1

Lazy Learning for Predictive Toxicology based on a Chemical Ontology

Eva Armengol and Enric Plaza

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

Summary. Predictive toxicology Predictive toxicology is the task of building models capable of determining, with a certain degree of accuracy, the toxicity of chemical compounds. We discuss several machine learning methods that have been applied to build predictive toxicology models. In particular, we present two lazy learning lazy learning techniques applied to the task of predictive toxicology. While most ML techniques use structure relationship models to represent chemical compounds, we introduce a new approach based on the chemical nomenclature to represent chemical compounds. In our experiments we show that both models, SAR and ontology-based, have comparable results for the predictive toxicology task.

1.1 Introduction

Thousands of new chemicals are introduced every year in the market for their use in products such as drugs, foods, pesticides, cosmetics, etc. Although these new chemicals are widely analyzed before commercialization, the effects of many of them on human health are not totally known. In 1973 the European Commission started a long term program consisting on the design and development of toxicology and ecotoxicology chemical databases. The main idea of this program was to establish lists of chemicals and methods for testing their risks to the people and the environment. Similarly, in 1978 the American Department of Health and Human Services established the National Toxicology Program (NTP) with the aim of coordinating toxicological testing programs and developing standard methods to detect potentially carcinogenic compounds (see more information in www.ntp-server.niehs.nih.gov).

When a chemical compound is suspected to be toxic, is included in the NTP list in order to perform standardized experiments to determine its toxicity degree. Basically, there are two kinds of experiments: *in vitro* and *in vivo*. *In vitro* experiments are carried out on *salmonella* and the outcome are quantitative results of several physical-chemical parameters. *In vivo* experiments are performed on rodents (rats and mice), and there are, in turn, two

kind of experiments: short-term (90 days) and long-term (2 years). Usually, short-term experiments are performed as a means to obtain a first clue of the toxicity of a compound. It should be emphasized that to determine the toxicity of chemical compounds on rodents is an expensive process that, in addition, offers results that are not conclusive concerning the toxicity in humans.

The use of computational methods applied to the toxicology field could contribute to reduce the cost of experimental procedures. In particular, artificial intelligence techniques such as knowledge discovery and machine learning (ML) can be used for building models of compound toxicity (see [18] for an interesting survey). These models reflect rules about the *structure-activity relationships* (SAR) of chemical compounds. Such rules are used to predict the toxicity of a chemical compound on the basis of the compound's chemical structure and other known physical-chemical properties. The construction of this model is called *predictive toxicology*.

The Predictive Toxicology Challenge (PTC) was a competition held in 1990 with the goal of determining the toxicity of 44 chemical compounds based on both experiments in the lab and the predictive toxicology methods. The results of this challenge [4, 10] showed that the best methods are those taking into account the results of the short-term tests. A second challenge was announced in 1994. This challenge was mainly focused on using ML techniques and results can be found in [30]. The last challenge held in 2001 [19] was also focused on ML techniques and most of them used SAR descriptors. In this challenge most of authors proposed a relational representation of the compounds and used inductive techniques for solving the task.

Currently there still are two open questions in predictive toxicology: 1) the representation of the chemical compounds, and 2) which are the characteristics of a chemical compound that allows its (manual or automatic) classification as a potentially toxic. In this chapter we describe several approaches to both questions: we propose a representation of the chemical compounds based on the IUPAC (*International Union of Pure and Applied Chemistry*) chemical nomenclature and a *lazy learning technique* for solving the classification task.

1.2 Representation of chemical compounds

One of the most important issues for developing computational models is the representation of *domain objects*, in our case chemical compounds. In the toxicology domain, there are several key features of the molecule to be taken into account for predicting toxicity. First, there are some concerning to the basic elements of the molecule, such as number of atoms, bonds between atoms, positions, electrical charges, etc. Second, there are physical-chemical properties of the molecule such as lipophilic properties, density, boiling point, melting point, etc. Finally, there often exists prior information about the toxicity of a molecule, which was obtained from studies on other species using different experimental methods.

```
atom(tr339,1,o,-1).  atom(tr339,2,n,1).  atom(tr339,3,o,0).  atom(tr339,4,c,0).
atom(tr339,5,c,0).   atom(tr339,6,c,0).  atom(tr339,7,c,0).  atom(tr339,8,o,0).
atom(tr339,9,c,0).   atom(tr339,10,n,0). atom(tr339,11,c,0). atom(tr339,12,h,0).
atom(tr339,13,h,0).  atom(tr339,14,h,0). atom(tr339,15,h,0). atom(tr339,16,h,0).
atom(tr339,17,h,0).  bond(tr339,1,2,1).  bond(tr339,2,3,2).  bond(tr339,2,4,1).
bond(tr339,4,5,1).   bond(tr339,5,6,2).  bond(tr339,5,12,1). bond(tr339,6,7,1).
bond(tr339,6,13,1).  bond(tr339,7,8,1).  bond(tr339,7,9,2).  bond(tr339,8,14,1).
bond(tr339,9,10,1).  bond(tr339,9,11,1). bond(tr339,10,15,1). bond(tr339,10,16,1).
bond(tr339,11,17,1).
atomcoord(tr339,1,3.0918,-0.8584,0.0066).   atomcoord(tr339,2,2.3373,0.0978,0.006).
atomcoord(tr339,3,2.7882,1.2292,0.0072).    atomcoord(tr339,4,0.8727,-0.1152,-0.0023).
atomcoord(tr339,5,0.3628,-1.4003,-0.0094).  atomcoord(tr339,6,-1.0047,-1.6055,-0.0172).
atomcoord(tr339,7,-1.868,-0.5224,-0.0174).  atomcoord(tr339,8,-3.2132,-0.7228,-0.0246).
atomcoord(tr339,9,-1.355,0.7729,-0.0098).   atomcoord(tr339,10,-2.2226,1.8712,-0.0096).
atomcoord(tr339,11,0.018,0.971,0.0028).     atomcoord(tr339,12,1.0343,-2.2462,-0.0092).
atomcoord(tr339,13,-1.3998,-2.6107,-0.0234). atomcoord(tr339,14,-3.4941,-0.7673,0.8996).
atomcoord(tr339,15,-3.1824,1.7311,-0.0147). atomcoord(tr339,16,-1.864,2.7725,-0.0043).
atomcoord(tr339,17,0.419,1.9738,0.0087).
```

Fig. 1.1. Representation of the chemical compound TR-339 using Horn clauses.

