Dissimilarity measure for collections of objects and values

Petko Valtchev and Jérôme Euzenat

INRIA Rhône-Alpes
ZIRST, 655 av. de l'Europe, 38330 Montbonnot Saint-Martin, France
phone + 33 (0)4 76 61 53 75, fax + 33 (0)4 76 61 52 07
{Petko.Valtchev, Jerome.Euzenat}@inrialpes.fr

Abstract. Automatic classification may be used in object knowledge bases in order to suggest hypothesis about the structure of the available object sets. Yet its direct application meets some difficulties due to the way data is represented: attributes relating objects, multi-valued attributes, non-standard and external data types used in object descriptions. We present here an approach to the automatic classification of objects based on a specific dissimilarity model. The *topological* measure, presented in a previous paper, accounts for both object relations and the variety of available data types. In this paper, the extension of the topological measure on multi-valued object attributes, e.g. *lists* or *sets*, is presented. The resulting dissimilarity is completely integrated in the knowledge model TROPES which enables the definition of a classification strategy for an arbitrary knowledge base built on top of TROPES.

1 Introduction

The global aim of our study is the development of a strategy for automatic taxonomy building within object knowledge bases. Methods for inferring taxonomic structures, or classifications, first appeared within the numerical taxonomy paradigm, in statistics. Statistic classification is aimed at detecting regularities in sets of feature-described individuals. Feature values, mainly numerical, are used to establish a proximity function on individuals. The classification methods tend to group highly similar individuals into clusters and, in some cases, organize clusters hierarchically. The automatic classification may be used to discover the conceptual structure of the specific domain where data comes from [19].

Such structure-detecting methods may be useful for domains where large amounts of data are processed, like databases and knowledge-based systems. In fact, the extraction of structural knowledge from databases, in particular by means of clustering, is one of the goals of the recently emerged *data mining* field [14].

Our own concern is the introduction of such techniques within object formalisms. The classification task has to be carried out within the knowledge base, i.e. in the context in which the data is stored and manipulated. Therefore, the complexity of the object description languages has to be successfully dealt with.

The necessity of classifying more complex data motivated the constitution of the *conceptual clustering* paradigm as an extension of numerical taxonomy in machine learning [13]. In contrast to the statistical methods conceptual clustering ones work on symbolic and structured data and put the emphasis on the constitution of intentional

descriptions, a concept, for each cluster. Various conceptual clustering methods on different kinds of data description formalisms have been reported since: attribute-value-like [8], first order logic [2] and graph formalisms [11]. Other approaches like *concept formation* [10] or Bayesian classification [6] rely on probabilistic considerations about feature values when grouping individuals.

For the purposes of object taxonomy inference, a proximity-based approach seems to be well suited [3]. Therefore, a key problem to address is the definition of a proximity model which fits object descriptions. This means, in particular, that the model should be able to process the variety of object features and inter-object relations admitted by the concrete formalism.

We propose a generic dissimilarity model, the *topological* measure, which is universally applicable both on features and relations. Its basic principle is the use of the hierarchical structure of a domain to assess the proximity between domain elements. The model has been presented in a previous paper [18].

In the present paper, the extension of the topological measure on multi-valued object attributes is discussed. The paper starts by a motivating example of a domain which illustrates some specific features of the object formalisms like complex objects, rich data type sets and multi-valued attributes (Section 2). Then, for self-containment purposes, we recall the definition of the topological dissimilarity (Section 3). We also discuss the concrete functions in some typical cases of domain structure like *nominal*, *ordinal*, etc. Next, the extension of the topological dissimilarity on multi-valued attributes, *sets* and *lists*, is introduced (Section 4). Finally, the integration of the measure within a concrete object model, TROPES, and its taxonomy building tool T-TREE is presented (Section 5).

2 Motivating example

Electromyography is a set of electrophysiological technics which allow the neuromuscular diseases to be diagnosed. The electromyographic diagnosis is carried out from a systematic acquisition of numeric and symbolic data. It is decomposed into a set of well defined steps: formulation of hypotheses and specialized examination procedure suited to the patient treated, evaluation of procedure results, validation or questioning of the current hypothesis, elaboration of a conclusion. The domain of electromyography (EMG) is broad, covering more than a hundred existing diagnoses, and about four thousand tests of nervous or muscular structures.

