Usages of Generalization in Case-based Reasoning

Eva Armengol

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

Abstract. The aim of this paper is to analyze how the generalizations built by a CBR method can be used as local approximations of a concept. From this point of view, these local approximations can take a role similar to the global approximations built by eager learning methods. Thus, we propose that local approximations can be interpreted either as: 1) a symbolic similitude among a set of cases, 2) a partial domain model, or 3) an explanation of the system classification. We illustrate these usages by solving the Predictive Toxicology task.

1 Introduction

One of the main differences between eager and lazy methods used for concept learning is that the former generalizes a set of examples and builds a *global approximation* of a concept. Then, this global approximation is used for classifying unseen examples. Instead, lazy learning methods do not explicitly generalize the examples but they always use the complete set of examples. Thus, an unseen problem is classified according to its similitude to a subset of known examples. In this sense, lazy learning methods can be seen as building *local approximations* of concept [**?**] since the similar examples define an area around the new example which can be taken as a general description of that area. However sometimes the general knowledge, in the sense of global approximations of concepts, could also be useful inside lazy learning methods. PROTOS [**?**], one of the early Case-based Reasoning (CBR) systems, takes the idea of generalization commonly used on inductive learning methods to define categories of cases and also defines *exemplars* representing each category. Then, a new case is classified into a category if a match can be found between an exemplar and the new case. Notice that the exemplars play the same role as general descriptions of a class induced by some inductive learning method. Bergmann et al [**?**] proposed the idea of generalized cases, i.e. a case does not represent a single point of the problem-solution space but a subspace of it. The use of generalized cases can be seen as general descriptions of parts of the problem space.

In this paper we are interested in analyzing how generalizations can be used inside CBR. In particular, from both the literature and our experience we identified some usages that generalization can have in the context of CBR. Thus, a generalization can be taken as a representative of a subset of cases as in PROTOS or in the work of Bergmann et al. [?], but also it could be interpreted as a symbolic similitude of a subset of cases as we proposed in [?]. In addition, we also propose the hypothesis that a set of local approximations can be seen as a partial model of a domain. The idea is that a lazy method can produce a generalization that explains the classification of a new problem, in a sense similar to the explanations produced by *explanation-based learning methods* [?]. A set of such explanations can be seen as a partial model of the domain since that model is able to classify only a subset of the available cases.

The structure of this paper is the following. Firstly we briefly introduce LID the method that we used in our experiments. LID produces a generalization that we call *similitude term* and that serves as the basis for the analysis of generalizations inside CBR. In particular, in section **??** we describe how generalizations can be interpreted as a symbolic similitude among a subset of cases. Then, in section **??** we explain how generalizations produced by a lazy method can be used to build a lazy model of the domain. In section **??** we describe how lazy generalizations can be interpreted as the explanation of the classification proposed by a CBR method. Finally, in section **??** we describe an application domain where we applied all the usages of generalizations we described in the previous sections.

2 Lazy Induction of Descriptions

In this section we briefly describe a lazy learning method called *Lazy Induction of Descriptions* (LID) we introduced in [?]. LID determines which are the more relevant features of a problem p and searches in the case base for cases sharing these relevant features. The problem p is classified when LID finds a set of relevant features shared by a subset of cases all belonging to the same solution class C_i. Then LID classifies the problem as belonging to C_i (Fig. **??**). We call *similitude term* the description formed by these relevant features and *discriminatory set* the set of cases satisfying the similitude term. In fact, a similitude term is a generalization of both p and the cases in the discriminatory set.

The similitude term can be interpreted in several ways. Firstly, the similitude term can be seen as a partial discriminant description of C_i since all the cases satisfying the similitude term belong to C_i (according to one of the stopping conditions of LID). Therefore, the similitude term can be used as a generalization of knowledge in the sense of either PROTOS or inductive learning methods. On the other hand, because the similitude term contains the important features used to classify a problem, it can be interpreted as a justification or *explanation* of why the problem has been classified in C_i. Finally inside the context of multiagent systems, where agents collaborate for solving problems, similitude terms could be taken as the basis for both exchanging knowledge and negotiation. In next sections the different usages of similitude terms is explained.

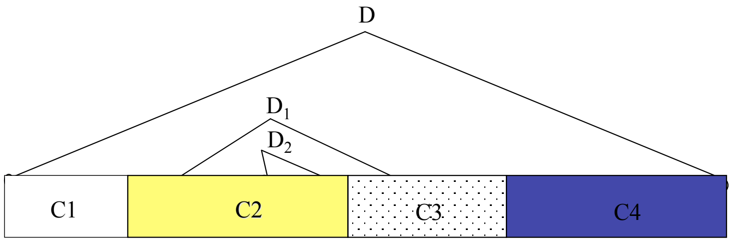


Fig. 1. From a description D that is satisfied by all the cases of the case base, LID builds successive specializations of D, until finding a similitude term (D_2 in this Fig.) that only is satisfied by cases of one class.

