Clustering sets of objects using concepts-objects bipartite graphs

Emmanuel Navarro†, Henri Prade†, and Bruno Gaume‡

†: IRIT, Université de Toulouse III,
118 Route de Narbonne; 31062 Toulouse Cedex 9, France
E-mail: navarro@irit.fr, prade@irit.fr

‡: CLLE-ERSS, Université de Toulouse II,
5, allées Antonio Machado; 31058 Toulouse Cedex 9, France
E-mail: gaume@univ-tlse2.fr

Abstract. In this paper we deal with data stated under the form of a binary relation between objects and properties. We propose an approach for clustering the objects and labeling them with characteristic subsets of properties. The approach is based on a parallel between formal concept analysis and graph clustering. The problem is made tricky due to the fact that generally there is no partitioning of the objects that can be associated with a partitioning of properties. Indeed a relevant partition of objects may exist, whereas it is not the case for properties. In order to obtain a conceptual clustering of the objects, we work with a bipartite graph relating objects with formal concepts. Experiments on artificial benchmarks and real examples show the effectiveness of the method, more particularly the fact that the results remain stable when an increasing number of properties are shared between objects of different clusters.

Key words: formal concept analysis, bipartite graph, graph clustering

1 Introduction

For making sense of complex data, one may need to cluster them, and if possible, to provide labels for the clusters. In this paper we are interested in data that take the form of a binary relation between a set of objects and a set of properties. Several families of approaches exist for such a task: one may use bi-clustering (or two-mode clustering) approaches [3], formal concept analysis (FCA for short) methods, and hybridization of them.

In previous work, the authors have emphasized the parallelism between FCA operators and two views of graph clustering, referring respectively to the search for maximal bi-cliques and to the search of maximal connected components [12]. Moreover, since the number of formal concepts is usually very large, we have proposed a preliminary approach for providing an approximate conceptual view of data by taking inspiration from the recent literature on graph clustering (often

called community detection problem). More precisely, we have proposed a two-step procedure: i) random walks are used for providing an approximate and more robust view of the formal context leading to a smaller number of formal concepts, ii) these concepts are then fused when they have a sufficient overlap [13]. However, this two-step method requires the tuning of threshold parameters.

In this paper we propose a new approach based on bipartite graphs between objects and concepts, rather than on bipartite graphs between objects and properties, as it was the case in the step i) of the previous method. Moreover no threshold are any longer needed. Our goal is now to look for a partition of the set of objects, while properties may remain shared between different clusters of objects. The paper is organized as follows. After a background on FCA and its bipartite graph counterpart (Section 2), we present the new approach in Section 3, and suggest a way of labeling the clusters of objects in Section 3.3. Experiments are reported in Section 4 that show the effectiveness of the method on artificial benchmarks and on a real dataset. Comparison with related works (Section 5) and concluding remarks (Section 6) end the paper.

2 Background: From formal concept analysis to clustering

In this section we first recall the standard notion of FCA, as well as the notion of independent sub-contexts, and then give their counterpart in the setting of bipartite graphs where we interpret them in clustering terms.

2.1 Formal concepts and independent subcontexts

Let R be a *binary relation* between a set $\mathbf{O}$ of objects and a set $\mathbf{P}$ of Boolean properties. We note $\mathcal{R} = (\mathbf{O}, \mathbf{P}, R)$ the tuple formed by these objects and properties sets and the binary relation. It is called a *formal context* [11]. The notation $(x, y) \in R$ means that object x has property y. Let $R(x) = \{y \in \mathbf{P} | (x, y) \in R\}$ be the set of properties of object x. Similarly, $R^{-1}(y) = \{x \in \mathbf{O} | (x, y) \in R\}$ is the set of objects having property y.

