PoBOC: an Overlapping Clustering Algorithm.
Application to Rule-Based Classification and Textual Data.

Guillaume Cleuziou and **Lionel Martin** and **Christel Vrain**[1]

Abstract. This paper presents the clustering algorithm PoBOC (Pole-Based Overlapping Clustering). It has two main characteristics: the number of final clusters is unknown *a priori* and PoBOC allows an object to belong to one or several clusters. Given a similarity matrix over a set of objects, PoBOC builds small and homogeneous sets of objects (the poles), and then it assigns the objects to the poles.

The clustering method is evaluated on two different research areas. First, on the Rule-Based Learning (RBL) task: classification rules are generated by organizing the instances of a class so that each cluster is covered with a single rule ; PoBOC is compared with different clustering methods and usual classifiers, on traditional datasets from the UCI repository. Otherwise, we observe the behaviour of PoBOC on the structuring of textual data in a semantic way. The efficiency of the proposed method on the two applications leads to conclude that PoBOC is also a general algorithm.

1 Introduction

Clustering is the task that consists in organizing a set of objects into classes, so that similar objects belong to the same cluster and dissimilar ones belong to different clusters. Several ways of clustering have been explored in the past (hierarchical, partitioning, graph-based, density-based methods,...) in various perspectives of application as for instance image segmentation and document clustering tasks [7]. Most of the clustering algorithms are *hard-clustering* techniques, each object is assigned to a single cluster. Conversely, the *fuzzy-clustering* methods [2] propose an organization in which each object participates to the definition of each cluster. This last approach is well-known for the richness of its description compared to hard-clustering methods, however, hard clusters are usually prefered for the simplicity of their definitions in a post-processing perspective.

We propose in this study a method which can be viewed as a compromise between *hard* and *fuzzy-clustering* approaches. Rather than assigning an object to only one cluster, this approach allows an object to belong to one or several clusters ; final clusters thus intersect. This type of algorithm is sometimes denoted as *soft-clustering* but we prefer using the term of *overlapping-clustering*. Most of clustering algorithms concern *hard* or *fuzzy*-clustering methods, nevertheless various overlapping clustering ones have been proposed. We can mention: partitioning methods with the axial-k-means algorithm [9], hierarchical methods with the pyramidal clustering [14] or clustering based on probabilistic models such as AutoClass [5]. The axial-k-means method concerns only data observed in a numerical space, pyramidal clustering does not allow a cluster to overlap with more than two other clusters and the clusters induced with Autoclass are often essentially elliptical in shape. The algorithm PoBOC (Pole-Based Overlapping Clustering) does not present the previous limitations, furthermore the number of clusters is not given *a priori*, like for Autoclass.

The evaluation of clustering algorithms is a difficult task which is usually solved by graphical validations or using quality criterion such as inter/intra-cluster similarities. We propose here an evaluation on two different applications. In the first one, PoBOC is used to organize events of a class in a supervised learning perspective. Thus, the proposed method is evaluated on the semantic clustering task which consists in structuring lexical units in a semantic perspective.

The paper is organized as follows: Section 2 presents a formal description of the clustering algorithm PoBOC ; Section 3 and 4 are respectively devoted to classification rules learning and semantic terms structuring. Finally, Section 5 presents a conclusion of this study and proposes perspectives of research.

[1] Laboratoire d'Informatique Fondamentale d'Orléans, Orléans, France
email: {cleuziou,martin,cv}@lifo.univ-orleans.fr

2 PoBOC : Pole-Based Overlapping Clustering

2.1 The clustering algorithm PoBOC

The algorithm PoBOC (Pole-Based Overlapping Clustering) takes as input a similarity matrix and builds a hierarchy of concepts in which an object may belong to several concepts. The four main steps are: (1) the search for *poles*, (2) the construction of a membership matrix of objects to poles, (3) the assignment of objects to one or several poles and (4) a hierarchical organization of the obtained groups.

