Lazy Induction of Descriptions for Relational Case-based Learning

Eva Armengol EVA@IIIA.CSIC.ES
Enric Plaza ENRIC@IIIA.CSIC.ES
IIIA - Artificial Intelligence Research Institute,
CSIC - Spanish Council for Scientific Research,
Campus UAB, 08193 Bellaterra, Catalonia (Spain).

Abstract

Reasoning and learning from cases are based on the concept of similarity often estimated by a distance. This paper presents LID, a learning technique adequate for domains where cases that are best represented by relations among entities. LID is able to 1) define a *similitude term*, a symbolic description of what is shared between a problem and precedent cases; and 2) assess the importance of the relations involved in a similitude term with respect to the purpose of correctly classifying the problem. The paper describes two application domains of relational case-based learning with LID: marine sponges identification and diabetes risk assessment.

1. Introduction

Reasoning and learning from cases is based on the concept of similarity. Often similarity is estimated by a distance (a metric) or a pseudo-metric. In addition to this, an assessment of which properties are "important" or "relevant" in the similarity is needed. This approach proceeds by a pairwise similarity comparison of a *problem* with every *precedent case* available in a case base; then one case (or k cases) with biggest (bigger) similarity is (are) selected. This process is called the *retrieval* phase in Case-based Reasoning (CBR), and also plays a pivotal role in lazy learning techniques like Instance-based Learning (IBL) and k-nearest neighbor. In classification tasks, the solution class of the *problem* is inferred from the solution class of the precedent case(s) selected.

However, distance-based approaches to case retrieval are mainly used for propositional cases, i.e. cases represented as attribute-value vectors. We are interested in this paper in learning tasks where cases are best represented in a scheme that uses relations among entities. We will call this setting *relational case-based learning*. One option to achieve case-based learning in a relational setting is to adapt the process of pairwise similarity comparison by defining a distance that works upon relational instances. This approach is taken in "relational IBL" (Bohnebeck et al., 1998) where cases are represented as collections of Horn clauses (see related work on section §6).

The approach taken in this paper is different from pairwise similarity comparison based on metrics or pseudo-metrics. Basically, in our approach, similarity between two cases is understood as that which they "share". In addition, we need to be able to evaluate if what they share is what is important (or to which degree they share what is important). This paper presents a technique called LID (for Lazy Induction of Descriptions) for relational case-based learning. LID is based on two main notions: 1) similarity is constructed as a symbolic description of what is shared between precedent cases and a specific *problem* to be classified, and 2) there is some assessment function to help the system decide which relations among entities are "important" or "relevant" to be shared with the precedent cases.

The representation formalism used to represent cases, feature terms, is presented in §2. In feature terms, entities are typed by *sorts* and relations among entities are represented by *features*. Then, in §3, we introduce the framework of relational case-based learning and the main building blocks that we will use in §4 to describe the LID method. In §5 two application domains of relational case-based learning with LID are described: marine sponges identification and diabetes risk assessment. The paper closes with the sections on related work and conclusions.

2. Representation of the cases

LID handles cases represented as feature terms. *Feature Terms* (also called feature structures or ψ-terms)

are a generalization of first order terms. The difference between feature terms and first order terms is the following: a first order term, e.g. $f(x, y, g(x, y))$ can be formally described as a tree and a fixed tree-traversal order. In other words, parameters are identified by position. The intuition behind a feature term is that it can be described as a labelled graph i.e. parameters are identified by name. A formal definition of feature terms is the following:

Given a signature $\Sigma =< S, \mathcal{F}, \leq >$ (where S is a set of sort symbols that includes $\perp$; $\mathcal{F}$ is a set of feature symbols; and $\leq$ is a decidable partial order on S such that $\perp$ is the least element) and a set ϑ of variables, we define *feature terms* as an expression of the form:

$$\psi ::= X : s[f_1 \doteq \Psi_1 \ldots f_n \doteq \Psi_n] \qquad (1)$$

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

Sorts have an informational order relation ($\leq$) among them, where $\psi \leq \psi'$ means that ψ has less information than ψ' or equivalently that ψ is more general than ψ'. The minimal element ($\perp$) is called *any* and it represents the minimum information. When a feature has unknown value it is represented as having the value *any*. All other sorts are more specific that *any*.

A *path* $\pi(X, f_i)$ is defined as a sequence of features going from the variable X to the feature f_i.

