete-link and other more recently introduced methods. The experimental results over databases of UCI repository and the AFP and TDT2 news collections show the usefulness and capability of our method to reveal different levels of information hidden in datasets.

1. Introduction

The development of technology and computing has enabled the processing of large datasets and every day it becomes more necessary to have tools to carry out this task. The exploratory analysis of the data, looking for an underlying structure that allows to manipulate it more efficiently and effectively, is often an obligatory task. In this regard, data clustering is a powerful tool. The clustering approach can be divided into two main groups: non-hierarchical or partitional and hierarchical [1]. The non-hierarchical approach produces only one partition of data whereas the hierarchical approach produces a sequence of nested partition of data. The k-means algorithm [2], expectation maximization algorithm [3], based on graph theory algorithms such as β_0-connected components and β_0-compact sets [4], among others, are examples of non-hierarchical algorithms; whereas single-link algorithm [5,6], complete-link algorithm [7,6] and commute time for grouping [8], are some examples of hierarchical clustering algorithms. The number of clustering algorithms are reported in literature is large. However, neither a clustering algorithm nor a list of clustering algorithms exist that are capable of discovering the subjacent structure in any given data collection. Due to this, and to the little information with which in most occasions we count about the characteristics and generic properties of these methods is that becomes very difficult to choose one of them when we want to classify objects in a real given context. This problematic topic is referred to in literature as: user dilemma [1]. Besides that the clustering results, as several other pattern recognition tasks, can be affected by the data representation [9], the manner in which similarity between the objects is measured [9], assumptions made about the shape and the size of the clusters [10,11], and so on. Due to these reasons, data clustering is an ill-posed problem, and any prior knowledge about the data and the clustering algorithms could be decisive for achieving success in the development of this task [12].

In this paper, we focus on the hierarchical approach. "A hierarchical clustering method is a procedure for transforming a proximity matrix into a sequence of nested partitions" [13]. The hierarchical clustering algorithms [14–16] have a greater importance since they provide several data-views at different levels of abstraction. However, aside from the previously mentioned issues, there are two disadvantages in the majority of the traditional hierarchical methods [17]. Firstly, let us note that in the majority of these methods obtaining a specific hierarchy level is conditioned by all of the previous levels. Secondly, to obtain each hierarchic level, a sole clustering criterion is used. These two aspects, in many cases, reduce the functionality of these hierarchical methods and limit its applications. On [18] the authors declare some of the deficiencies of the hierarchical methods to face regionalization problems, which are linked to the previously mentioned aspects. We particularly have various reasons to claim that these aspects are really disadvantageous for

* Corresponding author.
E-mail addresses: jyrkoc@gmail.com (J. Correa-Morris), despinosa@cenatav.co.cu (D.L. Espinosa-Isidrón), nadiozhin@gmail.com (D.R. Álvarez-Nadiozhin).

0031-3203/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2010.01.019

these methods. The first aspect is entirely related with efficiency, in the sense that it relates to computational costs and the algorithm execution time. If, in a given situation, we were interested in obtaining a specific level of hierarchy which can be obtained independently, we can optimize the problem resolution. On the other hand, the second aspect is more related with the efficacy of the method. In order to illustrate the idea we have, we need to first answer the following question: What is the role of the similarity function, and what is the role of the clustering criterion in the process of unsupervised classification? The measurement of similarity is responsible for quantifying the "alikeness" between the objects by looking at the features that the specialist in the area considers as determinant. On its part, the grouping criterion is in charge of utilizing the measurement of similarity to discover certain common properties in sub-collections of objects that are differentiated from the rest and give place to the formation of clusters. As such, if one criterion is utilized to obtain each one of the levels of the hierarchy, then the interpretation of two different levels of the hierarchy is limited to the measurement of similarity. Since only one clustering criterion is utilized, to obtain two levels of the hierarchy one must increase or decrease the thresholds of similarity to consider; whereas, if various criteria are used along with the thresholds of similarity, we have the information that gives us each one of the corresponding levels of criteria. Further, if the relationships between the criteria are known, then we are able to obtain more diverse information of the relations between the different levels. Needless to say, utilizing different levels permits us to better explore the measurement of similarity searching for links between the objects within the same level of the hierarchy; further, it permits us to explore the inner level relations. Because of these reasons, we begin to think of algorithms whose fundamental objective is to obtain a sequence of nested partitions that will also incorporate the functions mentioned above. Although these algorithms are a particular case of hierarchical algorithms, we will refer to them as nested partition algorithms to reassure that different criteria can be utilized to obtain the hierarchy as well as each individual level within the hierarchy can be obtain independently.

Analogous methods have been reported in literature for optimization issues [19–21]. The main goal of *nested partition methods for optimization* problems is to accelerate the search for the global optimum. With this aim, the properties of the target functions and the feasible region are used in order to focus the greatest computation effort on those regions which there are higher possibilities of global optimum is. Those methods have been used in data mining and pattern recognition problems [22–25]. In [22] the nested partition methods were used for variable selection problems, whereas in [23,24], are applications for texture analysis and speaker recognition, respectively. In [25] a study of common aspects between data mining and operation research is done. What is common to all these applications of nested partition methods for optimization in data mining and pattern recognition tasks is that all these problems have to be conceived as explicit optimization problems.

In this article we propose a nested partition algorithm to solve problems of unsupervised classification. Given that we keep in mind to apply it to document analysis, such as news and polls, in which the databases have a large quantity of data and updates occur frequently, we have decided to present an incremental version of this algorithm. Our method is based upon different clustering criteria, which have been previously alluded in literature [4], as well as utilized as the basis for the development of various incremental algorithms [26–28]. The principal predecessor of this work can be found in [17] in which a particular case of the discussed algorithm is exposed, can be considered the root of this methodology. With the intention of clarifying the general properties of each one of these criteria, as well as the relationship that exist within each other, we conducted a study whose results we presented in form of lemmas, propositions, and theorems. Not only does this method formalize the results, but it also allows understanding, in detail, the function of the algorithm and what is behind every step of it. In our opinion, this can help in deciding whether or not it is convenient to use in a determined problem.

In addition to the Introduction, this paper is organized into six sections as follows. In Section 2 some definitions and basic notions are presented. Section 3 is dedicated to the discussion of the main property and relationships of the clustering criteria on which our method is based. The study of the sensitivity of these criteria to the addition of new objects to the dataset is made in Section 4. In Section 5 the algorithm is detailed and its pseudocode is exposed. The experimental results with several dataset of different nature are presented and discussed in Section 6. The last section is devoted to the concluding remarks.

2. Similarity spaces

Suppose that $\mathcal{U}'$ is a set of real objects (data universe) and through a mathematic modeling process is obtained a set $\mathcal{U}$ of object descriptions in terms of a feature set $R=\{f_1,f_2,\ldots,f_n\}$. That is, there is an operator $I:\mathcal{U}'\to\mathcal{U}$ which associates to each object O its description $x(O)$ in terms of features set R; being $x(O)$ the mathematic entity which represents the real object O. This process of mathematic modeling is very important in every task of pattern recognition.

Once the objects are represented, we have to find a manner to measure the similarity between the objects (similarity function). Formally, let $\mathcal{S}\subseteq\mathcal{U}'$ be an object sample set and $\mathcal{X}\subseteq\mathcal{U}$ is the set of its representations, then a function $\Gamma:\mathcal{X}\times\mathcal{X}\to L$ is called a similarity (dissimilarity) function if and only if Γ satisfies the following conditions:

1. L is a field (see [29]);
2. there is a total order relation $\leq$ defined on L which is compatible with the field structure of L (usually $L=\mathbb{R}$ and $\leq$ is the less than relation);
3. $Range(\Gamma)\subseteq L$ has a least element m and a greatest element M;
4. $x=y\Rightarrow\Gamma(x,y)=M(=m)$.

Besides, if $\forall x,y\in\mathcal{X}$, $\Gamma(x,y)=\Gamma(y,x)$ it says that Γ is a symmetric similarity function. The pair $(\mathcal{X},\Gamma)$ is called *similarity* (*dissimilarity*) *space* and $\mathcal{S}$ is called the *support* of $(\mathcal{X},\Gamma)$. In this paper, we only consider similarity spaces $(\mathcal{X},\Gamma)$ such that the similarity function Γ is symmetric.

For each similarity (dissimilarity) function a $\beta_0\in L$ must exist such that if $x,y\in\mathcal{X}$ and $\Gamma(x,y)\geq\beta_0(\leq\beta_0)$, then x and y are very similar objects and reciprocally.

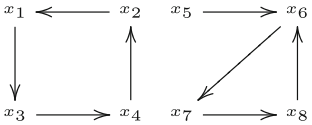
3. Clustering criteria: a nested partition

Given a graph $G=(V,E)$ we mean by path a sequence of vertexes $x_1, x_2,\ldots, x_m$ such that for every i from 1 to $m-1$ it has $(x_i,x_{i+1})\in E$. Henceforth, we use the following notations:

- If G is a directed graph (the edges have an orientation), the arrow $(x,y)\in E$ is denoted by $\overrightarrow{xy}$ and $x\to y$ meaning that exists an arrow from x to y. Moreover, the set of directed paths is denoted by $DP(G)$. An element of $DP(G)$ connecting the elements x and y is denoted by p, and $o(p)$, $d(p)$ denoting the origin and destiny of p, respectively.

A strongly connected component $G'=(V_{G'},E_{G'})$ in a directed graph $G=(V,E)$ is a subgraph of G such that $V_{G'}\neq\emptyset$ and for all $x,y\in V_{G'}$ there are directed paths $p,p'\in DP(G)$ with $o(p)=d(p')=x$, $o(p')=d(p)=y$.

Example 1. Let us consider the graph G as follows:

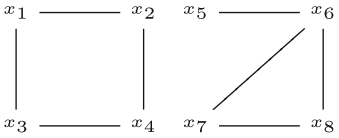


The strongly connected components of the graph G are the subgraphs $G_1=(V_{G_1},E_{G_1})$ with $V_{G_1}=\{x_1,x_2,x_3,x_4\}$ and $E_{G_1}=\{\overrightarrow{x_1x_3},\overrightarrow{x_3x_4},\overrightarrow{x_4x_2},\overrightarrow{x_2x_1}\}$, $G_2=(V_{G_2},E_{G_2})$ with $V_{G_2}=\{x_6,x_7,x_8\}$, $E_{G_2}=\{\overrightarrow{x_7x_8},\overrightarrow{x_8x_6},\overrightarrow{x_6x_7}\}$, and $G_3=(V_{G_3},E_{G_3})$ where $V_{G_3}=\{x_5\}$ and $E_{G_3}=\emptyset$. Observe that there is a directed path from x_5 to x_6, x_7 and x_8, respectively; however, there is not a direct path from any of these vertexes to x_5. Hence, $\{x_5\}$ is an unitary strongly connected component of the graph G.

- If G is a non-directed graph, the edge $(x,y)\in E$ is denoted by xy and $x\sim y$ meaning that x and y are adjacent. Besides that the set of paths is denoted by $P(G)$. An element of $P(G)$ connecting the elements x and y is denoted by $p(x,y)$. Observe that, in this case, the edges and therefore, the paths do not have a determined origin and destiny, is only important the existence of an edge or a path connecting the vertexes.

A connected component in a non-directed graph G is a subgraph $G'=(V_{G'},E_{G'})$ such that $V_{G'}\neq\emptyset$ and for all $x,y\in V_{G'}$ there is path $p\in P(G)$ connecting x and y.

Example 2. Let us consider the graph G of Example 1 removing the orientation of the arrows. Thus,



The graph G has two connected components given by the subgraphs $G_1=(V_{G_1},E_{G_1})$ where $V_{G_1}=\{x_1,x_2,x_3,x_4\}$ $E_{G_1}=\{x_1x_3, x_3x_4,x_4x_2,x_2x_1\}$, and $G_2=(V_{G_2},E_{G_2})$ with $V_{G_2}=\{x_5,x_6,x_7,x_8\}$, $E_{G_2}=\{x_7x_8,x_8x_6,x_6x_7,x_5x_6\}$.

Suppose that $(\mathcal{X},\Gamma)$ is a similarity space. By a β_0-*similarity graph* $G_{(\mathcal{X},\Gamma,\beta_0)}=(\mathcal{X},E_{(\mathcal{X},\Gamma,\beta_0)})$ we understand an undirected graph whose node set is $\mathcal{X}$ and $E_{(\mathcal{X},\Gamma,\beta_0)}=\{x_ix_j\in\mathcal{X}\times\mathcal{X}/\Gamma(x_i,x_j)\geq\beta_0\}$.

The directed graph $G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}=(\mathcal{X},E^{\max}_{(\mathcal{X},\Gamma,\beta_0)})$ with

$$E^{\max}_{(\mathcal{X},\Gamma,\beta_0)}=\left\{\overrightarrow{x_ix_j}\in\mathcal{X}\times\mathcal{X}/\max_{\substack{x_t\in\mathcal{X}\\ x_t\neq x_i}}\Gamma(x_i,x_t)=\Gamma(x_i,x_j)\geq\beta_0\right\}$$

is called *maximum* β_0-*similarity graph*. Further, $\overline{G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}}$ denotes the graph obtained from $G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}$ removing the arrow orientation.

Next, we shall see the notion of clustering function and the specific clustering functions that we utilize as the basis for the development of the algorithm proposed in this paper.

Definition 1. A parametric clustering function on $\mathcal{X}$ is a function

$$c:D\subseteq\mathcal{S}_{\mathcal{X}}\times\Omega_1\times\Omega_2\times\cdots\times\Omega_s\to\mathbb{P}_{\mathcal{X}},$$

where the sets Ω_i, $i=1,2,\ldots,s$ are sets of parameters, $\mathcal{S}_{\mathcal{X}}$ is the set of all possible similarity functions on $\mathcal{X}$ and $\mathbb{P}_{\mathcal{X}}$ is the set of all possible partitions of $\mathcal{X}$.

Definition 1 says that once that a similarity function Γ on a dataset $\mathcal{X}$ and a proper parameters of clustering function c are fixed, then c yields a partition of $\mathcal{X}$. From so on, $\Omega_1=\bigcup\{L/\exists\Gamma\in\mathcal{S}_{\mathcal{X}},\Gamma:\mathcal{X}\times\mathcal{X}\to L\}$. In this paper, we consider the following four clustering functions:

- The clustering function $c_1:D\subseteq\mathcal{S}_{\mathcal{X}}\times\Omega_1\to\mathbb{P}_{\mathcal{X}}$ which associates to each similarity function Γ on $\mathcal{X}$ and to each similarity threshold β_0 the partition $\mathcal{P}_1^{\beta_0}$ whose elements are the connected components of $G_{(\mathcal{X},\Gamma,\beta_0)}$. This clustering criterion assures that in the cluster of an object x there are also all objects whose similarity value with x is greater than the threshold β_0.
- The clustering function $c_2:D\subseteq\mathcal{S}_{\mathcal{X}}\times\Omega_1\to\mathbb{P}_{\mathcal{X}}$ which associates to each similarity function Γ on $\mathcal{X}$ and to each similarity threshold β_0 the partition $\mathcal{P}_2^{\beta_0}$ whose elements are the connected components of $\overline{G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}}$. This clustering criterion guarantees that given an object x then the cluster of x contains those elements $y\in\mathcal{X}$ *satisfying at least one* of the following conditions: (i) y is an object more similar to x or (ii) x is an object more similar to y.
- The clustering function $c_3:D\subseteq\mathcal{S}_{\mathcal{X}}\times\Omega_1\to\mathbb{P}_{\mathcal{X}}$ which associates to each similarity function Γ on $\mathcal{X}$ and to each similarity threshold β_0 the partition $\mathcal{P}_3^{\beta_0}$ whose elements are the strongly connected components of $G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}$. This clustering criterion ensures that given an object x then the cluster of x contains those elements $y\in\mathcal{X}$ *satisfying simultaneously the two* following conditions: (i) y is an object more similar to x or (ii) x is an object more similar to y.
- The clustering function $c_4:D\subseteq\mathcal{S}_{\mathcal{X}}\times\Omega_1\to\mathbb{P}_{\mathcal{X}}$ which associates to each similarity function Γ on $\mathcal{X}$ and to each similarity threshold β_0 the partition $\mathcal{P}_4^{\beta_0}$ whose elements are the completely connected components of $G_{(\mathcal{X},\Gamma,\beta_0)}$ in the sense of complete-link method.

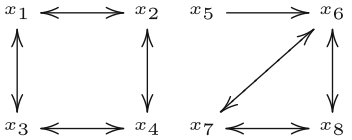
The strongly connected components of $G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}$ are characterized by the following results.

Lemma 1. *For all cycle* $x_{i_1},x_{i_2},\ldots,x_{i_q},x_{i_1}$ *in* $G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}$, $x_{i_1},x_{i_q},x_{i_{q-1}},\ldots,x_{i_1}$ *is also a cycle in* $G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}$.

