An Incremental Clustering Algorithm Based on Compact Sets with Radius α

Aurora Pons-Porrata[1], Guillermo Sánchez Díaz[2], Manuel Lazo Cortés[3], and Leydis Alfonso Ramírez[1]

[1] Center of Pattern Recognition and Data Mining, University of Oriente, Patricio Lumumba s/n, C.P. 90500 Santiago de Cuba, Cuba
aurora@app.uo.edu.cu, leydis@csd.uo.edu.cu
[2] Center of Technologies Research on Information and Systems, UAEH, Carr. Pachuca-Tulancingo Km. 4.5, C.P. 42084, Pachuca, Hgo., Mexico
sanchezg@uaeh.reduaeh.mx
[3] Institute of Cybernetics, Mathematics and Physics, 15 No. 551 Vedado, C.P. 10400, Havana, Cuba
mlazo@icmf.inf.cu

Abstract. In this paper, we present an incremental clustering algorithm in the logical combinatorial approach to pattern recognition, which finds incrementally the β_0-compact sets with radius α of an object collection. The proposed algorithm allows generating an intermediate subset of clusters between the β_0-connected components and β_0-compact sets (including both of them as particular cases). The evaluation experiments on standard document collections show that the proposed algorithm outperforms the algorithms that obtain the β_0-connected components and the β_0-compact sets.

1 Introduction

In some areas such as finance, banking, engineering, medicine and geosciences the amount of stored data has had an explosive increase [1]. In these areas, there are many instances where the description of objects is non-classical; that is, features are not numerical or exclusively categorical, and sometimes, with missing values (mixed data). Data Mining and Knowledge Discovery on Databases areas process data in order to extract knowledge from data sets [9]. An important tool to extract knowledge is clustering. Several non incremental techniques to obtain clusters of a mixed data set have been proposed [2].

On the other hand, static clustering methods (non incremental algorithms) mainly rely on having the whole object set ready before applying the algorithm. Unlike them, the incremental methods are able to process new data as they are added to the collection. Nowadays, there are many problems that require a clustering of dynamic object collections such as topic detection and tracking, web mining and others.

In the Logical Combinatorial Pattern Recognition approach some clustering criteria have been proposed [6]. These clustering criteria were used to solve

M. Lazo and A. Sanfeliu (Eds.): CIARP 2005, LNCS 3773, pp. 518–527, 2005.

real problems [4], using classical algorithms which generate and store similarity matrix between objects.

However, these algorithms are inapplicable when the data set is large or dynamic. For some of these criteria, incremental algorithms to process large data sets have been developed [8, 7]. The first one finds the β_0-connected components, but it could obtain clusters with low internal cohesion. The second one obtains the β_0-compact sets, but it could generate a great number of cohesive and small clusters.

In this paper, we use the β_0-compact sets with radius α [5]. This clustering criterion allows generating an intermediate subset of clusters between the β_0-connected components and the β_0-compact sets (including both of them as particular cases). Thus a new incremental algorithm in order to generate the β_0-compact sets with radius α of an object collection is introduced.

2 Some Basic Concepts

Let $U = \{O_1, \ldots, O_m\}$ be the universe of objects in study, described in terms of features $R = \{x_1, \ldots, x_n\}$. Besides, let $\beta(O_i, O_j)$ be a similarity function between objects O_i and O_j, and β_0 a similarity threshold defined by the user.

Definition 1. *We say that objects O_i and O_j are β_0-similar if $\beta(O_i, O_j) \geq \beta_0$. If $\forall\, O_j \in U$, $\beta(O_i, O_j) < \beta_0$ then O_j is a β_0-isolated object.*

Notation: Let us denote $\nu_i = max\{\beta(O_i, O_t)/O_t \in U \wedge O_t \neq O_i \wedge \beta(O_i, O_t) \geq \beta_0\}$.