In the literature, there are two approaches to represent chemical compounds: 1) those representing a compound as a vector of molecular properties (*propositional representation*), and 2) those explicitly representing the molecular structure of a compound (*relational representation*). In the follow sections we briefly explain these representations (details can be found at www.informatik.uni-freiburg.de/~ml/ptc/) and then we will introduce our own representation based on the chemical ontology used by the experts.

SAR and Qualitative SAR (QSAR) use equation sets that allow the prediction of some properties of the molecules before the experimentation in the laboratory. In analytical chemistry, these equations are widely used to predict spectroscopic, chromatographic and some other properties of chemical compounds. There is a number of commercial tools allowing the generation of these descriptors: CODESSA [22], TSAR (Oxford molecular products, www.accelrys.com/chem/), DRAGON (www.disat.inimib.it/chm/Dragon.htm), etc. These tools represent a chemical compound as a set of attribute value pairs. This kind of representation is called *propositional* in ML. For instance, the description of a car using propositional description is the following: {(size, medium), (builder, BMW), (model, 250), (color, white)}.

In addition to the knowledge about a particular compound, it is also useful to handle general chemical knowledge, what is called *background knowledge* in ML. Automatic methods that use background knowledge often consider compounds as a structure composed of substructures. This kind of representation is called *relational* because an object is represented by the relationships between their component elements. For instance, a car can be described composed of subparts like the chassis and the engine. In turn, each one of these parts can be described by their own subcomponents.

A form of relational representation is *logic programming*, that represents the relations among elements by a set of predicates. Thus, a set of predicates can be used to establish the relationship among the atoms of a molecule and

also handle basic information about the compounds (such as molecular weight, electrical charge, etc). Figure 1.1 shows the representation of the chemical compound TR-339 (the *2-amino-4-nitrophenol*) of the NTP data set. In this representation, there are three predicates:

- *atom(C, A, E, V)* gives information about an atom. C is the chemical compound where the atom belongs; A is the number of the atom in the chemical compound; E is the chemical element; and V is the electrical charge of the atom. For instance, atom(tr339, 1, O, -1) is the atom 1 of the compound tr339, It is an oxygen, and its charge is -1.
- *bond(C, A1, A2, B)* indicates the kind of bond between two atoms. C is the chemical compound where the bond belongs; $A1$ and $A2$ are the atoms of the compound connected by the bound; B is the kind of bond: simple, double or triple. For instance, bond(tr339, 9, 10, 1) is a simple bound of the chemical compound tr339 that connects the atoms 9 and 10.
- *atomcoord(C, A, X, Y, Z).* It gives the spatial coordinates of the compound atoms. C is the chemical compound, A is the atom and X, Y and Z are the spatial coordinates. For instance, atomcoord(tr339, 1, 3.0918, -0.8584, 0.0066) indicates that the atom 1 of the compound tr339 has as coordinates (3.0918, -0.8584, 0.0066).

Figure 1.1 represents the compound TR-339 with 17 atoms (3 oxygen, 2 nitrogen, 6 carbon and 6 hydrogen); there are double bonds between atoms 2 and 3; 5 and 6; 7 and 9; and 4 and 11 (see Fig. 1.1); and the rest of bonds are simple.

The representation introduced in [8] has a different approach: the compounds are organized according to their active centers (chemically identified with weak bonds). Active centers are atoms or groups of atoms responsible of the reactivity of the compound with biological receptors (for instance, toxicity). With this approach, each resulting part of the compound receives a code, therefore the chemical substances are represented as a string of codes. The Viniti's group [8] proposed the *fragmentary code of substructure superposition* (FCSS) language, allowing the description of chemical compounds as a set of substructures containing the active centers. The elements of the FCSS language are chains of carbon pairs that begin and end with the descriptors of active centers. For instance, the chemical compound TR-339 described in FCSS is the following code: *9 6,06 0700151 0700131 0700331 1100331 0200331 0764111 0263070 0262111.*

1.2.1 Representation using the Chemical Ontology

The representation of chemical compounds we propose is the *chemical ontology* based on the terminology used by chemists, i.e the IUPAC nomenclature (www.chem.qmul.ac.uk/iupac/). Also we take into account the experience of previous research (specially the works in [17, 15, 8]) since we represent a

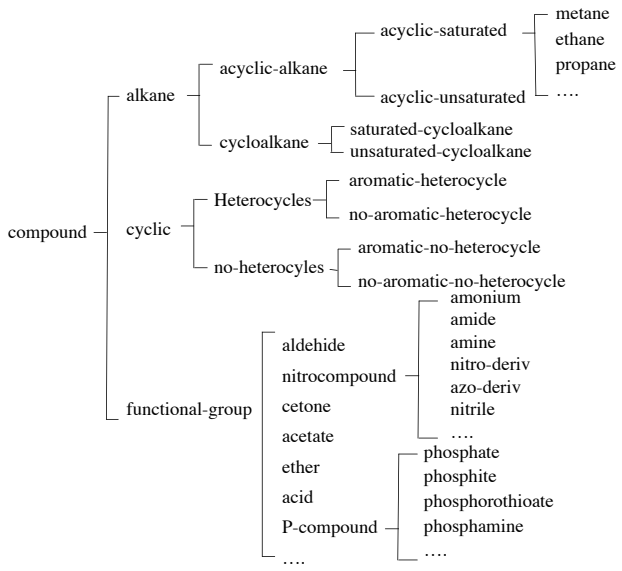


Fig. 1.2. Partial view of the Toxicology ontology

chemical compound as a structure with substructures. Our point is that there is no need to describe in detail the properties of individual atom properties in a molecule when the domain ontology has a characterization for the type of that molecule. For instance, the *benzene* is an aromatic ring composed by six carbon atoms with some well-known properties. While using SAR models would represent a given compound as having six carbon atoms related together (forming an aromatic ring), in our approach we simply state that the compound is a benzene (abstracting away the details and properties of individual atoms).