Proof. Since for j from 1 to $q-1$, $x_{i_j}\to x_{i_{j+1}}$ and $x_{i_q}\to x_{i_1}$, we have that $\Gamma(x_{i_1},x_{i_2})\leq\Gamma(x_{i_2},x_{i_3})\leq\cdots\leq\Gamma(x_{i_q},x_{i_1})\leq\Gamma(x_{i_1},x_{i_2})$. Hence $\Gamma(x_{i_1},x_{i_2})=\Gamma(x_{i_2},x_{i_3})=\cdots=\Gamma(x_{i_q},x_{i_1})$. □

Corollary 1. *Let* N *be a strongly connected component of* $G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}$ *and* $x,x'\in N$ *such that* $x\to x'$, *then* $x'\to x$.

In order to get a better comprehension to the previous results about the strongly connected components of a maximum similarity graph, let us consider the graph G of Example 1. In this graph, the sequence of vertexes $x_1\to x_3\to x_4\to x_2\to x_1$ is an oriented cycle in G. If G is a maximum similarity graph associated to the similarity function Γ and the similarity threshold β_0, then, by definition of these graphs, since $x_3\to x_4$, for all $x\in V$, $\Gamma(x_3,x_4)\geq\Gamma(x_3,x)$; in particular $\Gamma(x_3,x_4)\geq\Gamma(x_3,x_1)$. Analogously, since $x_4\to x_2$ then, for all $x\in V$, $\Gamma(x_4,x_2)\geq\Gamma(x_4,x)$; and therefore, $\Gamma(x_4,x_2)\geq\Gamma(x_4,x_3)\geq\Gamma(x_3,x_1)$. By using the same argument for x_2 and x_1, we obtain that $\Gamma(x_1,x_3)\leq\Gamma(x_3,x_4)\leq\Gamma(x_4,x_2)\leq\Gamma(x_2,x_1)\leq\Gamma(x_2,x_3)$. Hence, these values are exactly the same value. In view that $x_1\to x_3$ and $\Gamma(x_1,x_2)=\Gamma(x_1,x_3)$ then $x_1\to x_2$. By using the same argument for x_2, x_3 and x_4, we obtain that $x_1\to x_2\to x_4\to x_3\to x_1$. This is exactly the idea of the proof of Lemma 1. An analogous reasoning can be used with the sequence of vertexes x_6, x_7, x_8, x_6. We conclude from this that if G is a maximum similarity graph, then G has the form:



(1)

The double arrow (one arrow for each direction) between consecutive vertexes in a directed cycle is a very important property of the maximum similarity graphs. An immediate consequence of this result is the following proposition which relate the strongly connected components of the maximum similarity graph with the connected components of certain graph.

Proposition 1. *The set of strongly connected components of the directed graph* $G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}$ *is the same that the set of connected components of the non-directed graph* $\overline{\overline{G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}}} = (\mathcal{X}, \overline{\overline{E}})$ *where*

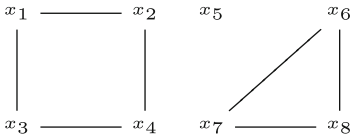
$$\overline{\overline{E}}_{(\mathcal{X},\Gamma,\beta_0)} = \{xy / \overrightarrow{xy} \in E^{\max}_{\beta_0} \wedge \overrightarrow{yx} \in E^{\max}_{\beta_0}\}.$$

Proof. Suppose that N is a strongly connected component of $G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}$. In view of definition of strongly connected component in a graph, for all pair of elements $x, y \in N$ there are directed paths of $G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}$ p and p' such that $o(p) = d(p') = x$, $d(p) = o(p') = y$. Suppose that $p = x_1, x_2, \ldots, x_m$ then, as a consequence of Corollary 1, $p'' = x_m, x_{m-1}, \ldots, x_1$ is also a directed path in $G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}$. Hence, for all $i = \overline{1,m}$ $x_i \sim x_{i+1}$ in $\overline{\overline{G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}}}$ and there is a connected component of $\overline{\overline{G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}}}$ containing x and y. Thus, for all strongly connected component N of $G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}$ there is a connected component C of $\overline{\overline{G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}}}$ such that $N \subseteq C$.

Reciprocally, suppose that C is a connected component of $\overline{\overline{G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}}}$, then for all $x, y \in C$ there is a path $p(x,y) = y_1, y_2, \ldots, y_s$. By construction of the graph $\overline{\overline{G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}}}$, for i from 1 to $m-1$, $y_i \sim y_{i+1}$ in $\overline{\overline{G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}}}$ iff $y_i \to y_{i+1} \wedge y_{i+1} \to y_i$ in $G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}$. As a consequence $p = y_1$, $y_2, \ldots, y_s$ and $p' = y_s, y_{s-1}, \ldots, y_1$ are directed paths of $G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}$ and hence, there is a strongly connected component N of $G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}$ containing x and y. It follows that $C \subseteq N$. □

In view of this proposition we can compute the strongly connected components of the graph $G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}$ by computing the connected components of the graph $\overline{\overline{G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}}} = (\mathcal{X}, E)$.

Example 3. If $G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}$ is the graph given in (1) then $\overline{\overline{G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}}} = (\mathcal{X}, E)$ has the form



Observe that there is no edge connecting x_5 to x_6 in $\overline{\overline{G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}}}$ because there is an arrow from x_5 to x_6 in $G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}$ but, there is no arrow from x_6 to x_5. Double arrow is needed between two objects in $G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}$ for to exist an edge between them in $\overline{\overline{G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}}}$.

3.1. Relationship with single-link and complete-link algorithms

In this section, we analyze some relationships between the criteria introduced in the previous section and the linkage algorithms. We restrict our analysis to single and complete link algorithm and in future works we will dedicate special attention to Haussdorff-linkage algorithm [16] which assumes that data lie into a metric space and the clusters are non-empty compact (in topological sense) subsets.

We shall show that every level of the hierarchy obtained using the single-link algorithm can be obtained through the connected components of $G_{(\mathcal{X},\Gamma,\beta_0)}$ for some value to the threshold β_0.

Proposition 2. *Let* $(\mathcal{X}, \Gamma)$ *be the similarity space. Suppose that* β_0 *is a similarity threshold and* ℓ *is a list of inter-object similarities for all different unordered pairs of objects sorted in descending order. Then if*

$$\overline{\Gamma} = \min_j \{\Gamma_j \in \ell / \beta_0 \leq \Gamma_j\},$$

the partition $\mathcal{P}_1^{\beta_0}$ *whose elements are the connected components of* $G_{(\mathcal{X},\Gamma,\beta_0)}$ *is the same partition obtained at a level corresponding to similarity value* $\overline{\Gamma}$ *of the single-link algorithm* (*see* [1,13,16] *for a suitable explanation of single-link algorithm*).

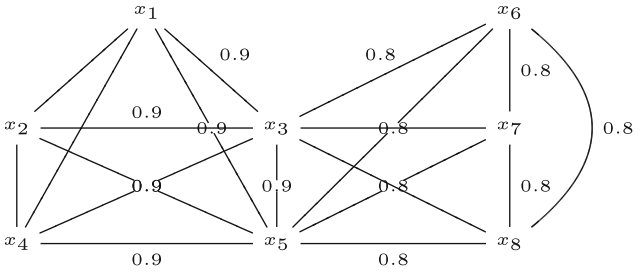
Proof. Let $\overline{\ell} = \Gamma_1, \Gamma_2, \ldots, \overline{\Gamma}$ be the list ℓ truncated in $\overline{\Gamma}$. The single-link algorithm begin with the partition formed by all unitary subsets of $\mathcal{X}$. Thus, each element belong to a unique cluster and each cluster contains a unique element. After that if $P_j = \{C_1, C_2, \ldots, C_{k_j}\}$ is the partition obtained at level j, then at level $j+1$ two clusters C_s and C_l of P_j are placed in same cluster of P_{j+1} if

$$\Gamma(C_s, C_l) = \max_t \max_i \Gamma(x_t, y_i) \geq \Gamma_{j+1}, \quad x_t \in C_s, \ y_i \in C_l. \tag{2}$$

Observe that the clusters C_s and C_l of P_j satisfying (2) iff there are an object $x \in C_s$ and an object $y \in C_l$ such that $\Gamma(x,y) \geq \Gamma_{j+1}$. In view that the list $\overline{\ell}$ is sorted in decreasing order, we have that $\Gamma_{j+1} \geq \overline{\Gamma} \geq \beta_0$. This implies that the edge $xy \in E_{(\mathcal{X},\Gamma,\beta_0)}$ and therefore, the partition obtained for the single-link algorithm with similarity value $\overline{\Gamma}$ is exactly those whose clusters are the connected components of the graph $G_{(\mathcal{X},\Gamma,\beta_0)}$. □

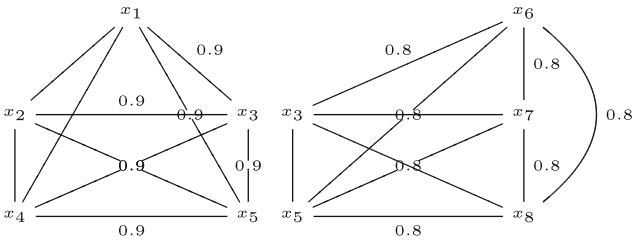
Let us concentrate now on the completely connected components of $G_{(\mathcal{X},\Gamma,\beta_0)}$ in the sense of complete-link algorithm. It is known that the completely connected components (cliques) of a graph do always not produce a partition of the vertexes set of the current graph, but they produce a cover. Let us see an example:

Example 4. Consider the following graph whose vertexes set is $\{x_1, x_2, \ldots, x_8\}$:



The two completely connected components of this graph are the maximally complete subgraphs showed below in the graphics (a) and (b), respectively. These completely connected components produce the following cover over the vertexes set of the considered graph: $\{\{x_1, x_2, x_3, x_4, x_5\}, \{x_3, x_5, x_6, x_7, x_8\}\}$. However, this is not the result of complete-link algorithm. Let us see how

complete-link algorithm works.



If we do a simple inspection to the explanation of complete-link algorithm made in [13], we can see that if some object belongs to more than one completely connected component, then complete-link assigns it to the cluster whose elements belong to the completely connected component which emerged first. But if they emerged at the same time, then complete-link assigns this object randomly to some one of these clusters. Therefore, if the considered graph above if a β_0-similarity graph in some level of complete-link, then this result will be the partition: $\{\{x_1, x_2, x_3, x_4, x_5\}, \{x_6, x_7, x_8\}\}$.

3.2. Nestedness

Consider the set $\mathbb{P}_{\mathcal{X}}$ of all possible partitions of the similarity space $(\mathcal{X}, \Gamma)$. Let $\trianglelefteq$ be the relationship "nested in" over $\mathbb{P}_{\mathcal{X}}$; that is, if $\mathcal{P}, \mathcal{P}' \in \mathbb{P}_{\mathcal{X}}$ then $\mathcal{P} \trianglelefteq \mathcal{P}'$ iff for all element $C' \in \mathcal{P}'$ there are elements $C_{i_1}, C_{i_2}, \ldots, C_{i_k} \in \mathcal{P}$ such that $C' = \bigcup_{j=1}^{k} C_{i_j}$.

In this section, every result is given for a similarity space $(\mathcal{X}, \Gamma)$.

Proposition 3. *Let $\beta_0, \beta_0' \in L$ be two similarity thresholds such that $\beta_0 \leq \beta_0'$ and consider $\mathcal{P}_1^{\beta_0}$ and $\mathcal{P}_1^{\beta_0'}$ be the two partition of $\mathcal{X}$; corresponding to the connected components of $G_{(\mathcal{X},\Gamma,\beta_0)}$ and $G_{(\mathcal{X},\Gamma,\beta_0')}$ respectively, then $\mathcal{P}_1^{\beta_0'} \trianglelefteq \mathcal{P}_1^{\beta_0}$.*

Proof. Taking into account that two arbitrary objects belonging to the same connected component of the graph if and only if there is a path between them, it is suffice to prove that every path of $G_{(\mathcal{X},\Gamma,\beta_0')}$ is also a path of $G_{(\mathcal{X},\Gamma,\beta_0)}$. Let $p = x_1, x_2, \ldots, x_m$ be an arbitrary path in $G_{(\mathcal{X},\Gamma,\beta_0')}$. This implies that $x_1 \sim x_2 \sim \cdots \sim x_m$ in $G_{(\mathcal{X},\Gamma,\beta_0')}$, and therefore, for i from 1 to $m-1$ $\Gamma(x_i, x_{i+1}) \geq \beta_0'$. But, $\beta_0' \geq \beta_0$, then $\Gamma(x_i, x_{i+1}) \geq \beta_0$ and, as an immediate consequence we have that $x_1 \sim x_2 \sim \cdots \sim x_m$ in $G_{(\mathcal{X},\Gamma,\beta_0)}$ and hence, p is also a path of $G_{(\mathcal{X},\Gamma,\beta_0)}$. □

The previous proposition has not been enounced on the same form in others works; however, equivalents results can be found in [4,1]. Besides that analogous propositions are obtained if we replace $\mathcal{P}_1^{\beta_0}$ and $\mathcal{P}_1^{\beta_0'}$ by $\mathcal{P}_2^{\beta_0}$ and $\mathcal{P}_2^{\beta_0'}$, respectively, or by $\mathcal{P}_3^{\beta_0}$ and $\mathcal{P}_3^{\beta_0'}$, respectively, or by $\mathcal{P}_4^{\beta_0}$ and $\mathcal{P}_4^{\beta_0'}$, respectively. So far, in this section, the results of nestedness have been obtained considering the same clustering criterion and different similarity threshold value. For next results in this section, we consider different clustering criteria and the same similarity threshold value.

Proposition 4. *Let β_0 be a similarity threshold value and consider the partitions $\mathcal{P}_1^{\beta_0}, \mathcal{P}_2^{\beta_0}, \mathcal{P}_3^{\beta_0}$ of $\mathcal{X}$. Then the following statements hold*;

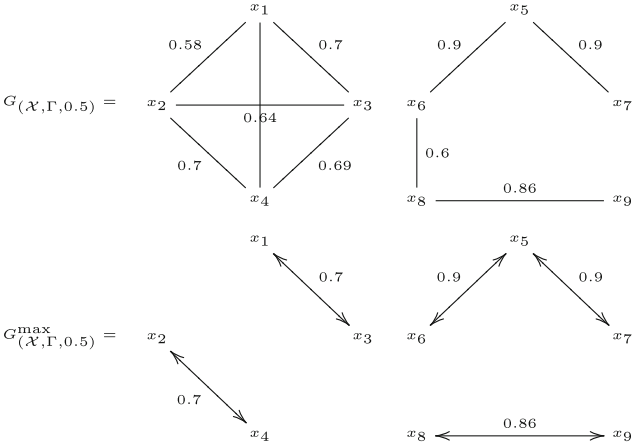
1. $\mathcal{P}_3^{\beta_0} \trianglelefteq \mathcal{P}_2^{\beta_0}$,
2. $\mathcal{P}_2^{\beta_0} \trianglelefteq \mathcal{P}_1^{\beta_0}$,
3. $\mathcal{P}_3^{\beta_0} \trianglelefteq \mathcal{P}_1^{\beta_0}$,
4. $\mathcal{P}_4^{\beta_0} \trianglelefteq \mathcal{P}_1^{\beta_0}$.

Proof.

