ARTIFICIAL INTELLIGENCE IN
MEDICINE

Volume 51 No 2 February 2011 ISSN 0933-3657

SPECIAL ISSUE:
Advances in Case-Based Reasoning
in the Health Sciences

Guest Editors:
Isabelle Bichindaritz and Stefania Montani

Classification of melanomas *in situ* using knowledge discovery with explained case-based reasoning

Eva Armengol *

IIIA - Artificial Intelligence Research Institute, CSIC - Spanish National Research Council, Campus UAB s/n, 08193 Bellaterra, Catalonia, Spain

ARTICLE INFO

ABSTRACT

Objective: Early diagnosis of melanoma is based on the ABCD rule which considers asymmetry, border irregularity, color variegation, and a diameter larger than 5 mm as the characteristic features of melanomas. When a skin lesion presents these features it is excised as prevention. Using a non-invasive technique called dermoscopy, dermatologists can give a more accurate evaluation of skin lesions, and can therefore avoid the excision of lesions that are benign. However, dermatologists need to achieve a good dermatoscopic classification of lesions prior to extraction. In this paper we propose a procedure called LazyCL to support dermatologists in assessing the classification of skin lesions. Our goal is to use LazyCL for generating a domain theory to classify melanomas *in situ*.

Methods: To generate a domain theory, the LazyCL procedure uses a combination of two artificial intelligence techniques: case-based reasoning and clustering. First LazyCL randomly creates clusters and then uses a lazy learning method called lazy induction of descriptions (LID) with leave-one-out on them. By means of LID, LazyCL collects explanations of why the cases in the database should belong to a class. Then the analysis of relationships among explanations produces an understandable clustering of the dataset. After a process of elimination of redundancies and merging of clusters, the set of explanations is reduced to a subset of it describing classes that are "almost" discriminant. The remaining explanations form a preliminary domain theory that is the basis on which experts can perform knowledge discovery.

Results: We performed two kinds of experiments. First ones consisted on using LazyCL on a database containing the description of 76 melanomas. The domain theory obtained from these experiments was compared on previous experiments performed using a different clustering method called self-organizing maps (SOM).

Results of both methods, LazyCL and SOM, were similar. The second kind of experiments consisted on using LazyCL on well known domains coming from the machine learning repository of the Irvine University. Thus, since these domains have known solution classes, we can prove that the clusters build by LazyCL are correct.

Conclusions: We can conclude that LazyCL that uses explained case-based reasoning for knowledge discovery is feasible for constructing a domain theory. On one hand, experiments on the melanoma database show that the domain theory build by LazyCL is easy to understand. Explanations provided by LID are easily understood by domain experts since these descriptions involve the same attributes than they used to represent domain objects. On the other hand, experiments on standard machine learning data sets show that LazyCL is a good method of clustering since all clusters produced are correct.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

Early diagnosis and surgical excision are the main goals in the secondary prevention of cutaneous melanoma. Nowadays, the diagnosis of melanoma is based on the ABCD rule [1] which considers four clinical features commonly observed in this kind of tumour: asymmetry, border irregularity, color variegation, and a diameter larger than 5 mm. Although most melanomas are correctly diagnosed following this rule, a variable proportion of melanomas do not comply with these criteria. The current procedure when a suspicious skin lesion appears is to excise and then to analyse it by means of a biopsy. The result of the biopsy usually allows an accurate determination of the malignancy of the lesion.

Dermoscopy is a non-invasive technique introduced by dermatologists two decades ago which provides a more accurate

evaluation of skin lesions, and can therefore, avoid the excision of lesions that are benign. However, dermatologists need to achieve a good dermatoscopic classification of lesions prior to extraction [2]. Hofmann-Wellenhof et al. [3] suggested a classification of benign melanocytic lesions. Argenziano et al. [4] hypothesized that dermoscopic classification may be better than the classical clinico pathological classification of benign melanocytic lesions (nevi). Currently, there is no dermoscopic classification for melanomas located in trunk and extremities. In the era of genetic profiling, molecular studies including microarrays suggest that there is more than one type of melanoma in these sites. The aim of the present study is to help dermatologists in the classification of early melanomas (melanomas in situ) based on dermoscopical characteristics. Dermatologists have defined several dermatoscopic classes of melanoma in situ based on the dermatoscopic features. Dermatopathologists also suggest another classification based on histological features. In particular, our approach consists in using case-based reasoning inside a new knowledge discovery procedure in order to provide dermatologists with a classification theory for melanomas.