3 Generalizations as Symbolic Similitude

Similarity among cases is one of the key issues of lazy methods in general and of CBR in particular. The usual approach to assess this similarity is by defining similarity measures. Since features defining domain objects can have different relevance concerning the classification task, some of these measures allow us to to weigh the features differently. Emde and Wettscherek [**?**] analyzed how the similarity measure influences the result of Instance-based Learning algorithm [**?**].

Eager learning methods induce discriminant descriptions of classes, i.e. they build descriptions with features that are only satisfied by examples belonging to one of the classes. For instance, an inductive learning method such as ID3 [**?**] produces a decision tree where each path from the root to a leaf gives the pairs attribute-value that are important to classify an example as belonging to a class C_i. Notice that, in fact, a path is a general and discriminant description d_i of C_i that can be interpreted as a *symbolic similitude* among the cases in C_i. In other words, d_i contains the features shared by a set of examples belonging to C_i.

What is the role of the similitude term produced by LID? On one hand, LID classifies a new problem p as belonging to a class C_i *because* it is similar to a subset of cases in C_i that share some features that have been considered as the most important for the classification. Therefore, in this sense the similitude term plays the role of symbolic similitude as the paths of a decision tree. On the other hand, because LID is a lazy method, that similitude term shows the similitude of the particular problem p to the subset of cases belonging to C_i that satisfy the similitude term.

4 Lazy Generalizations for building Lazy Domain Models

Lazy learning methods classify a new problem based on the similarity among that problem and a subset of known cases. Commonly, once the system proposes the solution, all the generalizations used to achieve the solution are rejected.

The justification of this is that any generalization is constructed based on the new problem. Our point is that, although these generalizations define a local approximation to the concept defined by the new problem, they can be useful for solving other problems inside such area. Therefore, as well as a CBR system is solving new problems, it can store all the local approximations supporting the classification of these problems. The set of such approximations can be seen as a *partial* model of the domain. The partiality of that domain comes from the fact that each local approximation is build from a subset of examples instead of being a model including all the known examples as in eager learning methods.

This lazy way to build a domain model can be useful in domains such as Predictive Toxicology [?] or some medical problems, where experts are interested in finding models about the domain. The usual tool in such domains is an eager learning method inducing general domain knowledge. The main problem of these approaches is that sometimes the models have to be induced from a set of cases with high variability and the result is a set of rules that are too general to be useful for classification. An example of a lazy construction of a domain model is the *Lazy Decision Trees (LDT)* proposed by [?]. Differently from pure eager techniques, LDT build a decision tree in a lazy way, i.e. each time that a new problem has to be classified, the system reuses, if possible, the existing tree. Otherwise, a new branch classifying the new problem is added to the tree. Notice that, in fact, the decision tree represents a general model of a domain and LDT builds it in a lazy way. The main difference between inductive learning methods and LDT is that the former generalize from all the examples of a class whereas the latter takes into account only the characteristics of the problem at hand.

A similar idea is behind the method C-LID [?]. C-LID is implemented on top of LID by storing the similitude terms provided by LID and using them as domain knowledge useful for solving new problems. C-LID uses 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. The caching policy of C-LID states that a similitude term D is stored 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.

The assumption of C-LID is that the similitude term is a partial description of the solution class in the same sense as in inductive learning methods. Thus the set of patterns stored by C-LID can be seen (an used) as a domain model, even if this model is partial because it does not cover all the available examples.

5 Generalizations and Explanations

Explaining the outcome of a CBR system has been an issue of growing interest in recent years. In 2004 was the first workshop on explanations in the framework of the EWCBR held in Madrid [?]. The focus of this workshop was to analyze how CBR applications from very different domain explain their result to the user. Then, in 2005 Roth-Berghofer and his colleagues organized an international workshop in the framework of the AAAI conference [?] with the same focus: to analyze different forms to explain the results. In the latter workshop the scope was not only CBR but authors participating in it coming from very different fields.

Focusing on CBR, in particular in recommender systems, the most common form of explanation is to show the user the set of cases that the system has assessed as the most similar to the new case at hand. Nevertheless some authors agree that in some situations this may not be a good explanation. For instance, McSherry [?] argues that the most similar case (in addition to the features that have been taken as relevant for selecting that case) also has features that could act as arguments against that case. For this reason, McSherry proposes that the explanation of a CBR system has to explicitly distinguish between the case features in favor of an outcome and the case features against it. In this way, the user could decide about the final solution of the problem. A related idea, proposed in [?], is to use the differences among cases to support the user in understanding why some cases do not satisfy some requirements.