Figure 1.2 shows a partial view of the chemical ontology we used for representing the compounds in the Toxicology data set. This ontology is based on the chemical nomenclature which, in turn, is a systematic way of describing molecules. In fact, the name of a molecule, when the standard nomenclature is used, provides to the chemist with all the information needed to graphically represent its structure. According to the chemical nomenclature rules, the name of a compound is usually formed in the following way: *radicals' names + main group*. Commonly, the *main group* is the part of the molecule that is either the largest or that located in a central position; however, there is no general rule to establish them. *Radicals* are groups of atoms usually smaller than the main group. A main group can have several radicals and a radical can, in turn, have a new set of radicals. Any group of atoms could be main group or radical depending on their position or relevance on the molecule, i.e.

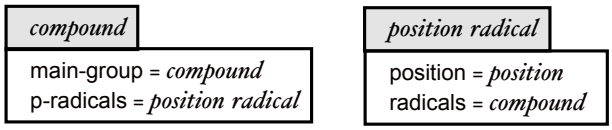


Fig. 1.3. Features corresponding to sorts *compound* and *position-radical*.

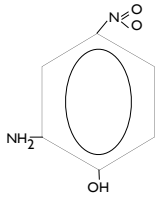
the benzene may be the main group in one compound and a radical in some other compounds.

The implementation of this representation is done using the *feature terms* formalism introduced in [1]. This formalism organizes concepts into a hierarchy of *sorts* (as that of Fig. 1.2), and represent descriptions and individuals as collections of features (functional relations). 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 value is not known it is represented as having the value *any*. All other sorts are more specific that *any*. The most general sort in Fig. 1.2 is *compound*. This sort has three subsorts: *alkane, cyclic* and *functional-group*, which in turn, have other subsorts. The sort *methane* is more specific than the sort *acyclic-alkane*; while the sorts *methane* and *ethane* are not directly comparable.

Each sort has a collection of features characterizing the relations for this sort. For instance, Fig. 1.3 shows that the sort *compound* has two features: main-group and p-radicals. The values of the feature main-group have to be of the sort *compound*, while the feature p-radicals has values of sort *position-radical*. The sort *position-radical* (Fig. 1.3) has, in turn, two features: radicals and position. The feature radicals has values of sort *compound* (since radicals themselves are compounds). The feature position indicates where the radical(s) is bound to the main group.

Fig. 1.4 shows the representation of the chemical compound TR-339, *2-amino-4-nitrophenol*, using feature terms. TR-339 has a benzene as main group and a set of three radicals: an *alcohol* in position one; an *amine* in position two; and a *nitro-deriv* in position four. Notice that this information has been directly extracted from the chemical name of the compound following the nomenclature rules.

This kind of description has the advantage of being very close to the representation that an expert has of a molecule from the chemical name. We have translated, with the support of a chemist, the compounds of the NTP data set to this representation based on the chemical ontology. A shortcoming of the representation based on the chemical name of a compound is the existence of synonymous names. Currently, we have selected one of the possible names and we codified the compound with feature terms using this selected name.



$$TR\text{-}339 = \left[\begin{array}{l} \underline{compound} \\ \mathsf{main\text{-}group} \doteq benzene \\ \mathsf{p\text{-}radicals} \doteq \left[\begin{array}{l} \underline{position\text{-}radical} \\ \mathsf{position} \doteq one \\ \mathsf{radicals} \doteq \left[\underline{compound} \; \mathsf{main\text{-}group} \doteq alcohol\right] \end{array}\right] \\ \qquad\qquad \left[\begin{array}{l} \underline{position\text{-}radical} \\ \mathsf{position} \doteq two \\ \mathsf{radicals} \doteq \left[\underline{compound} \; \mathsf{main\text{-}group} \doteq amine\right] \end{array}\right] \\ \qquad\qquad \left[\begin{array}{l} \underline{position\text{-}radical} \\ \mathsf{position} \doteq four \\ \mathsf{radicals} \doteq \left[\underline{compound} \; \mathsf{main\text{-}group} \doteq nitro\text{-}deriv\right] \end{array}\right] \end{array}\right]$$

Fig. 1.4. Representation of TR-339, *2-amino-4-nitrophenol*, with feature terms.

1.3 The predictive toxicology Task

The NTP data set contains reports of experiments on chemical compounds in order to establish whether they are carcinogenic. Each experiment is performed in two species: rats and mice. Moreover, because the carcinogenic activity of the compounds has proved to be different in both species and also among the sex of the same species, some computational approaches take separately the results of the experiments having, in fact, four data sets: male rats (MR), female rats (FR), male mice (MM) and female mice (FR). The chemical compounds can be classified in each data set into two solution classes: *positive* (i.e., when the compound is carcinogenic) and *negative* (i.e., when the compound is not carcinogenic).

The goal of predictive toxicology is to develop models able to predict whether a chemical compound is toxic or not. The construction of these models by computer assisted techniques takes into account the toxicity observed in some molecules to extract theories about the toxicity on families of molecules. Early systems focused on predictive toxicology were DEREK [27] and CASE [23]. DEREK is a knowledge-based system based on a set of rules describing relations between structural features and their associated toxicity. To determine the toxicity of a new compound, DEREK compares this new compound with all the compounds of the knowledge base. CASE has a base of substructures labeled as *active* or *inactive* according to their toxicity. Thus, to determine the toxicity of a new compound, CASE extracts all its possible substructures and labels each one as active or inactive using the base of substructures. Then CASE uses statistical techniques to determine the global toxicity of the new compound.