Case-based reasoning (CBR) methods predict the classification of a problem based on its similarity to already solved cases. One of the key points of CBR systems is the measure used to assess the similarity between cases, since the final classification of a new problem depends on it. Related to this issue is the fact that results should be clearly understood by the system's user; otherwise, he may not be fully convinced of the results produced by the system. For this reason, in recent years there has been an increasing interest in approaches addressed to explaining CBR results in a satisfactory way (see [5]). One of these approaches is the lazy induction of descriptions (LID) method we introduced in [6]. During the problem solving process LID builds an explanation justifying the classification of a new problem (see Section 2.1). This explanation is, in fact, a generalization of the relevant attributes shared by both problem and cases. In [7] we argued that generalizations can be seen as explanations since they commonly contain problem features useful for classifying problems. This is the case of prototypes from PROTOS [8], generalized cases [9], and lazy decision trees [10]. Also, explanation-based learning (EBL) methods [11] generalize a particular example to obtain a domain rule that can be used for solving unseen problems. Our point is that explanations produced by lazy learning methods like LID should be considered as domain rules in the same way as generalizations are. Thus, the set of explanations could be considered as a lazy domain theory.

Commonly, domain theories are built using eager learning methods (such as ID3 [12]). Eager learning methods build discriminant descriptions for classes, and so the union of these discriminant descriptions covers all the space of known examples. In contrast, lazy domain theories cover only zones around each new problem; therefore, this may result in "holes" in the description of the domain (see Fig. 1). In [13] we compared lazy domain theories formed by sets of explanations from LID with the eager theory built by the ID3 method [12]. In our experiments we showed that, for some domains, eager and lazy theories have similar predictive ability. The difference is that because the explanations that make up the lazy domain theory are more specific than eager rules, there is a high percentage of unseen problems that the lazy theory cannot classify although the classification, when it is proposed, is usually correct.

In the current paper we exploit the concept of lazy domain theory for knowledge discovery. Although lazy domain theories are formed by local rules, this information is very valuable to experts to obtain a picture of some parts of the domain. Frawley et al. [14] defined knowledge discovery as "the non-trivial extraction of implicit, unknown and potentially useful information from data". In fact, we want to support domain experts in building a domain theory, producing explanations (generalizations) that can be easily understood and giving them the opportunity to systematically analyze the classes proposed. However, knowledge discovery problems cannot be directly solved by means of either lazy or eager learning methods since most of them need to know the class label of the domain objects in advance. The most commonly used techniques in knowledge discovery are clustering methods whose goal is to analyze a set of objects and to build clusters based on the similarity between objects. Lazy learning methods cannot be used for clustering because the cases are not labeled. So how can a lazy learning method be used for knowledge discovery? Our proposal is to randomly cluster the domain objects and then consider these clusters as the solution classes. Because domain objects now belong to some class, a lazy learning method can be used to obtain explanations that can be seen as domain rules of a lazy domain theory. We call this procedure LazyCL and we use it to help dermatologists to define and describe classes of melanomas in situ. In fact, this is the main novelty of LazyCL; whereas most approaches combining both CBR and clustering techniques exploit the clustering to organize the case memory in order to make an efficient retrieval of past cases, LazyCL uses CBR for clustering, i.e., explanations produced by a lazy learning method are used as descriptions of clusters.