Formal concept analysis [11] defines two set operators, here denoted $(.)^{\Delta}$ and $(.)^{-1\Delta}$, called *intent* and *extent* operators respectively, s.t. $\forall Y \subseteq \mathbf{P}$ and $\forall\, X \subseteq \mathbf{O}$:

$$X^{\Delta} = \{y \in \mathbf{P} | \forall x \in X, (x, y) \in R\} \tag{1}$$

$$Y^{-1\Delta} = \{x \in \mathbf{O} | \forall y \in Y, (x, y) \in R\} \tag{2}$$

X^{Δ} is the set of properties possessed by all objects in X. $Y^{-1\Delta}$ is the set of objects having all properties in Y. These two operators induce an antitone Galois connection between $2^{\mathbf{O}}$ and $2^{\mathbf{P}}$. This means that the following property holds

$$X \subseteq Y^{-1\Delta} \Leftrightarrow Y \subseteq X^{\Delta}.$$

A pair such that $X^{\Delta} = Y$ and $Y^{-1\Delta} = X$ is called a *formal concept*[11]. X is its extent and Y its intent. In other words, a formal concept is a pair (X, Y)

such that X is the set of objects having all properties in Y and Y is the set of properties shared by all objects in X. It can be shown that formal concepts correspond to *maximal pairs* (X, Y) such that

$$X \times Y \subseteq R.$$

A recent parallel between formal concept analysis and possibility theory[8] has led to emphasize the interest of an other remarkable set operator $(.)^{\Pi}$, and their two respective duals. The new operator and the already defined intent operator can be written as follows, $\forall X \subset \mathbf{O}$:

$$X^{\Pi} = \{y \in \mathbf{P} | R^{-1}(y) \cap X \neq \emptyset\} \quad (3)$$

$$X^{\Delta} = \{y \in \mathbf{P} | R^{-1}(y) \supseteq X\} \quad (4)$$

Note that (4) is equivalent to the definition of operator $(.)^{\Delta}$ in (1). X^{Π} is the set of properties that are possessed by at least *one* object in X. X^{Δ} is the set of properties shared by all objects in X.

Operators $(.)^{-1\Pi}$, $(.)^{-1\Delta}$ are defined similarly on a set Y of properties by substituting R^{-1} to R and by inverting $\mathbf{O}$ and $\mathbf{P}$. $(Y)^{-1\Pi}$, $(Y)^{-1\Delta}$ are respectively, the set of objects having at least one property in Y and the set of objects that have all the properties in Y.

This new operator lead to consider a new connection[9] that corresponds to pairs (X, Y) such that $X^{\Pi} = Y$ and $Y^{-1\Pi} = X$ (while $(.)^{\Delta}$ leads to formal concepts, as already said). Pairs (X, Y) such that $X^{\Pi} = Y$ and $Y^{-1\Pi} = X$ do not define formal concept, but *independent sub-contexts*. Indeed, it has been recently shown[9] that pairs (X, Y) of sets exchanged through the new connection operator, are subsets such that

$$(X \times Y) \cup (\overline{X} \times \overline{Y}) \supseteq R,$$

just as formal concepts correspond to maximal pairs (X, Y) such that

$$X \times Y \subseteq R.$$

In Figure 1, two examples of formal concepts are the pairs $(\{a1, a2, a3, a4, b1\}, \{2, 7\})$ and $(\{c1, c2\}, \{4, 5, 6, 8\})$. On the other hand, if we forget the fact that the object $a2$ verify the property 10, the pairs $(\{a1, a2, a3, a4, b1, b2, b3, b4, c1, c2\}, \{1, 2, 3, 4, 5, 6, 7, 8\})$ and $(\{d1, d2\}, \{9, 10, 11\})$ are two independent sub-contexts.

Thus, in the setting of formal concept analysis, by means of two companion connections, two key aspects of the idea of clustering are at work. On the one hand, independent sub-contexts are characterized, and on the other hand inside each sub-context, formal concepts (X, Y) are identified where *each* pair (x, y) such that $x \in X, y \in Y$ are in relation (while *no* pair (x, y) such that $x \in \overline{X'}, y \in Y'$ or $x \in X', y \in \overline{Y'}$ are in relation if (X', Y') and $(\overline{X'}, \overline{Y'})$ are two independent subcontexts). In particular, two formal concepts belonging to two different sub-contexts are clearly well-separated. The relation with clustering is made still clearer in the next sub-section by providing a bipartite graph reading of FCA.

