Similarity of Structured Cases in CBR

Eva Armengol and Enric Plaza

IIIA - Artificial Intelligence Research Institute.
CSIC - Spanish Council for Scientific Research.
Campus UAB, 08193 Bellaterra, Catalonia (Spain).
{eva,enric}@iiia.csic.es

Resum

Lazy learning methods are based on retrieving a set of cases similar to a new case. An important issue of these methods is how to estimate the similarity among a new case and the precedents. Most of work on similarities considers that the cases have a propositional representation. In this paper we present SHAUD, a similarity measure useful to estimate the similarity among relational cases represented using *feature terms*. Also we present some preliminary results of the application of SHAUD for solving classification tasks. In particular we used SHAUD for classifying marine sponges and for assessing the carcinogenic activity of the compounds in the Toxicology dataset.

1 Introduction

Lazy learning algorithms are based on retrieving a set of cases similar to a new case. A very important part of such algorithms is how evaluate the similarity of two cases in order to retrieve an adequate set of precedents. Most of lazy learning algorithms handle cases represented as vectors of pairs attribute-value, i.e. cases having a propositional representation. Usually, when the cases have a propositional representation, the similarity among them is assessed by comparing the similarity of the value of each atribute and then aggregating the similarities of all the attributes to obtain a global similarity of the cases.

Currently, there is most work on relational representation focus on inductive techniques. However, relational representation can also be useful for lazy learning techniques. An important approach to relational lazy learning algorithms is RIBL [8]. Recently, RIBL has been extended allowing representations with lists and terms [10]. In this new version of RIBL the similarity between cases is assessed using the standard similarities for numerical and discrete attributes, and a similarity based on the concept of *edit distance* for attributes with lists and terms.

The *feature terms* is a formalism that we used for representing the objects handled by different learning techniques ([1], [2]). In [3] we defined LAUD a similarity measure that assessess the similarity of two cases represented as feature terms. LAUD proved to be useful in the classification task. Nevertheless, LAUD can be improved since this similarity does not take into account the complete structure of the feature terms but only the leaves of this structure. In this paper we introduce SHAUD, the natural improvement of LAUD. The main goal in introducing SHAUD is to take benefit of all the structure provided by the feature terms. SHAUD assessess the similarity of two cases based on the complete structure of the cases (i.e. the leaves and the intermediate nodes of the feature terms). Thus, given two cases represented as feature terms, SHAUD distinguishes two parts in their structure: the one formed by the features and nodes present in both cases, and the part is formed by those features and nodes that are only present in one of the cases. For the common part SHAUD uses the sort hierarchy to compare the feature values. The resulting similarity is normalized using the whole structure of both cases.

The paper is organized as follows: the section 2 introduces the feature term formalism. Section 3 defines what is a relational case and then the SHAUD similarity is introduced. Section 4 shows some of the results obtained from the application of SHAUD in solving the classification task in two real domains: marine sponges and toxicology. Finally, section 5 discusses some of the previous work done on similitude with relational cases.

2 Relational Cases

We propose to represent the relational cases using the *feature terms* formalism introduced in [1]. Given a signature $\Sigma = \langle S, \mathcal{F}, \preceq \rangle$ (where S is a set of sort symbols that includes $\bot$; $\mathcal{F}$ is a set of feature symbols; and $\preceq$ is a decidable partial order on S such that $\bot$ is the least element) and a set ϑ of variables, a *feature term* is an expression of the form:

$$\psi ::= X : s[f_1 \doteq \Psi_1 \dots f_n \doteq \Psi_n]$$

where X is a variable in ϑ called the *root* of the feature term, s is a sort in S, $f_1 \dots f_n$ are features in $\mathcal{F}$, $n \geq 0$, and each Ψ_i is a set of feature terms and variables. When $n = 0$ we are defining a variable without features. The function *root(X)* returns the sort of the root.

A *path* $\pi(X, f_i)$ is defined as a sequence of features going from the variable X to the feature f_i. The *depth* of a feature f in a feature term ψ with root X is the number of features that compose the path from the root X to f, including f, with no repeated nodes.

A feature term can be seen as a labelled graph where the nodes are values (also represented as feature terms) and the edges are features. Given a maximum feature depth k, a *leaf feature* of a feature term is a feature f_i such that either 1) the depth of f_i is k or 2) the value of f_i is a term without features. We call *leaves (ψ, k)* the set of leaf features of a term ψ.

