Relational Case-based Reasoning for Carcinogenic Activity Prediction

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*)*

Abstract.

Lazy learning methods are based on retrieving a set of precedent 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. Usually, similarity measures require that 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*. We also present results of the application of Shaud for solving classification tasks. Specifically we used Shaud for assessing the carcinogenic activity of chemical compounds in the Toxicology dataset.

Keywords: Machine Learning, lazy learning methods, similarity assessment, feature terms, toxicology dataset

1. Introduction

Bioinformatics is a relatively new field that uses computer science techniques for analyzing biological data. There are many biological domains where automatic tools can be used in order to support the understanding of the data, such as the analysis of protein structure and function. One of the main problems in these domains is how to represent data. The chosen representation has to satisfy two conditions: 1) to capture the knowledge that the domain expert considers necessary for the task at hand and, 2) to be easily understandable by the domain expert. A second problem in dealing with real-world data is to elucidate which techniques may be useful for solving the task at hand.

In this paper we present our work on the Toxicology dataset. The task on this dataset is to predict carcinogenic activity of chemical compounds. During the Predictive Toxicology Challenge (PTC) held at 2001 in Freiburg (Germany) most authors proposed a relational representation of the compounds using inductive techniques for solving the task. We propose *feature terms* for representing chemical compounds and a lazy learning technique for solving the classification task. The feature term formalism has already been used in several applications (Armengol and Plaza, 2000) and has proved to be useful to represent knowledge and easy to understand by the domain expert.

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

In this paper we introduce Shaud, a new similarity measure capable of assessing the similarity between objects represented as feature terms. Given two cases represented as feature terms, Shaud distinguishes two parts in their structure: one formed by the features and nodes present in both cases, and another formed by those features and nodes that are only present in one of the cases. For the common part Shaud uses a hierarchy of sorts to compute the similarity of the attribute values. The resulting similarity values are aggregated and then normalized using the whole structure of both cases.

The paper is organized as follows: section 2 introduces the feature term formalism. Section 3 defines the Shaud similarity. Section 4 shows the results of Shaud in the toxicology domain. Specifically, section 4.1 explains in detail the representation of chemical compound using feature terms. Then, section 4.2 analyzes the results of Shaud on the Toxicology dataset. Finally, section 5 discusses some previous work on similarity measures for relational cases.

2. Representation of Relational Cases

We propose to represent the relational cases using the *feature terms* formalism introduced in (Armengol and Plaza, 2000). This formalism organizes concepts into a hierarchy of *sorts*, and represent descriptions and individuals as collections of features (functional relations) called feature terms. *Feature terms* (also called *feature structures* or *ψ-terms*) are a generalization of first order terms. The intuition behind a feature term is that it can be described as a labelled graph. The edges of the graph are labelled with feature symbols and the nodes are the sorts of the feature values.

Let us illustrate the feature terms above with an example. The feature term showed in Figure 1 represents the description of a marine sponge. The *root* of this feature term is *s364*, the sorts are written in *italics* and underlined (for instance, *sponge, external-features, growing,*

$$s364 = \left[\begin{array}{ll} \multicolumn{2}{l}{\underline{\mathit{sponge}}} \\ \mathsf{external\text{-}features} \doteq & \left[\begin{array}{ll} \multicolumn{2}{l}{\underline{\mathit{external\text{-}features}}} \\ \multicolumn{2}{l}{\mathsf{body\text{-}size} \doteq \mathit{small}} \\ \multicolumn{2}{l}{\mathsf{touch} \doteq \mathit{hispid}} \\ \mathsf{growing} \doteq & \left[\begin{array}{l} \underline{\mathit{growing}} \\ \mathsf{grow} \doteq \mathit{encrusting} \\ \mathsf{form} \doteq \mathit{digitate} \\ \mathsf{peduncle} \doteq \mathit{no} \end{array}\right] \\ \multicolumn{2}{l}{\mathsf{hollow} \doteq \mathit{no}} \\ \multicolumn{2}{l}{\mathsf{osc} \doteq \mathit{absent}} \end{array}\right] \\ \mathsf{spiculate\text{-}skeleton} \doteq & \left[\begin{array}{ll} \multicolumn{2}{l}{\underline{\mathit{spiculate\text{-}skeleton}}} \\ \multicolumn{2}{l}{\mathsf{chemical} \doteq \mathit{silica}} \\ \multicolumn{2}{l}{\mathsf{architecture} \doteq \mathit{reticulate}} \\ \mathsf{megascleres} \doteq & \left[\begin{array}{l} \underline{\mathit{megas\text{-}form}} \\ \mathsf{smooth\text{-}form} \doteq \mathit{spherotylostyle} \\ \mathsf{acanthose} \doteq \mathit{yes} \\ \mathsf{ornamentation} \doteq \begin{array}{l} \mathit{smooth} \\ \mathit{spines} \end{array} \end{array}\right] \\ \multicolumn{2}{l}{\mathsf{microscleres} \doteq \mathit{isochelatoxa}} \end{array}\right] \end{array}\right]$$

Figure 1. Representation of a sponge using feature terms.

etc.), some features are external-features, form, megascleres, etc. Notice that the feature ornamentation is set-valued.

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 minimal element ($\perp$) is called *any* and it represents the minimum information. When a feature has an unknown value it is represented as having the value *any*. All other sorts are more specific that *any*. Figure 2 shows the sort/subsort hierarchy for the values of the feature megascleres. The most general sort allowed for the values of the feature megascleres is *megas-form* and there are several subsorts (e.g. *triaena, style, caltrop,* etc). In turn, some of these subsorts (e.g. *triaena, style, tylote*) have subsorts.

A *path* $\rho(X, f_i)$ is defined as a sequence of features going from the variable X to the feature f_i. An example of path is $\rho($*s364*, acanthose$)$ that represents the path from the root to the leaf feature acanthose, i.e. the sequence of features spiculate-skeleton, megascleres, acanthose. We will note a path with a dot notation, e.g. *s364*.spiculate-skeleton.megascleres.acanthose.

The semantic interpretation of feature terms induces 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 the information in ψ is also contained in ψ'.