	1	2	3	4	5	6	7	8	9	10	11
a1	×	×	×				×	×			
a2		×	×				×	×		×	
a3	×	×					×	×			
a4	×	×	×				×				
b1		×	×	×	×		×				
b2		×	×		×						
b3			×	×	×						
b4		×	×	×	×						
c1				×	×	×		×			
c2				×	×	×	×	×			
d1									×	×	×
d2									×	×	

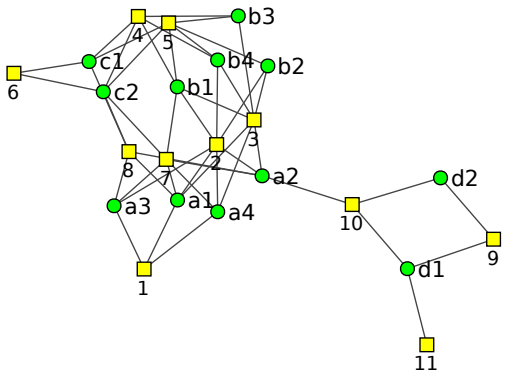


Fig. 1. A formal context R and the corresponding bipartite graph.

2.2 Formal concept analysis, bipartite graphs and clustering

For every formal context $\mathcal{R} = (\mathbf{O}, \mathbf{P}, R)$, one can build an undirected bi-graph $\mathcal{G} = (V_o, V_p, E)$ s.t. there is a direct correspondence between: the set of objects $\mathbf{O}$ and a set V_o of "o-vertices", the set of properties $\mathbf{P}$ and a set V_p of "p-vertices", and between the binary relation R and a set of edges E. In other words, there is one o-vertex for each object, one p-vertex for each property, and one edge between an o-vertex and a p-vertex if and only if the corresponding object possesses the corresponding property (according to R).

The operators $(.)^{\Pi}$ and $(.)^{\Delta}$ can then be rewritten in the following way:

$$X^{\Pi} = \cup_{x \in X} \Gamma(x) \tag{5}$$

$$X^{\Delta} = \cap_{x \in X} \Gamma(x) \tag{6}$$

where $\Gamma(x)$ denotes the set of neighbors of the vertex x. These notations are interesting since only the neighborhood of vertices of X is involved. It permits to immediately understand operators $(.)^{\Pi}$ and $(.)^{\Delta}$ in terms of neighborhood in the bi-graph: X^{Π} is the union of neighbors of vertices of X whereas X^{Δ} is the intersection of these neighbors. The same expressions apply to $(.)^{-1\Pi}$ and $(.)^{-1\Delta}$, changing X by Y (and x by y).

The connections induced by $(.)^{\Delta}$ and $(.)^{\Pi}$ can also be understood in the graph setting framework: the first connection corresponds to maximal bi-cliques whereas the second one two maximal connected components [12]. Indeed on the bi-graph $\mathcal{G} = (V_o, V_p, E)$, with $X \subseteq V_o$ and $Y \subseteq V_p$, we have:

Proposition 1 *$X = Y^{-1\Delta}$ and $Y = X^{\Delta}$, iff $X \cup Y$ is a maximal bi-clique.*

Proposition 2 *For a pair (X, Y) the two following propositions are equivalent:*

1. *$X = Y^{-1\Pi}$ and $Y = X^{\Pi}$ and there is no strict subset $X' \subset X$ and $Y' \subset Y$ such that $X' = Y'^{-1\Pi}, Y' = X'^{\Pi}$.*
2. *$X \cup Y$ is a maximal connected component (which counts at least 2 vertices).*

It is worth noticing that the two connections correspond to extreme definitions of what a cluster (or a community) could be:

1. a group of vertices with *no link missing inside*.
2. a group of vertices with *no link with outside*.

One the one hand a maximal bi-clique is a maximal subset of vertices with a maximal edge density. Vertices cannot be moved closer, and in that sense one can not build a stronger cluster. On the other hand, a set of vertices disconnected from the rest of the graph can not be more clearly separated from other vertices. It corresponds to another type of cluster. In fact, only the smallest of such sets are really interesting, and they are nothing else than maximal connected components. These two extreme definitions were already pointed out for clusters in unipartite graphs [19].

