Conceptual Clustering of Complex Objects: A Generalization Space based Approach

Isabelle BOURNAUD and Jean-Gabriel GANASCIA

LAFORIA-IBP, Université Paris 6
4, place Jussieu - Boite 169
75252 Paris Cedex 05 - France
email: {bournaud, ganascia}@laforia.ibp.fr

Abstract. A key issue in learning from observations is to build a classification of given objects or situations. *Conceptual clustering* methods address this problem of recognizing regularities among a set of objects that have not been pre-classified, so as to organize them into a hierarchy of concepts. Early approaches have been limited to unstructured domains, in which objects are described by fixed sets of attribute-value pairs. Recent approaches in structured domains use a first order logic based representation to represent complex objects. The problem addressed in this paper is to provide a basis for the analysis of complex objects clustering represented using conceptual graphs formalism. We propose a new clustering method that extracts a hierarchical categorization of the provided objects from an explicit space of concepts hierarchies, called *Generalization Space*. We give a general algorithm and expose several complexity factors. This algorithm has been implemented in a system called COING. We provide some empirical results on its use to cluster a large database of Chinese characters.

1 Introduction

A key issue in learning from observations is to build a classification of given objects or situations. This kind of inductive learning, where the objects have not been pre-classified by a teacher, is often called *unsupervised learning*. It consists in searching for regularities in the training objects, so as to organize them into categories [6] [12] [24]. Conceptual clustering and traditional techniques developed in cluster analysis and numerical taxonomy differ in two major aspects. Firstly, the conceptual clustering problem is not only to build a partition of the set of objects into separated clusters, but also to associate a *characterization* to each cluster acquired. This characterization, called intensional description of cluster [30], has to be expressed in the terms used to describe the objects. Secondly, a function of individual objects, but is dependent on concepts that describe the clusters, i.e. the intentional descriptions of the clusters [4].

More precisely, conceptual clustering has been defined by Michalski and Stepp [24] as:
Given:

- a set of objects and their associate descriptions

Find:

- clusterings that group these objects into *concepts*;
- an intentional definition for each concept,
- a hierarchical organization for these concepts.

The classification scheme resulting of a conceptual clustering process usually corresponds to a *concept hierarchy*. The task of *concept formation* is very similar to that of conceptual clustering, with the added constraint that learning be incremental [14].
Early systems in unsupervised concept learning have been limited to unstructured domains, in which objects are described by fixed sets of attribute-value pairs, like in CLUSTER/2 [24], COBWEB [9] or UNIMEM [20]. However, objects frequently have some natural structure, i.e. have components and relations among these components. Attribute-value based representations are not well suited to represent such complex objects [30]. Recent approaches in structured domains use a first order logic (FOL) based representation to represent complex objects. As examples, we may refer to systems such as CLUSTER/S [29] or LABYRINTH [30]. The major problem with such representations is that of the number of matching, i.e. the number of different ways two descriptions structurally match. In FOL based language, there is potentially an exponential number of matchings [27]. One way to limit the matching complexity is to restrict the expressiveness of the language used to characterize clusters [21]. This type of restriction is easier to express and to formalize in a graph-based formalism than in a FOL-based representation [32]. However, no work has analyzed in depth the use of conceptual graphs in conceptual clustering.

The problem addressed in this paper is thus to provide a basis for the analysis of complex objects clustering represented using conceptual graphs formalism [28]. Conceptual graphs are formally defined with a model theoretic semantics. They have all the expressive power of logic but are more intuitive and readable, with a smooth mapping to natural language [8]. Moreover, conceptual graphs are often used in data knowledge representation.
The main differences between our approaches and related works are first that the complex objects are represented using the conceptual graphs formalism, and secondly, the conceptual clustering process consists in searching through an *explicit* space of clusters of objects, called *Generalization Space*. A Generalization Space is an inheritance network, whose construction is based on a restricted graphs' generalization. We propose to extract from a particular Generalization Space different classifications corresponding to hierarchical organizations of the acquired concepts.