The notion of *pole* is central in our approach: given a set of objects $X = \{x_1, \ldots, x_n\}$, a pole is a subset corresponding to an homegeneous area appearing in a region having an uniform density. The construction of poles is perfomed from the similarity graph:

Definition 2.1 *Let $X = \{x_1, \ldots, x_n\}$ and S a similarity matrix defined on $X \times X$. The* **similarity graph***, denoted by $G_S(X, V)$, is the graph defined by the set of vertices X and the set of edges V such that $(x_i, x_j) \in V$ iff:*

$$s(x_i, x_j) \geq max\{\frac{1}{n}\sum_{x_k \in X} s(x_i, x_k), \frac{1}{n}\sum_{x_k \in X} s(x_j, x_k)\} \quad (1)$$

We say that x_i is **connected to** *x_j if $(x_i, x_j) \in V$.*

In this definition, an edge exists between x_i and x_j if their similarity is greater than both the average similarity between x_i and the whole set of objects and the average similarity between x_j and the whole set of objects. This definition avoids to specify a threshold corresponding to a minimum similarity value and allows to take into consideration the density around each object.

Definition 2.2 *Let $G_S(X,V)$ be the similarity graph over a set of objects X. A* **pole** *P_k is a subset of X such that the sub-graph $G_S(P_k, V(P_k))$ is a clique-graph, i.e. $\forall x_i \in P_k, \forall x_j \in P_k$ $(x_i,x_j) \in V(P_k)$).*
($V(P_k)$ is the set of vertices (x_i,x_j) such that $x_i \in P_k$ and $x_j \in P_k$).

Given :	$X = \{x_1,\ldots,x_n\}$ the set of objects S the similarity matrix over $X \times X$
Initialization :	Build the similarity graph $G_S(X,V)$
Step 1 :	Build the set of poles $\mathcal{P} = \{P_1,\ldots,P_l\}$ with $\forall i \in \{1,\ldots,l\}$ $P_i \subseteq X$
Step 2 :	Build the membership matrix U where $u(P_i,x_j) = \frac{1}{\lvert P_i\rvert}\sum_{x_k \in P_i} s(x_j,x_k)$
Step 3 :	For each $x_j \in X$, call `assign`(x_j,$\mathcal{P}$)
Step 4 :	Let $\mathcal{C}$ be the set of groups $\{C_1,\ldots,C_l\}$ such that : $C_i = \{x_j \in X \mid x_j$ has been assigned to $P_i\}$ Build a hierarchical organization of $\mathcal{C}$

Table 1. PoBOC : soft-clustering algorithm.

Table 1 summarizes the PoBOC algorithm. We can notice that the membership function ($u(P_i,x_j)$) is not a probabilistic function (the sum of the membership values of an object to all the poles can be $\neq 1$). In the following section, we detail the heuristics for the construction of poles, the assignment method and the construction of the hierarchical organization of the obtained groups.

2.2 Heuristics in PoBOC

Heuristics for the construction of poles
Our definition of pole is close to both the notion of *core* proposed in [3] and the notion of *fuzzy-center* (centroid or medoid) used in fuzzy clustering. The construction of poles requires to build a set of cliques in the similarity graph. Searching for maximal cliques in a graph is a NP-complete problem, so we propose to use the "Best-in" heuristic [4]: a clique is obtained starting from a single vertex and repeatedly adding the nearest neighbor (vertex) until it is not possible to find a vertex x connected to each vertex of the clique under construction. By the way, the clique-graph obtained is an approximation of the maximal complete sub-graph which contains the given vertex.

The construction of the set of poles is obtained by repeating the construction of a pole: this process requires to choose a starting vertex and then to add connected vertices.

The first vertex chosen x^1 is the one having the lower average similarity with other objects, among the set of vertices having at least one connected vertex. Let $G_S(X,V)$ be the similarity graph:

$$x^1 = Argmin_{x_i \in E} \frac{1}{|X|} \sum_{x_j \in X} s(x_i,x_j) \qquad (2)$$

where E is the set of vertices having at least one connected vertex.