There are two families of methods currently used to solve the predictive toxicology task: statistics and ML. A widely used statistical method is re-

gression analysis of molecular descriptors. This technique finds equations that correlate the toxicity of a compound with some physical-chemical properties [21] or with the presence of some functional groups [7]. Probabilistic reasoning such as Bayesian networks has also been widely used to build classifiers [32] or in combination with other techniques like multi-way recursive partitioning [25] and artificial neural networks [6, 5].

The focus of the second PTC [30] was to use ML to address the predictive toxicology problem. From the ML point of view, the goal of the predictive toxicology is a classification task, i.e. toxic compounds are classified as belonging to the *positive* class and non-toxic compounds are classified as belonging to the *negative* class. Moreover, the classification task has to be solved separately for each data set (MR, FR, MM and FM).

The majority of this work was concerned with using inductive techniques to construct toxicity models. Given a solution class C, a set of examples P belonging to C, and a set of examples N that do not belong to C, the goal of inductive learning techniques is to build a general description d of C such that 1) d is satisfied by all the examples in P, and 2) d is not satisfied by the examples in N.

Some inductive techniques build decision trees as predictive classifiers. The representation of the compounds is propositional (in the form of attribute value pairs) and the attributes are the values of molecular properties (molecular weight, physical-chemical properties, etc) and results of toxicity of some other tests. The main shortcoming of decision trees is the propositional representation of the compounds due to two reasons: 1) the high number of descriptors for a compound, and 2) the fact that not all them are equally relevant in order to predict the toxicity. Most approaches use ML and statistical methods to select feature subsets.

A widely used relational learning technique is Inductive Logic Programming (ILP). The main idea of ILP is to induce general descriptions explaining a set of examples represented using logical predicates. The first ILP program used to induce SAR models was PROGOL [29]; it was applied to a set of 230 aromatic and heteroaromatic nitro compounds and the resulting model was compared with models obtained by both linear regression and neural networks with backpropagation. PROGOL's results were very encouraging since the final rules were more understandable than those obtained using the other methods.

Other relational representation approaches consider a compound as a group of substructures instead of sets of atoms. These approaches consider that if a substructure has known toxic activity, then a compound having this substructure can also have toxic activity. Pfahringer and Gini proposed a more abstract representation of the chemical compounds using the concept of functional groups (similar to the chemical ontology we use, see Sect. 1.2.1). This abstraction improves the search process since it represents substructures rather than describing each atom and atom bonds.

Several authors [14, 17, 11] represent the compounds as labeled graphs and this allows the use of graph search algorithms for detecting frequent substructures of the molecules in the same class. Following this approach, SUBDUE [20] discovers substructures beginning with substructures matching a single vertex in the graph and extending them by selection of the best substructure in each iteration. At the end of the process, SUBDUE has a hierarchical description of the data in terms of the discovered substructures. SMILES [31], also following this approach, detects the set of molecular substructures (subgraphs) more frequently occurring in the chemical compounds.

There are also hybrid approaches, such as the one proposed by Gini et al [16]. This approach combines the toxicity results given by a set of fragments of structures with an artificial neural network that uses descriptors of the chemical compounds. Thus, first the authors defined a set of fragments that experts recognize as structures responsible for carcinogenicity. Then they developed a module that searches in the chemical compound structure for the presence of one or more of these fragments. On the other hand, they also used an artificial neural network that assessed the carcinogenicity of a chemical compound taking into account its molecular descriptors. Finally, an ILP module is used to combine the toxicity assessment of the two modules.

A problem with inductive techniques is that the high variability of chemical compounds poses great difficulties to find general rules describing the classes appropriately. In the next section we will introduce our work on lazy learning techniques for predictive toxicology.

1.3.1 Lazy Learning Techniques

Inductive learning techniques try to extract general rules describing the cases in each class. This kind of techniques has some difficulties in dealing with domains, like toxicology, where entities are subject to high variability. Lazy learning techniques, on the other hand, are based on the retrieval of a set of solved problems (*cases*) similar to a specific problem. A critical issue in lazy learning is the evaluation of similarity between two cases, as this forms the basis for identifying a suitable set of cases or 'promising' candidates. Several authors use the concept of similarity between chemical compounds: HazardExpert [12] is an expert system that evaluates the similarity of two molecules based on the number of common substructures; Sello [28] also uses the concept of similarity but the representation of the compounds is based on the energy of the molecules.

Shaud

When the domain objects have a propositional representation, the similarity between two objects is assessed by computing the similarity of attributes and then aggregating their similarities to obtain a global measure of the similarity of the objects. Shaud is a similarity measure able of assessing the similarity

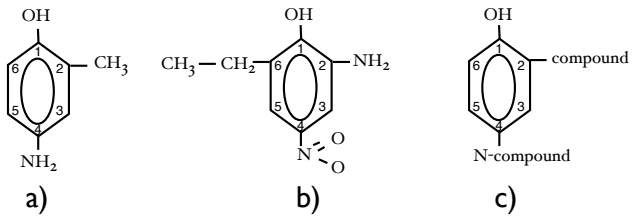


Fig. 1.5. a) 2-methyl-4 aminophenol. b) 2-amino-4-nitro-6-ethanophenol. c) structure shared by the chemical compounds a) and b). *compound* and *N-compound* are the most specific sort (*lub*) of the radicals in the respective positions, according to the sort/subsort hierarchy in Fig. 1.2

between structured objects represented as feature terms. Given two objects Shaud distinguishes two parts in their structure: one formed by the features present in both objects, called the *shared structure*; and another formed by those features that are only present in one of the objects (but not the other) called the *unshared structure*. For instance, Fig. 1.5 shows that the molecules a) and b) have in common the structure c). In this example, the unshared structure is only the radical *ethane* in position six of the molecule b).

Shaud [2, 3] assesses the similarity of two feature terms by computing the similarity of the shared structure and then normalizing this value taking into account both the shared and the unshared structure. The comparison of the shared structure is performed element by element comparing the position of their sorts into the sort/subsort hierarchy in the following way:

$$S(sort(\psi^1), sort(\psi^2)) = \begin{cases} 1 & \text{if } sort(\psi^1) = sort(\psi^2) \\ 1 - \frac{1}{M} level(lub(sort(\psi^1), sort(\psi^2)) & \text{otherwise} \end{cases}$$

The idea is that the similarity between two values depends on the level of the hierarchy (see Fig. 1.2) where their least upper bound (*lub*) is situated in the sort hierarchy: the more general *lub(*v_1, v_2*)* the smaller is the similarity between v_1 and v_2. M is the maximum depth of the sort hierarchy.