In section 2, we briefly review the classical clustering methods and present our approach based on the notion of Generalization Space. In the following section we describe a method called MSG, originally developed by Mineau [25], to build a Generalization Space. In section 4, we describe the conceptual clustering method proposed on an example and give a general algorithm used in an implemented system called COING. The last section presents some experimental results on conceptual clustering of Chinese characters.

2 Conceptual clustering

2.1 Related approaches

Different clustering methods have been developed in the area of research on unsupervised learning. On the one hand, the classic *agglomerative* clustering methods build a binary tree in a "bottom-up" manner. They begin with as many categories as the number of objects. They compute the similarity between all pairs of categories, and merge the two most similar categories into a single category. The merging process is repeated until all objects have been placed in a single category representing the root of the tree. Examples of systems based on this method are KBG [1] or MK10 [31].
On the other hand, the result of *divisive* methods is a tree built in a "top-down" manner. These methods begin with a single category containing all the objects, and repeatedly subdivide this category until all categories contain only a single object. The partitioning algorithm ("Nuées dynamiques") developed by Diday [5], and which has been used as a basis for developing the CLUSTER's family of systems, uses a top-down approach. Divisive methods are also used in concept formation approaches [14].

These two methods have been both used in Numerical Taxonomy and in Machine Learning. Although they differ in the building process, they both can be viewed as a search through a *non explicitly* built space of concept hierarchies [11] [14]. This is summarized on figure 1.

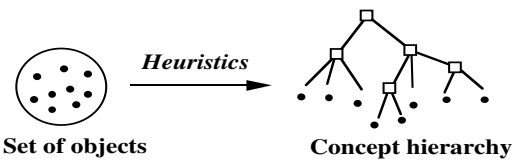


Fig. 1. Classical approaches principle

2.2 A new approach

We propose a new approach to conceptual clustering, called COING, which consists in searching a concept hierarchy of a set of objects through a space of concept hierarchies *explicitly* built: the Generalization Space. This approach is based on a representation of complex objects using conceptual graphs [28]. We will first introduce some terminology and an example used along this article.

Knowledge representation Our work being related to the MSG developed by Mineau [25], we have considered the same restrictions on the conceptual graphs: existential graphs where only binary relations are used. Such graphs may be decomposed without ambiguity as a set of triplets $< concept_1, relation, concept_2 >$. We call *conceptual arc* or *arc* such a triplet of a conceptual graph. Throughout this paper, we will use as an example a set of objects to be clustered (see figure 2). This set of objects, taken from [14], provides a good example of an application of our approach in spite of the fact that the objects do not require the full expression power of the conceptual graphs formalism. In this

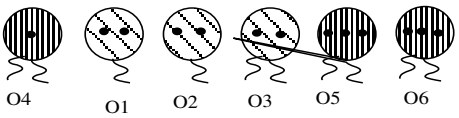


Fig. 2. An example of set of objects

example, we represent each object using conceptual graphs; figure 3 presents the representation of an object.

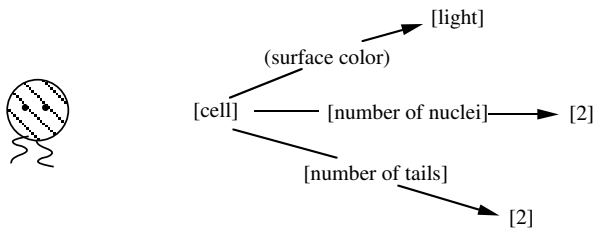


Fig. 3. Representation of t

The term *concept* will be used within two contexts:

- an *acquired concept* in the conceptual clustering paradigm designs the entity "cluster + cluster description",
- a *domain concept* corresponds to a concept of the conceptual graphs formalism.