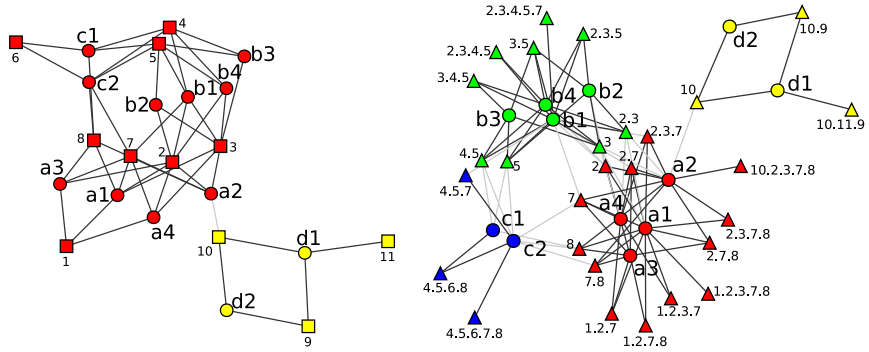
3 Looking for meaningful clusters of objects

In this section we motivate the need for a new clustering procedure which enables us to obtain meaningful clusters of objects, even if the objects in different clusters share many properties.

3.1 Preliminary discussion

As said in the introduction, our primary purpose is to cluster the set of considered objects into distinct subsets on the basis of their properties. However the application of a graph clustering method on the bipartite graph (associated to the formal context) generally fails. It is due to the fact that the method when tentatively gathering objects in separate clusters, often fails to do it since objects in different potential clusters usually share many common properties. In other words, bipartite graph clustering looks for a partition of the graph vertices. When applied to the object-properties graph it puts into correspondence subsets of objects with subsets of properties, i.e. they look for a partition of objects and a partition of properties such that each set of objects is in correspondence with a set of properties. This is illustrated on the Figure 2(a) for the formal context example of Figure 1. As can be seen, the method isolates the cluster $\{d1, d2\}$, but fails to discriminate more, leaving the rest of the objects in the same cluster. Indeed, it will have been desirable to separate these remaining objects in 3 clusters, namely $\{a1, a2, a3, a4\}$, $\{b1, b2, b3, b4\}$ and $\{c1, c2\}$, as revealed by a careful examination of the formal context of Figure 1.

Besides, it can be checked that there are 30 formal concepts in the formal context of Figure 1. Note that it is usually observed that FCA returns a rather large number of formal concepts, in particular with noisy data or when exceptions are present. Moreover there is no immediate way of using the lattice of concepts for building a partition of the objects. However, as can been seen in Figure 1 the 3 subsets of objects that the method have failed to separate (Figure 2(a)) form the "approximate" concepts $(\{a1, a2, a3, a4\}, \{1, 2, 3, 7, 8\})$,



(a) objects-properties graph, (b) objects-concepts graph

Fig. 2. For the relation given in Figure 1, results of Infomap [18] graph clustering method either on the basic objects-properties graph (a) or on the objects-concepts graph (b). On the two graphs circles are objects, for (a) squares are properties and for (b) triangles are concepts.

$(\{b1, b2, b3, b4\}, \{2, 3, 4, 5\})$ and $(\{c1, c2\}, \{4, 5, 6, 7, 8\})$. By approximate concepts [9], we mean that, up to a few missing crosses, we have large formal concepts (X, Y) (i.e., they correspond in Figure 1 to "approximate" $X \times Y$ rectangles). This suggests to investigate a "conceptual" clustering of the objects by dealing with the objects-formal concepts bipartite graph.

3.2 Clustering objects-concepts bipartite graphs

We now describe the method we propose more precisely. First, a preliminary step consists in building all the formal concepts associated to the objects-properties graph, using a formal concept extraction method, e.g. [10].

Second, a bipartite graph between objects and concepts is built such that each object $o \in \mathbf{O}$ is connected to a concept (X, Y) iff $o \in X$, then the corresponding edge is weighted by $w = |Y|$ the number of properties of the corresponding concept. This weighting is introduced in order to favor "large" concepts, which are expected to be more "meaningful". Indeed concepts with a small number of properties are likely to connect "too many" objects. Note that the top and bottom concepts are ignored, if they contain zero objects or zero properties.