For instance, in order to assess the similarity of the molecules a) and b) in Fig. 1.5, Shaud takes into account the structure shared by both molecules (c) and compares the elements composing that structure (Fig. 1.6). The similarity assessment of the shared structure is the following:

- the main group that is *benzene* in both molecules, therefore

$$S(benzene, benzene) = 1$$

- a radical in position 1 that is an alcohol in both molecules, therefore

$$S(alcohol, alcohol) = 1$$

$$c = \left[\begin{array}{ll} \underline{compound} & \\ \mathsf{main\text{-}group} \doteq benzene & \\ & \left[\begin{array}{l} \underline{position\text{-}radical} \\ \mathsf{position} \doteq one \\ \mathsf{radicals} \doteq \left[\begin{array}{l} \underline{compound} \\ \mathsf{main\text{-}group} \doteq alcohol \end{array}\right] \end{array}\right] \\ \mathsf{p\text{-}radicals} \doteq & \left[\begin{array}{l} \underline{position\text{-}radical} \\ \mathsf{position} \doteq two \\ \mathsf{radicals} \doteq compound \end{array}\right] \\ & \left[\begin{array}{l} \underline{position\text{-}radical} \\ \mathsf{position} \doteq four \\ \mathsf{radicals} \doteq \left[\begin{array}{l} \underline{compound} \\ \mathsf{main\text{-}group} \doteq N\text{-}compound \end{array}\right] \end{array}\right] \end{array}\right]$$

Fig. 1.6. Formal representation of molecule c) shown in Fig. 1.5

- a radical in position 2 that is a *methane* in the molecule a) and an *amine* in the molecule b), therefore

$$S(methane, amine) = 1 - \frac{1}{M} level(lub(methane, amine))$$

and since $lub(methane, amine) = compound$, M = 5, and *level(compound)* = 5 (see Fig. 1.2) then

$$S(methane, amine) = 1 - \frac{1}{5} level(compound) = 1 - \frac{1}{5} 5 = 0$$

- a radical in position 4 that is an amine in the molecule a) and a nitro-derivate (nitro-deriv) in the molecule b), therefore

$$S(amine, \text{nitro-deriv}) = 1 - \frac{1}{M} level(lub(amine, \text{nitro-deriv}))$$

and since *lub(amine, nitro-deriv)* = *N-compound*, M = 5 and *level(N-compound)* = 3 (see Fig. 1.2) then

$$S(amine, \text{nitro-deriv}) = 1 - \frac{1}{5} level(\text{N-compound}) = 1 - \frac{1}{5} 3 = 0.4$$

Because these are simple molecules where the radicals themselves have no radicals, the similarity of the common part is

$$S(benzene, benzene) + S(methane, amine) + S(amine, nitro\text{-}deriv) = 2.4$$

Then, this value is normalized by the total number of nodes (those of the shared structure plus those of the unshared structure), i.e., $S(a, b) = \frac{2.4}{5} = 0.48$.

Table 1.1. Distribution of the examples in the four PTC data sets and the accuracy results obtained by two of the authors presented at the PTC compared with the accuracy of Shaud with $k = 5$ and the MC and CSA criteria.

	Composition			PTC		Acc (k = 5)	
data set	+	−	total	Ohwada	Boulicaut	MC	CSA
MR	81	125	206	55.59	55.88	48.06	**62.13**
FR	66	140	206	65.43	**68.66**	49.03	64.08
MM	63	139	202	64.11	63.49	60.40	**64.85**
FM	78	138	216	**63.69**	60.61	59.72	62.50

We have performed several experiments using the *k-nearest neighbor* (*k-NN*) algorithm [13]. Given a new problem p, the k-NN algorithm retrieves the k most similar cases and classifies p into the class resulting of the aggregation of the classes where the k cases belong. There are two key issues in the k-NN algorithm: the similarity measure and the aggregation. In our experiments, we took Shaud as similarity and the majority class (MC) for aggregation (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 the majority criterion with different values of k did not provide a satisfactory accuracy. We proposed the *Class Similarity Average* (CSA) criterion [3], a domain-independent criterion that takes into account the similarity of the k most similar cases and also the solution class where they belong.

For each compound p to be classified, Shaud yields the similarity between p and each one of the k most similar cases. 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 p. More formally, let p be the compound to be classified and R_k the set of the k cases most similar to p according to the Shaud results. Each case $c_i \in R_k$ has the following data associated: 1) the structural similarity s_i between p and c_i, i.e. $s_i = \mathsf{Shaud}(p, c_i)$; and 2) for each data set (i.e. MR, FR, MM and FM) the compound c_i is *positive* or *negative*.

For each data set, let A^+ be the set containing cases $c_i \in R_k$ with positive activity, and A^- be the set containing 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.

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

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.*

Table 1.1 shows the results of using k-NN with $k = 5$ both MC and the CSA criteria together with the accuracy of two methods presented by [26, 9] in the PTC. Notice that the accuracy using the CSA criterion is higher than using MC. Also, the accuracy taking separately positive and negative examples is

more balanced using the CSA criterion. In particular, for the MR data set, the accuracies using MC are $Acc^+ = 35.80$ and $Acc^- = 56$ whereas the accuracies using CSA are $Acc^+ = 55.55$ and $Acc^- = 66.40$.

Lazy Induction of Descriptions

Lazy Induction of Descriptions (LID) is a lazy concept learning technique for classification tasks in case-based reasoning (CBR). LID determines which are the more relevant features of a problem and searches in the case base for cases sharing these relevant features. The problem is classified when LID finds a set of relevant features shared by a subset of cases all them belonging to the same solution class C_i. Then LID classifies the problem as belonging to C_i. We call *similitude term* the structure formed by these relevant features and *discriminatory set* the set of cases satisfying the similitude term. The similitude term is a feature term composed of a set of features shared by a subset of cases belonging to the same solution class.