Generalization Space Structure A Generalization Space (GS) is an inheritance network. In this network, a node n is a pair (Cov(n), Desc(n)) where Cov(n), the *coverage* of n, is the set of objects covered by n; and Desc(n), the *description* of n, corresponds to the common characteristics of nodes of Cov(n) (i.e. the set of the common arcs).
A node p is a *father* of a node n iff: Desc(p) is more general than Desc(n) (i.e. arcs of Desc(p) are arcs that are more general than arcs of Desc(n)) and Cov(n) is strictly included in Cov(p). The set of m fathers of a given node n are noted F(n)=$\{f_1(n), f_2(n), ..., f_k(n), ..., f_m(n)\}$. For example, F(O3)={n2, n3} and F(O2)={n4} on figure 4. We call *elementary node* a node covering exactly one of the provided objects, and *generalizing node* any other node.
The Generalization Space is different from the "generalization hierarchy" introduces by Sowa [28]. Indeed, in the GS, nodes' descriptions have not to be a connected graph but are only a set of arcs [26], possibly disconnected. Figure 4 presents the generalization space obtained with the set of objects presented in figure 2.
A Generalization Space may have more than one root node. In order to guaran-

Fig. 4. The Generalization Space for the set of objects of fig.2: GS1

tee that the GS has a unique root, and to provide the GS with an upper lattice structure, we define, as in Mineau [25], a *minimal description* which generalizes all the objects' descriptions. The root of the GS corresponds to the root of the concept hierarchy that will be extracted from the GS.

2.3 Clustering objects using a GS

Any set of nodes of the GS, that cover all the provided objects, may be considered a priori as a classification of the set of objects. In that way, the GS may be viewed as a *space of classifications*. In fact, the GS contains a sub-set of all the

potential classifications of the training objects, according to the restriction used in the building of GS (see section 3.1.). COING's approach consists in extracting one concept hierarchy from the GS. This approach is summarized in the figure 5.

Fig. 5. Coing's principle

To build a GS from a set of objects represented using conceptual graphs, we use the MSG method developed by Mineau in [25]. This method is briefly sketched in the next section. In contrast to Mineau who considers the GS as a classification by itself, our goal is to extract from the GS one particular classification scheme. The main classification schemes that one meets in the literature are partition (mutually exclusive categories), clumping (overlapping categories), hierarchy with overlap, or hierarchy without overlap [4] [11]. As in major conceptual clustering methods, we are interested in finding the latter kind of structure, i.e. concept hierarchies [14] [23]. Indeed, hierarchical organization is very natural for many domains [19]. Moreover, hierarchical structures have proven to be useful for organizing and managing large databases because of their searching efficiency [7] [22]. Finally they are the easiest to be interpreted by the experts.

To the best of our knowledge, no work has yet dealt with the problem of extracting a concept hierarchy from an explicitly built space of concept hierarchies. Section 4 presents more precisely COING's approach to this problem.

3 Generalization Space

3.1 Building a GS

Building the GS requires to group into clusters all objects that have similarities. One way to "evaluate" similarity between objects is to generalize the objects by searching the common features subsuming the objects. Generalizing objects, in the representation used, requires to match graphs, which is a problem known to be NP complete [13].
Among the several approaches developed to find a partial solution to this problem, one with a low complexity is the one developed by Mineau in the MSG [25]. This approach consists in searching the largest sub-sets of common *arcs* and not the largest common *sub graphs*. In doing so, it ignores the fact that any two arcs

may be connected. In other words, a graph is considered as a set of arcs; *matching graphs* is viewed as *matching arcs*. It consists in restricting the generalization language to a one-to-one matching [18]. As a result of this restriction, the complexity of the graph matching process is polynomial in the number of objects and arcs. Nevertheless, the GS does not contain all the existing generalizations of objects (only the ones that may be expressed in this restricted language).

3.2 Size of the GS

Let N be the number of objects and k the average number of arc in objects' descriptions.
The GS does not contain as many nodes as the number of all possible sub-sets of objects, but only nodes for sub-sets of objects having common features. The GS nodes are obtained in grouping in the same node all the arcs covering the same objects. The number of arcs in the GS, elementary and generalized arcs, is less or equal than $2^3 \times N \times k$ [25]. Since the GS nodes contain at least one arc, the number of nodes of the GS (GS size, noted $|GS|$) is linear with the number of objects:

$$|GS| \leq 8 \times N \times k$$

Our method to build the GS differs from the one proposed by Mineau since we introduce background knowledge, represented as a hierarchy of the domain concepts, allows the descriptions of the generalizing nodes to contain arcs which do not explicitly appear in the objects' descriptions. For example, the domain concepts "black" and "grey" may be generalized by the concept "dark". The size of the GS obtained with this method is linear with the number of objects, given by the following formula: (where P is the maximum depth of the domain hierarchy).

$$|GS| \leq P^3 \times N \times k$$

In Mineau's work, all concepts get generalized in "?"; the maximum depth of the domain hierarchy, P, is thus equal to 2.