The paper is structured as follows. In the next section we explain the general procedure of LazyCL. Section 3 describes experiences when using LazyCL to discover a classification of melanomas in situ. Section 4 describes experiences when
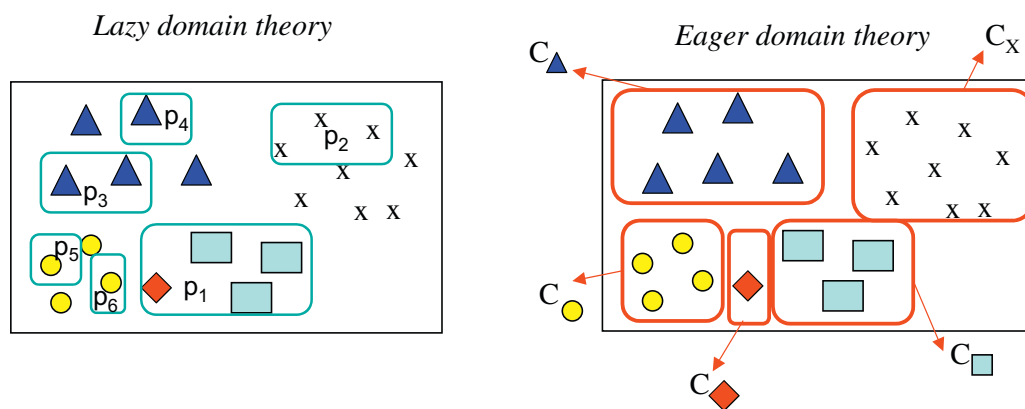

Fig. 1. Descriptions produced by eager methods cover all the space of known examples whereas the union of descriptions produced by lazy learning methods could not cover all the space of known examples.
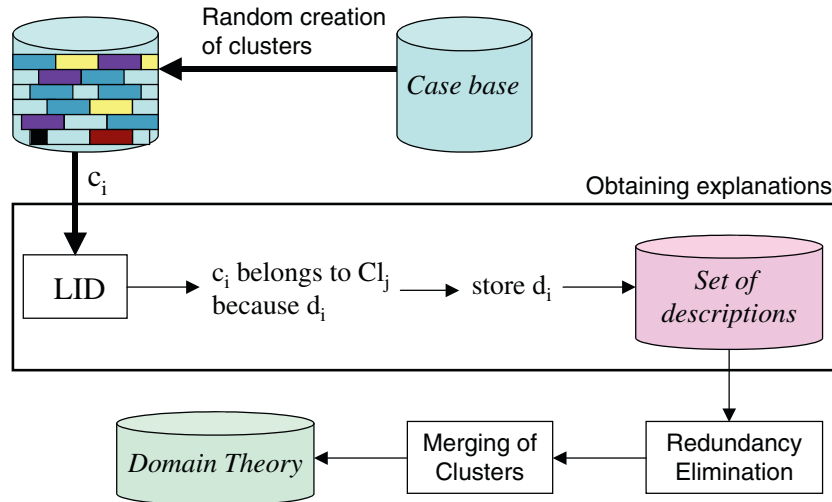
Fig. 2. Scheme of the LazyCL performance.

using LazyCL on some standard data sets of the machine learning repository from the Irvine University (UCI repository). Section 5 compares some methods used for knowledge discovery with our approach. Section 6 is devoted to conclusions and future work.

2. LazyCL: a procedure using explanations for knowledge discovery

Let us suppose the following scenario. Domain experts have available a set of object descriptions (cases) and they hypothesize about the existence of several classes of such objects. These classes would be reasonable from the experts' point of view, and so it is necessary to give an explanation of the clustering. This expected explanation would have a form similar to the symbolic descriptions given by eager learning methods. However, in this scenario, the use of a supervised learning method is not possible because the classes of objects are not known. For this reason, we propose LazyCL as a new procedure for clustering and knowledge discovery producing discriminant descriptions of clusters according to the following four steps (see Fig. 2):

(1) *Creation of clusters.* Disjoint clusters are randomly created to obtain a case base where each case c_i has associated a solution class Cl_k. Let C be a set of unlabeled cases, the procedure used to create the clusters is the following:
  (a) Generate a random natural number $N \leq \mathrm{Card}(C)$ (it will be used as the cardinality of a cluster).
  (b) Let Cl_i be a cluster name. Select randomly N cases of C and associate them to Cl_i.
  (c) Delete the cases of Cl_i from C ($C' = C - Cl_i$).
  (d) Repeat from step (a) until all the cases of C have been included in some cluster.
(2) *Obtaining explanations.* In this second step, the LID method is used with leave-one-out to obtain explanations. Each case c_i is

classified by LID as belonging to a cluster Cl_j. In addition, LID gives an explanation d_j of such classification (see details in Section 2.1). A possible situation is that the method cannot univocally classify a case into a cluster; in other words, the method gives the description (explanation) d_j satisfied by cases of two or more clusters. In such situation, a new cluster Cl_j is created and the description d_j will be associated with it (see Fig. 3). The result of this second step is that each cluster Cl_i (either from the original clustering or newly created during the problem solving process) has associated one or more explanations $d_{i1} \ldots d_{in}$ describing it.
(3) *Redundancy elimination* step to detect relationships between descriptions. Let us suppose that d_{ji} and d_{jk} are descriptions of a cluster and that d_{ji} is more general than d_{jk}. In such situation the description d_{jk} (the most specific of both) is rejected. For instance, let us suppose that a cluster from the glass data set of the UCI repository [15] is described by the following two descriptions:

- d_1: (Ri $\in$ [1.52, 1.73]) and (Na $\leq$ 14.065) and (Al $\leq$ 1.39) and (K $\in$ [0.055, 0.615]) and (Ca $\in$ [8.315, 10.075]).
- d_2: (Ri $\in$ [1.52, 1.73]) and (K $\in$ [0.055, 0.615]) and (Ca $\in$ [8.315, 10.075]).

During the redundancy elimination step d_1 will be eliminated since it is more specific than d_2 and, therefore, all the cases satisfying d_1 will also satisfy d_2. Notice that at the end of this step, each cluster Cl_j is described by a set of descriptions $d_{j1} \ldots d_{jn}$ where each d_{ji} describes a subset of the cases included in the cluster.
(4) *Merging of clusters.* The process is the same as the performed during the redundancy elimination step, but now it is applied on descriptions of different clusters. In other words, the goal of this step is to detect relationships between the descriptions of
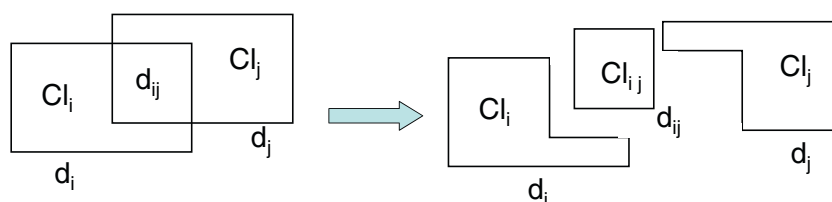

Fig. 3. When case c_i is classified as belonging to Cl_i and Cl_k justifying this classification by an explanation d_{ij}, then a new cluster described by d_{ij} is created.
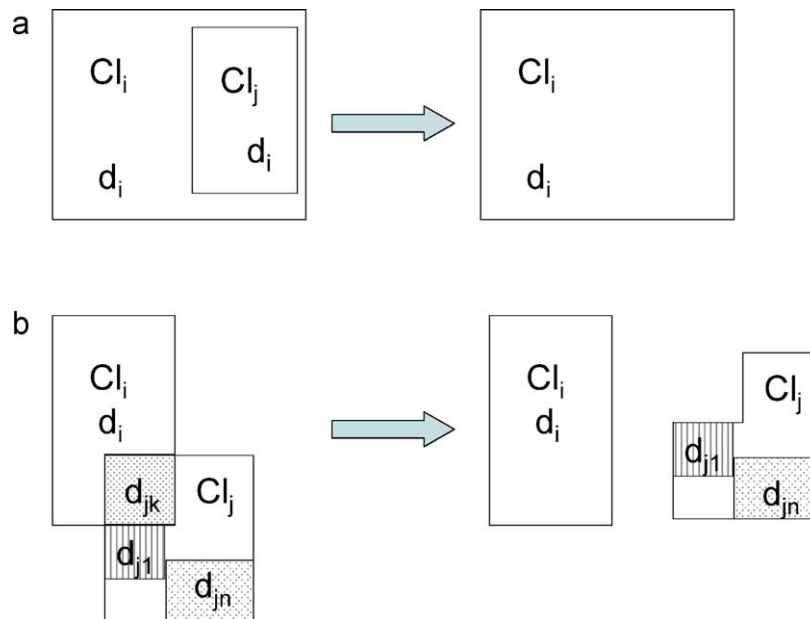
Fig. 4. Merging of clusters. (a) When the description of Cl_i is more general than the description of Cl_j, this latter cluster is rejected. (b) When a description d_{jk} of a cluster Cl_i is more specific than a description d_i of Cl_i then d_{jk} is rejected.