Given two feature terms, there are several similitude terms, LID builds the similitude term with the most relevant features. The relevance of a feature is heuristically determined using the *López de Mántaras* (LM) distance [24]. The LM distance assesses how similar two partitions are in the sense that the lesser the distance the more similar they are (see Fig. 1.7). Each feature f_i of an example induces a partition P_i over the case base according to the values that f_i can take in the cases. On the other hand, the LM considers the *correct partition* P_c that is the partition where all the cases contained into a partition set belong to the same solution class.

Given two partitions P_A and P_B of a set S, the distance among them is computed as follows:

$$LM(P_A, P_B) = 2 - \frac{I(P_A) + I(P_B)}{I(P_A \cap P_B)}$$

where $I(P)$ is the information of a partition P and $I(P_A \cap P_B)$ is the mutual information of two partitions.

In our case, the distance measure is applied to compute the distance among a partition generated by a feature and the correct partition. The correct partition P_c has two classes, one containing the positive examples (examples in C_k) and the other containing the negative examples (those not in C_k). Thus, for each feature f_i, there is a partition P_i of the case base B according to the values of f_i. Each partition P_i is compared with the correct partition P_c using the López de Mántaras distance. The most discriminatory feature f_d is that producing a partition P_d having the minimum distance $LM(P_d, P_c)$ to the correct partition P_c.

Let P_c be the correct partition and P_i and P_j the partitions induced by features f_i and f_j respectively. We say that the feature f_i is *more discriminatory than* the feature f_j iff $LM(P_i, P_c) < LM(P_j, P_c)$. In other words, when

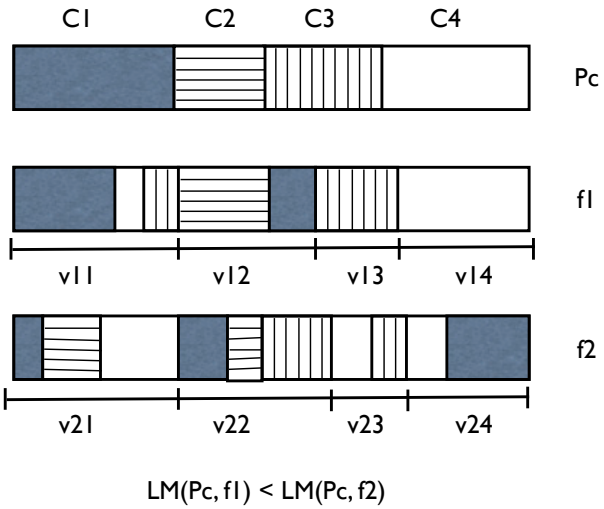
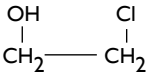


Fig. 1.7. Intuitive idea of the LM distance. The case base B contains precedents belonging to four solution classes. The partition induced by the feature $f1$ is more similar to the correct partition Pc than the partition induced by $f2$.

$$D^1 = \left[\begin{array}{l}\underline{compound} \\ \text{p-radicals} \doteq \left[\begin{array}{l}\underline{position\text{-}radical} \\ \text{radicals} \doteq \left[\begin{array}{l}\underline{compound} \\ \text{main-group} \doteq alcohol\end{array}\right]\end{array}\right]\end{array}\right]$$



2-chloroethanol

$$D^2 = \left[\begin{array}{l}\underline{compound} \\ \text{main-group} \doteq ethane \\ \text{p-radicals} \doteq \left[\begin{array}{l}\underline{position\text{-}radical} \\ \text{radicals} \doteq \left[\begin{array}{l}\underline{compound} \\ \text{main-group} \doteq alcohol\end{array}\right]\end{array}\right]\end{array}\right]$$

Fig. 1.8. Similitude terms build by LID to classify the *2-chloroethanol* in the negative class for male rats.

a feature f_i is more discriminatory than another feature f_j the partition that f_i induces in B is closer to the correct partition P_c than the partition induced by f_j. Intuitively, the most discriminatory feature classifies the cases in B in a more similar way to the correct classification of cases. LID uses the *most discriminatory than* relationship to estimate the features that are most relevant for the purpose of classifying a new problem.

Now, we will illustrate the performance of LID (see algorithm in Fig. 1.9) to assess toxicity of the *2-chloroethanol* (the TR-275 in the PTC case base) for male rats. LID inputs are $S_D = B$ of chemical compounds, a similitude term D initialized to the most general feature term (i.e. the most general description),

```
Function LID (Δ_D, p, D, C)
    if stopping-condition(Δ_D)
        then return class(Δ_D)
        else f_d := Select-feature (p, Δ_D, C)
             D' := Add-feature(f_d, D)
             Δ_D' := Discriminatory-set (D', Δ_D)
             LID (Δ_D', p, D', C)
    end-if
end-function
```

Fig. 1.9. The LID algorithm. D is the similitude term, Δ_D is the discriminatory set of D, C is the set of solution classes, $class(\Delta_D)$ is the class $C_i \in C$ to which all elements in Δ_D belong.

the description of the *2-chloroethanol*, and the set S_D (the discriminatory set associated to D) that contains all the cases that satisfy the structure described by D. Initially $S_D = B$ since D is satisfied by all the cases in B.

The first step of LID is to check whether all the cases in Δ_D belong to the same solution class. Since this stopping condition is not satisfied at the beginning, the second step is to specialize D. The specialization D^1 of D is built by adding to D the path p-radicals.radicals.main-group with main-group taking value *alcohol*, as in the *2-chloroethanol* (see Fig. 1.8). The discriminatory set Δ_{D_1} contains now 42 cases subsumed by D^1, i.e. those compounds in Δ_D having a radical alcohol. Next, LID is recursively called with D^1 and Δ_{D_1}.