Sorts have an informational order relation ($\preceq$) among them, where $\psi \preceq \psi'$ means that ψ has less information than ψ' or equivalently that ψ is more general than ψ'. The semantic interpretation of feature terms brings an ordering relation among feature terms that we call *subsumption.* Intuitively, a feature term ψ subsumes another feature term ψ' ($\psi \sqsubseteq \psi'$) when all information in ψ is also contained in ψ'. See (referencia) for a more formal definition of the subsumption.

Using the $\preceq$ relation, we can introduce the notion of *least upper bound (lub).* The *lub* of two sorts is the most specific sort generalizing both. Feature terms form a partial ordering by means of the subsumption relation. The *anti-unification* is defined over the subsumption lattice as an upper lower bound with respect to the subsumption ($\sqsubseteq$) ordering.

Intuitively, the anti-unification (AU) of two feature terms gives what is common to both (yielding the notion of generalization) and all that is common to both (the most specific generalization). Therefore, the AU of two feature terms F_1 and F_2 produces a feature term D that contains the features that are common to both F_1 and F_2. The values of the features in D have to satisfy the following conditions:

1. If a feature f has the same value v in both examples F_1 and F_2, the value of f in D is also v.
2. In a feature f has value of sort s_1 in F_1 and value of sort s_2 in F_2, the value of f in D is the most specific sort common to s_1 and s_2, i.e. the least upper bound of s_1 and s_2 in the $\preceq$ sort order.
3. otherwise, the examples F_1 and F_2 cannot be anti-unified.

The anti-unification of a set-valued feature with value the set V_1 in F_1 and the set V_2 as value in F_2 is a set $AU(V_1, V_2)$ whose elements are obtained in the following way. First the set $C = \{d_k = AU(x_i, y_j) | x_i \in V_1$ and $y_j \in V_2$ is built with $Card(C) = Card(V_1) \times Card(V_2)$. The set $AU(V_1, V_2)$ is the subset of C containing the feature terms d_k that does not subsumes any other feature term in C. The cardinality of the set $AU(V_1, V_2)$ must be $min\{Card(V_1), Card(V_2)\}$.

3 Similarity of Relational Cases

There are three aspects that we need to define in order to perform CBR on relational cases: 1) to define a *case* from a constellation of relations, 2) to assess the similarity of values, and 3) to define a way to assess similarity between cases.

A *case* is a term defined (in feature terms) by two parameters: a *root sort* and a *depth.* That is to say, assuming a "case base" expressed as a collection of feature terms, a case is a feature term whose root node is subsumed by the *root sort* and whose depth is at most *depth.* An example of case specification is $case[\textsf{root-sort} \doteq sponge, \textsf{depth} \doteq 4]$ in the marine sponges domain (see §4).

In this section we introduce a new similarity measure called SHAUD. This measure takes into account both the *common features* present in both input cases and the *relevant features* that are those features that occurs at least in one of the two cases (notice that the relevant information also includes the shared features). SHAUD compares the common part obtaining in this way an assessment of how similar are both feature terms. The non-common part provides information about the information that two feature terms could share.

A feature term has two kind of features: leaf features and intermediate features. When a feature is a leaf the similitude among its values is assessed using the basic similitude explained in section 3.1. Otherwise, the similitude of the values of an intermediate feature is assessed by the aggregation of the basic similitudes of the leaves as explained in section 3.2.

3.1 Elementary Similarity

Let f be a leaf feature common to the feature terms F_1 and F_2. Let v_1 be the value that f takes in F_1 and v_2 the value that f takes in F_2. When v_1 and v_2 are numerical values with range $[a, b]$ the similarity of v_1 and v_2 is computed, as usual, by means of the following expression:

$$\textit{sim-values}(v_1, v_2) = 1 - \frac{\mid v_1 - v_2 \mid}{b - a}$$

When v_1 and v_2 are symbolic, their similarity is computed using the hierarchy of the sorts S given by the subsumption relation. The idea is that the similarity between two values depends on the level of the hierarchy where their *lub* is situated with respect to the whole hierarchy: the more general *lub(v_1, v_2)* the greater is the distance between v_1 and v_2. Formally, let $s_f \in S$ be the most general sort that can take the values of a feature f. We consider s_f as the root of a subsort hierarchy, therefore the *depth* of s_f is 1. Given a subsort s of s_f (i.e. $s \preceq s_f$) we define the *level* of s as follows: $\textit{level}(s) = M - \textit{depth}(s) + 1$, where M is the maximum depth of the hierarchy of root s_f.