1. In view of Proposition 1, $\mathcal{P}_3^{\beta_0}$ is the partition of $\mathcal{X}$ whose elements are the connected components of the graph $\overline{G_{(\mathcal{X},\Gamma,\beta_0)}^{\max}}$. Observe also that the elements of $\mathcal{P}_2^{\beta_0}$ are the connected components of $\overline{G_{(\mathcal{X},\Gamma,\beta_0)}^{\max}}$, hence it suffices to prove that every path of $\overline{G_{(\mathcal{X},\Gamma,\beta_0)}^{\max}}$ is a path in $\overline{G_{(\mathcal{X},\Gamma,\beta_0)}^{\max}}$ ($P(\overline{\overline{G_{(\mathcal{X},\Gamma,\beta_0)}^{\max}}}) \subseteq P(\overline{G_{(\mathcal{X},\Gamma,\beta_0)}^{\max}})$). Let $p = x_1, x_2, \ldots, x_m$ be an arbitrary path in $\overline{G_{(\mathcal{X},\Gamma,\beta_0)}^{\max}}$. Then, for all for i from 1 to $m-1$ $x_i \to x_{i+1}$ and $x_{i+1} \to x_i$ in $G_{(\mathcal{X},\Gamma,\beta_0)}^{\max}$ (see Proposition 1) and we have that p is also a path in $G_{(\mathcal{X},\Gamma,\beta_0)}^{\max}$ and hence a path in $\overline{G_{(\mathcal{X},\Gamma,\beta_0)}^{\max}}$.
2. This proof is analogous to the previous. We shall prove that $P(\overline{G_{(\mathcal{X},\Gamma,\beta_0)}^{\max}}) \subseteq P(G_{(\mathcal{X},\Gamma,\beta_0)})$. We suppose that $p = x_1, x_2, \ldots, x_m$ is an arbitrary path in $\overline{G_{(\mathcal{X},\Gamma,\beta_0)}^{\max}}$. This means that for i from 1 to $m-1$ one of the two following conditions is satisfied in $G_{(\mathcal{X},\Gamma,\beta_0)}^{\max}$: $x_i \to x_{i+1}$ or $x_{i+1} \to x_i$, and therefore $\Gamma(x_i, x_{i+1}) \geq \beta_0$. But, if for i from 1 to $m-1$, $\Gamma(x_i, x_{i+1}) \geq \beta_0$ then p is path in $G_{(\mathcal{X},\Gamma,\beta_0)}$.
3. This follows since $\mathcal{P}_3^{\beta_0} \trianglelefteq \mathcal{P}_2^{\beta_0}$ and $\mathcal{P}_2^{\beta_0} \trianglelefteq \mathcal{P}_1^{\beta_0}$ imply that $\mathcal{P}_3^{\beta_0} \trianglelefteq \mathcal{P}_1^{\beta_0}$.
4. This proof is immediate of the fact that the elements of $\mathcal{P}_4^{\beta_0}$ are completely connected components whereas the elements of $\mathcal{P}_1^{\beta_0}$ are connected components. □

It is not hard to see that the pair $(\mathbb{P}_{\mathcal{X}}, \trianglelefteq)$ consisting of the set of all possible partitions of $\mathcal{X}$ and the relation $\trianglelefteq$ is a partially ordered set (poset). That is, the relation $\trianglelefteq$ on $\mathbb{P}_{\mathcal{X}}$ is reflexive ($\forall \mathcal{P} \in \mathbb{P}_{\mathcal{X}}, \mathcal{P} \trianglelefteq \mathcal{P}$), antisymmetric ($\mathcal{P} \trianglelefteq \mathcal{P}' \wedge \mathcal{P}' \trianglelefteq \mathcal{P} \Rightarrow \mathcal{P} = \mathcal{P}'$) and transitive ($\mathcal{P} \trianglelefteq \mathcal{P}' \wedge \mathcal{P}' \trianglelefteq \mathcal{P}'' \Rightarrow \mathcal{P} \trianglelefteq \mathcal{P}''$). Moreover, the poset $(\mathbb{P}_{\mathcal{X}}, \trianglelefteq)$ is a complete lattice (every subset of $\mathbb{P}_{\mathcal{X}}$ has both a least upper bound and a greatest lower bound in $\mathbb{P}_{\mathcal{X}}$). Proposition 3 assures that partitions obtained with the same clustering criterion and different values for the threshold belong to a chain of $(\mathbb{P}_{\mathcal{X}}, \trianglelefteq)$. Analogously, Proposition 4 assures that the partitions $\mathcal{P}_1^{\beta_0}, \mathcal{P}_2^{\beta_0}$ and $\mathcal{P}_3^{\beta_0}$ belong to a chain of $(\mathbb{P}_{\mathcal{X}}, \trianglelefteq)$ and the same for $\mathcal{P}_4^{\beta_0}$ and $\mathcal{P}_1^{\beta_0}$. However, we cannot say the same for the partitions $\mathcal{P}_2^{\beta_0}, \mathcal{P}_3^{\beta_0}$, and $\mathcal{P}_4^{\beta_0}$. The following examples illustrate these situations.

Example 5. Let us suppose that $\mathcal{X} = \{x_1, x_2, \ldots, x_9\}$ and the similarity values greater than $\beta_0 = 0.5$ are over the edges in the following similarity graphs:



In view of the clustering criteria definitions, we have that

$\mathcal{P}_1^{0.5} = \{\{x_1, x_2, x_3, x_4\}, \{x_5, x_6, x_7, x_8, x_9\}\}$,
$\mathcal{P}_2^{0.5} = \{\{x_1, x_3\}, \{x_2, x_4\}, \{x_5, x_6, x_7\}, \{x_8, x_9\}\}$,

$\mathcal{P}_3^{0.5}=\{\{x_1,x_3\},\{x_2,x_4\},\{x_5,x_6,x_7\},\{x_8,x_9\}\}$,
$\mathcal{P}_4^{0.5}=\{\{x_1,x_2,x_3,x_4\},\{x_5,x_6\},\{x_7\},\{x_8,x_9\}\}$.

Observe that $\mathcal{P}_3^{0.5}\trianglelefteq\mathcal{P}_2^{0.5}\trianglelefteq\mathcal{P}_1^{0.5}$ and $\mathcal{P}_4^{0.5}\trianglelefteq\mathcal{P}_1^{0.5}$, but the partitions $\mathcal{P}_2^{0.5}$ and $\mathcal{P}_3^{0.5}$ are not comparable with $\mathcal{P}_4^{0.5}$. However, let us consider the less upper bound and the greatest lower bound of the set $\{\mathcal{P}_2^{0.5},\mathcal{P}_4^{0.5}\}$ in the lattice $(\mathbb{P}_{\mathcal{X}},\trianglelefteq)$, which we denote by $\mathcal{P}_2^{0.5}\bigvee\mathcal{P}_4^{0.5}$ and $\mathcal{P}_2^{0.5}\bigwedge\mathcal{P}_4^{0.5}$, respectively, and, in this case, they are given by

$\mathcal{P}_2^{0.5}\bigvee\mathcal{P}_4^{0.5}=\{\{x_1,x_2,x_3,x_4\},\{x_5,x_6,x_7\},\{x_8,x_9\}\}$ and
$\mathcal{P}_2^{0.5}\bigwedge\mathcal{P}_4^{0.5}=\{\{x_1,x_3\},\{x_2,x_4\},\{x_5,x_6\},\{x_7\},\{x_8,x_9\}\}$.

Observe that $\mathcal{P}_2^{0.5}\bigwedge\mathcal{P}_4^{0.5}\trianglelefteq\mathcal{P}_3^{0.5}\trianglelefteq\mathcal{P}_2^{0.5}\bigvee\mathcal{P}_4^{0.5}\trianglelefteq\mathcal{P}_1^{0.5}$. This result can be formalized the following way:

Proposition 5. *In the lattice* $(\mathbb{P}_{\mathcal{X}},\trianglelefteq)$ *the following statements hold*:

1. *If* $\mathcal{P}_2^{\beta_0}\neq\mathcal{P}_3^{\beta_0}$ *then*
   (a) $\mathcal{P}_3^{\beta_0}\bigwedge\mathcal{P}_4^{\beta_0}\trianglelefteq\mathcal{P}_2^{\beta_0}\bigwedge\mathcal{P}_4^{\beta_0}\trianglelefteq\mathcal{P}_3^{\beta_0}\bigvee\mathcal{P}_4^{\beta_0}\trianglelefteq\mathcal{P}_2^{\beta_0}\bigvee\mathcal{P}_4^{\beta_0}\trianglelefteq\mathcal{P}_1^{\beta_0}$,
   (b) $\mathcal{P}_3^{\beta_0}\bigwedge\mathcal{P}_4^{\beta_0}\trianglelefteq\mathcal{P}_3^{\beta_0}\trianglelefteq\mathcal{P}_3^{\beta_0}\bigvee\mathcal{P}_4^{\beta_0}\trianglelefteq\mathcal{P}_2^{\beta_0}\bigvee\mathcal{P}_4^{\beta_0}\trianglelefteq\mathcal{P}_1^{\beta_0}$,
   (c) $\mathcal{P}_3^{\beta_0}\bigwedge\mathcal{P}_4^{\beta_0}\trianglelefteq\mathcal{P}_2^{\beta_0}\bigwedge\mathcal{P}_4^{\beta_0}\trianglelefteq\mathcal{P}_2^{\beta_0}\trianglelefteq\mathcal{P}_2^{\beta_0}\bigvee\mathcal{P}_4^{\beta_0}\trianglelefteq\mathcal{P}_1^{\beta_0}$,
   (d) $\mathcal{P}_3^{\beta_0}\bigwedge\mathcal{P}_4^{\beta_0}\trianglelefteq\mathcal{P}_3^{\beta_0}\trianglelefteq\mathcal{P}_2^{\beta_0}\trianglelefteq\mathcal{P}_2^{\beta_0}\bigvee\mathcal{P}_4^{\beta_0}\trianglelefteq\mathcal{P}_1^{\beta_0}$.
2. *If* $\mathcal{P}_2^{\beta_0}=\mathcal{P}_3^{\beta_0}$ *then*
   (a) $\mathcal{P}_2^{\beta_0}\bigwedge\mathcal{P}_4^{\beta_0}\trianglelefteq\mathcal{P}_2^{\beta_0}\trianglelefteq\mathcal{P}_2^{\beta_0}\bigvee\mathcal{P}_4^{\beta_0}\trianglelefteq\mathcal{P}_1^{\beta_0}$.

Proof. Suppose that $\mathcal{P}_2^{\beta_0}\neq\mathcal{P}_3^{\beta_0}$. Then, in view of Proposition 4, we have that $\mathcal{P}_3^{\beta_0}\trianglelefteq\mathcal{P}_2^{\beta_0}\trianglelefteq\mathcal{P}_1^{\beta_0}$ and $\mathcal{P}_4^{\beta_0}\trianglelefteq\mathcal{P}_1^{\beta_0}$. By definition of greatest lower bound $\mathcal{P}_3^{\beta_0}\bigwedge\mathcal{P}_4^{\beta_0}\trianglelefteq\mathcal{P}_3^{\beta_0}\trianglelefteq\mathcal{P}_2^{\beta_0}$, $\mathcal{P}_3^{\beta_0}\bigwedge\mathcal{P}_4^{\beta_0}\trianglelefteq\mathcal{P}_4^{\beta_0}$ and by virtue of $\mathcal{P}_2^{\beta_0}\bigwedge\mathcal{P}_4^{\beta_0}$ is the greatest lower bound of $\{\mathcal{P}_2^{\beta_0},\mathcal{P}_4^{\beta_0}\}$ we have that $\mathcal{P}_3^{\beta_0}\bigwedge\mathcal{P}_4^{\beta_0}\trianglelefteq\mathcal{P}_2^{\beta_0}\bigwedge\mathcal{P}_4^{\beta_0}$. Besides, $\mathcal{P}_3^{\beta_0}\bigwedge\mathcal{P}_4^{\beta_0}\trianglelefteq\mathcal{P}_4^{\beta_0}$ and $\mathcal{P}_4^{\beta_0}\trianglelefteq\mathcal{P}_2^{\beta_0}\bigvee\mathcal{P}_4^{\beta_0}$ imply that $\mathcal{P}_3^{\beta_0}\bigwedge\mathcal{P}_4^{\beta_0}\trianglelefteq\mathcal{P}_2^{\beta_0}\bigvee\mathcal{P}_4^{\beta_0}$ and we obtain $\mathcal{P}_3^{\beta_0}\bigwedge\mathcal{P}_4^{\beta_0}\trianglelefteq\mathcal{P}_2^{\beta_0}\bigwedge\mathcal{P}_4^{\beta_0}\trianglelefteq\mathcal{P}_3^{\beta_0}\bigvee\mathcal{P}_4^{\beta_0}$. Now, by using the definition of less upper bound we obtain that $\mathcal{P}_3^{\beta_0}\trianglelefteq\mathcal{P}_3^{\beta_0}\bigvee\mathcal{P}_4^{\beta_0}$, $\mathcal{P}_4^{\beta_0}\trianglelefteq\mathcal{P}_3^{\beta_0}\bigvee\mathcal{P}_4^{\beta_0}$ $\mathcal{P}_3^{\beta_0}\trianglelefteq\mathcal{P}_2^{\beta_0}\trianglelefteq\mathcal{P}_2^{\beta_0}\bigvee\mathcal{P}_4^{\beta_0}$ and $\mathcal{P}_4^{\beta_0}\trianglelefteq\mathcal{P}_2^{\beta_0}\bigvee\mathcal{P}_4^{\beta_0}$, and hence $\mathcal{P}_3^{\beta_0}\bigvee\mathcal{P}_4^{\beta_0}\trianglelefteq\mathcal{P}_2^{\beta_0}\bigvee\mathcal{P}_4^{\beta_0}$ of where it follows that $\mathcal{P}_3^{\beta_0}\bigwedge\mathcal{P}_4^{\beta_0}\trianglelefteq\mathcal{P}_2^{\beta_0}\bigwedge\mathcal{P}_4^{\beta_0}\trianglelefteq\mathcal{P}_3^{\beta_0}\bigvee\mathcal{P}_4^{\beta_0}\trianglelefteq\mathcal{P}_2^{\beta_0}\bigvee\mathcal{P}_4^{\beta_0}$. Finally, as $\mathcal{P}_4^{\beta_0}\trianglelefteq\mathcal{P}_1^{\beta_0}$ and $\mathcal{P}_2^{\beta_0}\trianglelefteq\mathcal{P}_1^{\beta_0}$ we have that $\mathcal{P}_2^{\beta_0}\bigwedge\mathcal{P}_4^{\beta_0}\trianglelefteq\mathcal{P}_1^{\beta_0}$, and therefore the statement 1(a) holds. The rest of the statements can be demonstrated in analogous manner. □

4. Partition sensibility. Preliminary algorithms

This section is devoted to study how the partitions $\mathcal{P}_1^{\beta_0},\mathcal{P}_2^{\beta_0}$ and $\mathcal{P}_3^{\beta_0}$ are modified when a new object x arrives to the dataset. That is, if a new object x arrives, several changes could occur on the configurations of these partitions. Our main goal in this section is to find rules for achieving a good update performance of these partitions.

Henceforth x will only denote the new object that arrives. The set $\mathcal{X}\bigcup\{x\}$ is denoted by $\mathcal{X}'$. Analogously, $\Gamma':\mathcal{X}'\times\mathcal{X}'\to L$ is a similarity function such that its restriction to $\mathcal{X}$ is Γ ($\Gamma'_{/\mathcal{X}}=\Gamma$). For the set $\mathcal{X}$, $\mathcal{P}_1^{\beta_0},\mathcal{P}_2^{\beta_0}$ and $\mathcal{P}_3^{\beta_0}$ continue denoting the partitions corresponding to the connected components of $G_{(\mathcal{X},\Gamma,\beta_0)}$, $\overline{G_{(\mathcal{X},\Gamma,\beta_0)}^{\max}}$ and $\overline{\overline{G_{(\mathcal{X},\Gamma,\beta_0)}^{\max}}}$, respectively. Analogously, for $\mathcal{X}'$, we denote these partitions with a prime mark.

The objects $x_j\in\mathcal{X}$ are divided into six sets attending to its relationship with the new object x and the similarity threshold β_0. These sets are:

$T_1(\beta_0)=\{x_j\in\mathcal{X}/\Gamma'(x,x_j)<\beta_0\}$. T_1 is the set of the objects such that its similarity value with x is less than β_0.

$T_2(\beta_0)=\{x_j\in\mathcal{X}/\Gamma'(x,x_j)\geq\beta_0\wedge\overrightarrow{x_jx}\notin E^{\max}_{(\mathcal{X}',\Gamma',\beta_0)}\wedge\overrightarrow{x,x_j}\notin E^{\max}_{(\mathcal{X}',\Gamma',\beta_0)}\}$. T_2 is the set of objects which are β_0-similar to x, but neither are the most β_0-similar object to x nor x is its most β_0-similar.

$T_3(\beta_0)=\{x_j\in\mathcal{X}/degree(x_j,G_{(\mathcal{X},\Gamma,\beta_0)})>0\wedge outdegree(x_j,G^{\max}_{(\mathcal{X}',\Gamma',\beta_0)})=1\wedge\overrightarrow{x_j,x}\in E^{\max}_{(\mathcal{X}',\Gamma',\beta_0)}\}$. Each object x_j belonging to T_3 is such that the objects that were its most β_0-similar in $\mathcal{X}$ are not anymore, and x becomes in the unique most β_0-similar object to these objects.

$T_4(\beta_0)=T_1(\beta_0)^C\bigcap T_2(\beta_0)^C\bigcap T_3(\beta_0)^C=T_5(\beta_0)\bigcup T_6(\beta_0)$.

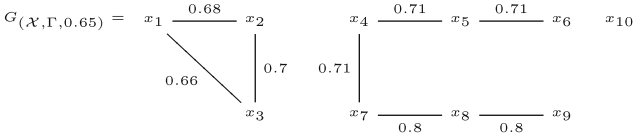
$T_5(\beta_0)=\{x\in T_4(\beta_0)/\overrightarrow{x_j,x}\in E^{\max}_{(\mathcal{X}',\Gamma',\beta_0)}\}$. The objects of T_5 are the objects of T_4 to which x is a most β_0-similar object.

$T_6(\beta_0)=\{x_j\in T_4(\beta_0)/\overrightarrow{x,x_j}\in E^{\max}_{(\mathcal{X}',\Gamma',\beta_0)}\}$. The objects of T_6 are the objects of T_4 which are most β_0-similar to x.

Observe that the sets T_1, T_2, T_3 and T_4 are pairwise disjoint. Henceforth, we denote by $[x]_1^{\beta_0},[x]_2^{\beta_0}$ and $[x]_3^{\beta_0}$ the connected component of $G_{(\mathcal{X}',\Gamma',\beta_0)}$, $\overline{G^{\max}_{(\mathcal{X}',\Gamma',\beta_0)}}$ and $\overline{\overline{G^{\max}_{(\mathcal{X}',\Gamma',\beta_0)}}}$ containing to x, respectively.

We begin with analysis for the case of $\mathcal{P}_1^{\beta_0}$. Let us consider a simple example.