The complexity of EMG examination procedure and the specific conditions in which it is carried out (tests are painful and unpleasant for the patient) make the implementation of a decision support system for the physician quite useful. MYOSYS [20] is a knowledge-based decision supporting system on EMG built on top of an object formalism. The system has been designed to assist the physician in different tasks ranging from symptom evocation to test choice. A base of already resolved EMG cases is integrated into MYOSYS. The way cases are modeled as complex objects is described below.

2.1 EMG case model

When a real-world domain is modeled with object knowledge formalisms the domain entities are represented as objects. Entity features are modeled by object attributes. Attribute values belong to a specific domain, data type or object set.

In the EMG field, entities are divided into several *concepts*: EMG case, clinical data, hypothesis, test, EMG conclusion, etc. The objects that represent entities of the same concept are described through a fixed set of attributes. Thus, they form homogeneous groups, we shall further call those groups *object sorts*. EMG object sorts define attributes of various domain structure. For example, test results are mainly numerical values: floats or integers, but may also be expressed as ordinals. Nominal features are used to describe the general state of the examined anatomic structures, while anatomic structures, i.e. muscles and nerves, themselves constitute hierarchical domains. Moreover, normal values for tests are introduced in form of intervals.

The existing relations between domain entities are modeled through *object-valued* attributes, as opposed to *primitive* attributes, describing features. For example, each EMG case is characterized by its clinical data. In the model, the `cl-data` attribute of the EMG case sort takes its values in Clinical data. Objects-valued attributes give rise to *complex* objects. A sub-set of the EMG domain concepts together with their relations are shown on Fig. 1. As it is shown on the figure, relations may associate an entity to a group of other entities. Thus, an EMG examination includes several tests and may lead to a set of final conclusions. One-many relations are modeled through multi-valued object attributes which may be defined on primitive features as well (see Section 4).

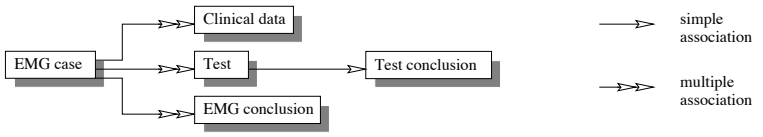

Fig. 1. Some EMG domain concepts and relations among them.

Objects are organized in class taxonomies. A class represents a group of objects, its instances. It is characterized in terms of the same attribute set as its instances. Class attributes describe sub-domains to which the attribute values of instances should belong. Usually, each object sort is assigned a class taxonomy. In the case of EMG, some of the domain concepts like hypothesis, clinical data, conclusion are subject to standardization so the underlying hierarchical structure is well known (see Fig. 2). In contrast, the sets of EMG tests and EMG cases admit no hierarchical structure a priori, since they are specific to a given physician or a hospital department. We tend to see here, the utility of clustering methods: they may be applied to extract a meaningful taxonomy from a set of objects, say EMG cases. The obtained structure can be useful both for extracting the expert knowledge from data and for optimizing the object storage.

In the following section, issues related to automatic classification of objects are discussed.

2.2 Classification-related problems

The construction of meaningful object taxonomies requires an efficient means for assessing objects. However, most of the numerical and conceptual clustering methods

are defined upon data representations which are much simpler than the above object model.

First, only a small set of data types, though variable, are admitted in the description of individuals. Compared to this, object formalisms use the whole variety of data types they inherited from programming languages: integers, reals, strings, Boolean; as well as other, less common types like date for example. Even more, some concrete object models [5] do not limit the set of the admitted data types: new types may be imported from outside. External types are introduced through an abstract data type (ADT), which represents a minimal interface of type management primitives (identity predicate, order predicate, etc.).

In addition, most of the existing clustering methods admit only primitive attributes in the description of individuals. However, the relations between objects represent an important part of the domain model. Thus, an object is characterized by the total set of its attributes, both primitive and object-valued. Clustering with relational data within a logical formalism has been studied in machine learning [2]. First attempts to adapt the method on objects have been reported in [3], but no universal approach exists so far.

Finally, objects are often described by means of multi-valued attributes. Such an attribute is defined upon a basic domain, object sort or primitive type, through a collection constructor, *set* or *list*. As collections may have variable cardinality, they cannot be compared directly. Some work on multi-valued features has been done in different fields within the frame of machine learning: in [12] the multiple associations between individuals has been studied for concept formation purposes, whereas the utility of set-valued primitive attributes in decision tree induction is discussed in [7].