Thus, the similarity $\textit{sim-values}(v_1, v_2)$ of two symbolic values v_1 and v_2 is 1 when $v_1 = v_2$, otherwise it is estimated using the following expression:

$$1 - \frac{1}{M}\textit{level}(\textit{lub}(v_1, v_2))$$

where $\frac{1}{M}\textit{level}(\textit{lub}(v_1, v_2))$ is a distance [3].

3.2 Structural Similarity

Given a feature term ψ, we call F_ψ the set of features of $\textit{root}(\psi)$. Let C_1 and C_2 be two feature terms and $CS(C_1, C_2) = F_{C_1} \cap F_{C_2}$ the set containing the features that are common to $\textit{root}(C_1)$ and $\textit{root}(C_2)$; let $RS(C_1, C_2) = F_{C_1} \cup F_{C_2}$ be the set formed by the relevant features of $\textit{root}(C_1)$ and $\textit{root}(C_2)$; $NS(C_1, C_2) = RS(C_1, C_2) \setminus CS(C_1, C_2)$, and let *nnodes(v)* be a function that given a feature term returns the number of nodes composing the feature term.

SHAUD associates to each feature $f_i \in CS(C_1, C_2)$ a tuple $T(f_i) = \langle s_i, w_i, r_i \rangle$ where s_i is a similitude, w_i is a measure of the common nodes to C_1 and C_2), and r_i is a measure of the total number of nodes present either in C_1 or C_2; below these two measures are explained in detail. Let v_1 be the value that f_i takes in C_1 and v_2 the value of f_i in C_2. The tuple $T(f_i)$ is computed as follows:

Case 1. When v_1 and v_2 have no common features, the tuple $\langle s_i, w_i, r_i \rangle$:

- $s_i = \textit{sim}(v_1, v_2)$ (see *sim* in section 3.1)
- $w_i = 1$
- $r_i = \textit{nnodes}(v_1) + \textit{nnodes}(v_2) - 1$

since w_i is the number of common nodes, $w_i = 1$ because the pair (v_1, v_2) corresponds to one common node and they have no common features (i.e. there is no more common structure). r_i is the number of nodes present in the terms v_1 and v_2 (up to the leaves defined by the depth we are considering).

Case 2. When neither v_1 nor v_2 are sets then let be $CS_{f_i} = CS(v_1, v_2)$, $RS_{f_i} = RS(v_1, v_2)$, $NS_{f_i} = RS(v_1, v_2) \setminus CS(v_1, v_2)$. We call $VS(v_1, v_2) = \{u_j | u_j = v_1.f_j \text{ or } u_j = v_2.f_j \text{ , } \forall f_i \in NS_{f_i}\}$, i.e. the set of values of the features in NS_{f_i}. Finally, let $T(CS_{f_i}) = \{T(f_j) = \langle s_j, w_j, r_j \rangle | f_j \in CS_{f_i}\}$ be the set of tuples associated to the features of CS_{f_i}.

The tuple associated with the feature f_i is computed as follows:

- $s_i = \frac{1}{r_i}\left[\textit{sim}(v_1, v_2) + \sum_{T(f_j) \in T(CS_{f_i})} s_j \cdot w_j\right]$
- $w_i = 1 + \sum_{T(f_j) \in T(CS_{f_i})} w_j$
- $r_i = 1 + \sum_{T(f_j) \in T(CS_{f_i})} r_j + \sum_{u_j \in VS(v_1, v_2)} \textit{nnodes}(u_j)$

w_i is the number of nodes common to v_1 and v_2, thus w_i aggregates the number of common nodes computed in lower levels by the tuples of the common features (plus 1, that counts the current common node corresponding to the pair (v_1, v_2) under consideration). r_i takes into account both the number of nodes that are common and the number of nodes that appear in only one of the feature terms. The value of r_i is computed aggregating the r values of the lower levels that are present in the tuples (for the common structure) and the number of nodes of the structure that is not shared using the *nnodes* function. The similitude s_i is normalized using r_i —i.e. the total number of nodes below the current node. In other words, the similitude of two feature terms is assessed taking into account all the structure present in the description of both feature terms.