Feature terms form a partial ordering by means of the subsumption relationship. We define the *anti-unification* operation over the subsumption lattice as a lower upper bound with respect to the subsumption ($\sqsubseteq$) ordering. Intuitively, the anti-unification (AU) of two

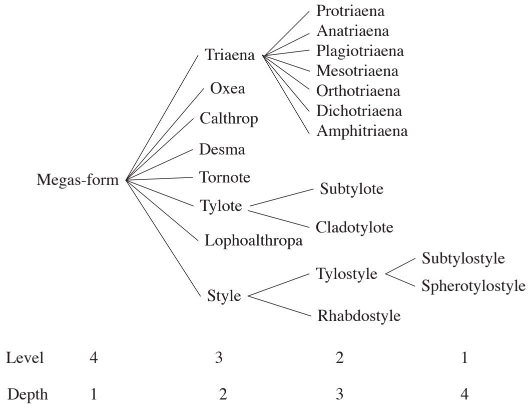


Figure 2. Part of the sort hierarchy of the feature *megascleres*.

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 ψ_1 and ψ_2 is a feature term D that contains the features that are common to both ψ_1 and ψ_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 ψ_1 and ψ_2, then the value of f in D is also v.

2. If a feature f has value of sort s_1 in ψ_1 and value of sort s_2 in ψ_2, then the value of f in D is the least upper bound (*lub*) of s_1 and s_2 in the $\preceq$ sort order.

Figure 3 shows the feature term *s-encrusting* representing sponges that have a spiculate skeleton and that grow in encrusting form. The sponge *s364* in Figure 1 is subsumed by this description (*s-encrusting* $\sqsubseteq$ *s364*) since all the information in *s-encrusting* is also contained in *s364* – although *s364* can have more (or more refined) information.

Because the features of feature terms can be set-valued, we have to define the anti-unification of two sets. Let f_k be a feature that takes the set V_1 as value in ψ_1 and the set V_2 as value in ψ_2. Intuitively, the AU of V_1 and V_2 has to produce as result a set $AU(V_1, V_2)$. The cardinality of the set $AU(V_1, V_2)$ is $MinCard = min(Card(V_1), Card(V_2))$ and each element in $AU(V_1, V_2)$ is the AU of a value of V_1 and a value of V_2 (obtaining the most specific combination).

The elements in $AU(V_1, V_2)$ are obtained as follows. First the set $C = \{(x_i, y_j) \mid x_i \in V_1 \text{ and } y_j \in V_2\}$ is obtained. Then the AU of each pair in C is computed. Finally, the set $AU(V_1, V_2)$ contains the

$$\text{s-encrusting} = \left[\begin{array}{l} \underline{\textit{sponge}} \\ \textsf{external-features} \doteq \left[\begin{array}{l} \underline{\textit{external-features}} \\ \textsf{growing} \doteq \left[\begin{array}{l} \underline{\textit{growing}} \\ \textsf{grow} \doteq \textit{encrusting} \end{array}\right] \end{array}\right] \\ \textsf{spiculate-skeleton} \doteq \textit{spiculate-skeleton} \end{array}\right]$$

Figure 3. Description of marine sponges with spiculate skeleton that grow encrusting.

$MinCard$ most specific compatible combinations of values. Given the feature terms $u_1 = AU(x, y)$ and $u_2 = AU(x', y')$ we say that u_1 and u_2 are *compatible* when $x \neq x'$ and $y \neq y'$. Otherwise u_1 and u_2 are *incompatible*. Intuitively, two feature terms in $AU(V_1, V_2)$ are compatible if they both have been obtained from the AU of different values. This means that the values of the sets to be anti-unified have been used only once. A more detailed explanation on feature terms, subsumption and anti-unification can be found in (Armengol and Plaza, 2000).

3. Similarity of Relational Cases

In this section we explain how to evaluate the similarity between cases represented as feature terms. For this purpose we introduce a new similarity measure called Shaud. The main idea of Shaud is to assess the similarity between two feature terms taking into account their structure. When comparing the structure of two feature terms ψ^1 and ψ^2 (see Figure 4), there are two parts that have to be taken into account: 1) the part of the structure that is common to both ψ^1 and ψ^2, called the *shared structure* (shown by colored nodes in Figure 4); and 2) the part of the structure that is present ψ^1 but not in ψ^2 and vice versa, called the *unshared structure* (shown by white nodes in Figure 4). Shaud assesses the similarity of two feature terms ψ^1 and ψ^2 by computing the similarity of the shared structure and then normalizing this similarity value taking into account both the shared and the unshared structure, as follows:

$$\mathsf{Shaud}(\psi^1, \psi^2) = \frac{sim_E(sort(\psi^1), sort(\psi^2)) + sim_S(\psi^1, \psi^2)}{1 + \Omega(\psi^1, \psi^2)} \qquad (1)$$

where $sort(\psi^1)$ ($sort(\psi^2)$) is the sort of the root of the feature term ψ^1 (ψ^2 respectively); $sim_E(sort(\psi^1), sort(\psi^2))$ is the elementary similarity between sorts (explained in section 3.1) applied to the roots of ψ^1 and ψ^2; $sim_S(\psi^1, \psi^2)$ is the *structural similarity* (explained below); and $\Omega(\psi^1, \psi^2)$ (also explained below) is the total number of nodes appearing in both the shared and unshared structure of ψ^1 and ψ^2.

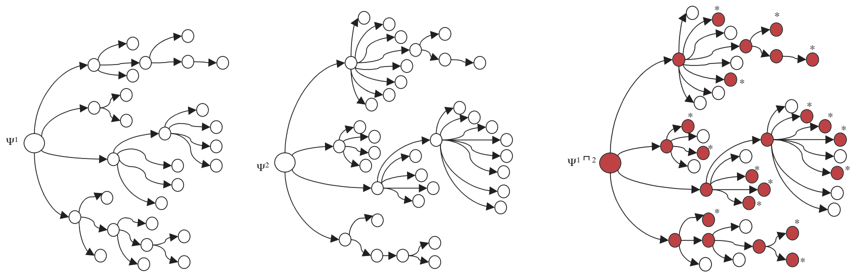


Figure 4. ψ^1 and ψ^2 are feature terms represented as graphs. $\psi^{1\sqcap 2}$ is a feature term containing both the *shared structure* (colored nodes) and the *unshared structure* (white nodes) of ψ^1 and ψ^2. Nodes market with "*" are *terminal nodes* of the shared structure.