There is a *path equality* when two paths point to the same value. Path equality is equivalent to variable equality in first order terms. For instance, in Figure 1a, the root of the feature term is X_1 of sort *person*. X_1 has three features: *address*, *spouse* and *interests*. The feature *address* has a value of sort address that has, in turn, two features: *street* and *city*. The feature *spouse* has a value (Y_1) of sort person that has only the feature *address*. Notice that the address of Y_1 is the same that the address of X_1, therefore there is path equality. Finally, the value of the feature *interests* of X_1 is a feature term without features.

The *depth* of a feature f in a feature term ψ with root X is the number of features that compose the path from the root X to f, including f, with no repeated nodes.

Given a particular maximum feature depth k, a *leaf feature* of a feature term is a feature f_i such that either 1) the depth of f_i is k or 2) the value of f_i is a term without features.

a)

$$X_1 = \begin{bmatrix} \underline{person} \\ \text{address} \doteq A_1 = \begin{bmatrix} \underline{address} \\ \text{street} \doteq Main \\ \text{city} \doteq London \end{bmatrix} \\ \text{spouse} \doteq Y_1 = \begin{bmatrix} \underline{person} \\ \text{address} \doteq A_1 \end{bmatrix} \\ \text{interests} \doteq Agents \end{bmatrix}$$

b)

$$X_2 = \begin{bmatrix} \underline{person} \\ \text{address} \doteq A_2 \\ \text{spouse} \doteq Z = \begin{bmatrix} \underline{person} \\ \text{address} \doteq A_2 \end{bmatrix} \end{bmatrix}$$

c)

$$X_3 = \begin{bmatrix} \underline{person} \\ \text{address} \doteq A_3 \\ \text{spouse} \doteq P_1 = \begin{bmatrix} \underline{person} \\ \text{address} \doteq A_3 \end{bmatrix} \\ \text{children} \doteq \begin{matrix} P_2 \\ P_3 \end{matrix} \end{bmatrix}$$

d)

$$X_4 = \begin{bmatrix} \underline{person} \\ \text{address} \doteq A_4 \\ \text{spouse} \doteq Y_4 = \begin{bmatrix} \underline{person} \\ \text{address} \doteq A_5 \end{bmatrix} \end{bmatrix}$$

Figure 1. Examples of four feature terms.

The semantic interpretation of feature terms brings an ordering relation among feature terms that we call *subsumption*. Intuitively, a feature term ψ subsumes another feature term ψ' ($\psi \sqsubseteq \psi'$) when all information in ψ is also contained in ψ'. In other words, a feature term ψ subsumes other feature term ψ' when the following conditions are satisfied: 1) the sort of $root(\psi')$ is either the same or a subsort of $root(\psi)$, 2) if F_ψ is the set of features of ψ and $F_{\psi'}$ is the set of features of ψ' then $F_\psi \subseteq F_{\psi'}$ and 3) the values of the features in F_ψ and $F_{\psi'}$ satisfy the two conditions above.

For instance, the feature term X_2 in Figure 1b represents a person that is married with a person Z and both live at the same address (A_2). Because all the features in X_2 are also present in X_1, we say that X_2 subsumes X_1. Instead, the feature term X_3 in Figure 1c that represents a person married with a person P_1 that live in the same address (A_3) and having children does not subsume X_1 (since X_1 has not the feature *children*).

As it can be seen in the example, path equality has to be preserved by the subsumption, i.e. when $\psi \sqsubseteq \psi'$, all path equalities in ψ also occur in ψ'. Notice that the feature term in Figure 1d representing X_4 that is married with Y_4 that lives in a different address is not subsumed by the feature term X_2 because the path equality between the addresses is not present.

In (Armengol & Plaza, 2000) can be found a more detailed explanation about the feature terms and the subsumption relation.

3. Relational Case-based Learning

There are three aspects that we need to define in order to perform CBR on relational cases: 1) to define a *case* from a constellation of relations, 2) to define a way to assess similarity between cases, and 3) to establish a degree of importance for relations involved in the similarity. These three aspects are explained in the following subsections.

3.1 Similarity between cases

A *case* is a term defined (in feature terms) by two parameters: a *root sort* and a *depth*. That is to say, assuming a "case base" expressed as a collection of feature terms, a case is a feature term whose root node is subsumed by the *root sort* and whose depth is at most *depth*. Examples of case specification are $case[\mathsf{root\text{-}sort} \doteq patient, \mathsf{depth} \doteq 5]$ in the diabetes domain and $case[\mathsf{root\text{-}sort} \doteq sponge, \mathsf{depth} \doteq 4]$ in the marine sponges domain (see §5).