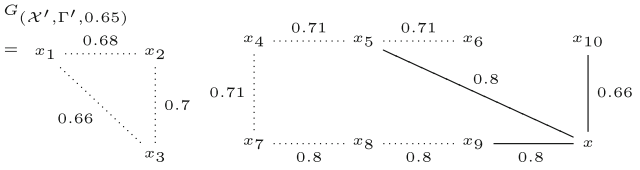
Example 6. Let $\mathcal{X}=\{x_1,x_2,\ldots,x_{10}\}$ be a database and let us suppose that the 0.65-similar graph $G_{(\mathcal{X},\Gamma,0.65)}$ of $(\mathcal{X},\Gamma)$ is given by



It is not difficult to see that

$\mathcal{P}_1^{0.65}=\{C_1^{0.65}=\{x_1,x_2,x_3\},C_2^{0.65}=\{x_4,x_5,x_6,x_7,x_8,x_9\},C_3^{0.65}=\{x_{10}\}\}$.

Now, suppose that a new object x arrives and the 0.65-similar graph associated to $\mathcal{X}$ become now for $\mathcal{X}'=\mathcal{X}\bigcup\{x\}$ in



Edges that emerge when x arrives are depicted by solid lines whereas existing edges are represented by dot lines. The clusters of $(\mathcal{P}_1^{0.65})'$ are the connected components of $G_{(\mathcal{X}',\Gamma',0.65)}$, hence

$(\mathcal{P}_1^{0.65})'=\{(C_1^{0.65})'=\{x_1,x_2,x_3\},(C_2^{0.65})'=[x]_1^{0.65}\}$.

By virtue of the definitions of the sets $T_i(\beta_0)$, $i=\overline{1,6}$ we have that:

$T_1(0.65)=\{x_1,x_2,x_4,x_7,x_8\},T_2(0.65)=\emptyset,T_3(0.65)=\{x_5\},T_4(0.65)=\{x_9,x_{10}\}$, $T_5(0.65)=\{x_9,x_{10}\}$ and $T_6(0.65)=\{x_9\}$.

Observe that every element of $C_1^{0.65}\in\mathcal{P}_1^{0.65}$ belongs to $T_1(0.65)$. This means that for all $x_i\in C_1^{0.65}$ we have $\Gamma(x_i,x)<0.65$. However, the clusters $C_2^{0.65}$ and $C_3^{0.65}$ have elements that are not in $T_1(0.65)$. Observe also that $C_1^{0.65}$ continues to be a cluster in $(\mathcal{P}_1^{0.65})'$, whereas $C_2^{0.65}$ and $C_3^{0.65}$ are not.

The conclusions which we reached in the example are quite natural if we keep in mind that the clustering criterion c_1 only

consider those similarity values that exceeds the threshold. The following lemma shows that such conclusions really are general.

Lemma 2. *Let C^{β_0} be an element of $\mathcal{P}_1^{\beta_0}$, then*:

1. *$C^{\beta_0} \subseteq T_1(\beta_0)$ if only if $C^{\beta_0} \in (\mathcal{P}_1^{\beta_0})'$.*
2. *$[x]_1^{\beta_0} = \cup\{C \in \mathcal{P}_1^{\beta_0} : C \not\subseteq T_1(\beta_0)\} \cup \{x\}$.*
3. *If $\mathcal{P}_1^{\beta_0}$ has m clusters and k of then are contained in $T_1(\beta_0)$, then $(\mathcal{P}_1^{\beta_0})'$ has $m-k+1$ clusters.*

Proof. Since $\Gamma'/_{\mathcal{X}} = \Gamma$, we have the following:

- $\forall x_i, x_j \in C^{\beta_0}$ there is a path $p=x_1, x_2, \ldots, x_m$ of $G_{(\mathcal{X},\Gamma,\beta_0)}$ such that $x_1 = x_i$ and $x_m = x_j$. This means that $\Gamma(x_k, x_{k+1}) \geq \beta_0$ for $1 \leq k \leq m$. But, $\Gamma'(x_k, x_{k+1}) = \Gamma(x_k, x_{k+1})$ hence, p is also a path in $G_{(\mathcal{X}',\Gamma',\beta_0)}$.
- Now, if $x_i \in C^{\beta_0}$ and $\tilde{x}_i \notin C^{\beta_0}$, then there is a path $p(x_i, \tilde{x}_i)$ in $G_{(\mathcal{X}',\Gamma',\beta_0)}$ if only if p passes by x. For this, it is necessary and sufficient that $\Gamma'(x_i, x) \geq \beta_0$.

Remember that we have taken $T_1(\beta_0) = \{x_j \in \mathcal{X}/\Gamma'(x, x_j) < \beta_0\}$. Therefore, from two items above, we conclude that the elements of C^{β_0} are necessarily in one same cluster in $(\mathcal{P}_1^{\beta_0})'$ and, C^{β_0} is exactly a cluster in $(\mathcal{P}_1^{\beta_0})'$ if only if for $x_i \in C^{\beta_0}$ it has $\Gamma'(x_i, x) < \beta_0$. From this, we conclude statement 1. On the other hand note that if there is $x_i \in C^{\beta_0}$ such that $\Gamma'(x_i, x) \geq \beta_0$, then for each element of C^{β_0} there is a path connecting it with x. In this case, $C^{\beta_0} \subseteq [x]_1^{\beta_0}$ and we conclude statement 2. Statement 3 follows from 1 and 2. □

The following corollaries are immediate consequences of the previous lemma. The reason this matters is that they will be heavily used in subsequent algorithms. Omit them could hinder the understanding of the algorithms.

Corollary 2. *Let $\beta_0, \beta_1 \in L$ be similarity thresholds such that $\beta_0 \leq \beta_1$. For each $C^{\beta_1} \in \mathcal{P}_1^{\beta_1}$, there is a unique $C^{\beta_0} \in \mathcal{P}_1^{\beta_0}$ such that $C^{\beta_1} \subseteq C^{\beta_0}$ (see Proposition 3). If $C^{\beta_0} \subseteq T_1(\beta_0)$ then, $C^{\beta_1} \subseteq T_1(\beta_1)$.*

Corollary 3. *Let $\beta_0 \in L$ be similarity threshold. For each $Z^{\beta_0} \in \mathcal{P}_2^{\beta_0}$, there is a unique $C^{\beta_0} \in \mathcal{P}_1^{\beta_0}$ such that $Z^{\beta_0} \subseteq C^{\beta_0}$ (see Proposition 4). If $C^{\beta_0} \subseteq T_1(\beta_0)$ then, $Z^{\beta_0} \subseteq T_1(\beta_0)$.*

Corollary 4. *Let $\beta_0 \in L$ be similarity threshold. For each $N^{\beta_0} \in \mathcal{P}_3^{\beta_0}$, there is a unique $C^{\beta_0} \in \mathcal{P}_1^{\beta_0}$ such that $N^{\beta_0} \subseteq C^{\beta_0}$ (see Proposition 4). If $C^{\beta_0} \subseteq T_1(\beta_0)$ then, $N^{\beta_0} \subseteq T_1(\beta_0)$.*

Next, we concentrate on to describe an algorithm for updating a hierarchy of nested partitions $\mathcal{P}_1^{\beta_n} \trianglelefteq \mathcal{P}_1^{\beta_{n-1}} \trianglelefteq \cdots \trianglelefteq \mathcal{P}_1^{\beta_0}$, where $\beta_0 \leq \beta_1 \leq \cdots \leq \beta_n$ is a increasing sequence of similarity thresholds. Observe that we are beginning by using only a clustering criterion c_1. After we see how to incorporate also the criteria c_2 and c_3. The algorithm we discuss now is actually the first stage of the general algorithm.

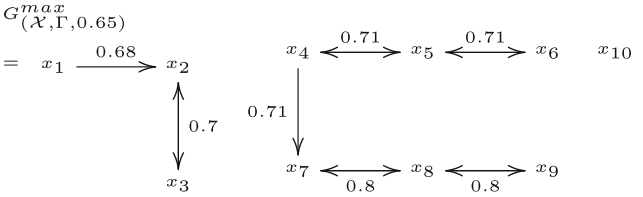
The underlying idea for this first part is to exploit the relationships obtained in Lemma 2 and Corollary 2 to develop the algorithm. Informally describing it, we would have: Once the similarity values between the new object x and each $x_i \in \mathcal{X}$ are calculated. The sets $T_i(\beta_0)$ are determined for each threshold, then we check for each cluster $C^{\beta_0} \in \mathcal{P}_1^{\beta_0}$ if it is contained in $T_1(\beta_0)$ and, we use Lemma 2 to determine their status. If $C^{\beta_0} \subseteq T_1(\beta_0)$, according to Lemma 2, it is concluded that C^{β_0} remains a cluster in $(\mathcal{P}_1^{\beta_0})'$ and, Corollary 2 ensures that each cluster $C^{\beta_1} \in \mathcal{P}_1^{\beta_1}$ such that $C^{\beta_1} \subseteq C^{\beta_0}$ is also a cluster in $(\mathcal{P}_1^{\beta_1})'$. If not, we add C^{β_0} to $[x]_1^{\beta_0}$ and, we repeat this analysis for each $C^{\beta_1} \subseteq C^{\beta_0}$. Next, we formally expose the pseudo-code for $n=1$.

Algorithm 1. Incremental nested partition algorithm: part 1

Input: $\mathcal{X}, x, \mathcal{P}_1^{\beta_0}, \mathcal{P}_1^{\beta_1}$
Output: $(\mathcal{P}_1^{\beta_0})', (\mathcal{P}_1^{\beta_1})'$
begin
  $(\mathcal{P}_1^{\beta_j})' \coloneqq \emptyset,\ [x]_1^{\beta_j} \coloneqq \{x\},\ j = 0, 1;$
  foreach x' ***in*** $\mathcal{X}$ **do**
    Compute $\Gamma(x, x')$ and determine the sets $\{T_i^{\beta_j}\}_{i=1}^{6},\ j = 0, 1;$
  foreach C^{β_0} ***in*** $\mathcal{P}_1^{\beta_0}$ **do**
    if $C^{\beta_0} \subseteq T_1^{\beta_0}$ **then**
      $(\mathcal{P}_1^{\beta_0})'.Add(C^{\beta_0});$
      foreach $C^{\beta_1} \subseteq C^{\beta_0}$ **do**
        $(\mathcal{P}_1^{\beta_1})'.Add(C^{\beta_1});$
    else
      $[x]_1^{\beta_0}.Add(C^{\beta_{0_1}});$
      foreach $C^{\beta_1} \subseteq C^{\beta_0}$ **do**
        if $C^{\beta_1} \subseteq T_1^{\beta_1}$ **then**
          $(\mathcal{P}_1^{\beta_1})'.Add(C^{\beta_1});$
        else
          $[x]_1^{\beta_1}.Add(C^{\beta_1});$
  $(\mathcal{P}_1^{\beta_0})'.Add([x]_1^{\beta_0});$
  $(\mathcal{P}_1^{\beta_1})'.Add([x]_1^{\beta_1});$
end

We shall now consider the case of $\mathcal{P}_2^{\beta_0}$. Analogously to the way we did before for $\mathcal{P}_1^{\beta_0}$, we begin with an illustrative example that raises suspicions of possible rules.

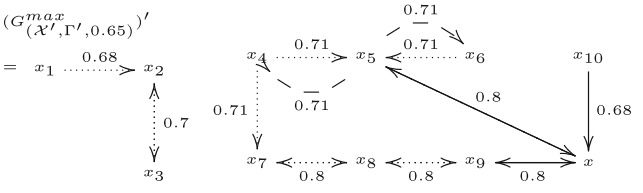
Really we continue with Example 6. Consider now the maximum β_0−similar graph associated to $\mathcal{X}$:



It is not difficult to see that

$$\mathcal{P}_2^{0.65} = \{Z_1^{0.65} = \{x_1, x_2, x_3\}, Z_2^{0.65} = \{x_4, x_5, x_6, x_7, x_8, x_9\}, Z_3^{0.65} = \{x_{10}\}\}.$$

When x arrives, this graph is updated the following way:



As we said above, the sets T_i (0.65), $i=\overline{1,6}$ are:

$T_1(0.65)=\{x_5,x_9\}^C, T_2(0.65)=\emptyset, T_3(0.65)=\{x_5\}$,
$T_5(0.65)=\{x_9,x_{10}\}$ and $T_6(0.65)=\{x_9\}$.

Observe that $Z_1^{0.65}\subseteq T_1(0.65)\bigcup T_2(0.65)$ and therefore, $Z_1^{0.65}\bigcap T_3(0.65)=\emptyset$ and $Z_1^{0.65}\bigcap T_4(0.65)=\emptyset$. Note also that $Z_1^{0.65}$ is both a cluster of $\mathcal{P}_2^{0.65}$ and a cluster of $(\mathcal{P}_2^{0.65})'$. However, $Z_2^{0.65}$ and $Z_3^{0.65}$ have not empty intersection with $T_3(0.65)$ and unlike $Z_1^{0.65}$, they do not continue to be clusters in $(\mathcal{P}_2^{0.65})'$. If we take into account that the criterion c_2 considers those similarity values exceeding the threshold and that they are maximum in at least one sense (i.e., $\Gamma(x_i,x_j)$ is the maximum similarity value for x_i or it is the maximum similarity value for x_j), then we see that the conclusions reached in the example are natural. These rules are satisfied in general as discussed in the following lemma.

Lemma 3. *Let Z^{β_0} be an element of $\mathcal{P}_2^{\beta_0}$, then*:

1. *$Z^{\beta_0}\subseteq T_1(\beta_0)\bigcup T_2(\beta_0)$ if only if $Z^{\beta_0}\in(\mathcal{P}_2^{\beta_0})'$.*
2. *$Z^{\beta_0}\bigcap T_3(\beta_0)=\emptyset$ and $Z^{\beta_0}\bigcap T_4(\beta_0)\neq\emptyset$ imply $Z^{\beta_0}\subseteq[x]_2^{\beta_0}$.*

Proof. Let us remember that $\overline{G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}}$ is obtained from it removing the orientation arrow in $G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}$. As a consequence, there is a path $p=x_1, x_2, \ldots, x_m$ in $\overline{G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}}$ iff for i from 1 to $m-1$ one of the following conditions is satisfied: $x_i\to x_{i+1}$ or $x_{i+1}\to x_i$ in $G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}$.

According to T_1 and T_2 are disjunct with respect to T_3, respectively, we have that $Z^{\beta_0}\subseteq T_1(\beta_0)\bigcup T_2(\beta_0)$ implies $Z^{\beta_0}\bigcap T_3(\beta_0)=\emptyset$. From this and the fact that $\Gamma'/_{\mathcal{X}}=\Gamma$ (i.e., the similarity values between elements of $\mathcal{X}$ are the same), we deduce that if $x_i,x_j\in\mathcal{X}$ and $x_i\to x_j$ in $G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}$ then, $x_i\to x_j$ in $G^{\max}_{(\mathcal{X},\Gamma',\beta_0)}$. In other words, if $Z^{\beta_0}\bigcap T_3(\beta_0)=\emptyset$ then, every arrow connecting two elements of Z^{β_0} in the graph $G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}$ is preserved in $G^{\max}_{(\mathcal{X}',\Gamma',\beta_0)}$. This ensures that Z^{β_0} is contained in some cluster of $(\mathcal{P}_2^{\beta_0})'$.