Explanations had received attention from the early rule-based systems, that explained the result by showing the user the chain of rules that produce the solution. Inductive learning methods can also explain their results by showing the general descriptions satisfied by the new problem. The explanation of a decision tree outcome could be formed by showing the conditions satisfied in the path from the root to a leaf used to classify a new problem. *Explanation-based learning (EBL)* [?] is a family of methods that build explanations by generalizing examples. In short, EBL solves a problem and then analyzes the problem solving trace in order to generalize it. The generalized trace is an *explanation* that, in fact, is used as a new domain rule for solving new problems. This explanation is represented using the same formalism as the problems, therefore it is perfectly understandable and usable by the system. In other words, the generalization of the process followed for solving a problem has been taken as explanation of the result and can be also used for solving future problems. Conceptually similar is the use that [?] makes of the similitude terms given by LID. The similitude term can be seen as a justification of the classification given by LID since it contains all the aspects considered as relevant to classify an example.

An explanation scheme for CBR based on the concept of *least general generalization* was introduced in [?]. The relation *more general than* ($\geq_g$) forms a lattice over a generalization language $\mathcal{G}$. Using the relation $\geq_g$ we can define the *least general generalization* or *anti-unification* of a collection of descriptions (either generalizations or instances) as follows:

- $AU(d_1, ..., d_k) = g$ such that $(g \geq_g d_1) \wedge ... \wedge (g \geq_g d_k)$ and not exists $(g' \geq_g d_1) \wedge ... \wedge (g' \geq_g d_k)$ such that $g >_g g'$

In other words the anti-unification g of a set of descriptions is the most specific generalization of these descriptions in the sense that there is no other generalization g' of all these descriptions that is more specific than g. The anti-unification is a description composed of all the properties shared by the descriptions. Therefore, the anti-unification can be seen as a symbolic description of the similarity among these descriptions.

Thus, descriptions resulting from the anti-unification of a collection of cases can be used to provide explanation of the classification of a new problem in CBR systems. Let us explain in more detail the explanation scheme based on the anti-unification concept we introduced in [**?**].

Let C be the set of cases that have been considered as the most similar to a problem p. For the sake of simplicity we assume that there are only two solution classes: $C_1 \subseteq C$ and $C_2 \subseteq C$ ($C = C_1 \cup C_2$). The explanation scheme is composed of three descriptions:

- AU^*: the anti-unification of p with all the cases in C. This description shows what aspects of the problem are shared by all the compounds in C, i.e. cases in C are similar to p because they have in common what is described in AU^*.
- AU_1: the anti-unification of p with the cases in C_1. This description shows what has c in common with the cases in C_1.
- AU_2: the anti-unification of p with the cases in C_2. This description shows what has p in common with the cases in C_2.

Thus the explanation of why a case p is in a class C_i is given by what p shares with the retrieved cases in that class. In other words, the anti-unification $AU(c_1 ... c_k, p)$ is an explanation of why the cases in C are similar to p, since it is a description of all that is shared among the retrieved cases and the new problem. Section **??** shows an example of ow this explanation scheme is used on the Predictive Toxicology task.

In the next section we explain in detail an application on Predictive Toxicology, where all the usages of generalizations explained in the previous sections have been applied.

6 A case study: Predictive Toxicology

In this section we explain the approach we introduced to solve the predictive toxicology task, i.e. to assess the carcinogenic activity of chemical compounds. This is a complex problem that most approaches try to solve using machine learning methods. The goal of these approaches is to build a general model of carcinogenesis from both domain knowledge and examples of carcinogen and non-carcinogen chemical compounds. Because these general models give not enough predictivity, we take a completely different vision of the problem. Our idea is that the low predictivity of the induced models is due to the high variability of the chemical compounds that produces overgeneralizations. Thus, we decided to take a lazy approach and to consider that the goal is to classify a chemical compound as carcinogen or non-carcinogen. Therefore all the efforts have to focus on the features allowing the classification of the chemical compound at hand. In other words, we do not try to build a general model of carcinogenesis as ML techniques do but we only try to classify a particular chemical compound.

Nevertheless, we benefit from the classification of that compound to build some patterns of carcinogenesis.