For the purposes of object taxonomy building, a proximity-based strategy seems to be a reasonable choice. It requires, however, the definition of proximity functions which satisfies the following criteria. First, proximity between objects should depend on each of their attribute values. In other words, the proximity function should take into account all the kinds of object attributes, in particular the object-valued and multi-valued ones. Next, each data type used in object descriptions has to be processed with the highest possible precision. Finally, the overall object proximity should remain of a low computational cost.

In the following, a dissimilarity model which meets the above requirements is presented. First, a generic function for single-valued object attributes is introduces. Then, the function is extended on collections of both primitive values and objects.

3 Topological dissimilarity

Providing a set of individuals I with a dissimilarity measure means defining a function $d : I \times I \longrightarrow \mathbb{R}_0^+$, which satisfies, for arbitrary $a, b \in I$: (i) $d(a,b) \geq 0$ (*positiveness*), (ii) $d(a,a) = 0$ (*minimalness*) and (iii) $d(a,b) = d(b,a)$ (*symmetry*). Most often, the dissimilarity measures are calculated on the features of the individuals. For this purpose, each feature is provided with a function to assess resemblances between values. Usually, these are *ad hoc* functions tied to the feature types (e.g. *nominal* or *ordinal*).

In the context of an object formalism with an extensible type system, i.e. where user-defined types are possible to import, such an *ad hoc* approach fails. A possible remedy could be to include a primitive for value resemblance computation in the mandatory interface for external types. For a representation formalism, this solution

seems to be rather restrictive. A reasonable alternative consists to define a generic function which applies to all data types, both built-in and external, admitted by the formalism. The function could be then overridden by a user-provided primitive, which fits better a particular data type. The topological dissimilarity model represents such a universal means for comparing members of a given domain. It is based on the fact that all domains share a common structure. In fact, for a given object attribute of a primitive domain D, the restrictions imposed by object classes on that attribute, let us call them *type expressions* like in [5], define sub-domains of D. The set of type expressions on D are naturally provided with an inclusion relationship called *sub-typing*. Sub-typing induces a partial order structure on D which is quite similar to the class taxonomy on an object set (see [18]). We use the classification scheme (CS) model to provide a formal description of the structural analogy between domains.

3.1 Classification Schemes

For the sake of compactness, we shall only insist on model's basic components (see [5] for details).

A classification scheme (CS) is defined over a domain, say D, provided with two languages: L_I, of individuals, and L_C, of categories. Individuals are interpreted as domain entities, whereas the categories have two different interpretations. The first one, called *abstract* interpretation (I_A), is the set of all entities the category may potentially represent. The second one, namely the *real* interpretation (I_R), includes only elements which are currently represented by the category. A *taxonomy*, with respect to one of the interpretations, is a partially ordered set of categories whereby the order respects the inclusion of interpretations. Now, a *classification scheme* $S = \langle L_C, C, \ll, \leq \rangle$ is composed of two taxonomies:

- $\langle L_C, \ll \rangle$ respects I_A; $\ll$ is called *sub-categorization criterion.*
- $\langle C, \leq \rangle$ respects I_R whereby $C \subseteq L_C$ and $\leq$ is called *sub-categorization relation.*

Both object sorts and primitive types may be seen as classification schemes. For example, in the EMG domain the integer type, used to encode the age of a patient, can be seen as a classification scheme. The admitted integer numbers constitute the language of individuals whereas categories correspond to integer intervals. On Fig. 2, the taxonomy of the classification scheme on the Clinical data sort is given. Categories representing object classes are given their standard names. Individuals, i.e. objects of the Clinical data sort, are drawn as rectangles and are attached to their most specific categories.

A classification scheme of an object sort may be obtained as *product* of the classification schemes of the object attributes. Thus, the EMG test sort may be seen as the product of all its attributes like test results, tested anatomic structure, test conclusion, etc. The same holds for EMG case, EMG conclusion and Clinical data. The reverse operation of the *CS product* is *projection*; it allows the separation of a sub-set of the product factors.

The uniform representation of both object sorts and primitive types offered by the CS model may be used in the definition of a proximity function. In the next section, a dissimilarity measure ($d : L_I \times L_I \longrightarrow \mathbb{R}_0{}^+$) is presented which is entirely based on the taxonomy structure of a CS.