Case 3. If either v_1 or v_2 is a set, SHAUD uses anti-unification because the idea is finding those pairs whose similarity is higher. For a pair $p = (x_j, y_k)$ of symbolic values the more specific their $lub(x_j, y_k)$ the higher is their similarity. Therefore we want to find the collection of pairs $\{p_1 \dots p_{min(n,m)}\}$ whose *lub*s are more specific: this is precisely the definition of anti-unification shown in §2. Therefore the anti-unification of the sets of values $v_1 = \{x_1 \dots x_n\}$ and $v_2 = \{y_1 \dots y_m\}$ provides the pairs that have the highest similarity.

Let AUS be the set of pairs whose anti-unification belong to the anti-unification of the sets v_1 and v_2, i.e. $AUS = \{(x, y) | x \in v_1 \wedge y \in v_2 \wedge AU(x, y) \in AU(v_1, v_2)\}$. For each pair $(x_l, y_l) \in AUS$, SHAUD associates a tuple $\langle s_l, w_l, r_l \rangle$ that is computed as in cases 1 and 2 above taking x_l as v_1 and y_l as v_2. Let $T(AUS)$ be the set of tuples of the pairs $(x_l, y_l) \in AUS$ and $US(v_1, v_2)$ the set containing the elements of v_1 or v_2 that have not been used in anti-unifying the sets v_1 or v_2 (i.e. the ones not present in any pair of AUS).

The tuple associated to the feature f_i is the following:

- $s_i = \frac{1}{r_i} \sum_{T(f_j) \in T(AUS)} s_j \cdot w_j$
- $w_i = \sum_{T(f_j) \in T(AUS)} w_j$
- $r_i = \sum_{T(f_j) \in T(AUS)} r_j + \sum_{z_k \in US(v_1, v_2)} nnodes(z_k)$

Let C_1 and C_2 be cases represented as feature terms. SHAUD assesses the similarity of both cases as the similarity s_i in *case 2*.

$$SHAUD(C_1, C_2) = \frac{1}{r_t} \sum_{T(f_i) \in T(CS(C_1, C_2))} s_i \cdot w_i$$

where

$$r_t = \sum_{T(f) \in T(CS(C_1, C_2))} r_i + \sum_{u_i \in VS(C_1, C_2)} nnodes(u_i)$$

In other words, the similarity of two cases is the aggregated similarity of the common part of both cases normalized by the total number of nodes used in the representation of the cases.

4 Experiments

We have used SHAUD for solving the classification task in several domains that we explain in next sections. In all them SHAUD is able to obtain acceptable classifications. The method followed for evaluating the predictivity of SHAUD is the leave-one-out method.

4.1 Marine Sponges Dataset

In this section we describe some experiments that use the similarity to identify the order of marine sponges. Marine sponges are relatively little studied and most of the existing species are not yet fully described. Main problems in the identification are due to the morphological plasticity of the species, to the incomplete knowledge of many of their biological and cytological features and to the frequent description of new taxa. Moreover, there is no agreement around the species delimitation since it is not clear how to characterize the taxa.

We used a case base containing 307 marine sponges belonging to three orders of the *demospongiae* class: *astrophorida, hadromerida* or *poecilosclerida*. The sponges are represented using feature terms. In each experiment we take out one sponge *sp* and then we compute the similarity of *sp* with each one of the remaining 306 sponges. Finally, *sp* is classified as belonging to the same order than the sponge estimated as more similar.

Figure 1 shows the results of these experiments, detailing the accuracy, and the number of correct and incorrect answers for each order. Thus, there are 95 sponges in the case-base belonging to the order *astrophorida*. For 93 of these sponges the similarity finds that the most similar sponge

				SHAUD	LAUD
order	N	Correct	Incorrect	%accuracy	%accuracy
astrophorida	95	93	2	97.89	92.63
hadromerida	117	113	4	96.58	92.30
poecilosclerida	95	83	12	87.37	90.53
TOTAL	307	289	18	94.14	91.86

Figure 1: Results of both LAUD and SHAUD to classify marine sponges.

is an *astrophorida*, i.e. they are correctly classified. Similarly, 113 of the 117 sponges of the order *hadromerida* and 83 of the 95 sponges of order *poecilosclerida* are correctly classified. Summarizing, from the 307 sponges of the case-base, 289 of them are correctly classified with respect the order where they belong. This represents an accuracy of 94, 14% that is better than the accuracy achieved by LAUD in the same dataset.