Given a feature term ψ, we note $F(\psi)$ the set of features $\{f_1 \ldots f_n\}$ of the root of ψ. Let $\psi^{1\sqcap 2}$ be the feature term obtained from the anti-unification of ψ^1 and ψ^2, i.e. $\psi^{1\sqcap 2}$ is formed by the part of the structure that is shared by both ψ^1 and ψ^2. The structural similarity over the feature term $\psi^{1\sqcap 2}$ is defined as follows:

$$sim_S(\psi^1, \psi^2) = \sum_{f_i \in F(\psi^{1\sqcap 2})} \mathsf{Shaud}(\psi^1.f_i, \psi^2.f_i) \cdot W(\psi^{1\sqcap 2}.f_i) \qquad (2)$$

The structural similarity sim_S is assessed over the shared features $f_i \in F(\psi^{1\sqcap 2})$. The term $\mathsf{Shaud}(\psi^1.f_i, \psi^2.f_i)$ is the result of applying Equation 1 to the values of f_i in ψ^1 and in ψ^2. The function W is explained below.

There is a particular case when all the values of the feature f_i are terminal nodes (for instance, nodes such as body-size, grow, form, or substrate in Figure 1). A node is *terminal* when it has no features, i.e., a terminal node ψ satisfies $F(\psi) = \emptyset$, consequently the value of the structural similarity sim_S is 0 (Equation 2).

The similarity of each feature value (in Equation 2) is weighted using the function W, where $W(\psi^{1\sqcap 2}.f_i)$ calculates the number of nodes of the substructure with root $\psi^{1\sqcap 2}.f_i$. Note that all the nodes in $\psi^{1\sqcap 2}.f_i$ are common to ψ^1 and ψ^2. W is calculated as follows:

$$W(\psi^{1\sqcap 2}.f_i) = card(\psi^{1\sqcap 2}.f_i) + \sum_{\psi_k \in \psi^{1\sqcap 2}.f_i} \;\sum_{f_j \in F(\psi_k)} W(\psi_k.f_j) \qquad (3)$$

Since the feature f_i may be set-valued, $card(\psi^{1\sqcap 2}.f_i)$ counts the number of values that $\psi^{1\sqcap 2}$ has in f_i. Let ψ_k be one of the values of $\psi^{1\sqcap 2}.f_i$, for each feature $f_j \in F(\psi_k)$ the function W is recursively applied in order to compute the number of nodes contained in the substructure with root ψ_k.

When ψ_k is a terminal node then $F(\psi_k) = \emptyset$. When all the feature values of f_i are terminal then $W(\psi^{1\sqcap 2}.f_i) = card(\psi^{1\sqcap 2}.f_i)$.

Notice in Equation 1 that the similarity of two feature terms is normalized by $1 + \Omega(\psi^1, \psi^2)$ where $\Omega(\psi^1, \psi^2)$ is the total number of nodes appearing in any of the two feature terms. The normalization term adds 1 to $\Omega(\psi^1, \psi^2)$ to count the root node shared by ψ^1 and ψ^2. The value of $\Omega(\psi^1, \psi^2)$ is computed using the expression $\Omega(\psi^1, \psi^2) = \omega(\psi^1) + \omega(\psi^2) - \omega(\psi^{1\sqcap 2})$ where the function $\omega(\psi)$ counts the number of nodes of the feature term ψ using the following expression:

$$\omega(\psi) = \sum_{f_i \in F(\psi)} W(\psi.f_i)$$

Since the sum $\omega(\psi^1)$ plus $\omega(\psi^2)$ counts twice the common nodes (those in $\psi^{1\sqcap 2}$), therefore it is necessary to subtract $\omega(\psi^{1\sqcap 2})$ from this sum.

When a node ψ is terminal the value of ω is 0 because, following the definition of terminal node, ψ has no substructure. Therefore the function Ω takes as value 0 when both ψ^1 and ψ^2 are terminal. Notice that when the value of a feature f_i belonging to the shared structure $\psi^{1\sqcap 2}$ is a terminal node in both feature terms then $\mathsf{Shaud}(\psi^1.f_i, \psi^2.f_i) = sim_E(sort(\psi^1.f_i), sort(\psi^2.f_i))$.

3.1. Elementary similarity

The elementary similarity sim_E of two values is assessed using different expressions according to the type of these values. In particular, three cases are distinguished to compute sim_E: numerical values (sim_N), symbolic values (sim_V) and sets (*sim-sets*) .

Given two feature terms ψ^1 and ψ^2, let f be a feature common to both feature terms. Let v_1 be the value that f takes in ψ^1 and v_2 the value that f takes in ψ^2. When v_1 and v_2 are numerical values with range $[a, b]$ the similarity $sim_E(v_1, v_2)$ is computed as follows:

$$sim_N(v_1, v_2) = 1 - \frac{\mid v_1 - v_2 \mid}{b - a} \tag{4}$$

When v_1 and v_2 are symbolic, their similarity is computed using the hierarchy of the sorts S given by the informational order relation between sorts. The idea is that the similarity between two values depends on the level of the hierarchy where their least upper bound (*lub*) is situated with respect to the whole hierarchy: the more general *lub*(v_1, v_2) the smaller is the similarity 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. The similarity $sim_E(v_1, v_2)$ of two symbolic values v_1 and v_2 will be estimated using the following expression:

$$sim_V(v_1, v_2) = \begin{cases} 1 & \text{if } v_1 = v_2 \\ 1 - \frac{1}{M} level(lub(v_1, v_2)) & \text{otherwise} \end{cases} \quad (5)$$

We the subtree of root s_f from the sort hierarchy S, therefore we can use the depth of that tree to define the similarity of two symbolic values. Moreover, given a subsort s of s_f (i.e. $s \preceq s_f$) we define the *level* of s as follows: $level(s) = M - depth(s)$, where M is the maximum depth of the subtree of root s_f and we are assuming that the *depth* of s_f is 0. In addition, we proved in (Armengol and Plaza, 2001b) that the expression $\frac{1}{M} level(lub(v_1, v_2))$ is a distance.