the different clusters. Let C_i and C_j be two clusters, their respective descriptions can be related in the following alternative ways:

- Let us suppose that Cl_i is described by only one description, namely d_i, and Cl_j is also described by only one description, namely d_j. If d_i is more general than d_j both the cluster Cl_j and the description d_j are rejected (Fig. 4a).
- Let us suppose that Cl_i is described by only one description, namely d_i, and Cl_j is described by several descriptions, namely $d_{j1} \ldots d_{jn}$. If one of these descriptions, say d_{jk}, is more general than d_i then both the cluster Cl_i and the description d_i are rejected.
- Let us suppose that Cl_i is described by only one description, namely d_i, and Cl_j is described by several descriptions, namely $d_{j1} \ldots d_{jn}$. If d_i is more general than d_{jk} then the description d_{jk} is eliminated from the description of Cl_j similarly to previous case (Fig. 4b).
- Let us suppose that Cl_i is described by several descriptions, namely $d_{i1} \ldots d_{in}$, and a cluster Cl_j is described by several descriptions, namely $d_{j1} \ldots d_{jn}$. If d_{jk} is more general than d_{ih} then the description d_{ih} is rejected as in the previous case. There is a dual situation when the description d_{ih} is more general than d_{jk}.

Finally, the descriptions of clusters, i.e., a provisional domain theory, are shown to experts who adjust them according to their knowledge. Note that this revision step is easy for the experts to carry out since the descriptions of clusters are represented with the same representation language as the used to describe domain objects.

The best situation for the clusters built by LazyCL is when they have empty intersection, since this means that all the known cases can be clearly separated. An interesting situation arises when some clusters have non-empty intersection, although their respective descriptions are not related. This situation can be interpreted, for example, in medical domains as a patient presenting a set of symptoms consistent with more than one disease. From a general point of view, this is an opportunity for the expert to discover relationships between classes. For instance, depending on the domain, a possible interpretation of this situation could be to

decide that clusters with non-empty intersection are actually the same cluster.

2.1. The lazy induction of descriptions method

In step 2, LazyCL may use any lazy learning method able to construct explanations of the result. In our implementation, LazyCL uses LID that determines which are the most relevant attributes of a problem and searches in a case base for cases sharing these relevant attributes. The problem p is classified when LID finds a set of relevant attributes 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. We call *similitude term* the description formed by these relevant attributes and *discriminatory set* the set of cases satisfying a similitude term. In fact, a similitude term is a generalization of both p and the cases in the discriminatory set.

Fig. 5 shows the LID algorithm (see a more detailed explanation of LID in [6]). Given a problem p, LID initializes D as a description with no attributes and the discriminatory set S_D as the set of cases satisfying D (initially the whole case base). When the stopping condition of LID is not satisfied, the next step is to select an attribute for specializing D. The specialization of a similitude term D is achieved by adding attributes to it. The selection of the most

$$
\begin{aligned}
&\underline{\text{Function}}\ \text{LID}\ (p,\ S_D,\ D,\ C)\\
&\qquad S_D := \text{Discriminatory-set}\ (D)\\
&\qquad \underline{\text{if}}\ \text{stopping-condition}(S_D)\\
&\qquad\qquad \underline{\text{then}}\ \underline{\text{return}}\ class(S_D)\\
&\qquad\qquad \underline{\text{else}}\ f_d := \text{Select-attribute}\ (p,\ S_D,\ C)\\
&\qquad\qquad\qquad D := \text{Add-attribute}(f_d,\ D)\\
&\qquad\qquad\qquad S_D := \text{Discriminatory-set}\ (D\ ,\ S_D)\\
&\qquad\qquad\qquad \text{LID}\ (S_D\ ,\ p,\ D\ ,\ C)\\
&\qquad \underline{\text{end-if}}\\
&\underline{\text{end-function}}
\end{aligned}
$$

Fig. 5. The LID algorithm: p is the problem to be solved, D is the similitude term, S_D is the discriminatory set associated with 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.

discriminatory attribute is heuristically done using the López de Mántaras' distance (LM) [16] over the candidate attributes (see Section 2.2). Let f_d be the attribute assessed as the most discriminatory. In such a situation, the specialization of D defines a new similitude term D' by adding to D the attribute f_d. After adding f_d to D, the new similitude term $D' = D + f_d$ satisfies a subset of cases in S_D, namely $S_{D'}$. Next, LID is recursively called with the discriminatory set $S_{D'}$ and the similitude term D'. The recursive call of LID uses $S_{D'}$ instead of S_D because the cases which are not satisfied by D' will not be satisfied by any further specialization.