The next vertices $\{x^2,\ldots,x^l\}$ are chosen in order to reduce the similarity with poles previously built:

$$x^k = Argmin_{x_i \in E} \frac{1}{k-1} \sum_{m=1,\ldots,k-1} \frac{1}{|P_m|} \sum_{x_j \in P_m} s(x_i,x_j) \qquad (3)$$

The process stops when the sum in the previous equation is greater than the average similarity of the whole set of objects. This heuristic determines the number of poles and then, the number of clusters.

The"soft" assignment method
This multi-assignment step (objects are assigned to poles) plays an important role in the construction of "overlapping-clusters" in PoBOC. The advantage of assigning an object to several clusters is well-known, the assignment method is often based on an arbitrary threshold applied on a fuzzy membership matrix obtained with a fuzzy-clustering method [8]. In this paper, we propose a new approach based on the relative similarity between objects and poles, defined as follows:

Definition 2.3 *Let $X = \{x_1,\ldots,x_n\}$ be the set of objects, let $\mathcal{P} = \{P_1,\ldots,P_l\}$ be the set of poles and U be the membership matrix on $\mathcal{P} \times X$ as defined in Table 1. Given an object x_j, we write $P_{j,1}$ the most similar pole for x_j($P_{j,1} = Argmax_{P_i \in \mathcal{P}}\, u(P_i,x_j)$), $P_{j,2}$ the second most similar pole for x_j and so on, $P_{j,l}$ is the least similar pole for x_j. The following condition* `ASSIGN`*(x_j,$P_{j,k}$) is used to test whether the object x_j is assigned to the pole $P_{j,k}$:*
`ASSIGN`*(x_j,$P_{j,k}$) iff one of the following properties is satisfied :*

i) k=1,
ii) $1 < k < l$, $u(P_{j,k},x_j) \geq \frac{u(P_{j,k-1},x_j)+u(P_{j,k+1},x_j)}{2}$ *and* $\forall k' < k$, `ASSIGN`*(x_j,$P_{j,k'}$).*

For each object, the set of poles is ordered with respect to its average similarity with the object. The property *i)* allows to assign each object to at least one pole (the most similar). The propertie *ii)* allows to assign an object x_j to a pole $P_{j,k}$ by considering the similarity with the previous pole ($P_{j,k-1}$) and the next pole ($P_{j,k+1}$) w.r.t the order associated to x_j. Thus, the assignment is not relative to a threshold but is based on the relative position (similarity) of the object w.r.t. the poles previously obtained. In the following, we call "group" the set of objects assigned to one pole.

Hierarchical organization of groups
The hierarchical organization allows to control the number of final clusters obtained from the set of groups. We will use this representation in our experiments in order to find a better characterization of the groups, and then to get a better set of clusters.

To get a hierarchical organization, we propose to apply the hierarchical agglomerative clustering method "single-link"[7], starting with the groups previously built $\mathcal{C} = \{C_1,\ldots,C_l\}$ where $C_i = \{x_j$ assigned to $P_i\}$. Since the similarity matrix is normalized, we have $\forall x_i \in X,\ s(x_i,x_i) = 1$ and we define the similarity between two groups by

$$sim(C_k,C_m) = \frac{1}{|C_k|.|C_m|} \sum_{x_i \in C_k} \sum_{x_j \in C_m} s(x_i,x_j) \qquad (4)$$

The organization is built as follows: the two most similar groups are agglomerated and this process is repeated until we get only one group. This organization is represented by a binary tree where the leaves correspond to the initial set of groups.

2.3 Discussion about PoBOC

The time complexity of PoBOC can be considered w.r.t. the 4 main steps of the algorithm given in Table 1 ; Step 4 is the most expensive

one, with a $o(k.n^2)$ complexity where n is the number of objects and k is the number of groups obtained after Step 3. This complexity is greater than the complexity of fast algorithms such that k-means or fuzzy-k-means ($o(k.n.t)$) but is lower than other methods such that the "max-link" agglomerative algorithm ($o(n^2.log\ n)$).