Definition 2. *We say that O_j is α-max β_0-similar to O_i if $\beta(O_i, O_j) \geq \beta_0$ and $\beta(O_i, O_j) \geq \nu_i - \alpha$. In other case, O_i is β_0-isolated and we do not consider ν_i.*

Definition 3. *We call β_0-maximum similarity reduced neighborhood with radius α of an object O_i, and we denote it by $N^0(O_i; \beta_0, \alpha)$, the following set:*

$$N^0(O_i; \beta_0, \alpha) = \{\, O_j \in U : O_j \neq O_i \wedge\ \beta(O_i, O_j) \geq \beta_0 \wedge [\beta(O_i, O_j) \geq (\nu_i - \alpha) \vee \beta(O_j, O_i) \geq (\nu_j - \alpha)] \,\}$$

From definition, $\forall\, O_i \in U, O_i \notin N^0(O_i; \beta_0, \alpha)$. We call β_0-*maximum similarity neighborhood with radius α of an object O_i*, and we denote it by $N(O_i; \beta_0, \alpha)$, the set $N^0(O_i; \beta_0, \alpha) \cup \{O_i\}$.

The set $N(O_i; \beta_0, \alpha)$ contains to O_i, all its α-max β_0-similar objects, and those objects for which O_i is an α-max β_0-similar object.

From definition 3, an interesting property is the following:

Proposition 1. $O_i \in N(O_j; \beta_0, \alpha) \equiv O_j \in N(O_i; \beta_0, \alpha)$.

Proof. It is sufficient with doing explicit the expressions:

$O_i \in N(O_j; \beta_0, \alpha)\ \equiv\ \beta(O_j, O_i) \geq \beta_0 \wedge [\beta(O_j, O_i) \geq (\nu_j - \alpha) \vee \beta(O_i, O_j) \geq (\nu_i - \alpha)]$

$O_j \in N(O_i;\beta_0,\alpha) \equiv \beta(O_i,O_j) \geq \beta_0 \wedge [\beta(O_i,O_j) \geq (\nu_i - \alpha) \vee \beta(O_j,O_i) \geq (\nu_j - \alpha)]$

As β is a symmetric function, the equivalence is fulfilled. □

Definition 4. *Let $\delta \subseteq U$, $\delta \neq \emptyset$, δ is a β_0-compact set with radius α with respect to (wrt) β and β_0 if:*

i) $O_i \in \delta \Rightarrow N(O_i;\beta_0,\alpha) \subseteq \delta$.

ii) $\forall O_i, O_j \in \delta\ \exists\{N_{s_1}, N_{s_2}, \ldots, N_{s_q}\} : N_{s_1} = N(O_i;\beta_0,\alpha) \wedge N_{s_q} = N(O_j;\beta_0,\alpha) \wedge N_{s_p} \cap N_{s_{p+1}} \neq \emptyset,\ \forall\, p \in \{1,\ldots,q-1\}$, *being* $\{N_{s_1}, N_{s_2}, \ldots, N_{s_q}\}$ *a set of β_0-maximum similarity neighborhoods with radius α of objects in δ.*

iii) If $\{O_i\} = N(O_i;\beta_0,\alpha)$ *then* $\delta = \{O_i\}$ *is a degenerate β_0-compact set with radius α wrt β and β_0.*

The first condition states that each object O_i in δ has its α-max β_0-similar objects and those objects for which O_i is an α-max β_0-similar object in δ. The second condition means that δ is the smallest set that holds the condition i).

We will denote by $\delta(O)$ the β_0-compact set with radius α which the object O belongs.

From now on, we will use the expression (β_0,α)-compact set instead of β_0-compact set with radius α.

For any β_0 and α values, (β_0,α)-compact sets generate a partition of the universe of objects in study.

β_0-compact sets and β_0-connected components [6] are particular cases of (β_0,α)-compact sets, taking $\alpha = 0$ and $\alpha = \beta_M - \beta_0$, being $\beta_M = max\{\nu_i\}_{O_i \in U}$ respectively [5]. For each of them, incremental algorithms have been developed [8, 7].

Definition 5. *We will call graph based on the α-max β_0-similarity according to β to the directed graph $\Gamma_{U,\beta,\beta_0,\alpha}$ whose vertices are the objects of U, and there is an arc from the vertex O_i to the vertex O_j if O_j is an α-max β_0-similar object to O_i. We will denote by $G_{U,\beta,\beta_0,\alpha}$ the undirected graph associated to $\Gamma_{U,\beta,\beta_0,\alpha}$.*

From the previuos definition, we obtain that $N^0(O_i;\beta_0,\alpha)$ coincides with the set of adjacent vertexes to O_i in the graph $G_{U,\beta,\beta_0,\alpha}$.