The cases in the discriminatory set Δ_{D_1} do not satisfy the stopping condition, i.e. some of them belong the positive class and some others belong to the *negative* class, therefore D^1 has to be specialized by adding a new discriminatory feature. Now most discriminatory feature is main-group. The specialization D^2 is built by adding main-group to D^1 with value *ethane* (see Fig. 1.8). LID is recursively called with the set Δ_{D_2} and the similitude term D^2.

The set Δ_{D_2} contains 6 cases all of them belonging to the negative class. Therefore LID terminates classifying the *chloroethanol* as belonging to the *negative* class and explaining it with the similitude term D^2 (shown in Fig. 1.8), i.e. because the compound is an ethane with a radical alcohol. This classification is supported by the 6 cases in Δ_{D_2}. The result of LID is the solution class C_i and a similitude term D^n. The similitude term D^n can be seen as an explanation of why the current problem p is in the solution class C_i. D^n is a *partial* description of C_i because, in general, not all cases in C_i satisfy D^n.

We conducted a series of experiments with the similitude terms to discover patterns in the Toxicology data set. These experiment had two steps: 1) use LID with the leave-one-out method in order to generate similitude terms for

Table 1.2. Accuracy of LID and C-LID on the PTC data set.

data set	# cases	LID	C-LID
MR	297	58.27	60.54
FR	296	63.09	66.97
MM	296	52.39	53.95
FM	319	52.36	56.60

classifying the cases; and 2) select a subset of these similitude terms. The first step yields a set of similitude terms that have been used for classifying some cases. The second step selects only those similitude terms that are totally discriminatory (we call them *patterns*). Some of the patterns detecting positive toxicity are also reported in the literature. For instance, LID founds that compounds with a radical chlorine are carcinogenic and Brautbar describes some experiments confirming the toxicity of chlorinated hydrocarbons.

As a second experiment, we defined Caching LID (C-LID), a lazy learning approach that reuses the patterns used for solving past problems in order to improve the classification of new problems in case based-reasoning (CBR). C-LID is implemented on top of LID by defining two policies: the caching policy and the reuse policy. The *caching policy* determines which similitude terms (patterns) are to be retained. The *reuse policy* determines when and how the cached patterns are used to solve new problems. In our experiments, the caching policy of C-LID states that a similitude term D will be cached if it is *univocal*, i.e. when all cases covered by a pattern belong to one class only. The *reuse policy* of C-LID states that patterns will be used for solving a problem p only when LID is unable to univocally classify p.

Thus, the experiment with C-LID has two phases: 1) a preprocessing of the case base in order to obtain some patterns to be cached; and 2) the problem solving phase that uses LID together with the cached patterns for classifying new problems. The preprocessing phase is done using the leave-one-out technique using the cases in the case base B. For each case $c \in B$, C-LID uses LID to classify c and generates a similitude term D_c. When D_c is univocal C-LID caches it. Thus, at the end of the preprocessing phase C-LID has obtained a set $M = \{D_1 \dots D_n\}$ of patterns. The reuse policy decides when to use these patterns during the problem solving phase.

The evaluation of the predictive accuracy of the methods has been made using 10-fold cross-validation. Table 1.2 shows the accuracy of LID and C-LID for each one of the data sets. Notice that C-LID improves the accuracy of LID in all the data sets showing that the caching policy is adequate. Notice that the caching policy stores only the similitude terms that are univocal, i.e. those subsuming cases belonging to only one solution class. With this policy C-LID takes into account only those patterns with clear evidence of a good discrimination among classes.

1.4 Conclusions

We have seen that the task of predicting the possible activity of molecules is a challenging one, from the chemist viewpoint and also the field of ML. From the chemist viewpoint it is interesting that automated techniques may be capable of predicting with some degree of accuracy the toxicity of chemical compounds that have not been synthesized. Predicting toxicity is a complex task for ML that requires thoughtful analysis of all dimensions involved.

We have summarily described several ML approaches to toxicity prediction, and we have highlighted the dimension of example representation. ML approaches that use a propositional representation (i.e., an example is represented by a vector of attribute value pairs) have problems for mapping the chemical model of chemical compounds based on SAR into vectors of attribute value pairs. Since this mapping ignores the structure itself, other ML approaches use relational learning techniques; specifically ILP maps the SAR models into a logic representation of examples and background knowledge. Our approach proposes a new kind of relational representation based on the chemical ontology that describes the compounds' structure in a more abstract way. The experiments have shown that the predictive performance of our methods (SHAUD and C-LID using the chemical ontology based representation) have comparable results to that of methods that use SAR models.

ML techniques are very dependent on the way examples are represented. The fact that ML techniques — using propositional SAR, relational SAR, and chemical ontology — achieve a similar performance in predicting toxicity implies that they possess a comparable information content in terms of the studied molecules. Nonetheless, toxicity prediction is a complex task for ML techniques, since their performance is just relatively good [19] while they can be very good for other tasks. Because there is no ML technique providing excellent results, a likely explanation is that the current representation of chemical compounds is not adequate. Notice that a compound can be toxic in a data set (say male rats) and not in another (say female mouse): since the representation of the examples is the same, and yet they have different solutions, this seems to indicate that there are external factors involved that are not represented in the examples themselves. An enriched characterization of the compounds would very likely improve the predictive accuracy of the ML techniques we have discussed here.

AcknowledgementsThis work has been supported by the SAMAP project (TIC2002-04146-C05-01). The authors thank Josep Lluís Arcos and Lluís Bonamusa for their support in the elaboration of this paper.