One of the advantages of PoBOC w.r.t. methods based on centroids (or medoids) is that each group is represented by a pole, i.e. a set of objects (instead of groups represented by one object for centroid-based metods), which is a less specific representation of a group. Moreover, we do not need to specify the expected number of groups.

3 Learning of classification rules with PoBOC

3.1 Motivations

Classification rules learning is a task which has already been long studied in the supervised learning field. It consists in generating, directly or not, a set of decision rules from a training set such as for each new event a prediction of a class is proposed according to learned rules. We can distinguish two main approaches constructing rules in the "attribute-value" formalism: decision lists and decision trees.

The first two methods are strongly based on an iterative search of a good selector which allows to separate positive observations from negative ones, with respect to a target class. In the case of decision trees methods (for instance C4.5 [13]), each attribute of the description space is viewed as a selector and is a candidate for the construction of a node. A rule corresponds to a path from the root to a leaf in the decision tree. On the other hand, the rules which appear in a decision list are generated directly, iteratively adding [*attribute*$\sharp$*value*] selectors, where $\sharp$ stands for a relational operator such as $=, \neq, <$... A well-known algorithm for decision list learning is CN2 [6]. The construction of a rule via the best selector heuristic does not allow to consider combination of two or more features as selectors, because of complexity problems. The combination of several features is usually better, all the same.

In the following, we $\mathcal{E}$ denotes the event space, E the set of training events, E_i the set of positive events for the class i and $X = \{x_1, \ldots, x_n\}$ the feature space over $\mathcal{E}$.

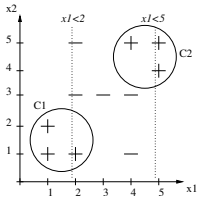

Figure 1. Multiple feature combination as selector.

Let us consider the set of events in the two dimensional space, proposed in Figure 1. The "+" denotes positive events and the "-" negative ones. The target class is clearly composed of two separated concepts C_1 and C_2. The search of the best selector [*attribute*$\sharp$*value*] over $X = \{x_1, x_2\}$ leads to $x_1 < 2$ or $x_1 < 5$ with most of usual quality measures[2]. In this way, final rules will not reveal the inside structure of the target class.

A method to avoid this problem is to consider the internal structure, by first clustering the positive events. Each cluster is then treated separatly and one rule has to be proposed for each cluster. On the previous example, the two clusters C_1 and C_2 can lead respectively to rules $x_1 < 3 \wedge x_2 < 3$ and $x_1 > 3 \wedge x_2 > 3$.

In the case of conceptual clustering [12], rules are generated by successive steps of specialization or generalization. From one event, or a small set of events, the general procedure consists in searching all the maximally general complexes[3] covering the positive events and not covering any negative one. Complexes are then reduced in order to find a disjoint clustering of the collection of events. One can notice that the quality of a conceptual clustering is usually measured with respect to the simplicity of the description and the fitting with training data. An other criteria is the *inter-cluster difference* which aims at favouring disjoint concept descriptions.

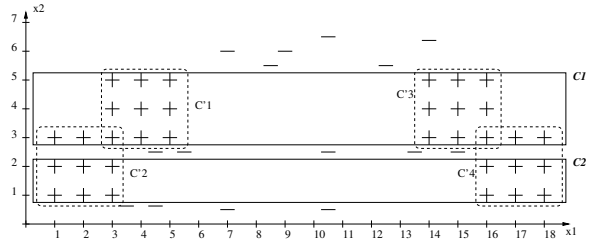

Figure 2. Example of conceptual clustering result.

In Figure 2, a possible result from conceptual clustering over the set of events leads to the two clusters C_1 and C_2 with the following descriptions : $x_2 \geq 3 \wedge x_2 \leq 5$ and $x_2 \geq 1 \wedge x_2 \leq 2$. These two complexes satisfy the quality criterion : descriptions are simple, disjoint and "fit well" with the training dataset. However, C_1 and C_2 do not match with the intuitive organization of positive events we observe. A pre-clustering step could allow to consider first the two separated clusters $\{C'_1 \cup C'2\}$ and $\{C'_3 \cup C'_4\}$ and then, the four non-disjointed clusters C'_1, C'_2, C'_3 and C'_4. This clustering better corresponds to the natural structure of the target class.