The estimation of the similitude between cases is one of the key issues of the lazy learning algorithms. Those techniques (such as IBL (Aha, 1997) and k-nearest neighbor) that use cases represented as attribute-value vectors define the *similitude* of two cases by means of a distance measure. The LID method that we present in section §4 uses a symbolic estimation of the similitude between cases. The intuition of a symbolic similitude is that of a description containing the features shared by the two cases. In feature terms we can formalize this intuition using the concept of subsumption.

We say that a term s is the *similitude term* of two cases c_1 and c_2 if and only if $s \sqsubseteq c_1$ and $s \sqsubseteq c_2$. In other words, the similitude term of two cases subsumes both cases.

In this framework, the task of similarity assessment is a search process over the space of similarity descriptions determined by the subsumption relation.

In general, however, we are not interested in a mere quantitative measure that tries to optimize the number of shared aspects. In CBR systems built for a specific task, we want two cases to be similar in those aspects that are relevant (important) for the task at hand. Therefore, we need a measure of "goodness" of a specific similitude term. Moreover, we want that this "goodness" measure satisfies two conditions: 1) it helps us to search into the space of similitude terms, and 2) it can be extracted from the examples of the case base.

The next subsection explains a technique to assess the importance of a feature using the examples present in the case base. This technique is a heuristic measure based on the López de Mántaras (RLM) distance. Then, §4 explains how LID incrementally builds a similitude term based on the RLM heuristic.

3.2 Relevance of attributes

Given a new example for classifying, the goal is to determine those features that are most relevant for the task. The relevance of a feature is heuristically determined using the RLM distance (López de Mántaras, 1991). The RLM distance assesses how similar are two partitions (in the sense that the lesser the distance the more similar they are). Let $F_l = \{f_1 \ldots f_n\}$ be the set of features of an example e that are leaf features (given a maximum depth k). Each feature $f_i \in F_l$ induces a partition P_i over the case base according to the values that f_i can take in the cases. Basically, the partition $P_i = \{S_{i1} \ldots S_{ik}\}$ induced by a feature f_i is built according to the number of different values that f_i takes in the cases in the case base. Thus, examples belonging to a set S_{ik} of the partition P_i have the same value v_k in the feature f_i.

The *correct partition* is a partition $P_c = \{C_1 \ldots C_k\}$ where all the cases contained into a set C_i belong to the same solution class.

For each partition P_i induced by a feature f_i, LID computes the RLM distance to the correct partition P_c. Formally, given two partitions P_i and P_c of the case base B, the RLM distance between them is computed as follows:

$$RLM(P_i, P_c) = 2 - \frac{I(P_i) + I(P_c)}{I(P_i \cap P_c)}$$

where

$$I(P_i) = -\sum_{j=1}^{n} p_j log_2 p_j; \quad p_j = \frac{|B \cap S_{ij}|}{|B|}$$

$$I(P_c) = -\sum_{k=1}^{m} p_k log_2 p_k; \quad p_k = \frac{|B \cap C_k|}{|B|}$$

$$I(P_i \cap P_c) = -\sum_{j=1}^{n}\sum_{k=1}^{m} p_{jk} log_2 p_{jk}; \; p_{jk} = \frac{|B \cap C_k \cap S_{ij}|}{|B|}$$

where $I(P_i)$ measures the information contained in the partition P_i; n is the number of possible values of the feature inducing P_i; p_i (p_k) is the probability of occurrence of class S_{ij} (C_k) i.e. the proportion of examples in B that belong to S_{ij} (C_k); $m = Card(P_c)$, i.e. the number of solution classes; $I(P_i \cap P_c)$ is the mutual information of two partitions; and p_{jk} is the probability of the intersection $C_j \cap S_{ik}$, i.e. the proportion of examples in B that belong to C_j and to S_{ik}.

Let P_c be the correct partition (i.e. the partition that correctly classifies the examples), 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 $RLM(P_i, P_c) < RLM(P_j, P_c)$. In other words, when 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 current problem.

4. LID

In this section we introduce a new method called LID (Lazy Induction of Descriptions). LID uses a case base containing classified cases. The goal of LID is to classify a problem as belonging to one of the solution classes. The main idea of LID is to determine which are the more relevant features of the problem and to search in the case base for cases sharing these relevant features. The problem is classified when LID founds a set of relevant features shared by a subset of cases belonging all of them to the same solution class. Then, the problem is classified into that solution class.