Let us now consider how to estimate the similarity sim_E for set-valued features. Let f be a set-valued feature such that $\psi_1.f = V_1$ and $\psi_2.f = V_2$ for some sets $V_1 = \{x_1 \ldots x_n\}$ and $V_2 = \{y_1 \ldots y_m\}$. We note $P_{1,2} = \{(x_i, y_j) | x_i \in V_1 \wedge y_j \in V_2\}$ the set of pairs from the Cartesian product of V_1 and V_2. If $card(V_1) = n$ and $card(V_2) = m$ the idea is to find $min(n, m)$ pairs with highest similarity. In other words, we want to find a collection of compatible pairs (see section 2) with highest similarity. Let us call P_{max} such collection. Thus, the similarity $sim_E(V_1, V_2)$ of the sets V_1 and V_2 is computed as follows:

$$sim\text{-}sets(V_1, V_2) = \frac{1}{max(n, m)} \sum_{(x_i, y_j) \in P_{max}} sim_E(x_i, y_j) \quad (6)$$

where sim_E is the elementary similarity defined in equations 4 and 5 and the similarity is normalized by the highest cardinality of both sets.

When the values of sets V_1 and V_2 are numeric, it is necessary to compute *sim-sets* for all possible combinations of the elements of the sets. Then, the similarity of the sets V_1 and V_2 is the highest value of *sim-sets*.

When sets V_1 and V_2 have symbolic values the idea is also the same: finding those pairs with the highest similarity. As we have seen in section 2, for a pair (x_i, y_j) of symbolic values the more specific their $lub(x_i, y_j)$ the higher is their similarity. Therefore we want to find a collection of $min(n, m)$ pairs whose *lub*s are the most specific. But this is precisely the definition of anti-unification shown in section 2.

Let $AU(V_1, V_2) = \{u_1 \ldots u_{min(n,m)}\}$ be the set resulting from the anti-unification of V_1 and V_2. Each $u_k \in AU(V_1, V_2)$ is the result of the anti-unification of a pair elements (x_i, y_j) such that $x_i \in V_1$ and $y_j \in V_2$. Thus, the pairs contained in the set $AU(V_1, V_2)$ are the most specific ones and, consequently they provide the highest similarity. Therefore the similarity of two sets V_1 and V_2 with symbolic elements is assessed using the equation 6 where P_{max} is the set $P_{max} = \{(x_i, y_j) | x_i \in V_1 \wedge y_j \in V_2 \wedge AU(x_i, y_j) \in AU(V_1, V_2)\}$.

4. The Toxicology Dataset

The Toxicology dataset has been provided by the US National Toxicology Program (NTP) (http://ntp-server.niehs.nih.gov) and has descriptions of around 500 chemical compounds that may be carcinogenic for two animal species: rats and mice. The carcinogenic activity of the compounds has proved to be different for both species and also for both sex in the same species. Therefore, there are in fact four datasets.

For the Toxicology dataset there are two open problems: 1) the representation of the chemical compounds, and 2) which are the characteristics of chemical compounds conducive to their (manual or automatic) classification regarding a *positive* or *negative* carcinogenic activity.

4.1. Representation of the Chemical Compounds

We explain in this section how to represent chemical compounds in order to be used by automatic tools. From a representational point of view, chemical compounds are sets of atoms with bonds between them. Some authors working on chemical datasets agree that the best representation for chemical compounds is the relational one (Blockeel et al., 2001; Dehaspe et al., 1998; Pfahringer, 2001) although other representations have also been tried (Blinova et al., 2001; Deshpande and Karypis, 2002). Details about these representations can be found in http://www.informatik.uni-freiburg.de/˜ ml/ptc/. A usual relational representation of compounds is using Horn clauses with three basic predicates: *atom*, *bond* and *atomcoord*, together with background knowledge (such as physico-chemical information of the molecule, molecular weigth, distance between atoms or information about rings).

Some authors use approaches that are not centered on the representation of specific atoms but on molecular substructures. For instance, (Gonzalez et al., 2000; Deshpande and Karypis, 2002) represent chemical compounds as labeled graphs, allowing the use of techniques that operate with graphs: (Gonzalez et al., 2000) use SUBDUE; (Chittimoori et al., 1999) and (Deshpande and Karypis, 2002) use SMILES (Weininger, 1988) to detect the set of molecular substructures (subgraphs) more frequently occurring in the chemical compounds of the Toxicology dataset.

The Viniti's group (Blinova et al., 2001) proposed the FCSS language, allowing the description of chemical compounds as a set of

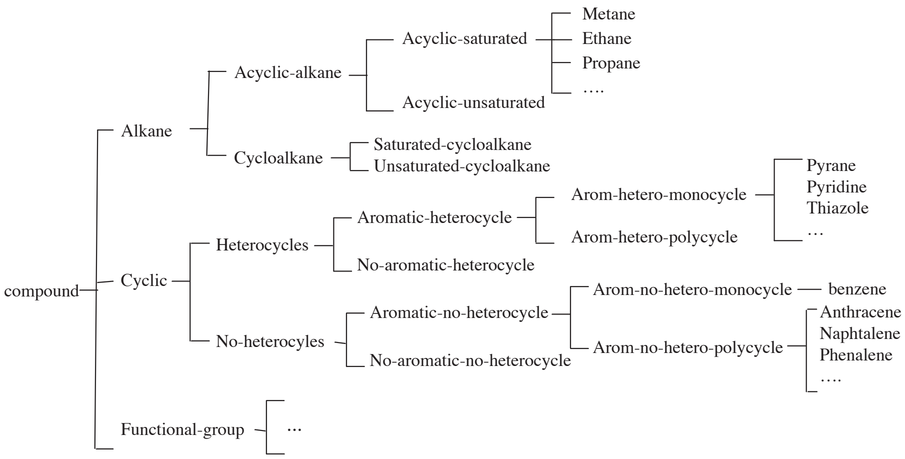


Figure 5. Partial view of the Toxicology ontology

substructures that are localization centers of π-electrons. Atoms or groups of atoms that are connected by π-electrons have a weak bond between them producing an activity point in the molecule. These points are called *description centers* and each one has and associated code. The elements of the FCSS language are chains of carbon pairs that begin and end with descriptor centers. For instance, the chemical compound TR-339 is represented as follows: Tr339 9 6,06 0700151 0700131 0700331 1100331 0200331 0764111 0263070 0262111.

We propose using a representation of chemical compounds based on the *chemical ontology* used by experts in chemistry. Also we take into account the experience of previous research, specially the works in (Gonzalez et al., 2000; Deshpande and Karypis, 2002; Blinova et al., 2001). We represent chemical compounds as a structure with substructures. The main difference between our approach and those taken in (Gonzalez et al., 2000; Deshpande and Karypis, 2002; Blinova et al., 2001) is that we use the chemical ontology that is implicit in the chemical nomenclature of the compounds. For instance, the *benzene* is an aromatic ring composed of six carbon atoms with some well-known properties. Our point is that it is not necessary to describe the individual atoms in benzene when *benzene* belongs to the domain ontology.