3.2 Method for class decomposition

We propose an algorithm to learn rules based on a class decomposition by clustering. For each class it proceeds as follows. If a single rule covering all positive events and covering no negative one exists, this rule is thus retained, else the process of decomposition starts. First, a similarity matrix is built over the set of positive events from the target class, then, clustering is performed over the set of positive events. Each cluster which can be covered by a single rule is considered as a natural concept for the class and the rule is stored. Clusters which do not satisfy to constraint are merged and a new clustering step is done over this subset of events. This procedure iterates until each cluster is covered. A formal presentation of this algorithm is proposed in Table 2.

The similarity measure (step 1) used in our experiments has been presented in [10]. It is based on the definition of a new language of description from the original one. This language is composed of $[attribute\sharp value \wedge \ldots \wedge attribute\sharp value]$ terms. In [10], the authors have shown that this measure provides good performances on nearest neighbour classifiers, whatever the type of features (quantitative and/or categorical).

Let us consider a set of positive events C_i, the test of the existence (step 3) of a single rule covering all the events of C_i and no negative

[2] With the *information gain* and *gain ratio* measures for instance.

[3] A complexe is a conjunction of selectors.

events, is based on the search of all the selectors $[attribute \sharp value]$ which cover C_i. If the conjunction of all these selectors covers no negative event, such a rule exists and it is then generated by an iterative addition of the best selectors.

Input :	$\mathcal{E}$ the event space, E the set of training events, E_i the set of positive events for class i,
Initialization :	$\mathcal{R} = \emptyset$ (the set of learned rules) $\mathcal{C}_0 = E_i$ (the set of events to process) $j = 0$ (an iteration identifier)
`Step` 1	Build S, the similarity matrix over E_i,
do	
`Step` 2	*Cluster*($\mathcal{C}_j$,S) $\rightarrow \{C_1, \ldots, C_{n_j}\}$, $j = j + 1$ and $\mathcal{C}_j = \emptyset$,
`Step` 3	for $s = 1, \ldots, n_{j-1}$ do, if $\exists R \mid \forall e \in C_s, e \in COV(R)$ *and* $\forall e \in E \backslash E_i, e \notin COV(R)$ then $\mathcal{R} \leftarrow \mathcal{R} \cup R$, else $\mathcal{C}_j \leftarrow \mathcal{C}_j \cup C_s$,
While $\mathcal{C}_j \neq \emptyset$	
Output :	$\mathcal{R}$ the set of learned rules.

N.B. $COV(R)$ corresponds to the set of events covered by the rule R.

Table 2. Decomposition algorithm, for one class.

Finally, when none of the clusters C_i is generalized with a rule (step 3), each cluster C_i is independently decomposed by the same algorithm. This last remark avoids the algorithm to get stuck in a loop, even if this situation is very rare.

3.3 Experimental results

We first compare the rule-based classifiers induced by six different clustering algorithms (Table 3). The classifier induced by PoBOC uses exactly the previous decomposition algorithm (Table 2) and, at each call of the *Cluster* function, the number of clusters is not given as output. Conversely, PoBOC is compared with *k-medoids* and *soft-k-medoids*. The last clustering approach corresponds to the well known *fuzzy-k-medoids* algorithm with a post-assignment[4] stage. Finally, the three last columns in Table 3 concern hierarchical agglomerative clustering methods for which a hierachical tree is built and scrolled through from the root to the nodes which are covered by a rule.

Classifiers are then evaluated on 10 famous datasets from the UCI repository [11] : audiology (AD), credit (CE), glass (GL), heart disease Switzerland (HDs), hepatitis (HE), iris (IR), soybean (SO), thyroid (TH), wine (WI) and zoology (ZO). Classification accuracies are obtained by the average of ten iterations of 10-fold cross-validation, and each method is evaluated on the same samples. When a test set is proposed(*), accuracy is computed only on this test sample.