Proposition 2. *The set of all (β_0,α)-compact sets of U coincides with the set of all connected components of graph $G_{U,\beta,\beta_0,\alpha}$.*

Proof. It is a direct consequence of definitions 4 and 5. Let $\delta = \{O_{i_1}, O_{i_2}, \ldots, O_{i_k}\}$ be a (β_0,α)-compact set of U. If $k = 1$, then $\delta = \{O_{i_1}\}$ is an isolated (β_0,α)-compact set, and $N^0(O_{i_1};\beta_0,\alpha) = \emptyset$, where O_{i_1} is an isolated vertex. Therefore, $\{O_{i_1}\}$ is a connected component in $G_{U,\beta,\beta_0,\alpha}$.

Now, if $k > 1$, where $N^0(O_{i_1};\beta_0,\alpha)$ is the adjacent vertex set of O_{i_1}, condition ii) of definition 4 guarantees that for any pair of objects $O_{i_l}, O_{i_j} \in \delta$, a path in $G_{U,\beta,\beta_0,\alpha}$ that connects these objects exists, and the associated subgraph of δ is connected.

In addition, condition i) of definition 4 guarantees that δ is not a subset of any connected component of graph $G_{U,\beta,\beta_0,\alpha}$, but δ is the same connected component. □

Definition 6. *Let $U' \subset U$ and $\delta \subset U'$ be a (β_0, α)-compact set of U'. Besides, let $O \in U \setminus U'$. We say that object O is connected with δ if there exists some object $O' \in \delta$ such that O is α-max β_0-similar to O' or O' is α-max β_0-similar to O.*

Proposition 3. *Let U', U and O be like above. If object O is not connected with δ, then δ is a (β_0, α)-compact set in $U' \cup \{O\}$.*

Proof. This is immediate from definition 6. As object O is not connected with δ, then $\delta \cup \{O\}$ does not satisfy (β_0, α)-compact set definition. □

Corollary 1. *If O is a β_0-isolated object, then the set of all (β_0, α)-compact sets in $U' \cup \{O\}$ is $\zeta \cup \{\{O\}\}$, where ζ is the set of all (β_0, α)-compact sets in U'. In this case, the graph $G_{U' \cup \{O\},\beta,\beta_0,\alpha}$ and the graph $G_{U',\beta,\beta_0,\alpha}$ differ in only one vertex.*

Let E be the set of edges of graph $G_{U,\beta,\beta_0,\alpha}$.

Proposition 4. *Let U', U, O and δ be like above. If O is connected with δ, and $E_{U',\beta,\beta_0,\alpha} \subset E_{U' \cup \{O\},\beta,\beta_0,\alpha}$ (i.e. new edges appear and no edge of $G_{U',\beta,\beta_0,\alpha}$ was broken by O), then $\delta \cup \{O\}$ is a (β_0, α)-compact set or it is a subset of a (β_0, α)-compact set in $U' \cup \{O\}$.*

Proof. As a consequence of the proposition 2, if δ is a (β_0, α)-compact set in U', then the subgraph associated to δ is a connected component of the graph $G_{U',\beta,\beta_0,\alpha}$. Let $\{O_{i_1}, O_{i_2}, \ldots, O_{i_r}\}$ be the objects connected with O by the new edges in the graph $G_{U' \cup \{O\},\beta,\beta_0,\alpha}$.

If $\{O_{i_1}, O_{i_2}, \ldots, O_{i_r}\} \subseteq \delta$, then O is added to this connected component, and therefore $\delta \cup \{O\}$ is a (β_0, α)-compact set in $U' \cup \{O\}$. Otherwise, if O is also connected with an object $O' \in \delta$, then $O' \in N(O; \beta_0, \alpha)$ and $\delta \cup \{O\}$ is not a (β_0, α)-compact set in $U' \cup \{O\}$, because it does not satisfy condition i) of definition 4. Nevertheless, as the subgraph associated to $\delta \cup \{O\}$ is connected, it is a subset of a (β_0, α)-compact set in $U' \cup \{O\}$. This (β_0, α)-compact set is the union of $\{O\}$ and all (β_0, α)-compact sets in U' to which O is connected. □

3 Incremental Clustering Algorithm

In this paper, we propose a new clustering algorithm that finds incrementally the (β_0, α)-compact sets of an object collection. This algorithm is based on the propositions explained above.