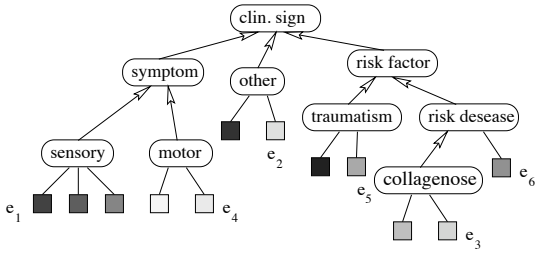

Fig. 2. A possible taxonomy on Clinical data.

3.2 Basic model

The *topological* dissimilarity is made up of of two functions: a *high level* function, d^t, computed on the objects and *low level* function, $\overline{\delta}$ that accounts for the field values.

The *low level function* δ is measured on the taxonomy of a given classification scheme $S = \langle L_C, C, \ll, \leq \rangle$. It may be roughly described as the shortest path between two individuals in the taxonomy. In fact, what is measured is the minimal sum of path lengths over paths leading to a common category. Thus, given e_1 and e_2 in L_I with $C^* = \{c | e_1, e_2 \in I_R(c)\}$:

$$\delta(e_1, e_2) = \min_{c \in C^*} [dist(e_1, c) + dist(e_2, c)]$$

where $dist(e, c)$ is the number of intermediate categories between individual e and the category c. The above function is further normalized:

$$\overline{\delta}(e_1, e_2) = \frac{\delta(e_1, e_2)}{\max_{x_1, x_2 \in \cup_C I_R(c)} \delta(x_1, x_2)}$$

The way $\overline{\delta}$ works is illustrated on the Clinical data taxonomy on Fig. 2. For example, the topological dissimilarity between the couple of objects $e1$ and $e2$ which represent symptoms of `sensory` and `motor` disorder respectively is $\delta(e_1, e_4) = 4$ since the shortest path between them is of length four. Since the maximal path-length in the above taxonomy is seven ($\delta(e_1, e_3) = 7$), the normalized topological dissimilarity is $\overline{\delta}(e_1, e_4) = 4/7$. Here are some other examples: $\overline{\delta}(e_1, e_2) = 5/7$, $\overline{\delta}(e_1, e_3) = 1$ and $\overline{\delta}(e_2, e_3) = 6/7$.

Both δ and $\overline{\delta}$ are valid dissimilarity indices. Moreover, δ may be extended to a category dissimilarity ($\delta_c : C \times C \longrightarrow \mathbb{R}_0^+$).

The *high level function* d^t is defined on a CS product, i.e. object sort, K of n direct factors (attributes). For individuals, say e and e', given with their corresponding unit projections (attribute values) e_i and e'_i ($i = 1 \ldots n$):

$$d^t{}_K(e, e') = Aggr_{i=1}^n \lambda_i \overline{\delta}(e_i, e'_i)$$

where the $Aggr$ is a generic aggregation operator and λ_i is the weight assigned to the i-th attribute of K. Various instanciations are imaginable for this operator, for example City bloc or other Minkowski metrics.

3.3 Concrete functions

Although defined on a graph structure, most often the computation of the topological measure does not require extensive graph search. In fact, for the most common domain structures, the generic model coincides with well known functions which are easy to compute directly on primitives included in the ADT. These functions are be implemented to speed-up the computation.

For instance, on a *nominal* domain D, like strings or symbols, the topological dissimilarity equals the reverse of the identity ($\overline{\delta} = 1 - id_D$). In fact, the classification scheme of such a domain is made up of all possible sets of elements, i.e. $L_C = 2^D$, since there is no reason to distinguish some of them. Thus, for a couple of individuals e_1 and e_2, the nearest common category corresponds to the set $\{e_1, e_2\}$. Consequently, all possible values for δ are 0 and 2 which yields after normalization $1 - id_D$.

When an *ordinal* domain D is considered, the categories can be intervals. Thus, the taxonomy is made up of all intervals on D with interval inclusion as sub-categorization relation. For a couple of values, δ accounts for the number of intervals between each value and the most specific common category. The latter is exactly the interval where those values are bounds. With some elementary computation one may show that $\delta(e_1, e_2) = 2 \times abs(ord(e_1) - ord(e_2))$. Consequently,

$$\overline{\delta}(e_1, e_2) = \frac{abs(ord(e_1) - ord(e_2))}{ord(max(D)) - ord(min(D))}$$

Since based on discrete structures, the topological dissimilarity is impossible to measure directly on a *continuous domain*. Therefore, we assume in this case $\overline{\delta}$ coincides with the normalized real number subtraction.