In the next sections we explain how we solved the problem. First we describe the predictive toxicology problem and a new representation of chemical compounds using feature terms. Then we describe how C-LID can be used as a lazy problem solving method but also as a form to build some domain knowledge. Finally, we detail how the system can explain the results to a chemist by means of the explanation scheme introduced in section **??**.

6.1 The Toxicology domain

Every year thousands of new chemicals are introduced 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 of 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 on 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, it is included in the NTP list in order to perform standardized experiments to determine its toxicity degree.

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 [**?**] for a survey).

6.2 Representation of chemical compounds

Predictive toxicology is a complex task for ML techniques. There is no ML technique providing excellent results [**?**], a likely explanation is that the current representation of chemical compounds is not adequate. The usual representation of chemical compounds is using *structure-activity relationship (SAR)* descriptors coming from commercial tools from drug design such as CODESSA [**?**], TSAR (Oxford molecular products, www.accelrys.com/chem/), DRAGON (www.disat.inimib.it/chm/Dragon.htm). By means of these descriptors a natural way to represent a chemical compound is as a set of attribute value pairs (propositional representation). A challenge on Predictive Toxicology held in 2001 [**?**] was focused on ML techniques and most contributions proposed a relational representation based on SAR descriptors and used inductive techniques for solving the classification task. Moreover the relational representation and the ILP techniques also allow the representation and use of chemical background knowledge.

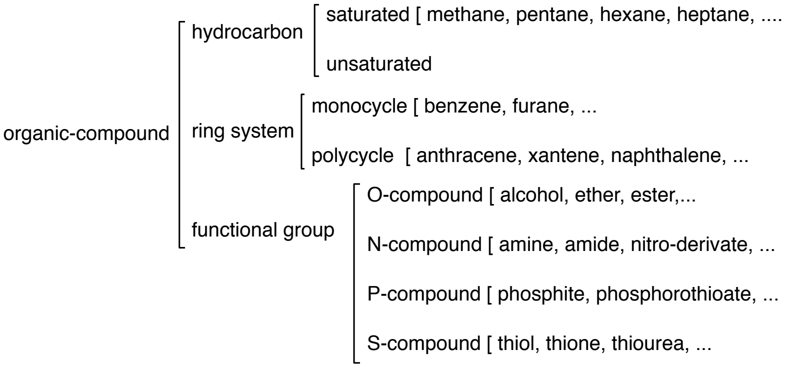
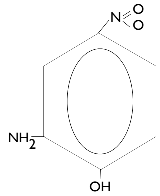

organic-compound
hydrocarbon
saturated [methane, pentane, hexane, heptane, ...
unsaturated
ring system
monocycle [benzene, furane, ...
polycycle [anthracene, xantene, naphthalene, ...
functional group
O-compound [alcohol, ether, ester,...
N-compound [amine, amide, nitro-derivate, ...
P-compound [phosphite, phosphorothioate, ...
S-compound [thiol, thione, thiourea, ...

Fig. 2. Partial view of the chemical ontology

Other approaches to represent chemical compounds have been proposed. For instance [**?**,**?**,**?**] 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. Particularly interesting are SUBDUE [**?**] and SMILES [**?**] that follow this approach. A completely different approach was introduced in [**?**] where the compounds are organized according to their active centers (chemically identified with weak bonds).

The representation of chemical compounds we propose is based on the chemical terminology, i.e the IUPAC (*International Union of Pure and Applied Chemistry*) nomenclature (www.chem.qmul.ac.uk/iupac/). Also we take into account the experience of previous research (specially the works in [**?**,**?**,**?**]) since we represent a 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 (like some relational representations based on SAR do) 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 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 **??** 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 the molecular structure of chemical compounds. In fact, the name of a molecule using the standard nomenclature, provides chemists 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 is located in a central position; however, there is



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

Fig. 3. Representation of the *2-amino-4-nitrophenol*, with feature terms.

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 a main group or a radical depending on their position or relevance on the molecule, i.e. the benzene may be the main group in one compound and a radical in some other compounds.

Figure **??** shows the representation of the chemical compound, *2-amino-4-nitrophenol*, using feature terms [**?**]. The *2-amino-4-nitrophenol* has a benzene as its 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 directly comes 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.

6.3 Assessing carcinogenic activity to chemical compounds

Inductive learning techniques applied to the Predictive Toxicology try to extract general rules describing the cases in each class. These kinds of techniques have some difficulties in dealing with domains, like toxicology, where entities are subject to high variability. The goal of predictive toxicology is to develop models able to predict whether or not a chemical compound is carcinogen. The construction of these models using inductive learning methods takes into account the toxicity observed in some molecules to extract theories about the carcinogenecity on families of molecules. Early systems focused on predictive toxicology were DEREK [?] and CASE [?]. PROGOL [?] was the first ILP program used to induce SAR models. PROGOL's results were very encouraging since the final rules were more understandable than those obtained using the other methods.