LID has two possible stopping situations: (1) all the cases in the discriminatory set belong to the same solution class, and (2) there is no attribute allowing the specialization of the similitude term. In the first situation, the final similitude term can be interpreted as a partial description of the class in the same sense as eager methods. Notice also that this partial description, in turn, can be interpreted as an explanation of why a problem may be classified as member of a class. In the second situation, the final similitude term shows a set of attributes shared by several classes.

The similitude term can be interpreted in several ways. A possible interpretation is that 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, EBL or inductive learning methods. Since the similitude term contains the important attributes used to classify a problem, it can also be interpreted as a justification or *explanation* of the problem classification. Notice that in any of the interpretations above, LID explanations can be taken by the system as domain rules since they contain the relevant attributes for classifying a problem.

2.2. The López de Mántaras distance

The LM distance [16] assesses how similar two partitions are in the sense that the lesser the distance the more similar they are. Each attribute 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 distance considers the *correct partition* P_c as the one 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 between them is computed as follows:

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

where $I(P_A) = -\sum_{k=1}^{n} p_k \cdot \log_2 p_k$ measures the information contained in the partition P_A; $I(P_A \cap P_B) = -\sum_{j=1}^{n} \sum_{k=1}^{m} p_{jk} \cdot \log_2 p_{jk}$ is the mutual information of the two partitions; n and m are the number of sets of the partitions P_A and P_B respectively; $p_k = |Cl_k|/|S|$ is the probability of occurrence of the class Cl_k, i.e., the proportion of elements in S belonging to Cl_k; and $p_{jk} = |Cl_k \cap Cl_j|/|S|$ is the probability of the intersection $Cl_j \cap Cl_k$, i.e., the proportion of elements in S belonging to Cl_j and Cl_k.

In our case, the LM distance is used to compute the distance between a partition generated by an attribute and the correct partition. The correct partition P_c has two classes, one containing positive examples of a class and other containing negative examples of that class. Thus, for each attribute f_i, there is a partition P_{fi} of the case base B according with the values of f_i. Each partition P_{fi} is compared with the correct partition P_c using the LM distance. Let P_{fi} and P_{fj} be the partitions induced by attributes f_i and f_j respectively. The attribute f_i is *more discriminatory than* the attribute f_j if $LM(P_{fi}, P_c) < LM(P_{fj}, P_c)$. In other words, when an

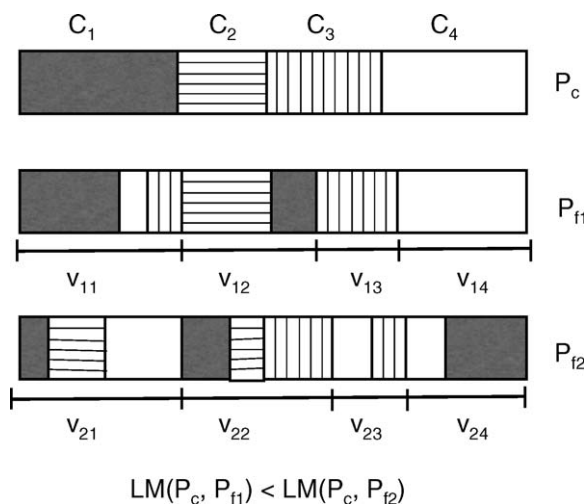$$\text{LM}(P_c, P_{f1}) < \text{LM}(P_c, P_{f2})$$

Fig. 6. Intuitive idea of the LM distance. The partition induced by f_1 is more similar to the correct partition P_c than the partition induced by f_2.

Table 1
Dermoscopic attributes describing melanomas *in situ*.