We can observe (Table 3) that the three agglomerative approaches are not appropriate to the data structuring task in a rules learning perspective. This phenomena is explained by a global disproportion between final clusters ; hierarchical trees are sometimes unbalanced so

[4] The same assignment method as in the PoBOC algorithm (cf. section 2)

DOM.	PoBOC	Soft-k-medoids	k-medoids	Complete linkage	Single linkage	Average linkage
AD*	81.5%^{1}	78.5%^{4}	80.4%^{3}	67.3%^{5}	80.8%^{2}	63.5%^{6}
CE	85.8%^{2}	85.7%^{3}	86.7%^{1}	32.2%^{5}	66.2%^{4}	30.7%^{6}
GL	69.8%^{1}	57.1%^{3}	64.9%^{2}	28.5%^{6}	43.2%^{4}	39.6%^{5}
HE	80.5%^{1}	76.3%^{3}	80.1%^{2}	47.7%^{6}	70.8%^{4}	60.7%^{5}
HDs	84.9%^{2}	84.0%^{3}	85.0%^{1}	65.7%^{6}	73.9%^{5}	83.1%^{4}
IR	95.9%^{3}	95.7%^{4}	95.3%^{5}	81.3%^{6}	96.4%^{1}	96.4%^{1}
SO*	85.2%^{1}	81.4%^{3}	83.9%^{2}	62.1%^{6}	70.4%^{4}	68.3%^{5}
TH	94.4%^{1}	94.4%^{1}	93.4%^{3}	61.1%^{6}	91.2%^{4}	80.4%^{5}
WI	95.8%^{1}	93.7%^{3}	94.7%^{2}	87.9%^{6}	89.7%^{4}	89.3%^{5}
ZO	89.7%^{3}	89.5%^{5}	89.9%^{1}	88.9%^{6}	89.8%^{2}	89.7%^{3}
Pos.	1.6	3.2	2.2	5.8	3.4	4.5

Table 3. Average accuracies and global average position. Superscripts denote position of the classifier with respect to the five others.

that clusters are not relevant in a conceptual structuring way. Thanks to its two main characteristics - discovering of an appropriate number of clusters and overlapping between them - PoBOC seems to be really more able of finding natural concepts in a dataset. It outperforms other clustering methods over six of the ten proposed datasets, and is among the two best methods over 8 domains.

DOM.	PoBOC	pFOIL	C4.5	1-NN
AD	81.5%± - ^{2}	65.4%± - ^{4}	84.6%± - ^{1}	76.9%± - ^{3}
CE	85.8%± - ^{2}	85.2%± - ^{3}	86.1%± - ^{1}	77.7%± - ^{4}
GL	69.8%±9.0^2	65.1%±10.2^4	69.2%±9.1^3	78.1%±9.9^1
HE	80.5%± - ^{1}	76.8%± - ^{4}	79.5%± - ^{2}	78.6%± - ^{3}
HDs	84.9%±9.9^3	81.7%±9.6^4	91.7%±8.9^1	86.1%±8.2^2
IR	95.9%±4.7^1	94.3%±5.1^4	95.1%±4.9^2	95.1%±4.5^2
SO	85.2%± - ^{2}	81.4%± - ^{3}	86.7%± - ^{1}	75.2%± - ^{4}
TH	94.4%±4.8^2	92.9%±5.3^4	93.0%±5.0^3	96.9%±3.4^1
WI	95.8%±4.9^1	90.6%±6.9^4	93.5%±5.8^3	95.1%±4.9^2
ZO	89.7%±7.0^4	90.2%±7.2^3	90.9%±6.6^2	94.3%±5.2^1
Pos.	2.0	3.7	1.9	2.3

Table 4. Average accuracies and standard deviations. Superscripts denote position of the classifier with respect to the three other classifiers.