The vertices of this bipartite graph are then partitioned by using the graph clustering Infomap method [18]. Infomap is recognized as one of the best methods of graph clustering [16]. It consists in searching for the clusters that best compress the description length of the trajectory of a random walk through the whole graph. This trajectory is described in a two-level way in function of the clusters: when the walker enters a cluster, the name of the cluster is used, but then only the name of the current vertex inside the cluster is retained. In this way, short length names may be used for naming different vertices that are in different clusters

leading to shorter trajectory descriptions, at the condition that clusters are such that random walkers tend to stay inside clusters. This intuitively fits the idea that random walkers are "trapped" when entering a cluster, since a cluster can only be weakly related to other clusters. This idea has been used in different manners in the recent graph clustering (or community detection) literature [19, 6]. Note that Infomap has not been specifically designed for bipartite graphs. However, nothing in the underlying mathematics is specific to uni-partite graphs either, and prevents to use it for bipartite graphs. Infomap does not specifically take into account the fact that the graph is bipartite. In fact, this is an advantage because we are looking for something which is a kind independent sub-contexts in the formal context defined by the relation linking objects and formal concepts. Thus, we obtain both a partition of objects and an *associated* partition of (formal) concepts.

As can be seen in Figure 2(b), the application of Infomap on the objects-concepts graph now yields the 4 expected clusters of objects, in the example of Figure 1.

3.3 Labeling clusters

In order to label each cluster of objects with a subset of relevant properties, we use the following simple method.

For each cluster of objects we look for two particular concepts: namely the concept (X^*, Y^*) which is associated with the largest subset of objects (of the corresponding objects cluster) and the concept (X_*, Y_*) which is associated with the smallest superset of objects. In formal terms, let $\mathcal{C} = (X, S)$ be a cluster of objects X with the associated set S of concepts, i.e. $S = \{(X'_1, Y'_1), (X'_2, Y'_2), ...\}$. Let be T the set of all formal concepts. Then we compute the two noticeable formal concepts that are defined as follows:

$$(X^*, Y^*) \in T \quad \text{s.t.} \quad \begin{cases} X^* \supseteq X \\ \nexists (X_j, Y_j) \in T \quad \text{s.t.} \quad X^* \supset X_j \supseteq X \end{cases} \tag{7}$$

$$(X_*, Y_*) \in T \quad \text{s.t.} \quad \begin{cases} X_* \subseteq X \\ \nexists (X_j, Y_j) \in T \quad \text{s.t.} \quad X_* \subset X_j \subseteq X \end{cases} \tag{8}$$

One can check that $X_* \subseteq X^*$, and $Y_* \supseteq Y^*$. Therefore the two sets of properties Y^* and Y_* can be used for labeling the cluster. Note that we are sure that all the properties of Y^* are shared by *all* the objects of the cluster.

4 Experiments and discussions

For evaluating (and illustrating) the proposed procedure we consider two kinds of benchmark, one generated artificially and a real example available in the literature.

4.1 Evaluation on artificial benchmarks

In order to build a benchmark for object clustering procedure, we built formal contexts in the following way. We take n groups of k objects, each group is associated with m_{own} properties that only objects of this group may satisfy, and with m_{shared} properties that may be verified by objects of s other groups. For each group of objects, an object of the group satisfies each property in the group with a probability μ. An example of such a context is given in Table 1.

Table 1. An example of formal context artificially generated by the procedure described in Section 4.1, with $n = 3$, $k = 3$, $m_{own} = 2$, $m_{shared} = 4$, $s = 1$, $\mu = 0.8$.

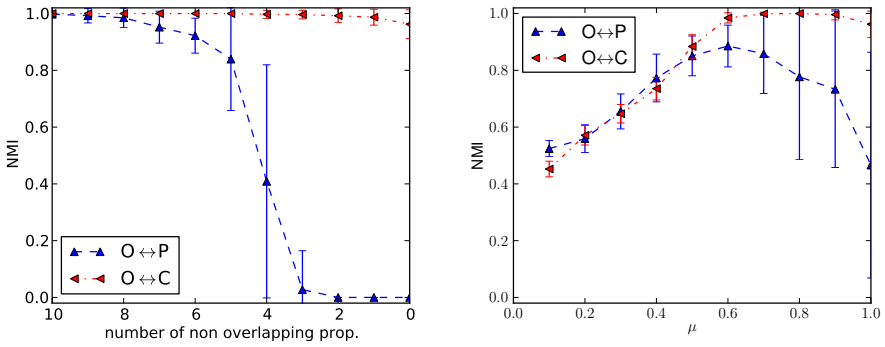
	A0	A1	B0	B1	C0	C1	AB0	AB1	BC0	BC1	CA0	CA1
a0	×						×	×			×	×
a1		×					×	×				×
a2	×	×					×	×			×	
b0				×				×	×	×		
b1			×	×			×	×	×	×		
b2			×				×		×	×		
c0						×			×	×	×	×
c1					×	×			×	×	×	×
c2					×	×			×	×		