4 Coing's approach

In [11], the authors explain that the conceptual clustering process is "composed of three distinct sub processes: the process of deriving a hierarchical classification scheme, the process of aggregating objects into individual classes; and the process of assigning conceptual descriptions to object classes". In our approach, the aggregation and characterization processes are made during the building of the GS. The remaining problem is to derive a classification scheme from the GS. The goal of COING's approach is to extract from the GS a classification, corresponding to a *hierarchical organization* of the acquired concepts. We shall briefly present a two-stage process for such an extraction. We first give a simple extraction algorithm. We then describe the criteria used to choose one hierarchy over another, and give the algorithm complexity.

4.1 A two-stage process

The process of extracting a hierarchy from the GS is composed of two stages: *skeleton selection* and *skeleton reduction*. Let n be the given node, $f_i(n)$ be a father of n and $s_j(n)$ a son of n; let Cov(n) be the coverage of n and Desc(n) its description.

Skeleton selection As previously said, the GS has an inheritance network structure, with a unique root thanks to the *minimal description* used during the GS's construction. Thus, each node of the GS, excluding the most general and its immediate sons, may have more than one father, i.e. may be generalized by more than one node of the GS (see object O3 of figure 4). In a hierarchy, each node has only *one* father. Extracting such a structure from the GS may be viewed as the problem of *choosing a unique father* for each node having more than one father. We call *skeleton*, the structure obtained after a single father has been chosen for each node of the GS.

The selection of a skeleton consists in:

- choosing for each node n a unique father $f_c(n)$ from its set ffffff

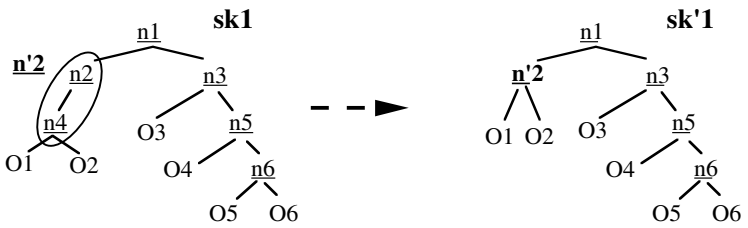


Fig. 7. The resulting skeleton for sk1 of fig.6

n4 have been grouped in a "new" node n'2. The updated content of the node n'2 is presented on figure 8.

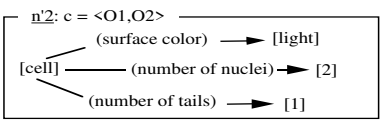


Fig. 8. Content of the node n'2 of fig.7 (different of node n2 of fig.4)

Skeleton reduction Given one skeleton sk_i, different *hierarchies* may be extracted. These hierarchies differ in the number of generalizing nodes they contain. In order to generate all the different hierarchies, we define an *absorption operation* over nodes. The process of restricting a skeleton to a hierarchy consists in choosing which nodes of the skeleton to absorb. Absorbing a node n into its father node $f_k(n)$ consists in propagating the description of n "down into" the descriptions of its sons that are not elementary nodes, and linking the sons of n to the absorbing node $f_k(n)$. The hierarchy h1 presented in figure 9 has been obtained from the skeleton sk'1 of figure 7 by absorbing the node n5 into its father node n3. The content of the node n"6 is presented on figure 10.