Given a case base B containing cases classified into one of the solution classes $C = \{C_1 \ldots C_k\}$ and a problem p, the goal of LID is to classify p as belonging to one of the solution classes. The problem and the cases in the case base are represented feature terms (§2).

We call *discriminatory set* the set $S_D = \{b_i \in B | D \sqsubseteq b_i\}$ that contains the cases of B subsumed by the similitude term D.

The main steps of the LID algorithm are shown in Figure 2. In the first call $LID(S_D, p, D, C)$ parameter S_D is initialized to B (the whole case base) and parameter D can be initialized to *any* or to a value $D = D^0$ (where $D^0 \neq any$) based on domain knowledge we may have (see an example in §5.2).

```
Function LID (S_D, p, D, C)
    if stopping-condition(S_D)
        then return class(S_D)
        else f_d := Select-leaf (p, S_D, C)
             D' := Add-path(π(root(p), f_d), D)
             S_D' := Discriminatory-set (D, S_D)
             LID (S_D', p, D', C)
    end-if
end-function
```

Figure 2. The LID algorithm. D is the similitude term, S_D is the discriminatory set of D, C is the set of solution classes, $class(S_D)$ is the class $C_i \in C$ to which all elements in S_D belong.

The specialization of a similitude term D is achieved by adding features to it. In principle, any of the features used to describe the examples could be a good candidate. Nevertheless, LID uses two biases to obtain the set F_l of features candidate to specialize the current similitude term D. First, of all possible features in $\mathcal{F}$, LID will consider only those features present in the problem p to be classified. As a consequence, any feature that is not present in p will not be considered as candidate to specialize D. The second bias is to consider as candidates for specializing D only those features that are leaf features of p (see section 2).

The next step of LID is the selection of a leaf feature $f_d \in F_l$ to specialize the similitude term D. Let F_l be the set of leaf features candidates to specialize D. Selecting the most discriminatory leaf feature in the set F_l is heuristically done using the RLM distance which is explained in section 3.2. Let us call the most discriminatory feature f_d.

The feature f_d is the leaf feature of path $\pi(root(p), f_d)$ in problem p. The specialization step of LID defines a new similitude term D' by adding to the current similitude term D the sequence of features specified by $\pi(root(p), f_d)$. After this addition D' has a new path $\pi(root(D'), f_d)$ with all the features in the path taking the same value that they take in p. After adding the path π to D, the new similitude term $D' = D + \pi$ subsumes a subset of cases in S_D, namely the discriminatory set $S_{D'}$ (the subset of cases subsumed by D').

Next, LID is recursively called with the discriminatory set $S_{D'}$ and the similitude term D'. The recursive call of LID has $S_{D'}$ as first parameter (instead of S_D) because the cases that are not subsumed by D' will not be subsumed by further specialization. The process of specialization reduces the discriminatory set $S_D^n \subseteq S_D^{n-1} \subseteq \ldots \subseteq S_D^0$ at each step.

The stopping condition of LID, given the current simil-

itude term D, is that all the cases in its discriminatory set S_D belong to only one solution class $C_k \in C$. The similitude term D can be viewed as a *partial* description of C_k because it subsumes (in general) only a subset of the elements in C_k. LID gives D as an explanation of classifying p in C_k and S_D as the cases justifying that result.

5. Experiments

In this section we describe some applications where LID has been applied. In the first one LID is used to classify marine sponges and in the second one it is used to determine the risk of complications of diabetic patients.

5.1 Classification of marine sponges

Marine sponges are relatively little studied and most of the existing species are not yet fully described. Main problems in the identification are due to the morphological plasticity of the species, to the incomplete knowledge of many of their biological and cytological features and to the frequent description of new taxa. Moreover, there is no agreement around the species delimitation since it is not clear how the different taxa can be characterized.

The application of LID to this domain allows the classification of new specimens based on their similarity to specimens clearly classified in some taxa. This similarity, in turn, is based on the relevant features of the specimen that has to be classified.

LID uses a case base containing the description of 532 sponges belonging to 9 orders of the *Demospongiae* class: $C = \{$*astrophorida, dictyoceratida, halichondrida, hadromerida, haplosclerida, axinellida, poecilosclerida, dendroceratida, homosclerophorida*$\}$. We have applied LID to the identification of the order of new specimens. In the following we describe the complete process of LID to classify the specimen *s252* of Figure 3.