4.2 The Toxicology Dataset

The Toxicology Dataset has been provided by the US National Toxicology Program (NTP) (http://ntp-server.niehs.nih.gov). In this dataset there are descriptions of around 500 compounds that may be carcinogenic for two animal species: rats and mices. The carcinogenic activity of the compounds has proved to be different in both species and also among the sex of the same species. Therefore there are, in fact, four datasets. The compounds of the dataset can be classified in eigth solution classes according to the laboratory experiments: *positive*, *clear evidence*, *some evidence*, *equivocal*, *equivocal evidence*, *inadequate study*, *negative* and *negative evidence*. Nevetheless, most of the authors working on this dataset consider the classes *positive*, *clear evidence* and *some evidence* as the class "positive"; the classes *negative* and *negative evidence* as the class "negative"; and the compounds belonging to the other classes are removed.

For the Toxicology dataset there are two open problems: 1) the representation of the compounds, and 2) the classification of the compounds into *positive* or *negative* carcinogenic activity. With respect to the representation, there are several representations for the compounds (http://www.informatik.uni-freiburg.de/ ml/ptc/).

In our experiments, we have represented the compounds using *feature terms* (figure 2). This representation is based on the chemical name of the compound since we consider that the chemical name provides enough information for a good description of the molecular structure. For instance, the chemical name *2-amino-4-nitrophenol* describes a molecule having a *benzene* as main group and three radicals: an *alcohol* in the position 1, an *amine* in position 2 and a *nitro-derivate* in position 4. Figure 2 shows the translation of the *2-amino-4-nitrophenol* to feature terms. In this representation a compound is a feature term of sort *organic-compound* with two features or relations: main-group and radicals. The feature main-group contains the part of the molecule that is either the biggest or that located in a central position. The radicals feature is a set of feature terms of sort *radical* representing the groups of the molecule that are usually smaller than the main group. Each radical, in turn, has the features main-group and position. The position of a radical in the molecule is always described in relation with the main group. For example, the *2-amino-4-nitrophenol* has three radicals: an alcohol placed in the position 1 of the benzene (the main group of the molecule); an *amine* placed in the position 2; and finally, a *nitro-derivate* placed in the position 4 of the benzene.

There are several authors ([9]) that use data mining and machine learning techniques for solving the task of classifying the carcinogenic activity of a molecule. Most of authors use induction for obtain general rules classifying the compounds. The maximum accuracy obtained by the different methods is around 65% (Pfahringer obtained an accuracy of 70% using a votation system among the individual methods of the other authors). In fact, the correct classification score for human experts in the domain ranges from 28% to 78% ([12]).

Our goal is to investigate two issues in this dataset: 1) if a lazy learning approach is feasible for solving the classification task, and 2) if the current representation based on the chemical name of the compounds is sufficient. For this purpose we have performed several experiments using SHAUD

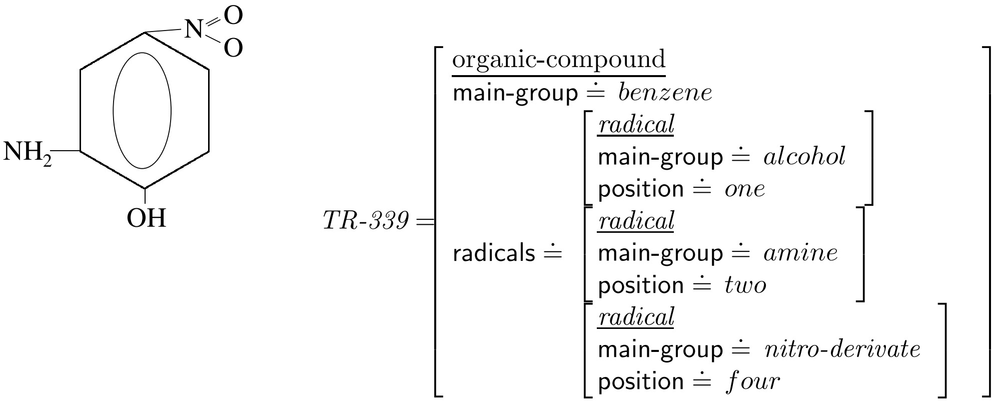

Figure 2: Representation of compound TR-339, *2-amino-4-nitrophenol*, with feature terms.