Now, because x is the unique element added to the database, new connections emerge only through x. Of course, there are no new connections with elements of Z^{β_0} if only if $Z^{\beta_0}\subseteq T_1(\beta_0)\bigcup T_2(\beta_0)$. Therefore, in this case, there is no cluster in $(\mathcal{P}_2^{\beta_0})'$ strictly containing Z^{β_0}. From this, Z^{β_0} is a cluster of $(\mathcal{P}_2^{\beta_0})'$ and we deduce 1. If $Z^{\beta_0}\not\subseteq T_1(\beta_0)\bigcup T_2(\beta_0)$, since $Z^{\beta_0}\bigcap T_3(\beta_0)=\emptyset$, we have that $Z^{\beta_0}\bigcap T_4(\beta_0)\neq\emptyset$ and hence, Z^{β_0} and x are contained in some cluster of $(\mathcal{P}_2^{\beta_0})'$. From this, statement 2 follows. □

Corollary 5. *Let $\beta_0,\beta_1\in L$ be similarity thresholds such that $\beta_0\leq\beta_1$. For each $Z^{\beta_1}\in\mathcal{P}_2^{\beta_1}$ there is a unique $Z^{\beta_0}\in\mathcal{P}_2^{\beta_0}$ such that $Z^{\beta_1}\subseteq Z^{\beta_0}$ (see Proposition 3). If $Z^{\beta_0}\subseteq T_1(\beta_0)\bigcup T_2(\beta_0)$ then, $Z^{\beta_1}\subseteq T_1(\beta_1)\bigcup T_2(\beta_1)$.*

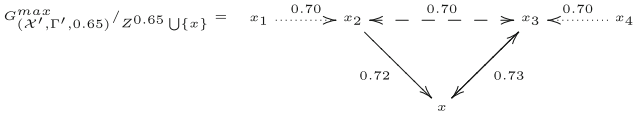
Corollary 6. *Let $\beta_0\in L$ be similarity threshold. For each $N^{\beta_0}\in\mathcal{P}_3^{\beta_0}$ there is a unique $Z^{\beta_0}\in\mathcal{P}_2^{\beta_0}$ such that $N^{\beta_0}\subseteq Z^{\beta_0}$ (see Proposition 4). If $Z^{\beta_0}\subseteq T_1(\beta_0)\bigcup T_2(\beta_0)$then, $N^{\beta_0}\subseteq T_1(\beta_0)\bigcup T_2(\beta_0)$.*

Note that in Lemma 3 the case in which $Z^{\beta_0}\bigcap T_3(\beta_0)\neq\emptyset$ is not considered. In this case, there is not an uniform behavior. In view that there are elements $x_i\in Z^{\beta_0}$ such that $x_i\to x$ in $(G^{\max}_{(\mathcal{X},\Gamma,\beta_0)})'$ and the elements of Z^{β_0} that were its most similar in $\mathcal{X}$ are not anymore (those elements $x_j\in\mathcal{X}$ such that $x_i\to x_j$ in $G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}$), all possibilities could occur. Let us see the following examples:

Example 7. Let us consider $Z^{\beta_0}=\{x_1,x_2,x_3,x_4\}$ such that the connection between its objects is represented in the following graph:

$G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}/_{Z^{0.65}}=$ $x_1 \overset{0.70}{\longleftrightarrow} x_2 \overset{0.70}{\longleftrightarrow} x_3 \overset{0.70}{\longleftrightarrow} x_4$

We suppose that x arrives and $G^{\max}_{(\mathcal{X}',\Gamma',0.65)}$ takes the following form:

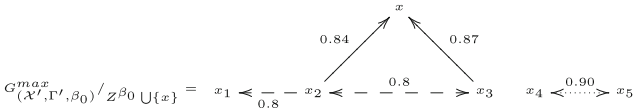


As before, edges that broken when x arrives are depicted by dashed lines whereas solid and dot lines represent the new edges and existing edges, respectively. Observe that in this case $T_3(\beta_0)=\{x_2,x_3\}$. However, although some connections are broken when x arrives, new connections that emerge through x, replacing the previous and the connectivity is maintained. Hence, in this case $Z^{\beta_0}\subseteq[x]_2^{\beta_0}$.

Example 8. Let us suppose $Z^{\beta_0}=\{x_1,x_2,x_3,x_4,x_5\}$ is such that

$G^{max}_{(\mathcal{X},\Gamma,\beta_0)}/_{Z^{\beta_0}}=$ $x_1 \overset{0.80}{\longleftrightarrow} x_2 \overset{0.80}{\longleftrightarrow} x_3 \overset{0.80}{\longleftrightarrow} x_4 \overset{0.90}{\longleftrightarrow} x_5$

when x arrives, we obtain



and therefore, $T_3(\beta_0)=\{x_2,x_3\}$. In this case, the elements x_2 and x_3 are going with x breaking the connectivity between x_1 and x_4. Therefore, two new clusters emerge: $\{x_1\}$ and $\{x_4,x_5\}$.

To summarize, we can say when $Z^{\beta_0}\bigcap T_3(\beta_0)\neq\emptyset$ can be two situations. First, the connections between elements of Z^{β_0} which are broken when x arrives are replaced by new connections through x which preserve the connectivity between elements of Z^{β_0}. Second, there are connections between elements of Z^{β_0} that are not replaced by any of the new ones formed. Therefore, the connectivity in Z^{β_0} is lost. In all elements of Z^{β_0} that are not connected with x, new clusters emerge. The set of such elements is denoted by $H_{Z^{\beta_0}}$.

Next, we describe two algorithms based on Lemmas 2 and 3 and their respective corollaries.

Suppose that we have a single value β_1 for the similarity threshold and two nested partitions $P_2^{\beta_1}\trianglelefteq P_1^{\beta_1}$. The first algorithm describes how these partitions are update when x arrives. Note that we are now using two clustering criteria: c_1 and c_2. For this, we check for each cluster $C^{\beta_1}\in\mathcal{P}_1^{\beta_1}$ if it is contained in $T_1(\beta_1)$ and we use Lemma 2 to determine their status. If $C^{\beta_1}\subseteq T_1(\beta_1)$, by using Lemma 2 we deduce that C^{β_1} remains a cluster in $(\mathcal{P}_1^{\beta_1})'$ and Corollary 3 and Lemma 3 ensures that each cluster $Z^{\beta_1}\in\mathcal{P}_2^{\beta_1}$ such that $Z^{\beta_1}\subseteq C^{\beta_1}$ is also a cluster in $(\mathcal{P}_2^{\beta_1})'$. If not, we add C^{β_1} to $[x]_1^{\beta_0}$ and, for each $Z^{\beta_1}\subseteq C^{\beta_1}$ we determine how it is related with each set T_i. Thus, if $Z^{\beta_1}\subseteq T_1(\beta_1)\cup T_2(\beta_1)$, then Lemma 3 ensure that Z^{β_1} remains a cluster in $(\mathcal{P}_2^{\beta_1})'$, but if $Z^{\beta_1}\cap T_3(\beta_1)=\emptyset$ and $Z^{\beta_1}\cap T_4(\beta_1)\neq\emptyset$ then, we using again Lemma 3 to conclude that $Z^{\beta_1}\subseteq[x]_2^{\beta_1}$. In the case that $Z^{\beta_1}\cap T_3(\beta_1)\neq\emptyset$, we determine the set $H_{Z^{\beta_1}}$ and compute the new clusters. Below, we formally expose the pseudo-code of this algorithm.

The second algorithm of this part aims to update a hierarchy of nested partitions $\mathcal{P}_2^{\beta_k}\trianglelefteq\mathcal{P}_2^{\beta_2}\trianglelefteq\cdots\trianglelefteq\mathcal{P}_2^{\beta_1}$, where $\beta_1\leq\beta_2\leq\cdots\leq\beta_k$ is a increasing sequence of similarity thresholds. This algorithm is based only on a clustering criterion c_2. This algorithm proceeds as follows. For each $Z^{\beta_1}\in\mathcal{P}_1^{\beta_1}$ we need to check which of the three possibilities is satisfied. If $Z^{\beta_1}\subseteq T_1(\beta_1)\cup T_2(\beta_1)$ then according to Lemma 3, Z^{β_1} remains a cluster in $(\mathcal{P}_2^{\beta_1})'$ and Corollary 5 assures that every $Z^{\beta_2}\subseteq Z^{\beta_1}$ is also a cluster in $(\mathcal{P}_2^{\beta_2})'$. Now, if $Z^{\beta_1}\cap T_3(\beta_1)=\emptyset$ and $Z^{\beta_1}\cap T_4(\beta_1)\neq\emptyset$ then by Lemma 3, $Z^{\beta_1}\subseteq[x]_2^{\beta_1}$ and for every $Z^{\beta_2}\subseteq Z^{\beta_1}$ we deduce by Corollary 5 that $Z^{\beta_2}\cap T_3(\beta_2)=\emptyset$. This

implies that $Z^{\beta_2} \subseteq T_1(\beta_2) \cup T_2(\beta_2)$ or $Z^{\beta_2} \cap T_4(\beta_2) \neq \emptyset$, which are situations described above, so we know how to proceed. But, if $Z^{\beta_1} \cap T_3(\beta_1) \neq \emptyset$, then we have to determine the set $H_{Z^{\beta_1}}$. If $H_{Z^{\beta_1}} = \emptyset$ then $Z^{\beta_1} \subseteq [x]_2^{\beta_1}$ and for each $Z^{\beta_2} \subseteq Z^{\beta_1}$, we have that $Z^{\beta_2} \subseteq T_1(\beta_2) \cup T_2(\beta_2)$ or $Z^{\beta_2} \subseteq [x]_2^{\beta_2}$. Instead, if $H_{Z^{\beta_1}} \neq \emptyset$ then $Z^{\beta_1} - H_{Z^{\beta_1}} \subseteq [x]_2^{\beta_1}$ and we have to compute the new clusters that emerge in $H_{Z^{\beta_1}}$. For every $Z^{\beta_2} \subseteq Z^{\beta_1}$ any of the three situations explained above is possible to occur. We have to determine which of them Z^{β_2} satisfies, and we proceed analogously as we have explained above. Below, the pseudo-code is given.

Now, we focus on the case of $\mathcal{P}_3^{\beta_0}$. Continuing Example 6, we have that the elements of $\mathcal{P}_3^{0.65}$ are:

$N_1^{0.65} = \{x_1\}$, $N_2^{0.65} = \{x_2, x_3\}$, $N_3^{0.65} = \{x_4, x_5, x_6\}$, $N_4^{0.65} = \{x_7, x_8, x_9\}$, and $N_5^{0.65} = \{x_{10}\}$.

This is easy to verify if we remember that a clustering criterion c_3 considers those similarity values exceeding the similarity threshold and that are maximum in the two senses (i.e., $\Gamma(x_i, x_j)$ is both the maximum similarity value of x_i and x_j). This means that only double arrows of $G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}$ are taken into account by the criterion c_3.

When x arrives, as we saw in the case of $\mathcal{P}_2^{\beta_0}$, the sets T_i are:

$T_1(0.65) = \{x_5, x_9\}^C, T_2(0.65) = \emptyset, T_3(0.65) = \{x_5\}, T_5(0.65) = \{x_9, x_{10}\}$, and $T_6(0.65) = \{x_9\}$.

Observe that none of the clusters $N_1^{0.65}$, $N_2^{0.65}$ and $N_5^{0.65}$ intersects neither $T_3(0.65)$ or $T_5(0.65) \bigcap T_6(0.65)$. Note also that these cluster remains clusters in $(\mathcal{P}_3^{\beta_0})'$. However, $x_5 \in N_3^{0.65} \bigcap T_3(0.65)$ and hence as $x \rightarrow x_5$ we have that $x_5 \in [x]_3$. In $N_3^{0.65} - T_3(0.65)$ only a new cluster emerge. This new cluster is unitary. On the other hand, $N_4^{0.65} \bigcap (T_4(0.65) \bigcap T_5(0.65)) \neq \emptyset$ and $N_4^{0.65}$ does not intersect $T_3(0.65)$. Note that $N_4^{0.65} \subseteq [x]_3$.

As we shall see in the following lemma, this behavior is typical.

Algorithm 2. Incremental nested partition algorithm: part 2

Input: $\mathcal{P}_1^{\beta_1}, \mathcal{P}_2^{\beta_1}$
Output: $(\mathcal{P}_1^{\beta_1})', (\mathcal{P}_2^{\beta_1})'$
begin
  $(\mathcal{P}_1^{\beta_1})' \coloneqq (\mathcal{P}_2^{\beta_1})' \coloneqq \emptyset$, $[x]_1^{\beta_1} \coloneqq [x]_2^{\beta_1} \coloneqq \{x\}$;
  foreach C^{β_1} ***in*** $\mathcal{P}_1^{\beta_1}$ **do**
    if $C^{\beta_1} \subseteq T_1^{\beta_1}$ **then**
      $(\mathcal{P}_1^{\beta_1})'.Add(C^{\beta_1})$;
      foreach $Z^{\beta_1} \subseteq C^{\beta_1}$ **do**
        $(\mathcal{P}_2^{\beta_1})'.Add(Z^{\beta_1})$;
    else
      $[x]_1^{\beta_1}.Add(C^{\beta_1})$;
      if $C^{\beta_1} \bigcap T_3^{\beta_1} = \emptyset \wedge C^{\beta_1} \bigcap T_4^{\beta_1} = \emptyset$ **then**
        foreach $Z^{\beta_1} \subseteq C^{\beta_1}$ **do**
          $(\mathcal{P}_2^{\beta_1})'.Add(Z^{\beta_1})$;
      if $C^{\beta_1} \bigcap T_3^{\beta_1} = \emptyset \wedge C^{\beta_1} \bigcap T_4^{\beta_1} \neq \emptyset$ **then**
        foreach $Z^{\beta_1} \subseteq C^{\beta_1}$ **do**
          if $Z^{\beta_1} \bigcap T_4^{\beta_1} = \emptyset$ **then**
            $(\mathcal{P}_2^{\beta_1})'.Add(Z^{\beta_1})$;
          else
            $[x]_2^{\beta_1}.Add(Z^{\beta_1})$;
      if $C^{\beta_1} \bigcap T_3^{\beta_1} \neq \emptyset$ **then**
        foreach $Z^{\beta_1} \subseteq C^{\beta_1}$ **do**
          if $Z^{\beta_1} \bigcap T_3^{\beta_1} = \emptyset \wedge Z^{\beta_1} \bigcap T_4^{\beta_1} = \emptyset$ **then**
            $(\mathcal{P}_2^{\beta_1})'.Add(Z^{\beta_1})$;
          if $Z^{\beta_1} \bigcap T_3^{\beta_1} = \emptyset \wedge Z^{\beta_1} \bigcap T_4^{\beta_1} \neq \emptyset$ **then**
            $[x]_2^{\beta_1}.Add(Z^{\beta_1})$;
          if $Z^{\beta_1} \bigcap T_3^{\beta_1} \neq \emptyset$ **then**
            if $H_{Z^{\beta_1}} = \emptyset$ **then**
              $[x]_2^{\beta_1}.Add(Z^{\beta_1})$;
            else
              Calculate the set
                $T^{Z^{\beta_1}}$ of new clusters that emerge in $H_{Z^{\beta_1}}$;
              $(\mathcal{P}_2^{\beta_1})'.Add(T^{Z^{\beta_1}})$;
              $[x]_2^{\beta_{0_n}}.Add(Z^{\beta_1} - H_{Z^{\beta_1}})$;
    $(\mathcal{P}_1^{\beta_1})'.Add([x]_1^{\beta_1})$;
    $(\mathcal{P}_2^{\beta_1})'.Add([x]_2^{\beta_1})$;
end

Algorithm 3. Incremental nested partition algorithm: part 3

Input: $\mathcal{P}_2^{\beta_1}, \mathcal{P}_2^{\beta_2}$
Output: $(\mathcal{P}_2^{\beta_1})', (\mathcal{P}_2^{\beta_2})'$
begin
  $(\mathcal{P}_2^{\beta_j})' = \emptyset,\ [x]_2^{\beta_j} = \{x\}\ j = 1, 2;$
  foreach $Z^{\beta_1} \in \mathcal{P}_2^{\beta_1}$ **do**
    if $Z^{\beta_1} \subseteq T_1^{\beta_1} \bigcup T_2^{\beta_1}$ **then**
      $(\mathcal{P}_2^{\beta_1})'.Add(Z^{\beta_1});$
      foreach $Z^{\beta_2} \subseteq Z^{\beta_1}$ **do**
        $(\mathcal{P}_2^{\beta_2})'.Add(Z^{\beta_2});$
    if $Z^{\beta_1} \bigcap T_4^{\beta_1} \neq \emptyset \wedge Z^{\beta_1} \bigcap T_3^{\beta_1} = \emptyset$ **then**
      $[x]_2^{\beta_1}.Add(Z^{\beta_1});$
      foreach $Z^{\beta_2} \subseteq Z^{\beta_1}$ **do**
        if $Z^{\beta_2} \bigcap T_4^{\beta_2} = \emptyset$ **then**
          $(\mathcal{P}_2^{\beta_2})'.Add(Z^{\beta_2});$
        else
          $[x]_2^{\beta_2}.Add(Z^{\beta_2});$
    if $Z^{\beta_1} \bigcap T_3^{\beta_1} \neq \emptyset$ **then**
      if $H_{Z^{\beta_1}} = \emptyset$ **then**
        $[x]_2^{\beta_1}.Add(Z^{\beta_1});$
        foreach $Z^{\beta_2} \subseteq Z^{\beta_2}$ **do**
          if $Z^{\beta_2} \subseteq T_1^{\beta_2} \bigcup T_2^{\beta_2}$ **then**
            $\mathcal{P}_2^{\beta_2}.Add(Z^{\beta_2});$
          else
            $[x]_2^{\beta_2}.Add(Z^{\beta_2});$
      else
        Calculate the set
          $T^{Z^{\beta_1}}$ of new clusters that emerge in $H_{Z^{\beta_1}}$;
        $(\mathcal{P}_2^{\beta_1})'.Add(T^{Z^{\beta_1}});$
        $[x]_2^{\beta_1}.Add(Z^{\beta_1} - H_{Z^{\beta_1}});$
        foreach $Z^{\beta_2} \subseteq Z^{\beta_1}$ **do**
          if $Z^{\beta_2} \subseteq T_1^{\beta_2} \bigcup T_2^{\beta_2}$ **then**
            $\mathcal{P}_2^{\beta_2}.Add(Z^{\beta_2});$
          if $Z^{\beta_2} \bigcap T_4^{\beta_2} \neq \emptyset \wedge Z^{\beta_2} \bigcap T_3^{\beta_2} = \emptyset$ **then**
            $[x]_2^{\beta_2}.Add(Z^{\beta_2});$
          if $Z^{\beta_2} \bigcap T_3^{\beta_2} \neq \emptyset$ **then**
            if $H_{Z^{\beta_2}} = \emptyset$
              $[x]_2^{\beta_2}.Add(Z^{\beta_2});$
            else
              Calculate the set
                $T_{Z^{\beta_2}}$ of new cluster that emerge in $H_{Z^{\beta_2}}$;
              $(\mathcal{P}_2^{\beta_2})'.Add(T_{Z^{\beta_2}});$
              $[x]_2^{\beta_2}.Add(Z^{\beta_2} - H_{Z^{\beta_2}});$
  $(\mathcal{P}_2^{\beta_j})'.Add([x]_2^{\beta_j})\ j = 1, 2;$
end

Lemma 4. *Let N^{β_0} be an element of $\mathcal{P}_3^{\beta_0}$, then*:

1. $N^{\beta_0} \bigcap T_3(\beta_0) = \emptyset$ *and* $N^{\beta_0} \bigcap T_5(\beta_0) \bigcap T_6(\beta_0) = \emptyset$ *if only if* $N^{\beta_0} \in (\mathcal{P}_3^{\beta_0})'$.
2. $N^{\beta_0} \bigcap T_3(\beta_0) = \emptyset$ *and* $N^{\beta_0} \bigcap T_5(\beta_0) \bigcap T_6(\beta_0) \neq \emptyset$ *imply* $N^{\beta_0} \subseteq [x]_3^{\beta_0}$.
3. $N^{\beta_0} \bigcap T_3(\beta_0) \neq \emptyset$ *implies* $N^{\beta_0} \bigcap T_5(\beta_0) \bigcap T_6(\beta_0) = \emptyset$. *Moreover, we have that* $\forall x' \in N^{\beta_0} \bigcap T_3(\beta_0)$ *such that* $x \nrightarrow x'$ *in* $G^{\max}_{(\mathcal{X}', \Gamma', \beta_0)}$, *it has that* $\{x'\} \in (\mathcal{P}_3^{\beta_0})'$. *In case that* $x' \in N^{\beta_0} \bigcap T_3(\beta_0)$ and $x \rightarrow x'$ *in* $G^{\max}_{(\mathcal{X}', \Gamma', \beta_0)}$, *then* $x' \in [x]_3^{\beta_0}$.