Table 4 presents a comparative study of different usual rule-based and instance-based classifiers. The classifier induced by PoBOC is first compared to the pFOIL method [1] which is a greedy approach based on the well-known FOIL algorithm, in first order logic formalism. The two first columns show an increase of the accuracy of the classifier when a data structuring step is proposed before the rule construction step (PoBOC) or not (pFOIL). One can also obseve that accuracies of PoBOC-based and decision-tree-based classifiers are comparable, since the classifier induced by PoBOC is better than C4.5 over five of the ten domains.

4 Textual data application

4.1 Motivations

In the field of textual data processing, structuring terms or textual units in a semantic way is difficult, because of the complexity and the nature of the relations between the terms. Considering a list of extracted terms from a corpus, an information processing task concerns the construction of themes or topics via a semantic concept organization. Clustering is thus the main tool which can help to discover topics, and we claim that PoBOC is appropriated, since semantic concepts are usually not disjoint. In the following experiment, we show the benefits of using PoBOC in order to identify subdomains from a list of specialized keywords.

4.2 Keywords structuring

A list of 38 keywords, proposed by the authors, has been extracted from scientific articles about three interconnected domains : Artificial Intelligence (AI), Web Technologies (WT) and Natural Language processing (NL). The three respective information resources are: the *Journal of Japanese Society for Artificial Intelligence (1997)*, the *International World Wide Web Conference (2002)* and the *International Conference on Language Resources and Evaluation (2000)*.

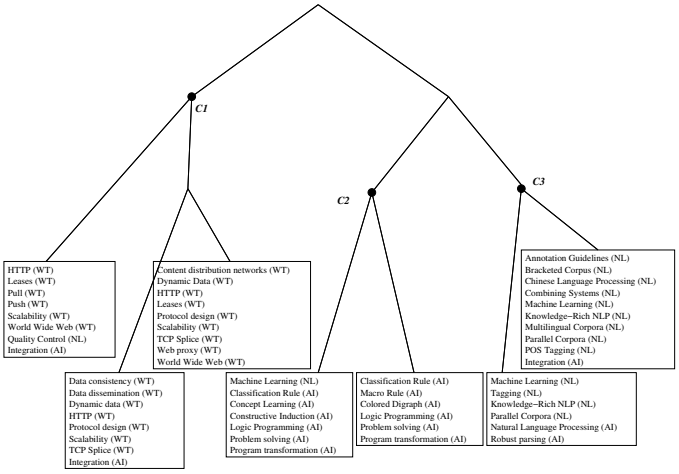

Figure 3. Hierarchical structuring of keywords with PoBOC

Then, a similarity matrix is built from the co-occurrences of the keywords on the web [15]. From this matrix, the keywords are organized in a totally unsupervised manner. Figure 3 gives the seven clusters obtained with PoBOC, organized in a hierarchical tree.

For each keyword, we give the corresponding domain to help the reader evaluation, although this information has not been used in the clustering process. We observe that the seven leaves of the tree correspond to rather pure clusters. Furthermore, some keywords such as *natural language processing* or *machine learning* have been put back in their correct context. Finally, the three initial topics are to be found in the top of the tree : C_1, C_2 and C_3 corresponding respectively to the Web Technologies, Artificial Intelligence and Natural Language domains. The average purity in the leaves is 88% and 86% in the three top nodes. A similar experiment with the $k-medoids$ clustering algorithm ($k = 3$) gives only about 69% of average purity[5].

Another interesting remark is about the nature of the intersections between clusters. Two different types of keywords are shared by several clusters : the inter-domain terms which are general and thus appear in many clusters (for instance *integration*) and the intra-domain ones which are general in only one domain and thus appear in several clusters about this topic (for instance *HTTP*).

5 Conclusion and perspectives

In this paper, we present the clustering algorithm PoBOC. This method can be seen as a compromise between hard-clustering and fuzzy-clustering approaches, providing an appropriate number of clusters which overlap. Furthermore, PoBOC combines advantages of the two previous mentioned approaches since it proposes simpler classes than fuzzy-clusters and the final organization is richer than hard-clustering structuring.