Following the algorithm of Figure 2, LID begins with the similitude term $D^0 = any$, the discriminatory set $S_D^0 = B$ and $p = s252$. Because the cases in S_D^0 belong to several orders, the similitude term D^0 has to be specialized. The first step of the specialization is to select the most discriminatory leaf of *s252*. The leaf features of *s252* (see Figure 3) are the set $F_l = \{$*body-size, consistency, grow, form, peduncle, hollow, osc, fixation, location, chemical, architecture, smooth-form, acanthose, ornamentation, ectosome*$\}$

LID uses the RLM distance to select the most discriminatory feature. Therefore, for each one of the leaf features $f_i \in F_l$, LID induces a partition of S_D^0 according to the values of f_i and measures the distance to the correct partition. Specifically, LID takes as the most discriminatory feature the leaf *smooth-form* that induces a partition with minimum distance to the correct partition (the classification of sponges according to their *order*).

The next step of LID (Figure 2) is to create a new similarity term D^1 by adding the path π(*sponge*, *smooth-form)* to the current similarity term D^0 (see Figure 4). Notice that the value of the leaf feature *smooth-form* in D^1 is the set $\{tylostyle, style, subtylostyle\}$ as in the specimen *s252*. LID is recursively called with the new discriminatory set S_D^1 and the new similitude term D^1.

$$s252 = \begin{bmatrix} \underline{sponge} \\ \text{external-features} \doteq \begin{bmatrix} \underline{external\text{-}features} \\ \text{body-size} \doteq small \\ \text{consistency} \doteq soft \\ \text{growing} \doteq \begin{bmatrix} \underline{growing} \\ \text{grow} \doteq \begin{matrix} erect \\ encrusting \end{matrix} \\ \text{form} \doteq branching \\ \text{peduncle} \doteq no \end{bmatrix} \\ \text{hollow} \doteq no \\ \text{osc} \doteq no \end{bmatrix} \\ \text{ecological-features} \doteq \begin{bmatrix} \underline{ecological\text{-}features} \\ \text{fixation} \doteq totally \\ \text{location} \doteq mediterranean \end{bmatrix} \\ \text{spiculate-skeleton} \doteq \begin{bmatrix} \underline{spiculate\text{-}skeleton} \\ \text{chemical} \doteq silica \\ \text{architecture} \doteq plumose \\ \text{megascleres} \doteq \begin{bmatrix} \underline{megascleres} \\ \text{smooth-form} \doteq \begin{matrix} style \\ subtylostyle \\ tylostyle \end{matrix} \\ \text{acanthose} \doteq yes \\ \text{ornamentation} \doteq \begin{matrix} smooth \\ spines \end{matrix} \end{bmatrix} \end{bmatrix} \\ \text{anatomy} \doteq \begin{bmatrix} \underline{anatomic\text{-}features} \\ \text{ectosome} \doteq nonvisible \end{bmatrix} \end{bmatrix}$$

Figure 3. Representation of a sponge using feature terms.

a)

$$D^1 = \begin{bmatrix} \underline{sponge} \\ \text{spiculate-skeleton} \doteq \begin{bmatrix} \underline{megascleres} \\ \text{smooth-form} \doteq \begin{matrix} style \\ subtylostyle \\ tylostyle \end{matrix} \end{bmatrix} \end{bmatrix}$$

b)

$$D^2 = \begin{bmatrix} \underline{sponge} \\ \text{spiculate-skeleton} \doteq \begin{bmatrix} \underline{megascleres} \\ \text{smooth-form} \doteq \begin{matrix} style \\ subtylostyle \\ tylostyle \end{matrix} \end{bmatrix} \\ \text{external-features} \doteq \text{growing} \doteq \begin{bmatrix} \underline{growing} \\ \text{grow} \doteq \begin{matrix} erect \\ encrusting \end{matrix} \end{bmatrix} \end{bmatrix}$$

Figure 4. Sequence of similitude terms constructed by LID for classifying sponge *s252*.