The algorithm stores the maximum β_0-similarity of each object O_i, and the set of objects connected to it in the graph $G_{U',\beta,\beta_0,\alpha}$, that is, the objects belonging to $N^0(O_i; \beta_0, \alpha)$. It stores, also, the similarity values with O_i for each object of $N^0(O_i; \beta_0, \alpha)$.

Every time that new object O arrives, its similarity with each object of existent (β_0, α)-compact sets is calculated and the graph $G_{U',\beta,\beta_0,\alpha}$ is updated. The arrival of O can change the current (β_0, α)-compact sets, because some new (β_0, α)-compact sets may appear, and others that already exist may disappear.

Therefore, after updating the graph $G_{U',\beta,\beta_0,\alpha}$ the (β_0, α)-compact sets are rebuilt starting from O, and the objects in the (β_0, α)-compact sets that become unconnected. The (β_0, α)-compact sets that do not include objects connected with O remain unchanged, by virtue of the Proposition 3. During the graph updating task the algorithm constructs the following sets:

ClustersToProcess: A (β_0, α)-compact set is included in this set if it has any object O_j that satisfies the following conditions:

1. The new object O is the most β_0-similar to O_j, and the objects that were α-max β_0-similar to O_j are not anymore; that is, its edges with O_j in the graph $G_{U',\beta,\beta_0,\alpha}$ are broken.
2. O_j had at least two α-max β_0-similar objects, in which its edges are broken, or O_j is α-max β_0-similar to at least another object in this (β_0, α)-compact set.

This set includes the (β_0, α)-compact sets that could lose its compactness when the objects with the previous characteristics are removed from the cluster. Thus, these (β_0, α)-compact sets must be reconstructed.

Example 1: Let be β_0=0.3 and α=0.1. As can be seen in Figure 1, the (β_0, α)-compact set C belongs to the set *ClustersToProcess*, because object O_1 satisfies the conditions mentioned above.

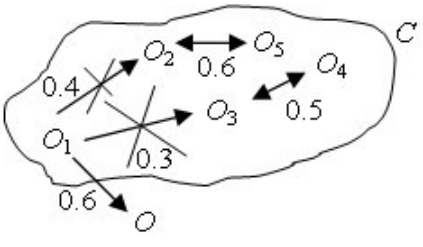

Fig. 1. A cluster that belongs to *ClustersToProcess*

ObjectsToJoin: An object O_j is included in this set if it satisfies the following conditions:

1. The new object O is the most β_0-similar to O_j, and the only object that was α-max β_0-similar to O_j is not anymore.
2. O_j is not α-max β_0-similar to any object of its (β_0, α)-compact set.

The objects in this set will be included in the same (β_0, α)-compact set as O, that is, $\delta(O)$. The (β_0, α)-compact set to which O_j belongs continues being a (β_0, α)-compact set when O_j is removed from it.

Example 2: Let be β_0=0.3 and α=0.1. The object O_1 belongs to the set *ObjectsToJoin*, as is illustrated in Figure 2. O_1 will belong to $\delta(O)$ and it must be removed from the (β_0, α)-compact set C. Also $C \backslash \{O_1\}$ is a (β_0, α)-compact set.

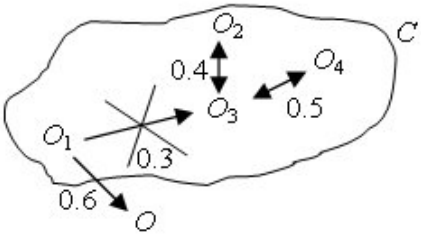

Fig. 2. An example of *ObjectsToJoin*

ClustersToJoin: A (β_0, α)-compact set is included in this set if it is not in *ClustersToProcess* and it has at least one object O_j that satisfies one of the following conditions:

1. O_j is α-max β_0-similar to the new object O.
2. O is α-max β_0-similar to O_j, and no edge of O_j in the graph $G_{U',\beta,\beta_0,\alpha}$ is broken.