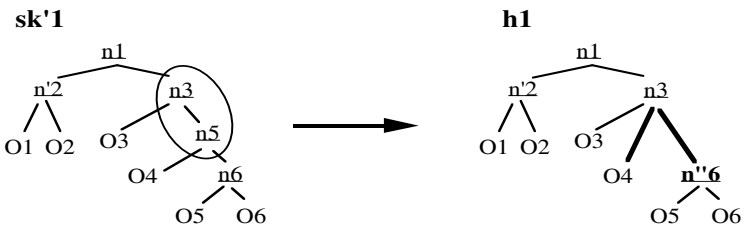
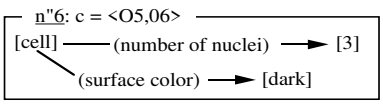


Fig. 9. One hierarchy extracted from sk'1: h1


n"6: c = <O5,06>
[cell]
(number of nuclei)
[3]
(surface color)
[dark]

4.3 Extraction criteria and computational complexity

The most classical criterion prefer one concept hierarchy over another one is the general criterion that may be expressed as: *choose categories that maximize similarity within categories and that concurrently minimize similarity between categories* [16] [17]. This criterion is in fact a good tradeoff between within-category similarity and inter-category dissimilarity of objects used in systems that do not build explicitly a space of concept hierarchies.
In our framework, we may define a set of criteria that will allow to define more explicitly the type of control desired on the building of the hierarchy. This approach may be particularly interesting in the context of conceptual clustering revision where potential users interactively request revisions of the provided hierarchy [2].

Choice for skeleton selection As previously explained, selecting a skeleton may be viewed as choosing a unique father for each node of the GS. Choosing a father for a node corresponds to decide which father is the "important" one, according to the clustering task. We have identified two types of criteria that may be used for choosing one father:

- a *preference function*, based on the coverage and the description of the nodes, that orders the father (for example the depth of the fathers in the GS). The preference function consists in choosing as father for a node the one with the minimum index value. Here are two different preference functions we have used: $f1(n) = \alpha \times Cov(n) + \frac{\beta}{Desc(n)}$ et $f2(n) = \frac{\alpha}{Cov(n)} + \beta \times Desc(n)$. The function $f1$ favors the nodes with small coverage and long descriptions, whereas the second function $f2$ favors the nodes with large coverage and short descriptions. The function $f2$ with $\alpha = \beta = 1$ has been used to extract sk'1 of GS1 of figure 4.
- a *preference bias*, based on user defined knowledge, that constrains the choice of father. This background knowledge may help to determine which features are relevant in a given situation [17]. For example, the relation "number of tails" is more important than "number of nuclei". It may also express some relations between objects and clusters; for example the fact that the user does not want the object O_i to be a son of the node p or objects O_i and O_j to have the same father.

In stage 1 of the algorithm, local criteria have been used to extract a skeleton. In stage 2, a global criterion is used to output one concept hierarchy.

Choice for skeleton reduction We have identified a set of parameters that describe the structure of a concept hierarchy: depth, width, maximum number of sons per node, ... These parameters have the great advantage to be easily interpretable by the user. We used these parameters to constrain the number of nodes to be absorbed. The choice of which nodes to absorb is based on an information measure criterion similar to that of *Category Utility* [16].

Computational complexity We will briefly give some time complexity results of the proposed method. Let N be the number of objects.

GS's construction

In the worst case, the time complexity for the construction of the GS in $\Theta(N^2)$.

Skeleton selection

Let SK be the set of potential skeletons for a GS. The size of SK is given by the following formula:

$$|SK| = \prod_{n_i \in GS} card(F(n_i))$$

In practice, this number may be small, e.g. the number of skeletons for the generalization space GS1 of the figure 4 is 2.

The proposed method does not explore all the possible skeletons. In fact, the time complexity of the skeleton extraction is, in the worst case, in $\Theta(N^2)$.

Skeleton reduction

The size of the skeleton (i.e. number of nodes) grows linearly with the number of objects, the time complexity of the reduction process is also growing linearly with the number of objects.

5 Experimentation

The algorithm presented above has been implemented in the system COING developed in LeLisp.

One of our domain of application of the COING's approach is related to Chinese characters. Chinese characters are complex structured objects that are naturally represented using conceptual graphs. Figure 11 presents a Chinese character represented using conceptual graphs. In an interdisciplinary research group, we