The Figures 3(a) and 3(b) present the results of the clustering on the objects-properties graph (the curve O $\leftrightarrow$ P, in blue) and on the objects-concepts graph (the curve O $\leftrightarrow$ C, in red). To evaluate the accuracy of our algorithm against the correct partition of objects we use the normalised mutual information (NMI). A value of 0 indicate that the two partitions are totally dissimilar, whereas a value of 1 indicate that the two partitions are identical. This is a commonly use measure in graph clustering literature [5]. Each point indicated the average value obtained on 50 realizations, the standard deviation is indicated by the vertical error bar on each point. As shown in Figure 3(a), the results remain stable with our approach when an increasing number of properties are shared between objects in different clusters, while it is not the case if we work with the objects-properties graph only.

4.2 The UCI Zoo dataset

The UCI Zoo dataset describes 101 animals on 16 Boolean-valued attributes and one numerical attribute (the number of legs). We transformed this numerical attribute in 7 Boolean attributes (no legs, one leg, two legs, ...). For each animal the type is indicated, there are 7 types of animals: mammal, bird, reptile, fishes, amphibians, insects, invertebrates. This data set can be downloaded from the UCI Machine Learning Repository[1].

[1] `http://archive.ics.uci.edu/ml/datasets/Zoo`



(a) Result in function of n_{own}, with $\mu = 0.8$ (b) Result in function of μ, with $m_{own} = 5$

Fig. 3. Normalised Mutual Information (NMI) [5] value of Infomap clustering method on objects-properties graph (curve O $\leftrightarrow$ P in blue) and on objects-concepts graph (curve O $\leftrightarrow$ C in red). Benchmark contexts are built with the following parameters: $n = 5, k = 10$, $m_{shared} = 6$, $s = 2$

Table 4.2 shows the result of the clustering over the objects-properties graph, while Table 4.2 shows the results on the objects-concepts graph. One can see that the clustering method fails on the objects-properties graph, whereas the partition given on the objects-concepts graph retrieves the types of animals almost exactly. Moreover, the labels Y^* and Y_* coincide in several cases.

5 Related works

We focus our discussion on the literature either related to bipartite graph clustering or to clustering of objects according to a binary relation between objects and properties. The first group makes explicit reference to the graph representation whereas the second one doesn't.

Let us start with representatives of the vast amount of literature in the second group. In [14], the authors use a measure of quality for clustering objects based on Kullback-Leibler entropy which is optimized by means of a genetic algorithm. However, such a black box method does not provide a means for labeling the clusters. In [1] A FCA-based method is proposed, where potentially interesting concepts are selected and then the underlying formal context is revised. It enables the extraction of new descriptors which allows for the reuse of concepts in an incremental way. This leads to a method taking inspiration from inverse resolution in inductive logic programming which enables the extraction of clusters with associated properties, in the Zoo dataset example. Note that there exist a lot of methods that look for bi-clustering (also named co-clustering, or two-mode clustering) which consist in finding a partition of objects that is in direct correspondence with a partition of properties, see [3] for a state of the art.

Table 2. Results of the clustering of the objects-concepts graph ($NMI = 0.81$)