Note that the third statement of Lemma 4 says that we do with the elements of N^{β_0} that are in $T_3(\beta_0)$, but there may be elements of N^{β_0} which are not in $T_3(\beta_0)$. In this case, we have to compute the new clusters that arise in $N^{\beta_0} - T_3(\beta_0)$.

Proof. As a consequence of Corollary 1, two objects $y_i, y_j \in \mathcal{X}'$ are in the same connected components of $\overline{G^{\max}_{(\mathcal{X}', \Gamma', \beta_0)}}$ iff there is directed paths $p = x_1, x_2, \ldots, x_m$ and $p' = x_m, x_{m-1}, \ldots, x_1$ in $G^{\max}_{(\mathcal{X}', \Gamma', \beta_0)}$ such that $o(p) = d(p')$, $d(p) = o(p')$ and, for i from 1 to $m-1$ it holds $x_i \rightarrow x_{i+1}$ and $x_{i+1} \rightarrow x_i$. Once remembered that we can prove the lemma.

Let N^{β_0} be an element of $\mathcal{P}_3^{\beta_0}$ such that $N^{\beta_0} \bigcap T_3(\beta_0) = \emptyset$. Taking into account that the similarity values between elements of N^{β_0} remain the same, we have that every arrow connecting a pair of objects of N^{β_0} in $G^{\max}_{(\mathcal{X}, \Gamma, \beta_0)}$ is preserved in $G^{\max}_{(\mathcal{X}', \Gamma', \beta_0)}$. It only remains to analyze whether N^{β_0} is contained in the class of x and whether it remains a cluster of $\mathcal{X}'$. In view of the initial comment of the proof, this is only possible if there is, at least, an object $x' \in N^{\beta_0}$ such that $x' \rightarrow x$ and $x \rightarrow x'$. Thus, $x' \in T_5(\beta_0) \bigcap T_6(\beta_0)$. Hence, if N^{β_0} is such that its intersection with $T_5(\beta_0) \bigcap T_6(\beta_0)$ is the empty set, then $N^{\beta_0} \in (\mathcal{P}_3^{\beta_0})'$, otherwise N^{β_0} is contained in $[x]_3^{\beta_0}$.

Let us see the last part. Let N^{β_0} be an element of $\mathcal{P}_3^{\beta_0}$ such that $N^{\beta_0} \bigcap T_3(\beta_0) \neq \emptyset$. This means that there is, at least, an object $x' \in N^{\beta_0}$ such that $x' \rightarrow x$ in $G^{\max}_{(\mathcal{X}', \Gamma', \beta_0)}$ and the elements of N^{β_0} that were its most similar in $\mathcal{X}$ are not anymore (those elements $x'' \in \mathcal{X}$ such that $x' \rightarrow x''$ in $G^{\max}_{(\mathcal{X}, \Gamma, \beta_0)}$). Hence, the unique object in $\mathcal{X}'$ which is the most similar to x' is x (if $x' \rightarrow x''$ in $G^{\max}_{(\mathcal{X}', \Gamma', \beta_0)}$ then $x'' = x$). Let us see now that $T_5(\beta_0) \bigcap T_6(\beta_0) = \emptyset$. Suppose that there is an object $x_j \in N^{\beta_0}$ which belongs to $T_5(\beta_0) \bigcap T_6(\beta_0)$. As x' and x_j belong to N^{β_0}, then there is a directed path $p = x_j, \ldots, x'$ in $G^{\max}_{(\mathcal{X}, \Gamma, \beta_0)}$ such that $o(p) = x_j$ and $d(p) = x'$. In general, we cannot assure that p is a path in $G^{\max}_{(\mathcal{X}', \Gamma', \beta_0)}$ because some of its elements could belong to $T_3(\beta_0)$. In this case, we can take x' as the element of p which belongs to $T_3(\beta_0)$ and is closest to x_j. For this reason, without a loss of generality, let us assume that p is a path in $G^{\max}_{(\mathcal{X}', \Gamma', \beta_0)}$. In view that $x_j \in T_5(\beta_0) \bigcap T_6(\beta_0)$ and $x' \in T_3(\beta_0)$ we have that $p' = xpx = x, x_j, \ldots, x', x$ is a cycle in $G^{\max}_{(\mathcal{X}', \Gamma', \beta_0)}$. Lemma 1 implies that $p'' = x, x', \ldots, x_j, x$ is also a cycle in $G^{\max}_{(\mathcal{X}', \Gamma', \beta_0)}$and therefore, there is an object $x_i \neq x$ such that $x' \rightarrow x_i$. This contradicts the fact that $x' \in T_3(\beta_0)$ and hence, $T_5(\beta_0) \bigcap T_6(\beta_0) = \emptyset$. Now, it is obvious that if $x' \in T_3(\beta_0)$ and $x \rightarrow x'$ then $x' \in [x]_3^{\beta_0}$; otherwise $\{x'\} \in (\mathcal{P}_3^{\beta_0})'$. □

Corollary 7. *Let $\beta_0, \beta_1 \in L$ be similarity thresholds such that $\beta_0 \leq \beta_1$. For each $N^{\beta_1} \in \mathcal{P}_2^{\beta_1}$ there is a unique $N^{\beta_0} \in \mathcal{P}_2^{\beta_0}$ such that $N^{\beta_1} \subseteq N^{\beta_0}$ (see Proposition 3). If $N^{\beta_0} \cap T_5(\beta_0) \cap T_6(\beta_0) = \emptyset$ and $N^{\beta_0} \cap T_3(\beta_0) = \emptyset$ then, $N^{\beta_1} \cap T_5(\beta_1) \cap T_6(\beta_1) = \emptyset$ and $N^{\beta_1} \cap T_3(\beta_1) = \emptyset$.*

Finally, we shall describe the last algorithms. Suppose that we have a single value β_1 for the similarity threshold and two nested partitions $P_3^{\beta_1} \trianglelefteq P_2^{\beta_1}$. The first algorithm describes how these partitions are update when x arrives. Note that we are again using two clustering criteria: c_2 and c_3. For this, we firstly check for each cluster $Z^{\beta_1} \in \mathcal{P}_2^{\beta_1}$ if it is contained in $T_1(\beta_1) \cup T_2(\beta_1)$ and, we use Lemma 3 to determine their status. If $Z^{\beta_1} \subseteq T_1(\beta_1) \cup T_2(\beta_1)$, by using Lemma 3 we deduce that Z^{β_1} remains a cluster in $(\mathcal{P}_2^{\beta_1})'$ and Corollary 6 and Lemma 4 ensure that each cluster $N^{\beta_1} \in \mathcal{P}_3^{\beta_1}$ such that $N^{\beta_1} \subseteq Z^{\beta_1}$ is also a cluster in $(\mathcal{P}_3^{\beta_1})'$. If not, we check if $Z^{\beta_1} \cap T_3(\beta_1) = \emptyset$ and $Z^{\beta_1} \cap T_4(\beta_1) \neq \emptyset$ and, by virtue of Lemma 3 we determine their status (see description of algorithms part 2 and part 3). In this case, Lemma 4 assures that for each $N^{\beta_1} \subseteq Z^{\beta_1}$ if $N^{\beta_1} \cap T_5(\beta_1) \cap T_6(\beta_1) \neq \emptyset$ then, $N^{\beta_1} \subseteq [x]_3^{\beta_1}$; otherwise $N^{\beta_1} \in (\mathcal{P}_3^{\beta_1})'$. In the case that $Z^{\beta_1} \cap T_3(\beta_1) \neq \emptyset$, we determine the set $H_{Z^{\beta_1}}$ and compute the new clusters. In this case, there are clusters $N^{\beta_1} \subseteq Z^{\beta_1}$ whose intersection with $T_3(\beta_1)$ is not empty, then Lemma 4 ensures that $N^{\beta_1} \bigcap T_5(\beta_1) \bigcap T_6(\beta_1) = \emptyset$ and $\forall x' \in N^{\beta_1} \bigcap T_3(\beta_1)$ such that $x \nrightarrow x'$ in $G^{\max}_{(\mathcal{X}', \Gamma', \beta_1)}$, we have that $\{x'\} \in (\mathcal{P}_3^{\beta_1})'$. In case that $x' \in N^{\beta_1} \bigcap T_3(\beta_1)$ and $x \rightarrow x'$ in $G^{\max}_{(\mathcal{X}', \Gamma', \beta_1)}$, then $x' \in [x]_3^{\beta_1}$. Next, if $N^{\beta_1} - T_3(\beta_1) \neq \emptyset$, we compute the new clusters that emerge in $N^{\beta_1} - T_3(\beta_1)$. The rest of elements $N^{\beta_1} \in \mathcal{P}_3^{\beta_1}$ with $N^{\beta_1} \subseteq Z^{\beta_1}$ satisfy one of the first two conditions given above. Below, we formally expose the pseudo-code of this algorithm.

Let us now suppose that we have two similarity thresholds $\beta_1 \leq \beta_2$, and the partition $\mathcal{P}_3^{\beta_2} \trianglelefteq \mathcal{P}_3^{\beta_1}$ obtained with the clustering criterion c_3 and the previous similarity values, respectively. The last algorithm updates these nested partitions when the object x arrives. This algorithm operates taking initially each $N^{\beta_1} \in \mathcal{P}_3^{\beta_1}$ with $N^{\beta_1} \cap T_3(\beta_1) = \emptyset$ and verifying if $N^{\beta_1} \cap T_5(\beta_1) \cap T_6(\beta_1)$ is the empty set or it is not. In the case that $N^{\beta_1} \cap T_5(\beta_1) \cap T_6(\beta_1) = \emptyset$, we have, by virtue of Lemma 4 that, N^{β_1} remains a cluster in $(\mathcal{P}_3^{\beta_1})'$. Moreover, for each $N^{\beta_2} \in \mathcal{P}_3^{\beta_2}$ such that $N^{\beta_2} \subseteq N^{\beta_1}$, we have, in view of Corollary 7 and Lemma 4 that, N^{β_2} remains a cluster in $(\mathcal{P}_3^{\beta_2})'$. Now, if $N^{\beta_1} \cap T_5(\beta_1) \cap T_6(\beta_1) \neq \emptyset$, then Lemma 4 assures that $N^{\beta_1} \subseteq [x]_3^{\beta_1}$. In this case, every $N^{\beta_2} \in \mathcal{P}_3^{\beta_2}$ such that $N^{\beta_2} \subseteq N^{\beta_1}$ satisfies that $N^{\beta_1} \cap T_5(\beta_1) \cap T_6(\beta_1)$ is the empty set or it is different of the empty set. Both situations were explained above; and this completes the case in which $N^{\beta_1} \cap T_3(\beta_1) = \emptyset$. On the other hand, when $N^{\beta_1} \cap T_3(\beta_1) \neq \emptyset$, then Lemma 4 ensures that $N^{\beta_1} \bigcap T_5(\beta_1) \bigcap T_6(\beta_1) = \emptyset$ and $\forall x' \in N^{\beta_1} \bigcap T_3(\beta_1)$ such that $x \nrightarrow x'$ in $G^{\max}_{(\mathcal{X}', \Gamma', \beta_1)}$, we have that $\{x'\} \in (\mathcal{P}_3^{\beta_1})'$. In case that $x' \in N^{\beta_1} \bigcap T_3(\beta_1)$ and $x \rightarrow x'$ in $G^{\max}_{(\mathcal{X}', \Gamma', \beta_1)}$, then $x' \in [x]_3^{\beta_1}$. Next, if $N^{\beta_1} - T_3(\beta_1) \neq \emptyset$, we compute the new clusters that emerge in $N^{\beta_1} - T_3(\beta_1)$. In this case, the clusters $N^{\beta_2} \subseteq N^{\beta_1}$ satisfy some of the situations considered above. We have to determine which of them N^{β_2} satisfies, and we proceed analogously as we have explained above. In Algorithm 5, we expose formally the pseudo-code.

Algorithm 4. Incremental nested partition algorithm: part 4

Input: $\mathcal{P}_2^{\beta_1}, \mathcal{P}_3^{\beta_1}$
Output: $(\mathcal{P}_2^{\beta_1})', (\mathcal{P}_3^{\beta_1})'$

begin
  $(\mathcal{P}_2^{\beta_1})' = (\mathcal{P}_3^{\beta_1})',\ [x]_2^{\beta_1} = [x]_3^{\beta_1} := \{x\}$
  foreach $Z^{\beta_1} \in (\mathcal{P}_2^{\beta_1})'$ **do**
    if $Z^{\beta_1} \subseteq T_1^{\beta_1} \bigcup T_2^{\beta_1}$ **then**
      $(\mathcal{P}_2^{\beta_1})'.Add(Z^{\beta_1})$;
      foreach $N^{\beta_1} \subseteq Z^{\beta_1}$ **do**
        $(\mathcal{P}_3^{\beta_1})'.Add(N^{\beta_1})$;
    if $Z^{\beta_1} \bigcap T_4^{\beta_1} \neq \emptyset \wedge Z^{\beta_1} \bigcap T_3^{\beta_1} = \emptyset$ **then**
      $[x]_3^{\beta_1}.Add(N^{\beta_1})$;
      foreach $N^{\beta_1} \subseteq Z^{\beta_1}$ **do**
        if $N^{\beta_1} \bigcap T_5^{\beta_1} \bigcap T_6^{\beta_1} = \emptyset$ **then**
          $(\mathcal{P}_3^{\beta_1})'.Add(N^{\beta_1})$;
        else
          $[x]_3^{\beta_1}.Add(N^{\beta_1})$;
    if $Z^{\beta_1} \bigcap T_3^{\beta_1} \neq \emptyset$ **then**
      if $H_{Z^{\beta_1}} = \emptyset$ **then**
        $[x]_2^{\beta_1}.Add(Z^{\beta_1})$;
      else
        Calculate the set
          $T^{Z^{\beta_1}}$ of new clusters that emerge in $H_{Z^{\beta_1}}$;
        $(\mathcal{P}_2^{\beta_1})'.Add(T^{Z^{\beta_1}})$;
        $[x]_2^{\beta_1}.Add(Z^{\beta_1} - H_{Z^{\beta_1}})$;
      foreach $N^{\beta_1} \subseteq Z^{\beta_1}$ **do**
        if $N^{\beta_1} \bigcap T_4^{\beta_1} \neq \emptyset \wedge N^{\beta_1} \bigcap T_3^{\beta_1} = \emptyset$ **then**
          if $N^{\beta_1} \bigcap T_5^{\beta_1} \bigcap T_6^{\beta_1} = \emptyset$ **then**
            $(\mathcal{P}_3^{\beta_1})'.Add(N^{\beta_1})$;
          else
            $[x]_3^{\beta_1}.Add(N^{\beta_1})$;
          if $N^{\beta_1} \cap T_3^{\beta_1} \neq \emptyset$ **then**
            foreach $x' \in N^{\beta_1} \bigcap T_3^{\beta_1}$ **do**
              if $x \rightarrow x'$ **then**
                $[x]_3^{\beta_1}.Add(x')$;
              else
                $(\mathcal{P}_3^{\beta_1})'.Add(\{x'\})$;
            if $N^{\beta_1} - T_3^{\beta_1} \neq \emptyset$ **then**
              Calculate the set
                $T_{N^{\beta_1}}$ the new clusters that emerge in
                $N^{\beta_1} - T_3^{\beta_1}$;
              $\mathcal{P}_3^{\beta_1}.Add(T_{N^{\beta_1}})$;
  $(\mathcal{P}_2^{\beta_1})'.Add([x]_2^{\beta_1})$;
  $(\mathcal{P}_3^{\beta_1})'.Add([x]_3^{\beta_1})$;
end