Now the discriminatory set S_D^1 contains 25 cases (specimens) subsumed by the similitude term D^1. These cases belong to the orders *hadromerida, axinellida* and *poecilosclerida*. Therefore, the similitude term D^1 has to be specialized by adding a new leaf feature (in fact, a path). As before, the most relevant feature is selected using the RLM distance. LID selects the feature *grow*, so the path $\pi(sponge, grow)$ is added to the similitude term D^1. Figure 4b shows the similitude term D^2 that is the specialization of D^1 by adding the path $\pi(sponge, grow)$ with the leaf feature *grow* taking the value the set $\{erect,\ encrusting\}$ as in sponge *s252*. The discriminatory set S_D^2 contains 10 cases subsumed by the similitude term D^2. Then LID is recursively called with S_D^2 and D^2.

All the cases in S_D^2 belong to the order *axinellida*. Therefore LID terminates and classifies the sponge *s252* as belonging to the order *axinellida*. LID explains this classification with the similitude term D^2 and justifies this inference with the cases in S_D^2. In other words, the sponge *s252* belongs to the order *axinellida* because both 1) it has megascleres of form *tylostyle, style,* and *subtylostyle*, and 2) it grows *erect* and *encrusting*.

5.2 Infarct risk assessment in diabetes

Diabetes mellitus is one of the most frequent human chronic diseases. There are two major types of diabetes: diabetes type I (or insulin-dependent) usually found in people being less than 40 years old, and diabetes type II (or non insulin-dependent) often developed in people over this age. Both forms of diabetes produce the same short-term symptoms (i.e. increase of thirst, and high blood glucose values) and long-term complications (i.e. blindness, renal failure, gangrene and amputation, coronary heart disease and stroke).

The main concern in the management of the diabetes is reducing the risk of a patient for developing a new long-term complication and the risk of progression in the complications already present. In fact, the risk for the complications is different whether the patient has diabetes type I than he has diabetes type II. Moreover, the risk is also different whether the patient already has some complication (*progression risk*) than he has no complications (*development risk*).

The application of LID to this domain allows to assess the risk of complications for a diabetic patient. In (Armengol & Plaza, 2001) there are some results of using LID to assess the risk of macro-complications such as infarct, stroke and amputations. In order to illustrate the performance of LID in the diabetes domain we will focus on the assessment of the infarct risk.

$$patient\text{-}71 = \begin{bmatrix} \underline{patient} & \\ \text{personal-data} \doteq & \begin{bmatrix} \underline{personal\text{-}data} & \\ \text{sex} \doteq & female \\ \text{birth-year} \doteq & 1965 \end{bmatrix} \\ \text{basic-diabetes-data} \doteq & \begin{bmatrix} \underline{basic\text{-}diabetes\text{-}data} & \\ \text{dm-type} \doteq & 2 \end{bmatrix} \\ \text{assessments} \doteq & \begin{bmatrix} \underline{assessments} & \\ \text{alcohol?} \doteq & false \\ \text{rhba1c} \doteq & low \\ \text{blindness} \doteq & true \\ \text{nephropathy} \doteq & false \\ \text{polyneuropathy} \doteq & false \\ \text{q-bp} \doteq & low \\ \text{smoker?} \doteq & false \\ \text{q-albumin} \doteq & low \\ \text{q-chol-total} \doteq & low \\ \text{macro-complications?} \doteq & false \end{bmatrix} \end{bmatrix}$$

Figure 5. Description of a diabetic patient: a 36 years old woman with diabetes type II, no macrovascular complications and low values of glycated hemoglobin, albumin, cholesterol and blood pressure (features *hba1c, q-albumin, q-chol-total* and *q-bp* respectively).

LID uses a case base containing 370 patients[1]. Some of these patients have diabetes type I and some others have diabetes type II. Moreover, some patients already have complications and some others have not. Let us suppose that LID is used to assess the risk of infarct for the *patient-71* described in Figure 5.

The LID method can be constrained in the search it performs using domain knowledge. In this domain, we are only interested in considering patients in the case base that share with our current patient the same type of diabetes and the fact of whether or not the patient has macro-vascular complications. This domain knowledge is given to LID as a schema that is used to extract the values of features *dm-type* and *macro-complications* from the current problem *patient-71* obtaining the similitude term of Fig. 6a.

The cases (patients) in S_D^0 have different infarct risk, therefore the similitude term D^0 has to be specialized. The first step is to select the most discriminatory leaf feature using the RLM distance. Specifically, the leaf feature *q-albumin* induces a partition of S_D^0 having the minimal distance to the correct partition. Now the correct partition C is the classification of the cases in S_D^0 according to their infarct risk degree.