for classifying a compound as having *positive* or *negative* carcinogenic activity. We used only the first 234 compounds of the dataset and, as other authors, we removed the compounds with activity *equivocal*, *equivocal evidence* and *inadequate study*; the classe *positive* is formed by the compounds having activity *positive*, *clear evidence* and *some evidence*; and the class *negative* is formed by the compounds having activity *negative* and *negative evidence*. Table 3 shows the total number of positive and negative compounds for each dataset.

	positive	negative	total	Acc (%)
MR	81	125	206	62.13
FR	66	140	206	64.08
MM	63	139	202	64.85
FM	78	138	216	62.50

Figure 3: Distribution of the examples in each dataset. MR = male rats; FR = female rats; MM = male mices; FM = female mices.

In the preliminary experiments we used the k-NN algorithm with SHAUD as distance and the leave-one-out method for evaluating the results. We experimented with different values for k and the best results are obtained for $k = 5$. One of the most usual criteria for classifying a new compound is to use the majority class (i.e. the new compound is classified as belonging to the same class than the most of the k retrieved precedents) but our experiments using this criteria do not provide a good accuracy (around the 50% in rats).

The criteria we used for classify a new compound is the following. Let c be the compound to be classified and R_k the set of the k precedents more similar to c according to the SHAUD results. Each precedent $c_i \in R_k$ has associated the following data:

1. The structural similarity s_i among c and c_i, i.e. $s_i = SHAUD(c, c_i)$, and

2. For each dataset (i.e. male rats, female rats, male mices and female mices), the compound c_i has a *positive* or *negative* carcinogenic activity.

For one of the datasets, let A^+ be the set containing the precedents $c_i \in R_k$ with positive activity in that dataset, and A^- be the set containing the precedents $c_i \in R_k$ with negative activity. From the sets A^+ and A^- we define *sim-pos* and *sim-neg* as the respective averages of the similarities of the positive and negative precedents retrieved, i.e. $\textit{sim-pos} = \frac{1}{|A^+|}\sum_{c_i \in A^+} s_i$ and $\textit{sim-neg} = \frac{1}{|A^-|}\sum_{c_i \in A^-} s_i$.

The carcinogenic activity of a compound c is obtained according to the following criteria:

if sim-pos < sim-neg then c has negative carcinogenic activity

else c has positive carcinogenic activity

Using $k = 5$ and this classification criteria we obtained the accuracy shown in table 3.

In the future we plan to experiment with the complete dataset using k-NN with SHAUD. Our plan is to explore the feasibility of using the existing techniques [14] allowing the determination of k for each individual class or for each individual case. Depending on the results of these experiments we also will revise the representation of the compounds. In the current representation we have not taken into account neither spatial information nor other informations of the molecule such as charge, relative positions of the atoms, etc. The necessary could be easyly included in the current representation by adding features to the *organic-compound* sort.

5 Related Work

Most of work on relational representation focus on inductive learning techniques. ILP [11] is a wide field of research focused on solving different aspects on *concept learning* using objects represented as Horn clauses. The relational representation, that is much more expressive than the propositional representation, has not been exploited in depth in lazy learning techniques. An important problem to be solved in lazy learning techniques is how to estimate the similarity among relational cases. Again, there are many work on estimating the similitude among propositional cases, nevertheless only a few works propose similarities for relational cases.

Some of these works [6, 7, 5, 13, 4] propose the notion of "structural similarity" for evaluating the similarity of relational cases. Thus, [7] use techniques of subtree isomorphism and subgraph isomorphism. Other authors such as [5] and [4] compute two kinds of similarity: the similarity among the elements of a same class (*intra-class*) and the similarity among of the classes among them (*inter-class*). In SHAUD, as in [3], both kinds of simlarity are implicitly considered in the representation of the feature terms and in the sort hierarchy of the nodes.

The closest work to SHAUD is RIBL in the version extended to lists and terms [10]. The main difference between SHAUD and RIBL is the case representation since RIBL uses Prolog clauses whereas SHAUD uses feature terms. Also, RIBL assumes that the objects are always described by a fixed set of features. In SHAUD we assume that some of the features describing a case may be unknown (i.e. they do not appear in the representation of the case). In fact, SHAUD assessess the similitude of two cases taking into account that there is a part that is no common to both cases.