Figure 5 shows part of the chemical ontology we have used to represent the compounds in the Toxicology dataset. This ontology is based on the IUPAC chemical nomenclature which, in turn, is a systematic way of describing molecules. In fact, the name of a molecule provides all the information needed to graphically represent the structure of the molecule. According to the chemical nomenclature rules, the name of a compound is formed in the following manner: *radicals' names + main group*. The *main group* is often the part of the molecule that is either the largest or the part located in a central position. However, there is no general rule for establishing it. *Radicals* are groups that are usually smaller than the main group. A main group can contain several radicals and a radical can, in turn, have a new set of radicals. Both main group and radicals are the same kind of molecules, i.e. the benzene may be the main group in one compound and a radical in some other compounds.

In our representation (see Figure 6) a chemical compound is represented by a feature term of sort *compound* described by two features: main-group and p-radicals. The values of the feature main-group belong to some of the sorts shown in Figure 5. The value of the feature p-radicals is a set whose elements are of sort *position-radical*. The sort *position-radical* is described using two features: radicals and position. The value of the feature radicals is of sort *compound*, as the whole chemical compound, since it has the same kind of structure (a main group with radicals). The feature position indicates where the radical is bound to the main group.

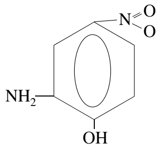
For example, the chemical compound TR-339, *2-amino-4-nitrophenol* (Figure 6), has a benzene[1] as main group and a set of three radicals: an *alcohol* in position one; an *amine* in position two; and a *nitro-derivate* in position four. Note that this information has been directly extracted from the chemical name of the compound following the nomenclature rules.

This kind of representation is very close to the representation that an expert has of a molecule from the chemical name. The main shortcoming of this representation is that the chemical nomenclature has ambiguities. In other words, a compound may have several names following the nomenclature rules. For instance, DDT can be formulated either as *1,1'-(2,2,2-trichloroethylidene)bis(4-chloro)-benzene* meaning that the main group is a benzene; or as the *1,1,1-trichloro-2,2-bis(p-chlorophenyl)-ethane* meaning that the main group is an ethane. This means that using feature terms we may have two alternative representations. In section 5 we discuss how this representation can be improved with the notion of multi-examples.

4.2. The Classification Task in the Toxicology Dataset

The NTP (National Toxicology Program) provides standardized chemical bioassays useful for identifying carcinogenic substances. Nevertheless, acquiring empirical evidence from these assays is very expensive

[1] The *phenol* is a benzene with a radical alcohol in position one



$$TR\text{-}339 = \left[\begin{array}{ll} \multicolumn{2}{l}{\underline{\textit{compound}}} \\ \multicolumn{2}{l}{\mathsf{main} \doteq \textit{benzene}} \\ \mathsf{p\text{-}radicals} \doteq & \begin{array}{l} \left[\begin{array}{l} \underline{\textit{position-radical}} \\ \mathsf{position} \doteq \textit{one} \\ \mathsf{radicals} \doteq \left[\begin{array}{l} \underline{\textit{compound}} \\ \mathsf{main} \doteq \textit{alcohol} \end{array}\right] \end{array}\right] \\ \left[\begin{array}{l} \underline{\textit{position-radical}} \\ \mathsf{position} \doteq \textit{two} \\ \mathsf{radicals} \doteq \left[\begin{array}{l} \underline{\textit{compound}} \\ \mathsf{main} \doteq \textit{amine} \end{array}\right] \end{array}\right] \\ \left[\begin{array}{l} \underline{\textit{position-radical}} \\ \mathsf{position} \doteq \textit{four} \\ \mathsf{radicals} \doteq \left[\begin{array}{l} \underline{\textit{compound}} \\ \mathsf{main} \doteq \textit{nitro-derivate} \end{array}\right] \end{array}\right] \end{array} \end{array}\right]$$

Figure 6. Representation of the compound TR-339 with feature terms.

and can take several years. Human experts in the toxicology domain achieve a classification score ranging from 28% to 78% (Pfahringer, 2001). For this reason, automatic techniques such as the detection of frequent substructures (Dehaspe et al., 1998; Deshpande and Karypis, 2002) and inductive learning methods can be useful support tools for predicting carcinogenic substances.

From the Machine Learning point of view, the problem of predicting carcinogenicity in compounds is a classification problem. In other words, the task to solve is to classify a chemical compound as either carcinogen (*positive* class) or non-carcinogen (*negative* class). In the Predictive Toxicology Challenge (PTC) (Helma and Kramer, 2003) several authors presented different approaches to solve this classification task. Most of them tried to induce rules that characterized these two classes. The maximum accuracy obtained by an individual method is around 65%. (Pfahringer, 2001) built a voting multi-classifier system among the individual methods of the other authors achieving an accuracy of 70%.

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 our approach for representing chemical compounds based on the chemists ontology is adequate. For this purpose we have performed several experiments using Shaud for classifying a compound as having *positive* or *negative* carcinogenic activity. We used only the first 234 compounds of the dataset and, as is often done (Helma and Kramer, 2003), we removed the compounds with both *equivocal* and *equivocal evidence* activity and those having *inadequate study*; the class *positive* is formed by the compounds having *positive* activity and both *clear evidence* and *some evidence* of activity; and the class *negative* is formed by the compounds having *negative* activity and *negative evidence* of activity. Thus, the case base used by Shaud contains 206 Male Rats (MR) cases (81 positives and 125 negatives); 206 Female Rats (FR) cases (66 positives and 140 negatives); 202 Male Mice (MM) cases (63 positives and 139 negatives); and 216 Female Mice (FM) cases (78 positives and 138 negatives).

In the experiments we used the k-NN algorithm with Shaud as similarity measure and the leave-one-out method for evaluating the results. One of the most usual criterion in k-NN is to use the majority class (i.e. the new compound is classified as belonging to the class that most of the k retrieved precedents belong to). However, our preliminary experiments using this criterion did not provide a good accuracy (see the accuracy for different values of k in Table I). Moreover, we observed that the accuracy in classifying negative compounds was higher than the accuracy in classifying positive compounds. For instance, in the male mice dataset, with $k = 4$, the accuracy in classifying positive compounds was 25%, whereas the accuracy in classifying negative compounds was 66.19%.