The second step adds the most discriminatory leaf feature to the current similitude term D^0. Figure 6b shows the specialization D^1 of D^0 by adding to D^0 the path π(*patient*, *q-albumin)* with *q-albumin* taking value *low* as in the *patient-71*. The discriminatory set S_D^1 contains 99 cases subsumed by D^1. Next LID is

[1]The cases correspond to real patients whose data have been provided by an expert diabetologist.

a)

$$D^0 = \begin{bmatrix} \underline{patient} \\ \text{basic-diabetes-data} \doteq \begin{bmatrix} \underline{basic\text{-}diabetes\text{-}data} \\ \text{dm-type} \doteq 2 \end{bmatrix} \\ \text{assessments} \doteq \begin{bmatrix} \underline{assessments} \\ \text{macro-complications?} \doteq false \end{bmatrix} \end{bmatrix}$$

b)

$$D^1 = \begin{bmatrix} \underline{patient} \\ \text{basic-diabetes-data} \doteq \begin{bmatrix} \underline{basic\text{-}diabetes\text{-}data} \\ \text{dm-type} \doteq 2 \end{bmatrix} \\ \text{assessments} \doteq \begin{bmatrix} \underline{assessments} \\ \text{macro-complications?} \doteq false \\ \text{q-albumin} \doteq low \end{bmatrix} \end{bmatrix}$$

c)

$$D^2 = \begin{bmatrix} \underline{patient} \\ \text{basic-diabetes-data} \doteq \begin{bmatrix} \underline{basic\text{-}diabetes\text{-}data} \\ \text{dm-type} \doteq 2 \end{bmatrix} \\ \text{assessments} \doteq \begin{bmatrix} \underline{assessments} \\ \text{macro-complications?} \doteq false \\ \text{q-albumin} \doteq low \\ \text{q-chol-total} \doteq low \end{bmatrix} \end{bmatrix}$$

Figure 6. Sequence of similitude terms constructed by LID for assessing the infarct risk of *Patient-71*.

recursively called with S_D^1 and D^1.

The cases in the discriminatory set S_D^1 have different risk of infarct, therefore the similitude term D^1 has to be specialized by adding a new discriminatory leaf feature. Now the selected leaf feature is *q-chol-total*. The specialization D^2 (see Figure 6c) has been obtained by adding the path π(*patient*, *q-chol-total)* to D^1. The leaf feature *q-chol-total* takes as value *low* as in the *patient-71*. LID is recursively called with the new discriminatory set S_D^2 and the new similitude term D^2.

The discriminatory set S_D^2 contains 21 cases belonging all them to the class of *moderate* risk of infarct. Therefore LID terminates assessing a *moderate* risk of infarct for *patient-71*. LID explains this classification with the similitude term D^2 and justifies this inference with the cases in S_D^2.

6. Related Work

There are two main lines of research closely related to our approach on similarity assessment in relational representations. On the one hand, there is research about similarity assessment of structured and complex cases in CBR; and, on the other hand, there is research on relational IBL.

Although a lot of work in similarity assessment of cases in CBR is focused on weighted distances among attribute value vectors there is also active research in establishing similarity estimates among structured representation of cases (Börner, 1994; Bunke & Messmer, 1994; Plaza, 1995). Some approaches share the idea of building a "structural similarity" (as our "similitude term" approach) but they use techniques subtree-isomorphism or subgraph isomorphism detection for building the description of this "structural similarity" (Bunke & Messmer, 1994). Another way to construct a similitude term is using antiunification (computing the most specific generalization of two cases) (Plaza, 1995), and later having a measure to assess which similitude term is better (e.g. using an entropy measure (Plaza et al., 1996)) to select the best precedent in the case-base.

A related approach to feature terms as case representation formalism is that of using DL (Description Logics) for this purpose; antiunification can also be used but some assessment measure is also needed, as shown in (Mantay & Moller, 1998) where a probabilistic interpretation of the "least common subsumer" is used. Another approach is using a "structural similarity" to guide adaptation phase of CBR (Börner, 1994) and not for the retrieval phase.

LID does not use antiunification to build the similitude term; instead it constructs the similitude term guided by an example-driven heuristic to assess the relevance of the features to be added to the similitude term. Finally, some CBR research work uses inductive techniques for case retrieval but it involves the construction of decision trees (as in the Inreca project (Auriol et al., 1995)) and only for cases that are attribute-value vectors.