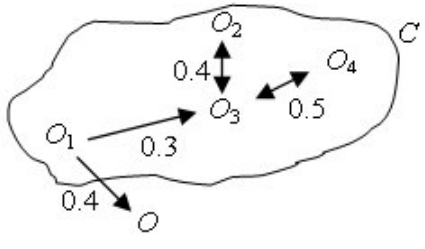

Fig. 3. A cluster that belongs to *ClustersToJoin*

All the objects in *ClustersToJoin* will be included in the same (β_0, α)-compact set as O. Notice that the clusters in *ClustersToJoin* are the (β_0, α)-compact sets that satisfy the Proposition 4.

Example 3: Let be β_0=0.3 and α=0.1. As can be seen in Figure 3, the (β_0, α)-compact set C belongs to the set *ClustersToJoin*, because the new object O is connected with it and no edge in C is broken.

3.1 The Incremental Algorithm

The main steps of the algorithm are the following:

1. Arrival of the new object O.
2. Updating of the graph $G_{U',\beta,\beta_0,\alpha}$.
   (a) For each object in the existent (β_0, α)-compact sets, its similarity with O is calculated.
   (b) The maximum β_0-similarity of each object in the graph $G_{U',\beta,\beta_0,\alpha}$, and the set of its α-max β_0-similar objects are updated.
   (c) The maximum β_0-similarity of O, and the set $N^0(O; \beta_0, \alpha)$ are determined.
   (d) The sets $ClustersToProcess$, $ClustersToJoin$ and $ObjectsToJoin$ are built.
   (e) Every time an object is added to $ObjectsToJoin$ it is removed from the (β_0, α)-compact set in which it was located before.
3. Reconstruction of the (β_0, α)-compact sets.
   (a) Let C be a set including O and all the objects included in the (β_0, α)-compact sets in $ClustersToProcess$.
   (b) Build the existing (β_0, α)-compact sets in C, and add them to the existing (β_0, α)-compact set list.
   (c) Add all the objects in $ObjectsToJoin$, and all the objects included in the (β_0, α)-compact sets of $ClustersToJoin$ to $\delta(O)$.
   (d) The (β_0, α)-compact sets in $ClustersToProcess$ and in $ClustersToJoin$ are removed from the existing (β_0, α)-compact set list.

The worst case time complexity of this algorithm is $O(n^2)$, since for each object, all the objects of existing clusters could be checked to find the most similar objects.

4 Evaluation

The effectiveness of the proposed clustering algorithm has been evaluated using four standard document collections, whose general characteristics are summarized in Table 1. Human annotators identified the topics in each collection.

In our experiments, the documents are represented using the traditional vectorial model. The terms of documents represent the lemmas of the words appearing in the texts. Stop words, such as articles, prepositions and adverbs are disregarded from the document vectors. Terms are statistically weighted using the term frequency. To account for documents of different lengths, the vector is normalized using the document length. We use the traditional cosine measure to compare the documents.

The source TREC was obtained of *http://trec.nist.gov*, TDT2 of *http://www.nist.gov/speech/tests/tdt.html*, and finally, Reuters-21578 of *http://kdd.ics.uci.edu*.

Table 1. Description of document collections

Collection	Source	N. of documents	N. of terms	N. of topics	Language
AFP	TREC-5	695	12330	25	Spanish
ELN	TREC-4	1997	39025	49	Spanish
TDT	TDT2	9824	55112	193	English
REU	Reuters-21578	10369	38367	120	English

There are many different measures to evaluate the quality of clustering. We adopt a widely used external quality measure: the Overall F1-Measure [3]. This measure compares the system-generated clusters with the manually labelled topics and combines the precision and recall factors. The higher the overall F1-measure, the better the clustering is, due to the higher accuracy of the clusters mapping to the topics.

Our experiments were focused on evaluating the quality of the clustering produced by GLC [8], Incremental Compact Clustering [7] and the proposed algorithm.