Lazy learning techniques, on the other hand, are based on the retrieval of a set of solved problems similar to a specific problem. Several authors use the concept of similarity between chemical compounds: HazardExpert [?] is an expert system that evaluates the similarity of two molecules based on the number of common substructures; Sello [?] also uses the concept of similarity but the representation of the compounds is based on the energy of the molecules.

We conducted a series of experiments focused on the use of lazy learning techniques for classifying chemical compounds. In [?] we report the results of using the *k-nearest neighbor* (*k-NN*) algorithm with Shaud as similarity measure. Results of these experiments show that our approach is comparable to results produced by inductive methods in terms of both accuracy and ROC analysis. We want to remark that our approach only handles information about the molecular structure of the chemical compounds whereas the other approaches use more information (SAR descriptors).

Clearly, in Predictive Toxicology the classification of a particular chemical compound its important, nevertheless, experts are also interested in finding a general model of carcinogenesis. In this sense, we think the use of C-LID can satisfy these expert's interests. On one hand it can classify a chemical compound and also justify this classification; on the other hand, it can produce general knowledge about carcinogenesis thanks to the similitude term. Thus, we conducted some experiments with a main goal: to build a (partial) model of carcinogenesis using C-LID. These experiments are composed of two steps: 1) using LID with the leave-one-out in order to generate similitude terms for classifying the cases; and 2) select a subset of these similitude terms to build a partial carcinogenesis model. We consider that the model is partial because given a class, we can only assure that the similitude term generated by LID is satisfied by a subset of compounds of that class. The idea behind these experiments comes from the observation of the similitude terms given by LID to justify the classification of a chemical compound (step 1). By analyzing these similitude terms we note that some of them are given several times and that they are good descriptions of carcinogen (or non-carcinogen) compound. This means that there are some features (those included in the similitude terms) that are good descriptors of a class since they are often used to classify compounds as belonging to that class. consequently, they can be used by C-LID as general domain knowledge for assessing the carcinogenic activity of new chemical compounds (step 2).

In [?] we report some domain knowledge contained in the carcinogenesis model built thanks to the similitude terms of LID and that have been successfully used by C-LID for predecting the carcinogenesis of unseen chemical compounds. 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 (www.expertnetwork.com/med2.htm) describes some experiments confirming the toxicity of chlorinated hydrocarbons. Nevertheless, there are other patterns whose positive carcinogenic activity is not clearly reported in the literature. An example of this are the chemical compounds with the polycycle *anthracene*. An analysis of the chemical compounds with *anthracene* included in the data set of the NTP shows that they are positive in rats, nevertheless there are no laboratory experiments confirming this result, even there are reports explaining that *anthracene* is a molecule with a high tendency to make associations with other molecules and these associations could easily be carcinogenic. Other patterns included in the partial domain knowledge built by C-LID concern the carcinogenecity of chemical compounds containing *epoxydes*, *bromine* and long *carbon* chains. Some of these patterns are confirmed by the experimental knowledge, therefore they could be directly included as rules of a model. Nevertheless, because C-LID is lazy it can include in the model some knowledge that is not general enough to be induced but that is true for a known subset of compounds (like the case of the long chains of *carbons*).

6.4 The explanation scheme

A common situation in toxicology is that chemical compounds with similar molecular structure have different carcinogenic activity. Therefore, the use of lazy learning methods, based on the similarity among structures of the compounds, can produce non univocal classifications. That is to say, a chemical compound can share some structural aspects with carcinogen compounds but it can also also share other aspects with no carcinogen compounds. Let C be the set of chemical compounds that have been considered by a lazy learning method (say k-NN) as the most similar to a compound c. Let $C^+ \subseteq C$ be the subset of positive (carcinogen) compounds and $C^- \subseteq C$ the subset of negative (non-carcinogenic) compounds ($C = C^+ \cup C^-$). In such situation the final prediction about the carcinogenic activity of c is taken using the *majority rule*, i.e. the compound is classified as belonging to the same class as the majority of the compounds in C. The application of the majority rule seems appropriate when there is a "clear" majority of compounds belonging to one of the classes. Nevertheless this is not always the case, consequently the result has to be explained to the user. In fact, more important than the classification should be to show the user the similitude that the compound has with compounds of both classes. In other words, if the user can analyze by themself the reasons that explain the classification of the compound in each one of the classes, then s/he could decide the final classification of the compound.