The low-level topological function applies to object sorts provided a taxonomy is available on them. In this case, the proximity between objects is assessed with respect to their mutual position within the taxonomy. In other words, $\overline{\delta}$ deals with objects as if they were atomic and no more products of attribute values. For example, when d^t is computed on a couple of EMG cases, $\overline{\delta}$ will be applied on each attribute. For a couple of Clinical data objects associated to the EMG cases, the graph distance between them within the taxonomy will be extracted by $\overline{\delta}$ as shown previously.

Disregarding object attributes means, in particular, that $\overline{\delta}$ explores the relational structure of the object sort set at depth one. Thus, the value of d^t on EMG cases depends on the respective EMG tests, but not on the Test conclusions.

We deliberately chose a measure based on the taxonomy. As a matter of fact, the existing taxonomy, created either manually or automatically, is a synthetic expression of the object sort (conceptual) structure. If it is meaningful (if it is not, then the whole modeling and/or clustering is meaningless), then a good dissimilarity approximates the object proximities induced by the taxonomy.

4 Multi-valued types

Multiple associations between entities lead to characterizations in terms of value collections instead of single values. For instance, chemical elements have several possible valences. When elements are modeled, the set of valences is to be associated to each of them. The nucleotide sequence associated to a gene is just another example.

Multiple associations are usually modeled through multi-valued object attributes. A multi-valued attribute is defined on a *basic* type, by means of a constructor: *set* or

list. Constructors apply to both primitive and object-valued attributes. For example, in the EMG domain, EMG case is assigned a set of tests.

When processing a multi-valued attribute difficulties arise due to the variable length of the collections. In machine learning, for example, set-valued features are often encoded through a set of single-valued ones and only rarely processed in a direct way (see [7] for a discussion).

4.1 Comparing collections

A generic dissimilarity model on multi-valued attributes should account for both the pairwise member dissimilarity and cardinality differences. One may imagine an exhaustive computation, leading to the average of all member pairwise dissimilarities. However, the obtained measure is not *minimal*, i.e. its values on identical collections are strictly non-negative.

In order for *minimalness* to be guaranteed, some preliminary selection of member pairs is necessary. More precisely, the set of selected pairs must satisfy: (*i*) a collection member may take part in at most one pair and (*ii*) the number of pairs is maximal. Those conditions define a *matching* between collections which is maximal in cardinality. In addition, we require the matching to minimize the total dissimilarity of the selected pairs.

4.2 Set-valued attributes

Let $S = \{e_1, e_2, ...e_k\}$ and $S' = \{e'_1, e'_2, ...e'_l\}$ be two sets over a basic domain D. Let also δ_D be a normalized dissimilarity measure on D.

Matching S and S' in the way described above means resolving the problem of an optimal matching in a weighted bipartite graph. Algorithms of $O(n^3)$ complexity for the problem have been reported in [1].

Let $M_{opt}(S, S') = \{(e_i, e'_j)\}$ be an optimal matching. When at least one of the sets is non empty, the set dissimilarity between S and S' may be defined as the average over the total dissimilarity of the matching and the unmatched elements taken with a maximal dissimilarity, i.e. 1.:

$$\delta_s(S, S') = \frac{\sum_{(e_i, e'_j) \in M_{opt}(S,S')} \delta_D(e_i, e'_j) + |l - k|}{\max(l, k)}.$$

If both S and S' are empty, then we set the result of the function to zero: $\delta_s(\emptyset, \emptyset) = 0$. The obtained measure is a valid dissimilarity index, since it is positive, symmetric and minimal.

For example, let $S = \{2, 7, 4, 3, 9\}$ and $S' = \{6, 4, 8, 1\}$ be sets constructed over an integer domain $D = [0, 10]$. An optimal matching is $M_{opt}(S, S') = \{(2, 1), (7, 6), (4, 4), (9, 8)\}$, consequently $\delta_s(S, S') = (3/10 + 1)/5 = 0,26$.