Algorithm 5. Incremental nested partition algorithm: part 5

Input: $\mathcal{P}_3^{\beta_1}, \mathcal{P}_3^{\beta_2}$
Output: $(\mathcal{P}_3^{\beta_1})', (\mathcal{P}_3^{\beta_2})'$
begin
  $(\mathcal{P}_3^{\beta_1})' = (\mathcal{P}_3^{\beta_2})' = \emptyset,\ [x]_3^{\beta_1} = [x]_3^{\beta_2} = \{x\}$;
  foreach $N^{\beta_1} \in \mathcal{P}_3^{\beta_1}$ **do**
    if $N^{\beta_1} \bigcap T_3^{\beta_1} = \emptyset$ **then**
      if $N^{\beta_1} \bigcap T_5^{\beta_1} \bigcap T_6^{\beta_1} = \emptyset$ **then**
        $(\mathcal{P}_3^{\beta_1})'.Add(N^{\beta_1})$;
        foreach $N^{\beta_2} \subseteq N^{\beta_1}$ **do**
          $(\mathcal{P}_3^{\beta_2})'.Add(N^{\beta_2})$;
      else
        $[x]_3^{\beta_1}.Add(N^{\beta_1})$;
        foreach $N^{\beta_2} \subseteq N^{\beta_1}$ **do**
          if $N^{\beta_2} \bigcap T_5^{\beta_2} \bigcap T_6^{\beta_2} = \emptyset$ **then**
            $(\mathcal{P}_3^{\beta_2})'.Add(N^{\beta_2})$;
          else
            $[x]_3^{\beta_2}.Add(N^{\beta_2})$;
    else
      foreach $x' \in N^{\beta_1} \bigcap T_3^{\beta_1}$ **do**
        if $x \to x'$ **then**
          $[x]_3^{\beta_1}.Add(x')$;
        else
          $(\mathcal{P}_3^{\beta_1})'.Add(\{x\})$;
      if $N^{\beta_1} - T_3(\beta_1) \neq \emptyset$ **then**
        Compute the set $T_{N^{\beta_1}})$ of new clusters that emerge in
        $N^{\beta_1} - T_3(\beta_1)$;
        $(\mathcal{P}_3^{\beta_1})'.Add(T_{N^{\beta_1}})$;
      foreach $N^{\beta_2} \subseteq N^{\beta_1}$ **do**
        if $N^{\beta_2} \bigcap T_5^{\beta_2} \bigcap T_6^{\beta_2} = \emptyset$ **then**
          if $N^{\beta_2} \bigcap T_3^{\beta_2}$ **then**
            $(\mathcal{P}_3^{\beta_2})'.Add(N^{\beta_2})$;
          else
            foreach $x'' \in N^{\beta_2}$ **do**
              if $x \to x''$ **then**
                $[x]_3^{\beta_2}.Add(x'')$;
              else
                $(\mathcal{P}_3^{\beta_2})'.Add(\{x''\})$;
              if $N^{\beta_2} - T_3(\beta_2) \neq \emptyset$ **then**
                Compute the set $T_{N^{\beta_2}}$ of new clusters that emerge in
                $N^{\beta_2} - T_3(\beta_2)$;
                $(\mathcal{P}_3^{\beta_2})'.Add(N^{\beta_2})$;
        else
          $[x]_3^{\beta_2}.Add(N^{\beta_2})$;
  $(\mathcal{P}_3^{\beta_j})'.Add([x]_3^{\beta_j}) j = 1, 2$;
end

5. The main algorithm

The proposed algorithm is based on the five preliminary algorithms exposed in the previous section and works roughly as follows:

Let $\beta_1, \beta_2, \ldots \beta_n, \beta_{n+1}, \ldots, \beta_{n+p}, \beta_{n+p+1}, \beta_{n+p+2}, \ldots, \beta_{n+p+s}$ be a decreasing sequence of similarity values. From this sequence, the input of our algorithm is the set $\mathcal{X}$, the new object x which arrives to the collection, and a sequence of nested partitions $\mathcal{P}_1^{\beta_1}, \mathcal{P}_1^{\beta_2}, \ldots, \mathcal{P}_1^{\beta_n}, \mathcal{P}_2^{\beta_n}, \mathcal{P}_2^{\beta_{n+1}}, \ldots, \mathcal{P}_2^{\beta_{n+p}}, \mathcal{P}_3^{\beta_{n+p}}, \mathcal{P}_3^{\beta_{n+p+1}}, \ldots, \mathcal{P}_3^{\beta_{n+p+s}}$ of $\mathcal{X}$, where

$$\mathcal{P}_i^{\beta_j} = \begin{cases} c_1(\Gamma, \beta_j), & i = 1 \wedge 1 \leq j \leq n; \\ c_2(\Gamma, \beta_j), & i = 2 \wedge n \leq j \leq n+p; \\ c_3(\Gamma, \beta_j), & i = 3 \wedge n+p \leq j \leq n+p+s. \end{cases}$$

When x arrives, the algorithm exploits the results exposed in previous sections for updating the hierarchy of clustering. The output of the algorithm is the updated sequence $(\mathcal{P}_1^{\beta_1})', (\mathcal{P}_1^{\beta_2})', \ldots, (\mathcal{P}_1^{\beta_n})', (\mathcal{P}_2^{\beta_n})', (\mathcal{P}_2^{\beta_{n+1}})', \ldots, (\mathcal{P}_2^{\beta_{n+p}})', (\mathcal{P}_3^{\beta_{n+p}})', (\mathcal{P}_3^{\beta_{n+p+1}})', \ldots, (\mathcal{P}_3^{\beta_{n+p+s}})'$ of nested partitions of $\mathcal{X}'$. This is, given a sequence of nested partitions where the n firsts are obtained by using the clustering criterion c_1, the next $p+1$ by using the criterion c_2 and the last $s+1$ by using the criterion c_3; the algorithm operates for update this sequence when x arrives. Observe that each level of this hierarchy can be obtained independently of other (by computing the connected component of the correspondent similarity graph) and, different clustering criteria are used. Then, the general steps are:

1. Compute the firsts $n-1$ updated partitions (yielded by the clustering criterion c_1 with different similarity thresholds) by using Algorithm 1.
2. Compute the nth and the $(n+1)$th updated partitions (yielded by the clustering criteria c_1 and c_2, respectively, both with the same similarity threshold) by using Algorithm 2.
3. Compute the update partitions from the $(n+2)$th until the $(n+s-1)$th partition (yielded by the clustering criterion c_2 with different similarity thresholds) by using Algorithm 3.
4. Compute the $(n+s)$th and the $(n+s+1)$th updated partitions (yielded by the clustering criteria c_2 and c_3, respectively, both with the same similarity threshold) by using Algorithm 4.
5. Compute the update partitions from the $(n+s+2)$th until the $(n+s+p)$th partition (yielded by the clustering criterion c_3 with different similarity thresholds) by using Algorithm 5.

Remark 1. Several previously reported algorithms can be seen as particular cases of the proposed algorithm. By example, if we limit ourselves only to Part 1 and we consider the full and increasingly ordered list of similarity values, then we obtain an incremental variant of the single-link algorithm; whereas, if we do a closer inspection to the algorithm proposed in [17], we can see that their algorithm is obtained from our algorithm if we take $n=1$ and $p=s=1$.

6. Experiments and results

This section is devoted to experimental tests. In order to show the suitability and capability of our method we describe the experiments in several databases. First, we focus on showing some characteristics of the clusterings obtained with our method.

For this, we utilized small databases of UCI repository [30]. In second place, we have chosen the topic detention problem, considering that the news collection are frequently updated and that hierarchical structuralization could provide a good detection of the topics.

6.1. Results and discussion with UCI repository database

The description of the databases used in this section is showed in Table 1. Either in Zoo or Abalone database, objects are described in terms of nominal, ordinal, and numerical features simultaneously, whereas in Breast Cancer Wisconsin, objects are only described in terms of numerical features. For Zoo and Abalone database, a weighted sum of δ-kernels (one for each feature) is used as similarity function, whereas for Breast Cancer Wisconsin, the polynomial kernel of degree 1 is used (see [31] for definition of polynomial kernels).

Taking into account the theoretical comparative analysis with single-link (SL) and complete-link (CL) algorithms that we have made during the development of the paper and the fact that several hierarchical clustering algorithms are enhancements of the single-link and complete-link algorithms, in this section, we compare the results of our method with the results of these algorithms.

There is a large amount of measures for evaluating the clustering results [32,33]. In order to evaluate our experimental results, we have chosen *F*-micro, *F*-macro, and *F*-overall measures. The *F*-measures [33] are external evaluation measures. That is, these functions measure how close the clustering yielded by the algorithm is to the "natural" clustering. For cluster C_i of the clustering yielded by the algorithm and a cluster C_j belonging to the "natural" clustering we can calculate the precision $P(i,j)=n_{ij}/n_j$ and recall $P(i,j)=n_{ij}/n_j$ factors, being $n_{ij}=|C_i \bigcap C_j|$,

Table 1
UCI repository databases description.

Database	Objects	Features	Classes
Zoo	101	16	7
Breast Cancer Wisconsin	569	32	2
Iris	150	4	2

Table 2
Algorithms performance using *F*-measures.

DB/algs.	SL	CL	INPM	SL	CL	INPM	SL	CL	INPM
Zoo	0.352	0.591	**0.809**	0.403	0.419	**0.821**	0.435	0.469	**0.881**
Breast Cancer	0.683	0.753	**0.789**	0.701	0.752	**0.742**	0.719	0.763	**0.810**
Iris	0.480	0.526	**0.756**	0.506	0.524	**0.749**	0.518	0.524	**0.749**

Table 3
Two levels of the hierarchy imposed by our method in Zoo database.

Animal	Type	CC	SC	Animal	Type	C	SCC	Animal	T	C	SCC
aardvark	1	1	1	buffalo	1	1	14	crab	7	6	26
bear	1	1	1	deer	1	1	14	starfish	7	6	26
girl	1	1	2	elephant	1	1	14	octopus	7	6	27
gorilla	1	1	3	giraffe	1	1	14	crayfish	7	6	28
wallaby	1	1	3	oryx	1	1	14	lobster	7	6	28
squirrel	1	1	3	mole	1	2	15	slug	7	6	29
fruitbat	1	1	4	opossum	1	2	15	worm	7	6	29
vampire	1	1	4	scorpion	7	3	16	crow	2	7	30
hare	1	1	5	bass	4	3	17	hawk	2	7	30
vole	1	1	5	catfish	4	3	17	duck	2	7	31
cavy	1	1	6	chub	4	3	17	swan	2	7	31
hamster	1	1	6	herring	4	3	17	flamingo	2	7	31
calf	1	1	7	piranha	4	3	17	ostrich	2	7	31
goat	1	1	7	stingray	4	3	18	rhea	2	7	31
pony	1	1	7	dogfish	4	3	19	kiwi	2	7	31
reindeer	1	1	7	pike	4	3	19	penguin	2	7	31
pussycat	1	1	8	tuna	4	3	19	vulture	2	7	31
boar	1	1	9	frog	5	3	20	gull	2	7	32
cheetah	1	1	9	frog	5	3	20	skimmer	2	7	32
leopard	1	1	9	newt	5	3	20	skua	2	7	32
lion	1	1	9	toad	5	3	20	lark	2	7	33
lynx	1	1	9	tuatara	3	3	20	pheasant	2	7	33
mongoose	1	1	9	slowworm	3	3	20	sparrow	2	7	33
polecat	1	1	9	pitviper	3	3	20	wren	2	7	33
puma	1	1	9	seasnake	3	3	21	tortoise	3	7	34
raccoon	1	1	9	carp	4	4	22	flea	6	8	35
wolf	1	1	9	haddock	4	4	23	termite	6	8	35
mink	1	1	10	seahorse	4	4	23	gnat	6	8	36
platypus	1	1	11	sole	4	4	23	ladybird	6	8	36
seal	1	1	12	chicken	2	5	24	housefly	6	8	37
sealion	1	1	12	dove	2	5	24	moth	6	8	37
dolphin	1	1	13	parakeet	2	5	24	honeybee	6	8	38
porpoise	1	1	13	clam	7	6	25	wasp	6	8	38
antelope	1	1	14	seawasp	7	6	25				

The level denoted by CC was obtained from the connected components of $\overline{G^{\max}_{(\mathcal{X},\Gamma,0.75)}}$ whereas that it denoted by SCC was obtained from the strongly connected components of $G^{\max}_{(\mathcal{X},\Gamma,0.75)}$.

$n_i = |C_i|$, and $n_j = |C_j|$. The *F*-measures combine the precision and recall factors for obtaining an evaluation of the quality of the clustering.

From left to right, Table 2 shows the *F*-micro, *F*-macro, and *F*-overall measure values for single-link, complete-link and our method, respectively. As it can be noticed, the value of *F*-measures for our proposed algorithm is in all cases greater than the values of the other considered algorithms, and moreover, they always show considerably highest values. From this, we can conclude that, in these cases, our algorithm captures the structure of these databases better than single-link and complete-link algorithms. Below, we show two levels captured by our algorithm in Zoo database.

Let us now concentrate on the results obtained using Zoo database. As it can be seen in Table 3, there are levels of the hierarchy imposed by our methods which capture almost exactly the real structure of the data. The levels corresponding to the criterion c_1 (connected components of $G_{(\mathcal{X},\Gamma,\beta_0)}$) does not reveal relevant information, in most of cases a unique clusters is given. However, the levels corresponding to the criterion c_2 (connected components of $G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}$) capture almost exactly the real structure of the data, whereas the criterion c_3 (strongly connected components of $G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}$) reveals substructures present in the data. By example, let us analyze the class of mammals (Type 1). Excepting for the mole and the opossum, the rest of mammals were assigned by the connected components of $G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}$ to the same cluster and there is a cluster whose unique elements are mole and opossum. On the other hand, the criterion c_3 capture almost perfectly the subclass of the felines as well as the aquatic mammals.

In general terms, the criterion c_1 tends to yield elongated and straggled cluster. Due to, this criterion tends not to offer relevant information in small databases. The defect of this criterion could be a result of the chaining effect it suffers. However, in large datasets this criterion could be very useful for capturing wide clusters as well as upper-structures. On the other hand, the c_2 criterion is, from our points of view, the best independent criterion of the three partition criteria analyzed in this paper.

The clusters obtained by this criterion are more compact and, in general, it tends to form clusters with higher intra-cluster similarity. By its part, c_3 criterion gives very small clusters. The elements belonging to the same cluster obtained with this criterion are very similar to each other. Our method exploits these features for capturing different structures present in the data. It offers a hierarchy of diverse clusterings which give a panoramic view of data. We think that this method is more versatile and flexible than single and complete-link.

6.2. News collections

Finally, we experimented our algorithm on topic detection problems. Several algorithms such as star algorithm (S) [34], the extended star algorithm (ES) [35], ACONS algorithm (AC) [36], algorithms based on compact sets [27], among other, are in the state of the art. Also, classical algorithms such as k-means, single and complete-link have been used in this task. In this section, we expose and compare the results of the algorithm proposed by us in this paper and the results of some of these algorithms.

In order to carry out this experiment, we use two databases: the AFP collection [37], which contains 695 articles published by the AFP agency during 1994, and the TDT2 collection version 4.0 [38], which contains 9824 news stories. A more detailed description of these databases is showed in Table 4.

Table 4
AFP and TDT news collections description.

Collection	Font	Documents	Terms	Topics
AFP	TREC-5	695	12 575	25
TDT	TDT2	9824	55 112	193

Table 5
Algorithms performance in AFP database.

F-measures/algs.	SL	CL	AC	S	ES	INPM
F-micro	0.580	0.620	0.681	0.700	0.704	**0.713**
F-macro	0.601	0.600	0.643	0.664	0.688	**0.734**
F-overall	0.616	0.642	0.672	0.699	0.711	**0.738**

Table 6
Algorithms performance in TDT database.

F-measures/algs.	SL	CL	AC	S	ES	INPM
F-micro	0.471	0.462	0.754	**0.782**	0.743	0.775
F-macro	0.501	0.482	0.714	**0.721**	0.708	0.694
F-overall	0.531	0.511	0.834	**0.836**	0.793	0.801

Here, we follows the traditional vector space model for obtaining the similarity space. The documents are represented by frequency terms vectors in which stop words (articles, prepositions, adverbs) are not considered. The polynomial kernel of degree 1 is used as similarity function. An excellent presentation of this model following a kernel method approach can be found in [39].

Analogously to previous section, we use *F*-measures for evaluating the results. These measures compare the system-generated clusters with the topics identified by human annotators. As it can be seen in previous section these measures combine the precision and recall factors. Tables 5 and 6 show the *F*-measures values in AFP and TDT databases, respectively.