Let us illustrate the complete explanation scheme with an example. The right hand side of Fig. **??** shows a chemical compound, namely *C-356*, for which we want to assess its carcinogenicity for male rats. The set C of retrieved cases (*retrieval set*), formed by five chemical compounds considered the most similar to *C-356* is also shown on the right hand side of Fig. **??**. The set C is divided

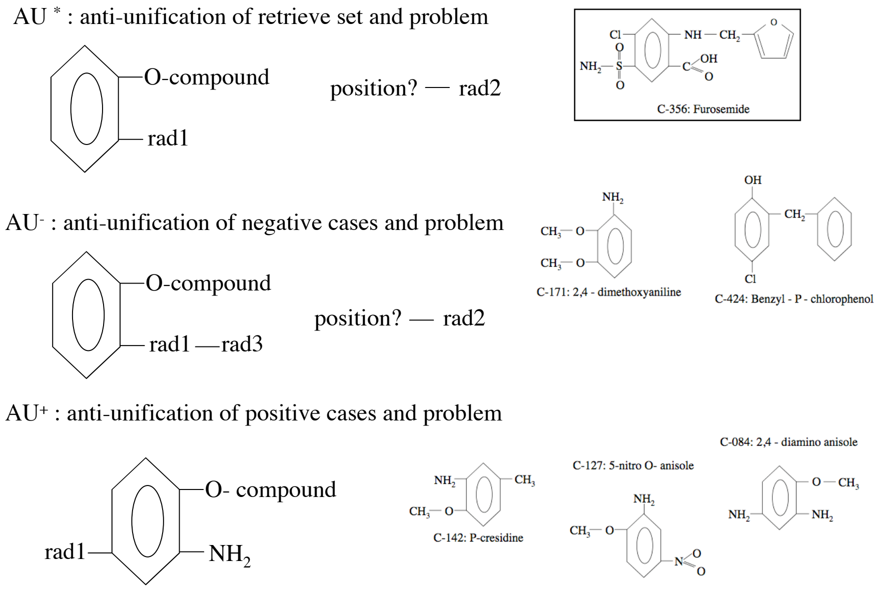

AU * : anti-unification of retrieve set and problem
O-compound
rad1
position? — rad2
Cl
NH — CH2
O
NH2
S
O
OH
C
O
C-356: Furosemide
AU- : anti-unification of negative cases and problem
O-compound
rad1 —rad3
position? — rad2
NH2
CH3 — O
CH3 — O
C-171: 2,4 - dimethoxyaniline
OH
CH2
Cl
C-424: Benzyl - P - chlorophenol
AU+ : anti-unification of positive cases and problem
O- compound
rad1
NH2
NH2
CH3
CH3 — O
C-142: P-cresidine
C-127: 5-nitro O- anisole
NH2
CH3 — O
N
O
O
C-084: 2,4 - diamino anisole
O — CH3
NH2
NH2

Fig. 4. AU^* is the chemical structure common to all the compounds in Fig. **??**. AU^- is the chemical structure common to *C-356* and the negative compounds (i.e. *C-424* and *C-171*). AU^+ is the chemical structure common to *C-356* and the positive compounds (i.e. *C-084, C-127* and *C-142*).

in $C^- = \{$*C-424, C-171*$\}$ and $C^+ = \{$*C-084, C-127, C-142*$\}$ according to the carcinogenic activity of the compounds.

Following our approach, the explanation scheme (left hand side of Fig. **??**) for chemical compound *C-356* is as follows:

- The description AU^* shows that *C-356* and the compounds in C have in common that they are all benzenes with at least three radicals: one of these radicals is a functional group derived from the oxygen (i.e. an alcohol, an ether or an acid) called *O-compound* in the figure; another radical (called *rad1* in the figure) is in the position next to the functional group (chemically this means that both radicals are in disposition *ortho*). Finally, there is a third radical (called *rad2* in the figure) that is in no specific position.
- The description AU^- shows that *C-356* and the chemical compounds in C^- have in common that they are benzenes with three radicals: one radical derived from an oxygen (*O-compound*), a radical *rad1* with another radical (*rad3* in the figure) in position *ortho* with the *O-compound*, and finally a third radical (*rad2*) with no specific position.
- The description AU^+ shows that *C-356* and the chemical compounds in C^+ have in common that they are benzenes with three radicals: one of the rad-

icals is derived from an oxygen (*O-compound*), another radical is an *amine* (NH_2) in position *ortho* with the *O-compound*, and a third radical (*rad1*) is at distance 3 of the *O-compound* (chemically this means that both radicals are in disposition *para*).