$Y^* = \{backbone, breathes, hair, milk, toothed\}$ $Y_* = \{backbone, breathes, hair, milk, tail, toothed\}$	
Mammals:	aardvark, lynx, leopard, bear, boar, puma, lion, cheetah, raccoon, mink, pussycat, mongoose, wolf, polecat, antelope, calf, elephant, oryx, goat, deer, reindeer, buffalo, pony, giraffe, vole, mole, hare, cavy, hamster, opossum, sealion, girl, wallaby, gorilla, fruitbat, squirrel, vampire
$Y^* = \{0legs\}$ $Y_* = \{0legs, aquatic, eggs\}$	
Fishes:	stingray, pike, piranha, catfish, herring, dogfish, tuna, chub, bass, sole, seahorse, carp, haddock
Invertebrates:	clam, seawasp
Reptiles:	seasnake
$Y^* = \{2legs, backbone, breathes, eggs, feathers, tail\}$ $Y_* = \{2legs, backbone, breathes, eggs, feathers, predator, tail\}$	
Birds:	flamingo, gull, skimmer, sparrow, wren, skua, hawk, crow, duck, vulture, lark, swan, pheasant, kiwi, rhea, ostrich, penguin
$Y^* = Y_* = \{4legs, eggs\}$	
Amphibians:	newt, frog2, frog1, toad
Reptiles:	tortoise, tuatara
Mammals:	platypus
Invertebrates:	crab
$Y^* = Y_* = \{0legs, aquatic, backbone, breathes, catsize, fins, milk, predator, toothed\}$	
Mammals:	porpoise, dolphin, seal
$Y^* = Y_* = \{6legs, breathes, eggs\}$	
Insects:	flea, ladybird, moth, gnat, wasp, honeybee, housefly, termite
$Y^* = Y_* = \{0legs, breathes, eggs\}$	
Reptiles:	slowworm, pitviper
Invertebrates:	worm, slug
$Y^* = Y_* = \{2legs, airborne, backbone, breathes, domestic, eggs, feathers, tail\}$	
Birds:	chicken, parakeet, dove
$Y^* = \{aquatic, eggs, predator\}$ $Y_* = \{6legs, aquatic, eggs, predator\}$	
Invertebrates:	crayfish, starfish, lobster
$Y^* = Y_* = \{8legs, breathes, predator, tail, venomous\}$	
Invertebrates:	scorpion
$Y^* = Y_* = \{8legs, aquatic, catsize, eggs, predator\}$	
Invertebrates:	octopus

Table 3. Results of the clustering of the objects-properties graph ($NMI = 0.02$)

Mammals:	aardvark, antelope, bear, boar, buffalo, calf, cavy, cheetah, deer, dolphin, elephant, fruitbat, giraffe, girl, goat, gorilla, hamster, hare, leopard, lion, lynx, mink, mole, mongoose, opossum, oryx, platypus, polecat, pony, porpoise, puma, pussycat, raccoon, reindeer, seal, sealion, squirrel, vampire, vole, wallaby, wolf
Birds:	chicken, crow, dove, duck, flamingo, gull, hawk, kiwi, lark, ostrich, parakeet, penguin, pheasant, rhea, skimmer, skua, sparrow, swan, vulture, wren
Fishes:	bass, carp, catfish, chub, dogfish, haddock, herring, pike, piranha, seahorse, sole, stingray, tuna
Invertebrates:	clam, crab, crayfish, lobster, octopus, scorpion, seawasp, slug, worm
Insects:	flea, gnat, honeybee, housefly, ladybird, moth, termite, wasp
Reptiles:	pitviper, seasnake, slowworm, tortoise, tuatara
Amphibians:	frog1, frog2, newt, toad
Invertebrates:	starfish

Another family of methods came from the literature concerning bipartite graph clustering. In [7] a spectral method is used for finding a partition of a bipartite graph that minimized the cut size, i.e. the number of edges running between clusters. The main drawback of such approaches is that the number of clusters has to be known in advance, and the methods tend to create clusters having almost the same size, which rarely makes sense with real data sets.

In [2], the authors proposed an adaptation of Newman modularity [4] to the case of bipartite graph. The Newman modularity is a measure of quality of a partition of a graph vertices, a relevant partitioning is usually found by optimizing this quality measure using various heuristics.

Most of these methods lead to a partition of objects and properties, and therefore do not manage to partition objects when properties are shared between many clusters. Note that this issue has been partially addressed in [17], where the authors proposed a measure of quality (inspired from the Newman modularity) of a bipartite graph clustering that allows the fact that there is no direct correspondence between properties cluster and object clusters.

Finally, note that in [15] the authors propose an approach that consists in partitioning a bipartite graph between objects and hypercliques (which can be understood as a set of properties that are satisfied by almost the same objects). This method is in a spirit similar to the method we proposed. However they use a partitioning method that amounts to minimizing a *cut* measure, which suffers from the main drawbacks as the one used in [7].

6 Conclusion

Starting with a binary relation linking objects and properties, formal concept analysis enables us to obtain formal concepts on the one hand, but also independent sub-contexts on the other hand, as recalled at the beginning of this paper.