In order to evaluate the method, we propose two really different application domains : the pre-processing of data in the perspective of classification rules learning, and the organization of textual data. Thus, we can observe that the clusters provided by PoBOC lead to a learning step more efficient than with other traditional hard or soft-clustering algorithm. It should be noticed that this pre-processing step allows to outperform greedy rules learning methods so that the accuracy of the classifier is comparable to decision-tree performances. Finally, PoBOC is also efficient for the textual data processing, because overlapping clusters enable to take account of the semantic complexity of the data.

A more complete study about overlapping clustering of words for text classification is under progress. Other research perspectives could be considered, one can notice for instance: clustering on spatial data and image processing.

[5] This percentage corresponds to an average over 50 iterations.

REFERENCES

[1] K. M. Ali and M. J. Pazzani, 'HYDRA: A noise-tolerant relational concept learning algorithm', in *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, ed., R. Bajcsy, pp. 1064–1071. Morgan Kaufmann, (1993).

[2] A. Baraldi and P. Blonda, 'A survey of fuzzy clustering algorithms for pattern recognition. II', *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, **29**, 786–801, (1999).

[3] Amir Ben-Dor, Ron Shamir, and Zohar Yakhini, 'Clustering gene expression patterns', *Journal of Computational Biology*, **6**(3/4), 281–297, (1999).

[4] I. Bomze, M. Budinich, P. Pardalos, and M. Pelillo, 'The maximum clique problem', in *Handbook of Combinatorial Optimization*, eds., D.-Z. Du and P. M. Pardalos, volume 4, Kluwer Academic Publishers, Boston, MA, (1999).

[5] P. Cheesman, J. Kelly, M. Self, J. Stutz, W. Taylor, and D. Freeman, 'Autoclass: A bayesian classification system', in *Proceedings of the Fifth International Conference on Machine Learning*, ed., MI. Ann Arbor, pp. 54–64. Morgan Kaufmann Publishers, (1988).

[6] Peter Clark and Tim Niblett, 'The cn2 induction algorithm', *Machine Learning*, **3**, 261–283, (1989).

[7] A. K. Jain, M. N. Murty, and P. J. Flynn, 'Data clustering: a review', *ACM Computing Surveys*, **31**(3), 264–323, (1999).

[8] Michael Kearns, Yishay Mansour, and Andrew Y. Ng, 'An information-theoretic analysis of hard and soft assignment methods for clustering', in *Proceedings of Uncertainty in Artificial Intelligence. AAAI*, pp. 282–293, (1997).

[9] Alain Lelu, 'Clusters and factors: neuronal algorithms for a novel representation of huge and highly multidimensional data sets', in *New Approaches in Classification and Data Analysis*, pp. 241–248. E. Diday, Y. Lechevallier & al. eds., Springer-Verlag, Berlin, (1994).

[10] Lionel Martin and Frédéric Moal, 'A language-based similarity measure', in *Machine Learning: ECML 2001, 12th European Conference on Machine Learning, Freiburg, Germany, September 5-7, 2001, Proceedings*, volume 2167 of *Lecture Notes in Artificial Intelligence*, pp. 336–347. Springer, (2001).

[11] C.J. Merz and P.M. Murphy, 'Uci repository of machine learning databases', (1998).

[12] R.S Michalski and R.E. Stepp, 'Learning from observation: Conceptual clustering', *Machine Learning: An Artificial Intelligence Approach*, **1**, 331–363, (1983).

[13] J. Ross Quinlan, *C4.5: programs for machine learning*, Morgan Kaufmann Publishers Inc., 1993.

[14] Oldemar Rodríguez and Edwin Diday, 'Pyramidal clustering algorithms in iso-3d project', in *Proceedings of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases, PKDD'00*, (2000).

[15] Peter D. Turney, 'Mining the web for synonyms: PMI-IR versus LSA on TOEFL', in *Machine Learning: ECML 2001, 12th European Conference on Machine Learning, Freiburg, Germany, September 5-7, 2001*, volume 2167 of *Lecture Notes in Artificial Intelligence*, pp. 491–502. Springer, (2001).