The δ_s function applies successfully to object sets as well. Indeed, let $S = \{e_1, e_2, e_6\}$ and $S' = \{e_4, e_5\}$ be two sets of EMG tests (see Fig. 2) associated to a couple of EMG cases. Their dissimilarity is $\delta_s(S, S') = (8/7 + 1)/3 = 5/7$ obtained with $M_{opt}(S, S') = \{(e_4, e_1), (e_5, e_6)\}$.

4.3 List-valued attributes

In the case of lists, the collection is provided with a sequential structure. The new structure implies some extra constraints for the matching procedure. Thus, for lists $L = \langle e_1, e_2, ...e_k \rangle$ and $L' = \langle e'_1, e'_2, ...e'_l \rangle$, a matching $M_l(L, L') = \{(e_i, e'_j)\}$ should preserve the order induced by the lists. In other terms, $M_l(L, L')$ should satisfy:

$$\forall (e_i, e'_j), (e_m, e'_n) i \leq m \Rightarrow j' \leq n'$$

The new kind of matching is more complex than the previous one. In fact, the sequential structure implies stronger dependencies between member pairs than in the previous case. For instance, $(e_1, e'_2) \in M_l$ implies $(e_2, e'_2) \notin M_l$ but also $(e_2, e'_1) \notin M_l$.

The task may be evaluated in the following way. First, with no loss of generality we may suppose $k < l$, the case $k = l$ being trivial. Then, for lists of different length the matching we are looking for may be seen as a mapping from the shorter list, L, to the longer one, L'. The number of all maps that preserve the list-induced order is C_l^k. In the worst case, this number is an exponential function of l. We use therefore a branch-and-bound algorithm exploring the space of all possible matchings, i.e. k-tuples on $[1, l]$, in lexicographic order.

The initial solution is provided by a greedy heuristic algorithm which implements a recursive divide-and-conquer strategy. At each step, it chooses the best pair, i.e. the one of lowest dissimilarity, between all possible pairs. For a given element of the shorter list, only matchings are considered which do not prevent other elements of the same list to be further matched. Thus, for e_i in L, only $e'_i, e'_{i+1}, .., e'_{l-k+i}$ will be taken into account. Once the best pair is fixed, it is added to the matching, its elements are extracted from their respective lists and each list is splited in two sub-lists: one to the left and one to the right of the extracted element. The algorithm is recursively applied on both pairs of respective lists; it stops with empty list. In the worst case, the complexity of the above procedure is $O(n^3)$.

For example, let $L_1 = \langle 5, 3, 6 \rangle$ and $L'_1 = \langle 2, 4, 6, 2, 7 \rangle$ be lists on the integer domain $D = [2, 12]$. For this couple of lists, the heuristic algorithm will provide a matching $M_l = \{(5, 2), (3, 4), (6, 6)\}$ with total dissimilarity between matched elements of 4/10. Now, when the *branch-and-bound* algorithm is applied with this initial solution, it will rapidly find the optimal matching $M_{lopt} = \{(5, 6), (3, 2), (6, 7)\}$ with total dissimilarity of 3/10.

Thus, the matching computed by the heuristic algorithm provides quite a high bound for the following search. Finally, the total dissimilarity between lists is completed in order to take into account the unmatched elements. Let $M_{opt}(L, L') = \{(e_i, e'_j)\}$ be an optimal matching obtained by the above procedure. With at least one non empty list, the list dissimilarity between L and L' will be:

$$\delta_l(L, L') = \frac{\sum_{(e_i, e'_j) \in M_l(S, S')} \delta_D(e_i, e'_j) + |l - k|}{\max(l, k)}.$$

Should both L and L' be empty, their dissimilarity is zero: $\delta_l(\emptyset, \emptyset) = 0$.

In the above example of integer lists, $\delta_l(L, L') = (3/10 + 2)/5 = 0,46$ with M_{lopt} and $\delta_l(L, L') = (4/10 + 2)/5 = 0,48$ with M_l.

With object lists $L = \langle e_1, e_6 \rangle$ and $L' = \langle e_3, e_4, e_2 \rangle$ (see Fig. 2), and matching $M_{lopt}(L, L') = \{(e_1, e_4), (e_6, e_2)\}$, the dissimilarity is $\delta_l(L, L') = (9/7 + 1)/3 = 16/21$.