Using the majority rule, the compound *C-356* will be classified as positive. The explanation scheme explicitly shows the user the similarities among the compound and the retrieved compounds (with known activity). Nevertheless, the user can also easily compare all the descriptions and analyze the differences between them. Thus, from AU^- and AU^+ the user is able to observe that the presence of the *amine* (NH_2) may hypothetically be a key factor in the classification of a compound as positive for carcinogenesis. Once the symbolic similarity description gives a key factor (such as the *amine* in our example), the user can proceed to search the available literature for any empirical confirmation of this hypothesis. In this particular example, a cursory search in the Internet has shown that there is empirical evidence supporting the hypothesis of *amine* presence in aromatic groups (i.e. benzene) being correlated with carcinogenicity [**?**], [**?**].

Notice that a similar explanation scheme could be proposed when using LID instead of the k-NN algorithm for solving the classification task. In such case the similitude term takes the role of the description AU^*, i.e. the anti-unification of the k cases similar to a problem p. The difference among both AU^* and the similitude term is that the former contains *all* that is shared by the k cases whereas the later contains only the relevant features used for classifying p. A detailed description of the use of the similitude term as explanation can be found in [**?**].

7 Conclusions

Lazy learning methods can build local approximations of concepts. In this paper we analyzed how these approximations can be used in a CBR method. In particular, we analyzed the usages as 1) symbolic similitude among a set of cases; 2) partial model of the domain, when they are stored to be used for solving new problems; and 3) as explanations, since they can be interpreted as the justification of the classification given by the system.

We show an example of these usages of generalization for solving the Predictive Toxicology task. Moreover, the generalization can also be used in the same terms in the context of multi-agent systems as is proposed by Ontañón and Plaza in [**?**]. In fact, these authors propose to use the generalizations build by a CBR method as a means for the communication of knowledge among the agents.

Acknowledgments

Author thanks Enric Plaza, Ramon López de Mántaras and Josep Lluis Arcos for all the helpful discussions during the elaboration of this paper. This work has been supported by the MCYT-FEDER Project MID-CBR (TIN2006-15140-C03-01)

References

1. D. W. Aha, D. Kibler, and M. K. Albert. Instance-based learning algorithms. *Machine Learning*, 6(1):37–66, 1991.
2. S. Ambs and H. G. Neumann. Acute and chronic toxicity of aromatic amines studied in the isolated perfused rat liver. *Toxicol. Applied Pharmacol.*, 139:186–194, 1996.
3. E. Armengol. Discovering plausible explanations of carcinogenecity in chemical compounds. In *Int. Conf. on Machine Learning and Data Mining MLDM-07*, number to appear in Lecture Notes in Artificial Intelligence. Springer-Verlag, 2007.
4. E. Armengol and E. Plaza. Bottom-up induction of feature terms. *Machine Learning*, 41(1):259–294, 2000.
5. E. Armengol and E. Plaza. Lazy induction of descriptions for relational case-based learning. In *Machine Learning: ECML-2001*, number 2167 in Lecture Notes in Artificial Intelligence, pages 13–24. Springer-Verlag, 2001.
6. E. Armengol and E. Plaza. Discovery of toxicological patterns with lazy learning. In V. Palade, R.J. Howlett, and L. Jain, editors, *KES-2003*, number 2774 in Lecture Notes in Artificial Intelligence, pages 919–926. Springer, 2003.
7. E. Armengol and E. Plaza. Relational case-based reasoning for carcinogenic activity prediction. *Artif. Intell. Rev.*, 20(1-2):121–141, 2003.
8. E. Armengol and E. Plaza. Remembering similitude terms in case-based reasoning. In *3rd Int. Conf. on Machine Learning and Data Mining MLDM-03*, number 2734 in Lecture Notes in Artificial Intelligence, pages 121–130. Springer-Verlag, 2003.
9. E. Armengol and E. Plaza. Symbolic explanation of similarities in case-based reasoning. *Computing and informatics*, 25(2-3):153–171, 2006.
10. R. Bergmann and A. Stahl. Similarity measures for object-oriented case representations. In *Proc. European Workshop on Case-Based Reasoning, EWCBR-98*, Lecture Notes in Artificial Intelligence, pages 8–13. Springer Verlag, 1998.
11. V. Blinova, D. A. Bobryinin, S. O. Kuznetsov, and E. S. Pankratova. Toxicology analysis by means of simple jsm method. In *Procs. of the Predictive Toxicology Challenge Workshop, Freiburg, Germany, 2001.*, 2001.
12. R. Chittimoori, L. Holder, and D. Cook. Applying the subdue substructure discovery system to the chemical toxicity domain. In *Procs. of the Twelfth International Florida AI Research Society Conference, 1999*, pages 90–94, 1999.
13. F. Darvas, A. Papp, A. Allerdyce, E. Benfenati, and G. Gini et al. Overview of different ai approaches combined with a deductive logic-based expert system for predicting chemical toxicity. In G.C. Gini and A.R. Katrizky, editors, *Predictive Toxicology of Chemicals: Experiences and Impacts of AI Tools*, pages 94–99. AAAI Press, 1999.
14. L. Dehaspe, H. Toivonen, and R. D. King. Finding frequent substructures in chemical compounds. In R. Agrawal, P. Stolorz, and G. Piatetsky-Shapiro, editors, *4th International Conference on Knowledge Discovery and Data Mining*, pages 30–36. AAAI Press., 1998.
15. M. Deshpande and G. Karypis. Automated approaches for classifying structures. In *Proc. of the 2nd Workshop on Data Mining in Bioinformatics.*, 2002.
16. W. Emde and D. Wettschereck. Relational instance based learning. In L. Saitta, editor, *Machine Learning - Procs. 13th International Conference on Machine Learning*, pages 122–130. Morgan Kaufmann Publishers, 1996.
17. J. H. Friedman, R. Kohavi, and Y. Yun. Lazy decision trees. In *AAAI/IAAI, Vol. 1*, pages 717–724, 1996.