For the Toxicology dataset, several authors defined specific criteria based on domain knowledge with the goal of increasing the classification accuracy of positive compounds, e.g. (Blockeel et al., 2001) used domain knowledge in the form of very specific rules (for instance *a molecule has positive carcinogenic activity if it contains bromine* as an attempt to improve the classification). Nevertheless, they did not achieve the improvement they expected.

In (Pfahringer, 2001) authors considered a different kind of criterion. They built a voting multi-classifier system between all the approaches presented in the PTC (Helma and Kramer, 2003). As in (Blockeel et al., 2001) they found that applying a simple voting method, most of the compounds were classified as negative. For this reason they decided to apply several alternative criteria: 1) a chemical compound will be classified as positive when at least one of the individual classifiers has classified it as positive; 2) a chemical compound will be classified as positive when at least two of the individual classifiers has classified it as positive; and 3) to use probabilities and adjust them to the training set distribution.

We experimented with Shaud adapting these criteria but adapting them to the k-NN approach. Specifically, we used the following criterion (C1): *a chemical compound is positive when at least one of the k retrieved precedents has positive activity.* Table I shows the results of applying C1: the accuracy in predicting positive compounds has increased but the error in predicting negative compounds has also increased.

We can improve the results of C1 with the following criterion (C2): *a compound is positive when at least two of the retrieved compounds are* *positive*. The criterion C2 (see Table I) increased both the overall accuracy and the accuracy in classifying negative compounds. Nevertheless, the accuracy for negative compounds is still lower than the achieved applying the majority criterion. In the next section we introduce a new criterion called Class Similarity Average.

4.3. Class Similarity Average

From these experiments, we can set up two goals: 1) to improve the accuracy in predicting positive compounds and 2) to preserve (or increase) the accuracy for the classification of negative compounds obtained in the previous results. For this reason, we propose a new classification criterion for k-NN called *Class Similarity Average* (CSA). CSA is not domain-dependent and improves the accuracy on both (positive and negative) classes.

For each compound c to be classified, Shaud yields the similarity between c and each one of the k most similar cases. Then CSA will compute the average of the similarity of the cases in the same class; then the class with higher average similarity is selected as solution for c. More formally, let c be the compound to be classified and R_k the set of the k cases most similar to c according to the Shaud results. Each case $c_i \in R_k$ has the following data associated: 1) the structural similarity s_i between c and c_i, i.e. $s_i = \mathsf{Shaud}(c, c_i)$; and 2) for each dataset (i.e. MR, FR, MM and FM) the compound c_i is *positive* or *negative*.

For each dataset, let A^+ be the set containing the cases $c_i \in R_k$ with positive activity, and A^- be the set containing the cases $c_i \in R_k$ with negative activity. From the sets A^+ and A^- we define sim^+ and sim^- as the respective averages of the similarities of positive and negative cases retrieved, i.e.

$$sim^+ = \tfrac{1}{|A^+|}\textstyle\sum_{c_i \in A^+} s_i \text{ and } sim^- = \tfrac{1}{|A^-|}\textstyle\sum_{c_i \in A^-} s_i$$

The carcinogenic activity of a compound c is obtained according to the following criterion (CSA): *if sim-pos < sim-neg then c has negative carcinogenic activity else c has positive carcinogenic activity.*

The results of k-NN with Shaud with CSA are shown in Table I. Notice that for $k > 3$ the CSA accuracy has improved with respect to the accuracy of the other criteria (specially for $k = 5$). The right part of Table I shows, for each criterion and for $k = 5$, the *sensitivity* (i.e. percentage of positive compounds correctly classified) and the *specificity* (i.e. percentage of negative compounds correctly classified). Notice that for criteria C1 and C2 the sensitivity is high and the specificity is low (especially using C1). This is due to the high probability of both criteria to classify a compound as positive. The overall accuracy of MC

Table I. The left part of the table shows the accuracy results using the majority criterion (MC) and the criteria C1, C2, and CSA for different values of k. The right part of the table are accuracy results of sensitivity and specificity for $k = 5$. Values in bold are the best for each k and each dataset.

	Set	k = 3	k= 4	k = 5	*sensitivity*	*specificity*
MC	MR	50.00	41.26	48.06	35.80	56.00
	FR	51.46	43.20	49.03	21.21	62.14
	MM	58.41	53.46	60.40	31.75	73.38
	FM	54.63	51.86	59.72	46.15	67.39
C1	MR	45.14	42.23	40.78	95.06	5.60
	FR	46.12	44.17	40.78	86.36	19.28
	MM	51.98	44.06	42.57	85.71	23.02
	FM	47.22	41.67	40.28	87.18	13.77
C2	MR	**56.31**	50.48	42.72	62.96	29.6
	FR	**59.71**	52.91	46.12	53.03	42.86
	MM	63.37	61.39	59.40	84.13	61.87
	FM	**61.57**	41.67	55.55	64.10	50.72
CSA	MR	55.82	**57.28**	**62.13**	55.55	66.40
	FR	58.74	**59.22**	**64.08**	51.51	70.00
	MM	**65.35**	**64.35**	**64.85**	53.97	69.78
	FM	59.94	**54.17**	**62.50**	53.84	67.39

is obtained from the specificity, since the sensitivity is below 47%. The reason is that since the most of the retrieved precedents are negative, MC tends to classify the compounds as negative. Instead, CSA gives higher values on both sensitivity and specificity. Our interpretation is that CSA is more strict than MC and gives more opportunities to classify a compound as positive.

Commonly, Machine Learning methods are compared using the accuracy of their performance on a dataset. Nevertheless, most participants in the PTC used the ROC (Receiver Operating Characteristic) curves for comparing their methods. These graphs allow a comparison of classifiers that is robust with respect to imprecise class distributions and misclassification costs. ROC curves are useful to show the tradeoff between true positives (TP) and false positives (FP) produced by a classifier. TP is the ratio between positive cases correctly classified and the total number of positive cases. Similarly, FP is the ratio between negative cases incorrectly classified and the total number of negative cases.