5 Classification strategy

The topological measure represents an efficient tool for building a taxonomy within an object formalism. However, in case of multiple object sorts and several taxonomies to infer, the application of the measure, requires a specific strategy. In the following, we describe the strategy we developed for the case of the TROPES knowledge model [15].

In a TROPES knowledge base, objects are instances of disjoint *concepts*. TROPES concepts correspond to what we called *object sorts*: their instances share the same set of attributes. Values of a given attribute belong to a specific data domain, either a primitive data type or an object concept. Furthermore, the type system integrated to TROPES supports encapsulated external types, introduced via abstract data types [4], as well as multi-valued types.

The model has been provided with a taxonomy building tool, T-TREE [9] which implements some numerical classification algorithms. Enhanced with the topological measure T-TREE is able to process objects with no restriction and thus can infer several taxonomies on disjoint concepts in the knowledge base.

In doing that, the set of concepts is considered as a graph. In fact, the knowledge base may be considered as a graph where concepts and ADT are vertices and attributes are edges (see Fig. 1 for a partial view on that structure). The obtained structure is a directed acyclic graph since for the time being we excluded mutual dependencies between concept characterizations. When classification has to be carried out on several concepts the global principle is to process each concept only after all its subordinated concepts, i.e. those related by object-valued attributes. In the case of the EMG domain, this means that if EMG cases have to be classified, then the EMG test concept should first be provided with a suitable taxonomy. This amounts to exploring the graph structure in a bottom-up manner, at each step inferring a taxonomy on a concept by referencing the taxonomies on the concept attributes.

Such an exploration has the following advantages. First, both object features and relations are taken into account, whereby the greatest attention is paid to the domain structure of each attribute. Next, object relations are dealt with at a reasonable cost. In fact, only direct attribute values are processed, their possible structure and further relations remaining hidden. For example, when classifying EMG cases, each EMG tests will be considered only as a member of its class in the test concept taxonomy. Finally, the taxonomic structure discovered at a particular level is reused on higher levels.

6 Related works

A dissimilarity measure based on graph distance has been discussed in [16]. The underlying measure is defined on a semantic net and accounts for the shortest path between a couple of nodes. Link directions and nature are not considered. Compared to that model, ours focuses on taxonomy, i.e. specialization links, and considers only up-going paths.

In [3], a possible way to compare complex objects has been presented. The proposed similarity function considers all relations between individuals to be reflexive and transitive. Thus, the similarity of a couple of objects is assumed to depend on the similarity of all related objects. When the model is applied to the EMG domain, for example, the proximity of a couple of EMG tests is computed with respect to proximity of the whole EMG cases, the EMG conclusions, etc. This additional information is not

unlikely to disturb the EMG test proximity assessment. It seems, therefore, that the topological measure is better adapted to real-world domains where relations are mainly non-reflexive. Yet both models should be compared experimentally in order to find out which one is better.

Issues on inter-individual associations, matching and clustering with multiple individual sets have been addressed for the first time in [17]. The paper presents an extension of the concept formation algorithm COBWEB [10] for structured domains. A possible way to further extend the basic concept formation approach on multiple associations between individuals is described in [12].

7 Conclusion

The automatic inference of object taxonomies is a special kind of analyzing data and extracting implicit knowledge from it. A straightforward way to build object taxonomies is to use an automatic classification method on object sets. The detection of meaningful object clusters requires a proximity measure which completely fits object descriptions.

We described here such a measure, called *topological* dissimilarity. The *topological* dissimilarity is a generic model which applies to any data domain used in object descriptions. It allows to handle a variety of data types: nominal, ordinal, continuous, etc. as well as to successfully explore the inter-object relations during classification.

Furthermore, we proposed an extension of the topological dissimilarity for multi-valued attributes, sets and lists. which utility has been exemplified in the EMG domain. The computation of a dissimilarity between collections requires a preliminary step of matching between collection members. Strategies for matching sets and lists have been discussed. The extended model is able to process primitive and object-valued attributes as well as single and multi-valued ones.

Finally, we presented a strategy for taxonomy inference based on the extended *topological* dissimilarity. The advantages of the described strategy are multi-fold: (*i*) classification is carried out directly within the knowledge base, (*ii*) the domain structure of different data types is respected and (*iii*) the existing taxonomies on object sorts are reused in the construction of new taxonomies on other sorts.

This new measure has been integrated into the taxonomy building module of the TROPES system and is currently under evaluation. Its comparison with other measures is a subject of future works and shall include studies on several fields of application.