In addition to this, relevant work is being carried on for transferring IBL techniques to relational learning, mainly in the ILP framework of a Horn clause representation of cases and background knowledge. The RIBL system (Emde & Wettschereck, 1996) first generates cases out of Horn clauses in a knowledge base, then calculates the distance among cases using an estimation of the relevance of predicates and attributes. An extension of this approach is the RIBL 2.0 system (Bohnebeck et al., 1998) that allows to compute distances between cases (whose representation include lists and terms) relying on the notion of "edit distance". In our approach we do not use any notion of "distance" among cases, but we indeed use a distance in the heuristic assessment of the importance of features that are to be used in the symbolic description of similitude. Moreover, our assessment is not based on a pairwise comparison of "problem vs. case" similarity but takes into account all cases that share a particular "structural similarity" embodied by those cases subsumed by the similitude term that LID builds.

7. Conclusions

We have developed a technique for case-based learning where cases are best represented as collection of relations among entities. The LID approach is based on a similarity assessment used by a heuristic search in a space of symbolic descriptions of similitude. Moreover, the symbolic description of similitude provides an explanation of the grounds on which a precedent case is selected from the case base as most relevant (or "similar") to the current problem. The symbolic similitude that classifies a problem subsumes a subset of the elements in a class, and as such it is just a partial description of that class. Indeed, this is the main difference between a lazy learning approach like LID and an eager approach as that of induction (see the INDIE method of bottom up induction of feature terms in (Armengol & Plaza, 2000)).

As for future work, we intend to explore a variation of LID that adaptively chooses a middle ground between the extreme points of lazy and eager approaches. Our assumption is that it is unlikely that a lazy (or eager) approach is always the best suited for all application domains. Thus, our aim will be to investigate in which situations it is useful to store (memorize) the partial class descriptions provided by LID an use them to solve new problems and in which situations is better to keep a purely lazy approach.

ACKNOWLEDGEMENTS

This work has been supported by Project SMASH (CICYT TIC96-1038-C04-01). The authors thank Dr. Marta Domingo and physician Albert Palaudàries for their assistance in developing the marine sponges and diabetes applications.

References

Aha, D. (Ed.). (1997). *Lazy learning*. Kluwer Academic Publishers.

Armengol, E., & Plaza, E. (2000). Bottom-up induction of feature terms. *Machine Learning Journal*, *41*, 259–294.

Armengol, E., & Plaza, E. (2001). Individual prognosis of diabetes long-term risks: A CBR approach. *Methods of Information in Medicine*, to appear.

Auriol, E., Wess, S., Manago, M., Althoff, K.-D., & Traphöner, R. (1995). Inreca: A seamless integrated system based on inductive inference and case-based reasoning. *Case-Based Reasoning Reseacrh and Development* (pp. 371–380). Springer-Verlag.

Bohnebeck, U., Horváth, T., & Wrobel, S. (1998). Term comparisons in first-order similarity measures. *Proc. of the 8th International Workshop on Inductive Logic Programming* (pp. 65–79). Springer Verlag.

Börner, K. (1994). Structural similarity as a guidance in case-based design. *Topics in Case-Based Reasoning: EWCBR'94* (pp. 197–208).

Bunke, H., & Messmer, B. T. (1994). Similarity measures for structured representations. *Topics in Case-Based Reasoning: EWCBR'94* (pp. 106–118).

Emde, W., & Wettschereck, D. (1996). Relational instance based learning. *Machine Learning - Proceedings 13th International Conference on Machine Learning* (pp. 122 – 130). Morgan Kaufmann Publishers.

López de Mántaras, R. (1991). A distance-based attribute selection measure for decision tree induction. *Machine Learning*, *6*, 81–92.

Mantay, T., & Moller, R. (1998). Content-based information retrieval by computing least common subsumers in a probabilistic description logic. *Proceedings of the ECAI Workshop Intelligent Information Integration*.

Plaza, E. (1995). Cases as terms: A feature term approach to the structured representation of cases. In M. Veloso and A. Aamodt (Eds.), *Case-based reasoning, ICCBR-95*, no. 1010 in Lecture Notes in Artificial Intelligence, 265–276. Springer-Verlag.

Plaza, E., López de Mántaras, R., & Armengol, E. (1996). On the importance of similitude: An entropy-based assessment. In I. Smith and B. Saltings (Eds.), *Advances in case-based reasoning*, no. 1168 in Lecture Notes in Artificial Intelligence, 324–338. Springer-Verlag.