In order to compare Shaud we evaluated its performance using ten-fold cross-validation. Since CSA (with $k = 5$) was the best method during the leave-one-out evaluation, we computed TP and FP for this configuration. The average of six 10-fold cross-validation runs gives the following results:

	MR	FR	MM	FM
accuracy (%)	55.27	58.24	59.13	58.06
TP	0.5621	0.5187	0.4955	0.5130
FP	0.4572	0.3716	0.3488	0.3766

Figure 7 shows ROC curves for each one of the datasets (male rats, female rats, male mice and female mice). Each point in the curve represents the (FP, TP) ratio of a method presented in the PTC. The line $x = y$ represents the strategy of randomly guessing the class and the point (0, 1) represents perfect classification. Therefore, a point in ROC space is better than another if TP is higher and FP is lower. The points shown in Figure 7 were assembled during the PTC from different techniques presented there. However, since these techniques did not effectively use the same set of data for training, ROC curves are more a rough estimate than an exact comparison. We show the point for the Shaud with CSA and $k = 5$. Shaud's (FP, TP) points are above the curve in two datasets (specially for MR) but it's still a competitive method for the other two.

5. Related Work

Most studies on the Toxicology dataset have used inductive techniques to build general rules for each class. Blockeel et al. (2001) argue that lazy techniques could be more useful than eager ones. The idea is that inductive techniques try to extract general rules describing the cases in each class. Nevertheless, due to the wide variety of chemical compounds finding general rules to appropriately describe the classes is very difficult. Instead, because lazy techniques are focused on each new problem to be solved, it could be easiest to classify new compounds. However, Blockeel has not experimented with lazy techniques.

Concerning the issue of representation, there are two ways of representing chemical compounds: 1) description of the characteristics of each atom of each compound, and 2) description of the overall structure of the compound. In the first group we find most of the works using Horn clauses for describing the compounds (Blockeel et al., 2001; Woo,

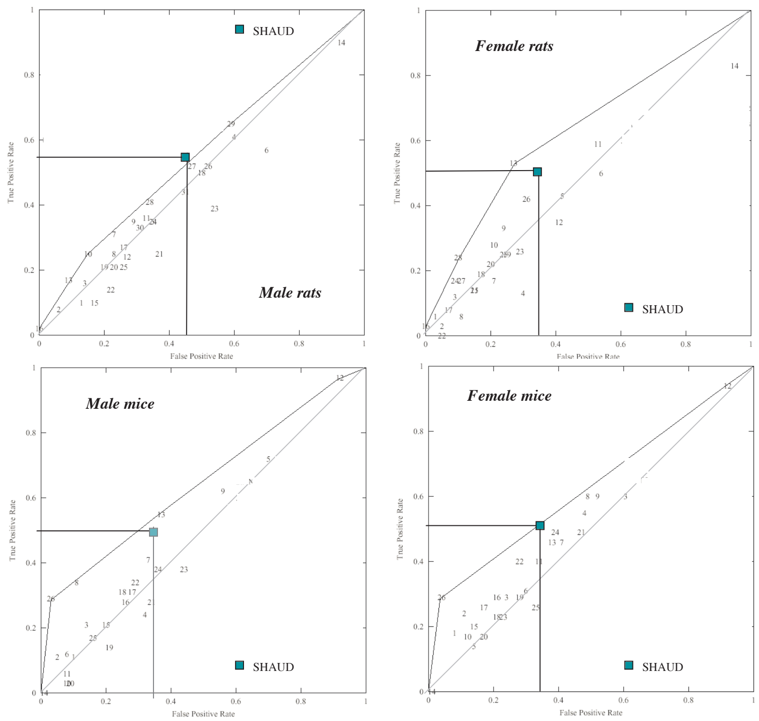


Figure 7. ROC curves corresponding to the approaches submitted to the PCT to which we added the result of Shaud for $k = 5$ and CSA.

2001; Ohwada et al., 2001). Often, compounds are represented as sets of predicates relating the atoms of the molecule and they also include information about the compounds (such as molecular weight, charge, etc). The second group of representations (Dehaspe et al., 1998; Gonzalez et al., 2000; Chittimoori et al., 1999) considers a compound as a structure. This approach is based on representing the compounds as graphs and then using graph techniques for detecting frequent substructures of the molecules included in each class. The representation of (Blinova et al., 2001) also belongs to the second group but has a distinct approach: compounds are segmented according to their activity points. Each segment has a code, and thus chemical compounds are represented as a string of codes. Our approach can be included in the second group since we use the chemical ontology for abstract molecules' representation.

The notion of *multi-examples* is useful when domain objects can be viewed in several ways. Specifically, (Dietterich et al., 1997) used *multi-examples* for determining the activity of a molecule, taking into account that a molecule has different isomers with different activity. As explained in section 4.1, chemical nomenclature allows synonym names for one compound. We intend to use the notion of *multi-examples* to manage synonymous descriptions of chemical compounds.

Most of the work on relational representation focuses on inductive learning techniques. However, relational representation can also be useful for lazy learning techniques. There is a group of techniques based on the notion of "structural similarity" that uses subtree-isomorphisms or subgraph isomorphisms to assess similarity. The approaches of (Bisson, 1995) and (Bergmann and Stahl, 1998) distinguish between inter-class and intra-class similarity. This is because they separate similarity among instances in the same class from similarity among classes. Inter-class similarity (Bergmann and Stahl, 1998) requires a hand-made assignment of "similarity degrees" to the class hierarchy, while Shaud defines a similarity over the hierarchy of sorts. Another difference is that both Shaud and (Bisson, 1995) support set-valued attributes, whereas (Bergmann and Stahl, 1998) does not.

RIBL (Emde and Wettschereck, 1996) is a relational lazy learning algorithm that has been recently extended to support representations of lists and terms (Horváth et al., 2001). In this new version of RIBL the similarity between cases is assessed using the standard similarity measures for numerical and discrete attributes, together with a similarity measure based on the concept of *edit distance* for attributes with lists and terms.

In (Plaza, 1995) and (Armengol and Plaza, 2001a) we described an approach where cases are represented as feature terms and where similarity is assessed through the notion of *similarity term*. The *similarity term* of two cases is defined as a feature term containing the features common to both cases that have been considered as the most relevant for classifying a new case. In both approaches the similarity is symbolic and not numerical.

In (Armengol and Plaza, 2001b) we defined Laud, a measure that assesses the similarity of two cases represented as feature terms. Laud proved to be useful in classification tasks. Nevertheless, Laud can be improved because it does not take into account the complete structure of the feature terms but only the leaves of this structure. Shaud, seems to be the natural improvement of Laud since it is able to take into account the complete structure provided by the feature terms. Indeed, Shaud assesses the similarity of two cases based on the complete structure of the cases (i.e. the leaves and the intermediate nodes of feature terms).