In [2] and [13] we proposed a similitude between cases represented as feature terms. Nevertheless in those works the similarity is not numerical but symbolic. We defined the *similarity term* as a feature term containing the features that has been more relevant in order to classify a case.

6 Conclusions

In this paper we introduced SHAUD, a new measure for assess the similarity of relational cases represented using feature terms. SHAUD estimates the similarity of relational cases taking into account both the estructure shared by the cases and the structure that they do not share. Thus, the similarity is estimated on the nodes of the description part that is common to the cases. Then this similarity value is normalized by the total number of nodes present in both cases.

In feature terms, there is an informational order relation ($\preceq$) between the sorts of the values. Using this relation we define the anti-unification operation in order to obtain the most specific generalization of two feature terms. The relation $\preceq$ and the anti-unification operation are used to compute the similitude of the symbolic values.

The preliminary results obtained from the application of SHAUD on two real datasets are satisfactory. Nevertheless, in the case of the Toxicology dataset it is necessary to analyze the representation more accurately.

Acknowledgements

This work has been supported by the Project IBROW (IST-1999-19005). The authors thank Josep LLuís Arcos and Lluis Bonamusa for their support in the elaboration of this paper.

References

[1] E. Armengol and E. Plaza. Bottom-up induction of feature terms. *Machine Learning Journal*, 41(1):259–294, 2000.

[2] E. Armengol and E. Plaza. Lazy induction of descriptions for relational case-based learning. In Luc De Raedt and Peter Flach, editors, *Machine Learning: ECML-2002*, number 2167 in Lecture Notes in Artificial Intelligence, pages 13–24. Springer-Verlag, 2001.

[3] E. Armengol and E. Plaza. Similarity assessment for relational cbr. In David Aha and Ian Watson, editors, *Case-based Reasoning Research and Development*, number 2080 in Lecture Notes in Artificial Intelligence, pages 44–58. Springer-Verlag, 2001.

[4] R. Bergmann and A. Stahl. Similarity measures for object-oriented case representations. In *Proc. European Workshop on Case-Based Reasoning, EWCBR-98*, Lecture Notes in Artificial Intelligence, pages 8–13. Springer Verlag, 1998.

[5] G. Bisson. Why and how to define a similarity measure for object based representation systems, 1995.

[6] K Börner. Structural similarity as a guidance in case-based design. In *Topics in Case-Based Reasoning: EWCBR'94*, pages 197–208, 1994.

[7] H Bunke and B T Messmer. Similarity measures for structured representations. In *Topics in Case-Based Reasoning: EWCBR'94*, pages 106–118, 1994.

[8] W. Emde and D. Wettschereck. Relational instance based learning. In Lorenza Saitta, editor, *Machine Learning - Proceedings 13th International Conference on Machine Learning*, pages 122 – 130. Morgan Kaufmann Publishers, 1996.

[9] C. Helma, R. King, S. Kramer, and A. Srinivasan. The predictive toxicology challenge 2000-2001. Technical report, ECML/PKDD-01, Freiburg, 2001.

[10] T. Horváth, S. Wrobel, and U. Bohnebeck. Relational instance-based learning with lists and terms. *Machine Learning Journal*, 43(1):53–80, 2001.

[11] Stephen Muggleton and Luc De Raedt. Inductive logic programming: Theory and methods. *Journal of Logic Programming*, 19–20:629–679, 1994.

[12] Bernhard Pfahringer. (the futility of) trying to predict carcinogenicity of chemical compounds. In C. Helma, R. King, S. Kramer, and A. Srinivasan, editors, *Proc. of the Workshop on Predictive Toxicology Challenge 2000-2001*, pages 1–11, 2001.

[13] Enric Plaza. Cases as terms: A feature term approach to the structured representation of cases. In M. Veloso and A. Aamodt, editors, *Case-Based Reasoning, ICCBR-95*, number 1010 in Lecture Notes in Artificial Intelligence, pages 265–276. Springer-Verlag, 1995.

[14] Dietrich Wettschereck and Thomas G. Dietterich. Locally adaptive nearest neighbor algorithms. In Jack D. Cowan, Gerald Tesauro, and Joshua Alspector, editors, *Advances in Neural Information Processing Systems*, volume 6, pages 184–191. Morgan Kaufmann Publishers, Inc., 1994.