References

1. Armengol, E. and E. Plaza: 2001a, 'Lazy induction of descriptions for relational case-based learning'. In: L. D. Reaedt and P. Flach (eds.): *ECML-2001. Freiburg.*

Germany. pp. 13–24.
2. 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.
3. Armengol, E. and E. Plaza: 2003, 'Relational case-based reasoning for carcinogenic activity prediction'. *Artificial Intelligence Review* **20**(1–2), 121–141.
4. Ashby, J. and R. Tennant: 1994, 'Prediction of rodent carcinogenicity for 44 chemicals: results'. *Mutagenesis* **9**, 7–15.
5. Bahler, D., B. Stone, C. Wellington, and D. Bristol: 2000, 'Symbolic, neural, and bayesian machine learning models for predicting carcinogenicity of chemical compounds.'. *J. of Chemical Information and Computer Sciences* **8**, 906–914.
6. Basak, S., B. Gute, G. Grunwald, D. Opitz, and K. Balasubramanian: 1999, 'Use of statistical and neural net methods in predicting toxicity of chemicals: a hierarchical QSAR approach'. In: G. Gini and A. Katrizky (eds.): *Predictive Toxicology of Chemicals: Experiences and Impacts of AI Tools.* pp. 108–111.
7. Benfenati, E., S. Pelagatti, P. Grasso, and G. Gini: 1999, 'COMET: the approach of a project in evaluating toxicity'. In: G. Gini and A. Katrizky (eds.): *Predictive Toxicology of Chemicals: Experiences and Impacts of AI Tools.* pp. 40–43.
8. Blinova, V., D. Bobryinin, V. Finn, S. Kuznetsov, and E. Pankratova: 2003, 'Toxicology analysis by means of simple JSM method'. *Bioinformatics* **19**(10), 1201–1207.
9. Boulicaut, J.-F. and B. Cremilleux: 2001, 'δ-strong classification rules for characterizing chemical carcinogens'. In: *Proceedings of the Predictive Toxicology Challenge Workshop, Freiburg, Germany, 2001.*
10. Bristol, D., J. Wachsman, and A. Greenwell: 1996, 'The NIEHS predictive toxicology evaluation project.'. *Environmental Health perspectives* **104**, 1001–1010.
11. 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.
12. Darvas, F., A. Papp, A. Allerdyce, E. Benfenati, G. Gini, M. Tichy, N. Sobb, and A. Citti: 1999, 'Overview of different AI approaches combined with a deductive logic-based expert system for predicting chemical toxicity'. In: G. Gini and A. Katrizky (eds.): *Predictive Toxicology of Chemicals: Experiences and Impacts of AI Tools.* pp. 94–99.
13. Dasarathy, B. V.: 1990, *Nearest neighbor (NN) norms: NN pattern classification techniques.* Washington; Brussels; Tokyo; IEEE computer Society Press.
14. 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 Int. Conf. on Knowledge Discovery and Data Mining.* pp. 30–36.
15. Deshpande, M. and G. Karypis: 2002, 'Automated approaches for classifying structures'. In: *Proc. of the 2nd Workshop on Data Mining in Bioinformatics.*
16. Gini, G., M. Lorenzini, E. Benfenati, R. Brambilla, and L. Malvé: 2001, 'Mixing a symbolic and subsymbolic expert to improve carcinogenicity prediction of aromatic compounds'. In: *Multiple classifier systems. 2th Intern. Workshop.* pp. 126–135.
17. Gonzalez, J., L. Holder, and D. Cook: 2000, 'Graph Based Concept Learning'. In: *AAAI.* p. 1072.
18. Helma, C., E. Gottmann, and S. Kramer: 2000, 'Knowledge Discovery and Data Mining in Toxicology'. *Statistical Methods in Medical Research* **9**, 329–358.

19. Helma, C. and S. Kramer: 2003, 'A survey of the Predictive Toxicology Challenge 2000-2001'. *Bioinformatics* pp. 1179–1200.
20. Holder, L., D. Cook, and S. Djoko: 1994, 'Substructure Discovery in the SUBDUE System'. In: *Proceedings of the AAAI Workshop on Knowledge Discovery in Databases*. pp. 169–180.
21. Karelson, M. and U. Maran: 1999, 'QSPR and QSAR models derived with CODESSA multipurpose statistical analysis software'. In: G. Gini and A. Katrizky (eds.): *Predictive Toxicology of Chemicals: Experiences and Impacts of AI Tools*. pp. 12–23.
22. Katritzky, A., R. Petrukhin, H. Yang, and M. Karelson: 2002, *CODESSA PRO. User's manual*. University of Florida.
23. Klopman, G.: 1984, 'Artificial intelligence approach to structure-activity studies: Computer automated structure evaluation of biological activity of organic molecules'. *Journal of the America Chemical society* **106**, 7315–7321.
24. López de Mántaras, R.: 1991, 'A distance-based attribute selection measure for decision tree induction'. *Machine Learning* **6**, 81–92.
25. Mello, K. and S. Brown: 1999, 'Combining recursive partitioning and uncertain reasoning for data exploration and characteristic prediction'. In: G. Gini and A. Katrizky (eds.): *Predictive Toxicology of Chemicals: Experiences and Impacts of AI Tools*. pp. 119–122.
26. 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.*
27. Sanderson, D. and C. Earnshaw: 1991, 'Computer prediction of possible toxic action from chemical structure: the DEREK system'. *Human and Experimental Toxicology* **10**, 261–273.
28. Sello, G.: 1999, 'Similarity, diversity and the comparison of molecular structures'. In: G. Gini and A. Katrizky (eds.): *Predictive Toxicology of Chemicals: Experiences and Impacts of AI Tools*. pp. 36–39.
29. Srinivasan, A., S. Muggleton, r.D. King, and M. Sternberg: 1994, 'Mutagenesis: ILP experiments in a non-determinate biological domain'. In: *Proceedings of the Fourth Inductive Logic Programming Workshop*.
30. Srinivasan, S., S. Muggleton, R. King, and M. Stenberg: 1997, 'The Predictive Toxicology Evaluation Challenge'. In: *IJCAI, Nagoya, Japan*. pp. 4–9.
31. Weininger, D.: 1988, 'SMILES a Chemical Language and Information System'. *J. Chem. Inf. Comput. Sci.* **28**(1), 31–36.
32. Wellington, C. and D. Bahler: 1999, 'Predicting rodent carcinogenicity by learning bayesian classifiers'. In: G. Gini and A. Katrizky (eds.): *Predictive Toxicology of Chemicals: Experiences and Impacts of AI Tools*. pp. 131–134.