6. Conclusions

In this paper we introduce a lazy learning technique for relational cases. Our approach incorporates domain knowledge in the task of learning to predict the carcinogenic effect of compounds. First, we use an ontology based on the the expert chemist to describe the compounds. Moreover, the sorts in this ontology are organized into a hierarchy that later is used in Shaud to estimate the similarity among cases. The similarity is assessed taking into account both the structure 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. Next, the elementary similarity values are aggregated and then normalized by the total number of nodes present in both cases.

Commonly, the classification task in Toxicology is solved using inductive techniques. We used the k-NN algorithm, taking Shaud as similarity measure, for solving the classification task in Toxicology. The difficulty in this domain is to predict when a compound is carcinogenic (positive activity), as has been ascertained in the Machine Learning methods that we have described in this article. We experimented with different values of k and also with different criteria for elaborating the solution from the k most similar cases. In the preliminary experiments we used the criteria of the majority class rule (commonly used in k-NN) as well as some domain-dependent criteria; the results show the difficulty of predicting the positive activity of chemical compounds.

Finally, we proposed a domain-independent criterion, called *Class Similarity Average*, for k-NN classification. We have shown that Shaud with CSA and $k = 5$ achieves an accuracy comparable to the other methods. Moreover the ROC curves showed that the performance of our approach is comparable to the best techniques applied to this dataset.

Acknowledgements
This work has been supported by the projects IBROW (IST-1999-19005) and SAMAP (TIC2002-04146-C05-01). The authors thank Josep Lluís Arcos and Lluís Bonamusa for their support in the elaboration of this paper.

References

Armengol, E. and E. Plaza: 2000, ‘Bottom-up Induction of Feature Terms’. *Machine Learning* **41**(1), 259–294.

Armengol, E. and E. Plaza: 2001a, ‘Individual Prognosis of diabetes Long-term Risks: A CBR Approach’. *Methods of Information in Medicine* pp. 46–51.

Armengol, E. and E. Plaza: 2001b, 'Similarity Assessment for Relational CBR'. In: D. W. Aha and I. Watson (eds.): *CBR Research and Development. Proceedings of the ICCBR 2001. Vancouver, BC, Canada.* pp. 44–58, Springer-Verlag.

Bergmann, R. and A. Stahl: 1998, 'Similarity Measures for Object-Oriented Case Representations'. In: *Proc. European Workshop on Case-Based Reasoning, EWCBR-98.* pp. 8–13, Springer Verlag.

Bisson, G.: 1995, 'Why and How to Define a Similarity Measure for Object Based Representation Systems'. In: *In Towards Very Large Knowledge Bases, IOS Press, Amsterdam.* pp. 236–246.

Blinova, V., D. A. Bobryinin, S. O. Kuznetsov, and E. S. Pankratova: 2001, 'Toxicology analysis by means of simple JSM method'. In: *Proceedings of the Predictive Toxicology Challenge Workshop, Freiburg, Germany, 2001.*

Blockeel, H., K. Driessens, N. Jacobs, R. Kosala, S. Raeymaekers, J. Ramon, J. Struyf, W. V. Laer, and S. Verbaeten: 2001, 'First order models for the Predictive Toxicology Challenge 2001'. In: *Proceedings of the Predictive Toxicology Challenge Workshop, Freiburg, Germany, 2001.*

Chittimoori, R., L. Holder, and D. Cook: 1999, 'Applying the Subdue Substructure Discovery System to the Chemical Toxicity Domain'. In: *Proceedings of the Twelfth International Florida AI Research Society Conference, 1999.* pp. 90–94.

Dehaspe, L., H. Toivonen, and R. D. King: 1998, 'Finding frequent substructures in chemical compounds'. In: R. Agrawal, P. Stolorz, and G. Piatetsky-Shapiro (eds.): *4th International Conference on Knowledge Discovery and Data Mining.* pp. 30–36, AAAI Press.

Deshpande, M. and G. Karypis: 2002, 'Automated approaches for classifying structures'. In: *Proc. of the 2nd Workshop on Data Mining in Bioinformatics.*

Dietterich, T., R. Lathrop, and T. Lozano-Perez: 1997, 'Solving the Multiple Instance Problem with Axis-Parallel Rectangles'. *AI Journal* **89**(1-2), 31–71.

Emde, W. and D. Wettschereck: 1996, 'Relational Instance Based Learning'. In: L. Saitta (ed.): *Machine Learning - Proceedings 13th International Conference on Machine Learning.* pp. 122–130, Morgan Kaufmann Publishers.

Gonzalez, J. A., L. B. Holder, and D. J. Cook: 2000, 'Graph Based Concept Learning'. In: *AAAI/IAAI.* p. 1072.

Helma, C. and S. Kramer: 2003, 'A survey of the Predictive Toxicology Challenge 2000-2001.'. *Bioinformatics* p. in press.

Horváth, T., S. Wrobel, and U. Bohnebeck: 2001, 'Relational Instance-based Learning with Lists and Terms'. *Machine Learning Journal* **43**(1), 53–80.

Ohwada, H., M. Koyama, and Y. Hoken: 2001, 'ILP-based rule induction for predicting carcinogenicity'. In: *Proceedings of the Predictive Toxicology Challenge Workshop, Freiburg, Germany, 2001.*

Pfahringer, B.: 2001, '(The Futility of) Trying to Predict Carcinogenicity of Chemical Compounds'. In: *Proceedings of the Predictive Toxicology Challenge Workshop, Freiburg, Germany, 2001.*

Plaza, E.: 1995, 'Cases as terms: A feature term approach to the structured representation of cases'. In: M. Veloso and A. Aamodt (eds.): *Case-Based Reasoning, ICCBR-95*, Lecture Notes in Artificial Intelligence. Springer-Verlag, pp. 265–276.

Weininger, D.: 1988, 'SMILES a Chemical Language and Information System'. *J. Chem. Inf. Comput. Sci.* **28**(1), 31–36.

Woo, Y.: 2001, 'Predictive Toxicology Challenge 2000-2001. A Toxicologist's view and Evaluation.'. In: *Proceedings of the Predictive Toxicology Challenge Workshop, Freiburg, Germany, 2001.*