As we can see, excepting for single-link and complete-link, the results of the algorithms considered in this experiment are comparable. The complete-link and single-link were overcame by the star algorithms and ACONS algorithm, as we expect from the experiments carried out by Aslam et al. [34] and results showed in [36]. On the other hand, the results of our algorithm are better than the results of all algorithms in the case of AFP database whereas that in TDT database are comparable with the results obtained by the star algorithms and better than all remaining algorithms.

In the case of the proposed algorithm, we can perceive that no criteria can capture a significant number of topics very well, due to the broadness. However, the incremental nested partition method used of the particular properties of each criterion for capturing topics with different broadness. As the connected components of $G_{(\mathcal{X},\Gamma,\beta_0)}$ obtains "straggly" clusters due to the chaining effect, it is better at capturing broad topics. The connected components of $\overline{G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}}$ tend to form clusters with higher intra-cluster similarity than the connected components of G_{β_0}, well suited for capturing not very wide topics. Finally, the strongly connected components of $G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}$ create the clusters with the highest intra-cluster similarity, those clusters tend to be very small, and it is better at capturing small topics.

Table 7
Algorithm's running time.

Algorithm	SL	CL	AC	S	ES	INPM
Cost	$O(\tau n^2)$	$O(\tau n^2)$	$O(\tau n^2)$	$O(\tau n^2)$	$O(\tau n^2)$	$O(\tau n^2)$

6.3. Computational issues

When dealing with incremental collections, we must consider the size that can reach them. If the size becomes considerably large, we could be motivated to choose an algorithm that have not the best performance in classification, but it is much more efficient from the computational point of view. For this reason we make a brief computational cost analysis of the algorithm presented in this paper. In order to simplify the analysis, we assume a sole similarity threshold β_0, $n=|\mathcal{X}|$, the cost of compute the values of Γ is denoted by τ and a graph based implementation.

The incremental nested partition method has three main steps:

1. Compare the new object x with each element of the collection $\mathcal{X}$.
2. Compute the similarity graphs $G_{(\mathcal{X}',\Gamma',\beta_0)}$, $\overline{G^{\max}_{(\mathcal{X}',\Gamma',\beta_0)}}$, $\overline{\overline{G^{\max}_{(\mathcal{X}',\Gamma',\beta_0)}}}$ from the same graphs for $\mathcal{X}$, respectively.
3. Compute the connected components for each one of three graph obtained in (2).

The step (1) is $O(\tau n)$. In the step (2), for each vertex x_i belonging to each graph, one of the following situations is given: (1) the set of adjacent vertexes is kept $O(1)$, (2) the adjacent vertexes of x_i are removed $O(1)$, and x become the unique adjacent vertex of x_i $O(1)$, (3) x is added to the set of adjacent vertexes of x_i $O(1)$. As we can see they are all constant operations conducted at the n vertices, and hence this step is $O(n)$. The computation of the connected component of each graph is a text book procedure which is $O(n+|E|)$ being E the edges set. For the graphs considered in this paper, we have $n\approx|E|$, because in general, they are sparse. Observe that the adjacent vertexes to a fix vertex x_i in the maximum similarity graphs are those with maximum similarity value, therefore x_i has in most cases a single adjacent vertex. This tells us that the step (3) has expected running time of $O(n)$. Hence, the procedure of add a new object is $O(\tau n)$ and the total cost for insert n objects is $O(\tau n^2)$.

Table 7 presents the running time of every algorithm utilized in the experiments. Observe that all of these algorithms are quadratics. The advantage of our method is precisely in incremental problems. Each time that a new object x arrives, the running time of the algorithm is linear with respect to the number of objects already present in the database when x arrives. All other algorithms repeat the same process each time a new object arrives, therefore the cost of inserting a new object remains quadratic.

Last experiment exemplified the behavior of the algorithm regarding to the time (measured in seconds) in the TDT database. The implementation of the algorithms was accomplished in the java programming language, in a PC with a microprocessor Intel (R) Core (TM) 2 Quad CPU with 2.5 GHz of speed. First, we count the running time of INPM algorithm adding the first $k*1000$ objects from the database TDT. Then, we calculate the running time of INPM algorithm adding 1000 objects, but assuming that we have n objects in the database. That is, in the first case we start with the empty database, while the second case the database already has n elements. Fig. 1 shows the results of this experiment, denoting by INPM(0) and INPM(n) the first case and second case, respectively. As it can be observed, in less than 30 s were inserted 9000 objects and the curve that is shown behaves like a quadratic function. However, the curve of the time to insert 1000 objects in the database already having n behaves like a linear function.

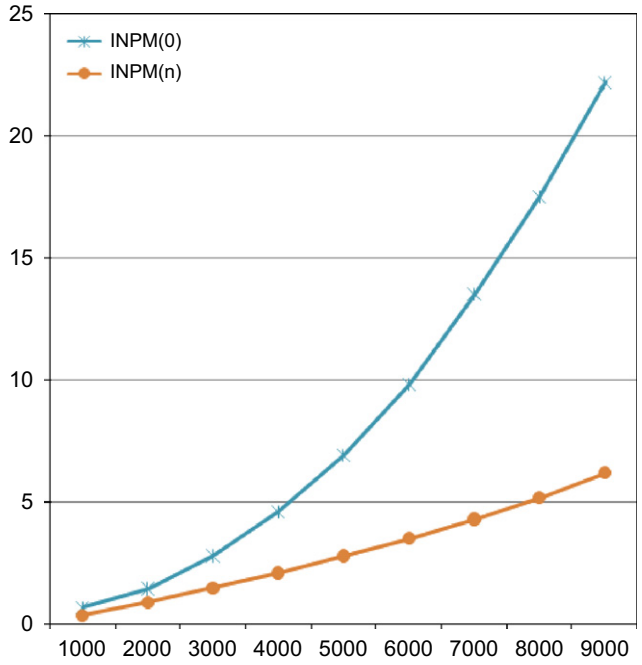


Fig. 1. Time in TDT database.

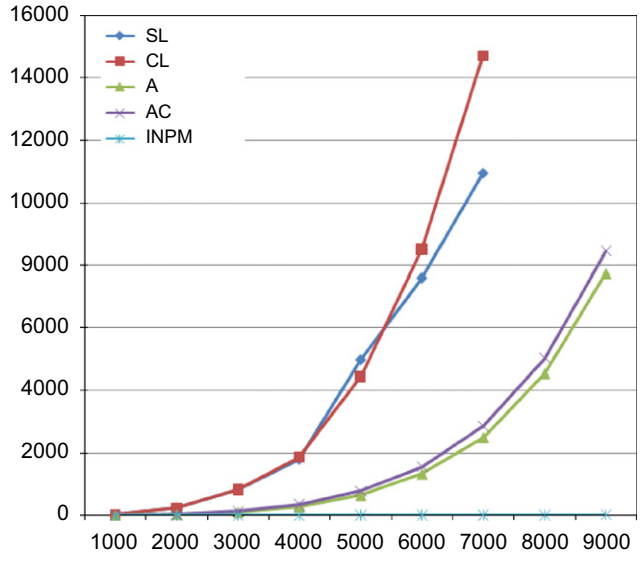


Fig. 2. Time in TDT database.

Finally, we use again TDT database to compare the real time cost of all those algorithms in the evaluation experiments. Fig. 2 shows the result of this experiment. As can be appreciated in this database our algorithm is considerably faster than other algorithms. This example illustrates the suitability of our algorithm in incremental problems and reinforces our previous analysis.

7. Conclusions and feature works

The need to extract underlaying information from dynamic datasets appears frequently in many real world problems. In this sense, the clustering algorithms are an essential tool, and within them, the hierarchical algorithms play an important role. Motivated by this problematic, in this paper we introduce an incremental nested partition method based on mathematical properties of three different partition criteria: the connected components of the graph $G_{(\mathcal{X},\Gamma,\beta_0)}$, the connected components of the graph $\overline{G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}}$ and the strongly connected components of the graph $G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}$. Having analyzed the results of this paper, we arrive to the following conclusions:

1. Unlike most of hierarchical algorithms, our method offers a sequence of nested partitions; each partition can be obtained from different partition criteria. Besides that our method exploits mathematical properties which are given in terms of lemmas, propositions, and theorems, for giving a theoretical ground for the use of these algorithms in applications such as biology taxonomy, image segmentation, database indexation, topic detection, regionalization, as well as in the geo-sciences in general. Moreover, it assures properties like uniqueness of the partitions in each level and the order invariance for their construction and allows a better understanding and interpretation of the results.
2. In view of Proposition 1 the partition of $\mathcal{X}$ obtained from the strongly connected components of the graph $G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}$ is the same partition imposed by the connected components of the graph $\overline{G^{\max}_{(\mathcal{X},\Gamma,\beta_0)}}$ and therefore, every level of the hierarchy of clusterings can be obtained if we compute the connected components of the respective graph. This fact also implies that the number of clusters of each level can be previously known without the execution of the procedure. The last is a trivial consequence of the number of connected components of a graph is exactly equal to the multiplicity of 0 as eigenvalue of its Laplacian matrix [40].
3. For any dataset, the hierarchy of clustering that single-link algorithm impose over it, it is always possible to be obtained with our algorithm. Moreover, experimental results show that the hierarchy of clusterings imposed by our algorithm is frequently better than the hierarchies of clusterings captured by single-link and complete-link algorithms.
4. The algorithm proposed in [17] is a particular case of our algorithm.
5. Experimental results into news collection show that the use of different partition criteria allow to capture topics with different broadness and to improve the obtained results with classical and recent algorithms for topics detection.

On the other hand, future works include three main tasks:

1. Extend the method to the fuzzy and conceptual clustering procedures.
2. Study the relationship of our method with single-link and complete-link using the cophenetic matrix and investigate the ultrametric inequality for the cophenetic proximities of our method.
3. Exploit the properties of the lattice of partitions for the obtention of hierarchies of clustering.

References

[1] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Comput. Surv. 31 (3) (1999) 264–323.
[2] J.A. Hartigan, M.A. Wong, Algorithm AS 136: a k-means clustering algorithm, Appl. Stat. 28 (1) (1979) 100–108.
[3] R.A. Redner, H.F. Walker, Mixture densities, maximum likelihood and the em algorithm, SIAM Rev. 26 (2) (1984) 195–239.
[4] J.F. Martínez, J. Ruiz-Shulcloper, M.S. Lazo, Structuralization of universes, Fuzzy Sets Syst. 112 (3) (2000) 485–500.
[5] P.H.A. Sneath, R.R. Sokal, Numerical Taxonomy: The Principles and Practice of Numerical Classification, W.H. Freeman, San Francisco, 1973.
[6] L. Kaufman, P. Rousseeuw, Finding Groups in Data—An Introduction to Cluster Analysis, in: Wiley Series in Probability and Statistics, Wiley, New York, 2005.
[7] C.K. Bayne, J.J. Beauchamp, C.L. Begovich, V.E. Kane, Monte Carlo comparisons of selected clustering procedures, Pattern Recognition 12 (2) (1980) 51–62.
[8] H. Qiu, E.R. Hancock, Clustering and embedding using commute times, IEEE Trans. Pattern Anal. Mach. Intell. 29 (11) (2007) 1873–1890.
[9] R.P.W. Duin, E. Pekalska, Open issues in pattern recognition, Comput. Recogn. Syst. 30 (2005) 27–42.
[10] R. Kothari, D. Pitts, On finding the number of clusters, Pattern Recognition Lett. 20 (4) (1999) 405–416.
[11] Y. Man, I. Gath, Detection and separation of ring-shaped clusters using fuzzy clustering, IEEE Trans. Pattern Anal. Mach. Intell. 16 (8) (1994) 855–861.
[12] A.P. Topchy, A.K. Jain, W.F. Punch, Clustering ensembles: models of consensus and weak partitions, IEEE Trans. Pattern Anal. Mach. Intell. 27 (12) (2005) 1866–1881.
[13] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.
[14] J. Liang, G. Li, Hierarchical clustering algorithm based on granularity, in: GRC '07: Proceedings of the 2007 IEEE International Conference on Granular Computing, IEEE Computer Society, Washington, DC, USA, 2007, pp. 429–431.
[15] O. Maqbool, H.A. Babri, Hierarchical clustering for software architecture recovery, IEEE Trans. Software Eng. 33 (11) (2007) 759–780.
[16] N. Basalto, R. Bellotti, F. De Carlo, P. Facchi, E. Pantaleo, S. Pascazio, Hausdorff clustering, ArXiv e-prints 801.
[17] J. Correa-Morris, J. Ruiz-Shulcloper, A. Pons-Porrata, D.L. Espinosa-Isidrón, Incremental nested partition method, in: ICPR, 2008, pp. 1–4.
[18] J.C. Duque, R. Ramos, J. Suriach, Supervised regionalization methods: a survey, Int. Regional Sci. Rev. 30 (3) (2007) 195–220.
[19] L. Shi, S. Ólafsson, Nested partitions method for global optimization, Oper. Res. 48 (3) (2000) 390–407.
[20] L. Shi, S. Ólafsson, Nested partitions method for stochastic optimization, Methodol. Comput. 2 (3) (2002) 271–291.
[21] A.K. Hartmann, A. Mann, W. Radenbach, Solution-space structures of (some) optimization problems, J. Phys. Conf. Series 9 (2008) 012011.
[22] S. Ólafsson, J. Yang, Intelligent partitioning for feature selection, INFORMS J. Comput. 17 (3) (2005) 339–355.
[23] V. Lakshmanan, V.E. DeBrunner, R. Rabin, Nested partitions using texture segmentation, in: SSIAI '02: Proceedings of the Fifth IEEE Southwest Symposium on Image Analysis and Interpretation, IEEE Computer Society, Washington, DC, USA, 2002, p. 153.
[24] A. Nowak, A. Wakulicz-Deja, S. Bachliński, Optimization of speech recognition by clustering of phones, Fundam. Inf. 72 (1–3) (2006) 283–293.
[25] S. Ólafsson, X. Li, S. Wu, Operation research and data mining, Eur. J. Oper. Res. 187 (3) (2002) 1429–1448.
[26] J. Ruiz-Shulcloper, G.S. Díaz, M.A. Abidi, Clustering in mixed incomplete data, Heuristics & Optimization for Knowledge Discovery, 2002, pp. 88–106.
[27] R. Gil-García, J.M. Badía-Contelles, A. Pons-Porrata, Dynamic hierarchical compact clustering algorithm, in: CIARP, 2005, pp. 302–310.
[28] A. Pons-Porrata, R. Berlanga-Llavori, J. Ruiz-Shulcloper, On-line event and topic detection by using the compact sets clustering algorithm, J. Intell. Fuzzy Syst. 12 (3,4) (2002) 185–194.
[29] F.W. Anderson, K.R. Fuller, Rings and Categories of Modules, Springer, Berlin, 1992.
[30] C.J. Merz, P.J. Murphy, D.W. Aha, Uci repository of machine learning database. University of California at Irvine, Department of Computer Sciences, 1996.
[31] B. Scholkopf, A.J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, MIT Press, Cambridge, MA, USA, 2002.
[32] L.I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, Wiley-Interscience, New York, 2004.
[33] C.J. Van Rijsbergen, Information Retrieval, second ed., Department of Computer Science, University of Glasgow, 1979.
[34] J.A. Aslam, E. Pelekhov, D. Rus, The star clustering algorithm for static and dynamic information organization, J. Graph Algorithms Appl. 8 (2004) 95–129.
[35] R. Gil-García, J.M. Badía-Contelles, A. Pons-Porrata, Extended star clustering algorithm, in: CIARP, 2003, pp. 480–487.
[36] A.G. Alonso, A.P. Suárez, J.E. Medina-Pagola, Acons: a new algorithm for clustering documents, in: CIARP, 2007, pp. 664–673.
[37] ⟨http://trec.nist.gov⟩.
[38] ⟨http://www.nist.gov/speech/tests/tdt.html⟩.
[39] J. Shawe-Taylor, N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge University Press, New York, NY, USA, 2004.
[40] J. Lurie, Review of spectral graph theory: by Fan R.K. Chung, SIGACT News 30 (2) (1999) 14–16.

About the Author—JYRKO CORREA MORRIS was born in Havana in 1982, he received his B. S. degree in Mathematic at Havana University in 2006. B.S. Jyrko Correa Morris was a young researcher at Advanced Technologies Applications Center for two years, until September 2008. At present, he is a doctoral student of the Institute for Pure and Applied Mathematics (IMPA), Rio de Janeiro, Brazil. The current pattern recognition research interests of B.S. Jyrko Correa are data clustering in particular cluster ensemble methods and issues related with the general theory of clustering. Besides that he is also interested in kernel methods and geometry.

About the Author—DUSTIN L. ESPINOSA ISIDRÓN was born in Havana in 1984, he received his B.S. degree in Computer Science at Havana University in 2008. Actually, he is a young researcher at Advanced Technologies Applications Center. His current research interests include pattern recognition and data mining.

About the Author—DENIS R. ÁLVAREZ NADIOZHIN was born in Russia in 1984, he received his B.S. degree in Computer Science at Havana University in 2008. At present, he is a professor at Mathematic Faculty of the Havana University. His current research interests include pattern recognition and data mining.