References

1. R.K. Ahuja, T.L. Magnanti, and J.B. Orlin, *Network Flows: Theory, Algorithms and Applications*, Prentice Hall, 1993.
2. G. Bisson, 'Conceptual clustering in a first order logic representation', in *Proceedings of the 10th European Conference on Artificial Intelligence, Vienna, Austria*, pp. 458–462, (1992).
3. G. Bisson, 'Why and how to define a similarity measure for object-based representation systems', in *Towards Very Large Knowledge Bases*, ed., N.J.I. Mars, pp. 236–246, Amsterdam, (1995). IOS Press.
4. C. Capponi, *Identification et exploitation des types dans un modèle de connaissances à objets*, Ph.D. dissertation, Joseph Fourier, Grenoble (FR), 1995.

5. C. Capponi, J. Euzenat, and J. Gensel, 'Objects, types and constraints as classification schemes', in *Proceedings of the 1st KRUSE symposium*, pp. 69–73, Santa Cruz (CA US), (1995).
6. P. Cheeseman, J. Kelly, M. Self, J. Stutz, W. Taylor, and D. Freeman, 'Autoclass: A bayesian classification system', in *Proceedings of the 5th Internatinal Conference on Machine Learning, Ann Arbor, MI*, pp. 54–56, (1988).
7. W. Cohen, 'Learning trees and rules with set-valued features', in *Proceedings of the 13th AAAI and 8th IAAI*, (1996).
8. F. Esposito, 'Conceptual clustering in structured domains: a theory guided approach', in *New Approaches in Classification and Data Analysis*, eds., E. Diday, Y. Lechevallier, M. Schader, P. Bertrand, and B. Burtschy, pp. 395–404, Berlin, (1994). Springer Verlag.
9. J. Euzenat, 'Brief overview of t-tree: the TROPES taxonomy building tool', in *Proceedings of the 4th ASIS SIG/CR classification research workshop*, pp. 69–87, Columbus (OH US), (1993).
10. D.H. Fisher, 'Knowledge acquisition via incremental conceptual clustering', *Machine Learning*, **2**, 139–172, (1987).
11. R. Godin, G.W. Mineau, and R. Missaoui, 'Incremental structuring of knowledge bases', in *Proceedings of the 1st KRUSE symposium*, pp. 179–193, Santa Cruz (CA US), (1995).
12. A. Ketterlin, P Gançarski, and J.J. Korczak, 'Hierarchical clustering of composite objects with variable number of components', in *Proceedings of the 5th International Workshop on Artificial Intelligence and Statistics*, eds., D. H. Fisher and P. Lenz, Fort Lauerdale (FL USA), (1995).
13. R. Michalski and R. Stepp, *Machine learning: an Artificial Intelligence approach*, volume I, chapter Learning from observation: conceptual clustering, 331–363, Tioga publishing company, Palo Alto (CA US), 1983.
14. G Piatetsky-Shapiro and W. Frawley, *Knowledge discovery in databases*, AAAI Press, 1991.
15. Sherpa project, *Tropes 1.0 reference manual*, INRIA Rhône-Alpes, Grenoble (FR), 1995.
16. R. Rada, H. Mili, E. Bicknell, and M. Blettner, 'Development and application of a metric on semantic nets', *IEEE Transactions on Systems, Man and Cybernetics*, **19**(1), 17–30, (1989).
17. K. Thompson and P. Langley, *Knowledge and experience in unsupervised learning*, chapter Concept formation in structured domains, 127–161, Morgan Kaufman, San Mateo (CA US), 1991.
18. P. Valtchev and J. Euzenat, 'Classification of concepts through products of concepts and abstract data types', in *Ordinal and symbolic data analysis*, eds., Y. Lechevallier E. Diday and O. Opitz, pp. 3–12, Heildelberg (DE), (1996). Springer Verlag.
19. B. van Cutsem, *Classification and dissimilarity analysis*, Lecture notes in statistics, Springer Verlag, New York, 1994.
20. D. Ziébelin, A. Vila, and V. Rialle, 'Neuromyosys a diagnosis knowledge based system for emg', in *Proceedings of the 12th International Congress of Medical Informatics in Europe*, Lisboa (PT), (1994).

This article was processed using the LaTeX macro package with LLNCS style