Then, the independent sub-contexts may be viewed as separating clusters of objects and properties, inside which formal concepts identify homogeneous families of objects. But due to noisy data, due to the existence of exceptions, and more generally due to the fact that the same property may be shared by a variety of objects, it is difficult to cluster a set of objects in a meaningful way directly on a formal context. In the paper, we have proposed to handle the problem in a new formal context where the properties are replaced the formal concept obtained from the initial formal context. Then we have shown on artificial benchmarks and on a real data set that looking for clusters in this higher level formal context makes possible to obtain clusters that can then be interpreted in terms of two nested sets of properties where the smallest one contains only properties that are shared by all the objects in the cluster. As can be seen on the real data set, the two nested sets of properties may be equal, and then a perfect characterization of the cluster is obtained. More experiments would be necessary to evaluate possible variants of this general approach.

References

1. M. Bain. Structured features from concept lattices for unsupervised learning and classification. In B. McKay and J. K. Slaney, editors, *15th Australian Joint Conference on Artificial Intelligence, Canberra, Australia*, volume 2557 of *LNCS*, pages 557–568. Springer, 2002.
2. M. J. Barber. Modularity and community detection in bipartite networks. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 76(6), December 2007.
3. S. Busygin, O. Prokopyev, and P. M. Pardalos. Biclustering in data mining. *Computers and Operations Research*, 35(9):2964–2987, September 2008.
4. A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Phys. Rev. E*, 70(6), Dec 2004.
5. L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09):P09008, 2005.
6. J. -C Delvenne, S. N Yaliraki, and M. Barahona. Stability of graph communities across time scales. *Proc. of the National Academy of Sciences of the USA*, 107(29):12755–12760, 2010.
7. I.S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–274, San Francisco, 2001. ACM.
8. D. Dubois, F. Dupin de Saint-Cyr, and H. Prade. A possibility theoretic view of formal concept analysis. *Fundamenta Informaticae*, 75(1):195–213, 2007.
9. D. Dubois and H. Prade. Possibility theory and formal concept analysis: Characterizing independent sub-contexts. *Fuzzy Sets and Systems*, 196:4–16, 2012.
10. H. Fu and E. Mephu Nguifo. A parallel algorithm to generate formal concepts for large data. In Peter W. Eklund, editor, *Concept Lattices, Prooc. 2nd Int. Conf. on Formal Concept Analysis (ICFCA 2004) Sydney*, volume 2961 of *LNCS*, pages 394–401. Springer, 2004.
11. B. Ganter and R. Wille. *Formal Concept Analysis*. Springer-Verlag, 1999.

12. B. Gaume, E. Navarro, and H. Prade. A parallel between extended formal concept analysis and bipartite graphs analysis. In E. Hüllermeier, R. Kruse, and F. Hoffmann, editors, *Computational Intelligence for Knowledge-Based Systems Design, Proc. 13th Inter. Conf. on Information Processing and Management of Uncertainty (IPMU 2010), Dortmund, June 28 - July 2*, volume 6178 of *LNAI*, pages 270–280. Springer, 2010.
13. B. Gaume, E. Navarro, and H. Prade. Clustering bipartite graphs in terms of approximate formal concepts and sub-contexts. *IJCIS*, 2012. To be published.
14. T. Gonçalves and F. Moura-Pires. An attribute redundancy measure for clustering. In R. E. Mercer and E. Neufeld, editors, *Advances in Artificial Intelligence, 12th Biennial Conference of the Canadian Society for Computational Studies of Intelligence, AI '98, Vancouver, Canada*, volume 1418 of *LNCS*, pages 273–284. Springer, 1998.
15. T. Hu, C. Qu, C. Lim T., S. Yuan Sung, and W. Zhou. Preserving patterns in bipartite graph partitioning. In *18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2006), Washington, USA*, pages 489–496. IEEE Computer Society, 2006.
16. A. Lancichinetti and S. Fortunato. Community detection algorithms: A comparative analysis. *Phys. Rev. E*, 80(5):056117, November 2009.
17. X. Liu and T. Murata. Evaluating community structure in bipartite networks. In A. K. Elmagarmid and D. Agrawal, editors, *Proceedings of the 2010 IEEE Second International Conference on Social Computing, SocialCom / IEEE International Conference on Privacy, Security, Risk and Trust, PASSAT 2010, Minneapolis, USA*, pages 576–581. IEEE Computer Society, 2010.
18. M. Rosvall and C. T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008.
19. S. E. Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.