Table 2. Quality results obtained by clustering algorithms

Collection	Algorithm	Parameters	Overall F1-measure
AFP	GLC	$\beta_0 = 0.33$	0.65
	Compact set	$\beta_0 = 0.1$	0.43
	Proposed algorithm	$\beta_0 = 0.25, \alpha = 0.02$	**0.68**
ELN	GLC	$\beta_0 = 0.38$	0.21
	Compact set	$\beta_0 = 0.15$	0.30
	Proposed algorithm	$\beta_0 = 0.22, \alpha = 0.002$	**0.31**
TDT	GLC	$\beta_0 = 0.5$	0.57
	Compact set	$\beta_0 = 0.24$	0.25
	Proposed algorithm	$\beta_0 = 0.45, \alpha = 0.02$	**0.61**
REU	GLC	$\beta_0 = 0.67$	0.32
	Compact set	$\beta_0 = 0.1$	0.14
	Proposed algorithm	$\beta_0 = 0.5, \alpha = 0.04$	**0.49**

The obtained results for each collection are shown in Table 2. Second column contains the values that produce best results. The entries that are boldfaced correspond to the method that performed the best in each document collection.

Several observations can be made by analyzing the results in Table 2. First, in most collections the algorithm GLC obtains better results than Compact algorithm. However, our algorithm overcomes them in all collections. Finally, the best value of β_0 parameter in our algorithm is always greater than the best value of the Incremental Compact Algorithm, but it is always lesser than the β_0 value of the GLC algorithm.

5 Conclusions

In this paper, a new incremental clustering algorithm has been introduced. This algorithm is based on the incremental construction of existing β_0-compact sets with radius α in the object collection. It handles a clustering criterion that generating an intermediate subset of clusters between the β_0-connected components and β_0-compact sets (including both of them as particular cases). In this sense, the proposed algorithm is more restrictive than GLC algorithm, and at the same time, is more flexible than Incremental Compact algorithm.

Our algorithm allows the finding of clusters with arbitrary shapes, the number of clusters is not fixed a priori and it does not impose any restrictions to the representation space of the objects. Another advantage of this algorithm is that the generated set of clusters is unique, independently on the arrival order of the objects.

Our experiments with standard document collections have demonstrated the validity of our algorithm for document clustering tasks. The proposed algorithm overcomes the GLC algorithm and the Incremental Compact algorithm in all document collections.

The new algorithm can be used in tasks such as information organization, browsing, topic tracking and new topic detection. Although we employ our algorithm to cluster document collections, its use is not restricted to this area, since it can be applied to any problem of Pattern Recognition where clustering mixed objects can appear.

As future work, we will study the inclusion of this clustering criterion as clustering routine in a dynamic hierarchical clustering algorithm.

Acknowledgements. This work was financially supported by Institutional Program Research of UAEH (Mexico).

References

[1] Fayyad, U.; Piatetsky-Shapiro, G.; Smyth, P. and Uthurusamy, R.: *Advances in knowledge discovery in databases*, Cambridge, MIT Press, 1996.

[2] Jain, K. and Dubes, R.: *Algorithms for clustering data*, Prentice Hall, 1998.

[3] Larsen, B. and Aone, C.: Fast and Effective Text Mining Using Linear-time Document Clustering. In *Proceedings of KDD'99*, San Diego, California, pp. 16–22, 1999.

[4] Lopez-Caviedez, M.: A cities stratification tool in risk zones for the healt. MSc. Thesis, UAEH, Pachuca, Hgo. Mexico, 2004 (in Spanish).

[5] Lopez-Caviedez, M. and Sanchez-Díaz, G.: A new clustering criterion in pattern recognition. *WSEAS Transactions on Computers* 3(3), pp. 558–562, 2004.

[6] Martínez Trinidad, J. F.; Ruiz Shulcloper, J. and Lazo Cortés, M.: Structuralization of universes. *Fuzzy Sets and Systems* 112 (3), pp. 485–500, 2000.

[7] Pons-Porrata, A.; Berlanga-Llavori, R. and Ruiz-Shulcloper, J.: On-line event and topic detection by using the compact sets clustering algorithm. *Journal of Intelligent and Fuzzy Systems* (3-4), pp. 185–194, 2002.

[8] Sanchez-Díaz, G. and Ruiz-Shulcloper, J.: Mid mining: a logical combinatorial pattern recognition approach to clustering in large data sets. In *Proc. VI Ibero-American Symposium on Pattern Recognition*, Lisboa, Portugal, pp. 475–483, 2000.

[9] Sarker, R.; Abbass, H. and Newton, C.: Introducing data mining and knowledge discovery. *Heuristics & optimization for knowledge discovery*, Idea Group publishing, pp. 1–12, 2000.