18. P. Gervás and K. M. Gupta. Explanations in cbr. In *Procs of the ECCBR 2004 Workshops*, pages 85–176. Dep. de Sistemas Informáticos y Programación, Universidad Complutense de Madrid, Madrid, Spain. TR 142-04, 2004.
19. J. A. Gonzalez, L. B. Holder, and D. J. Cook. Graph based concept learning. In *AAAI/IAAI*, page 1072, 2000.
20. C. Helma, E. Gottmann, and S. Kramer. Knowledge discovery and data mining in toxicology. *Statistical Methods in Medical Research*, 9:329–358, 2000.
21. C. Helma and S. Kramer. A survey of the predictive toxicology challenge 2000-2001. *Bioinformatics*, page in press, 2003.
22. A.R Katritzky, R. Petrukhin, H. Yang, and M. Karelson. *CODESSA PRO. User's manual*. University of Florida, 2002.
23. G. Klopman. 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, 1984.
24. Janet Kolodner. *Case-based reasoning*. Morgan Kaufmann, 1993.
25. K. McCarthy, J. Reilly, L. McGinty, and B. Smyth. Thinking positively - explanatory feedback for conversational recommender systems. In *Procs of the ECCBR 2004 Workshops. TR 142-04*, pages 115–124. Dep. de Sistemas Informáticos y Programación, Universidad Complutense de Madrid, Madrid, Spain, 2004.
26. D. McSherry. Explanation in recommendation systems. In *Procs. of the ECCBR 2004 Workshops. TR 142-04*, pages 125–134. Dep. de Sistemas Informáticos y Programación, Univ. Complutense de Madrid, Madrid, Spain, 2004.
27. T. M. Mitchell. *Machine Learning*. McGraw-Hill International Editions. Computer Science Series, 1997.
28. T. M. Mitchell, R. M. Keller, and S. T. Kedar-Cabelli. Explanation-based learning: A unifying view. *Machine Learning*, 1(1):47–80, 1986.
29. S. Ontañón and E. Plaza. Justification-based multiagent learning. In *Procs. 20th ICML*, pages 576–583. Morgan Kaufmann, 2003.
30. J. R. Quinlan. Induction of decision trees. *Mach. Learn.*, 1(1):81–106, 1986.
31. D.M. Sanderson and C.G. Earnshaw. Computer prediction of possible toxic action from chemical structure: the derek system. *Human and Experimental Toxicology*, 10:261–273, 1991.
32. G. Sello. Similarity, diversity and the comparison of molecular structures. In G.C. Gini and A.R. Katrizky, editors, *Predictive Toxicology of Chemicals: Experiences and Impacts of AI Tools*, pages 36–39. AAAI Press, 1999.
33. R. U. Sorensen. Allergenicity and toxicity of amines in foods. In *Procs. of the IFT 2001 Annual Meeting, New Orleans, Louisiana*, 2001.
34. A. Srinivasan, S. Muggleton, R.D. King, and M.J. Sternberg. Mutagenesis: Ilp experiments in a non-determinate biological domain. In *Procs. of the Fourth Inductive Logic Programming Workshop*, 1994.
35. D.J. Weininger. Smiles a chemical language and information system. *J. Chem. Inf. Comput. Sci.*, 28(1):31–36, 1988.