Attribute	Values
Pigment-Network	0: Absence; 1: typical; 2: atypical
Dots-and-Globules	0: Absence; 1: typical; 2: atypical
Streaks	0: Absence; 1: regular; 2: irregular
Regression-Structures	0: Absence; 1: presence; 2: peppering; 3: white-areas; 4: peppering-and-white-areas
Bloches	0: Absence; 1: regular; 2: irregular
Vessels	0: Absence; 1: dotted; 2: typical; 3: atypical; 4: dotted-atypical

attribute f_i is more discriminatory than another attribute f_j, the partition induced by f_i in B is closer to the correct partition P_c than the partition induced by f_j. Intuitively, the most discriminatory attribute classifies the cases in B in a more similar way to the correct classification of cases (see Fig. 6). The most discriminatory attribute f_d produces a partition P_d having the minimum distance $\text{LM}(P_{fd}, P_c)$ to the correct partition P_c.

3. Experimenting with LazyCL for classifying melanomas *in situ*

Dermatologists provided us with a database with descriptions of 76 melanomas *in situ* from the consensus of six experts (four dermatologists and two dermatopathologists).[1] The descriptions comprise three kinds of attributes: clinical, dermoscopic and histological. Clinical attributes are those referring to the patient (such as age and sex) and also to the characteristics of the melanoma (diameter, situation, etc.). Dermoscopic attributes (Table 1) are melanoma characteristics that dermatologists extract using epiluminiscence microscopy during an examination. Histological attributes are the ones obtained from the biopsy of an excised nevus. Because the dermatologist's goal is to identify skin lesions that are melanomas (or might be in the future), we ignore both clinical and histological attributes and use only dermoscopic attributes for cluster formation.

The first step of the LazyCL procedure associates a fictitious class label to each melanoma of the database. Fig. 7 shows that the available melanomas have been initially grouped in six clusters. Notice that because of the randomness of this step, each execution of LazyCL can begin with a different number of clusters. The second

---

[1] This consensus is important for describing melanomas and the meaning of the values of each attribute. The resulting description of melanomas must be widely agreed and recognized by the dermatology community.

clust-174

(obj-38 obj-67 obj-50 obj-40 obj-64 obj-34 obj-32 obj-31 obj-13
  obj-16 obj-28 obj-41 obj-24 obj-70 obj-69 obj-14 obj-19 obj-36
  obj-1 obj-68 obj-56 obj-9 obj-35 obj-3 obj-21 obj-46 obj-54
  obj-8 obj-59 obj-27 obj-23 obj-29 obj-55)

clust-175

(obj-65 obj-10 obj-62 obj-45 obj-15 obj-72 obj-0 obj-60 obj-4 obj-73)

clust-176

(obj-22 obj-7 obj-63 obj-58 obj-48 obj-66 obj-26 obj-53
  obj-12 obj-51 obj-5 obj-2 obj-43 obj-11 obj-49 obj-74 obj-33)

clust-177

(obj-47 obj-42 obj-25 obj-52 obj-30 obj-44)

clust-178

(obj-57 obj-6 obj-61 obj-18 obj-20 obj-71 obj-75)

clust-179

(obj-39 obj-37 obj-17)

Fig. 7. Initial clustering of the 76 melanomas of the database, randomly formed during the first step of LazyCL.

step in LazyCL is to use the LID method with leave-one-out in order to obtain both a classification for each melanoma and an explanation of the classification. The idea is to use these explanations as rules of the domain theory. Sometimes LID finishes, although the current similitude term D_n is satisfied by objects of several classes (second stopping condition mentioned in Section 2.1). Fig. 8 shows some outputs provided by LID in classifying objects. In particular *obj-49* is classified as belonging to *clust-176* since it satisfies the description characterizing that cluster. Object *obj-56* is classified as belonging to *clust-177* and there are four explanations justifying this classification. Finally, *obj-60* can be classified as belonging to two clusters: *clust-178* and *clust-176* and there are also two explanations of this classification. In the case of *obj-60*, LazyCL creates a new cluster label (*clust-191* in our example) described by the current explanations, where the current problem and also the known objects satisfying the explanation are included. At the end of the leave-one-out process, each (initial or newly created) cluster has one or more descriptions associated to characterize the objects included in it.

The *redundancy elimination step* of LazyCL allows the rejection of some descriptions due to the existence of generality-specificity relationships between them. For instance, the first description of cluster *clust-191* (see Fig. 9) is rejected because it is more specific than the second description of the same cluster (it contains the condition Dots-and-Globules = 2 which is not present in the second description). Similar relationships are detected between descriptions of several clusters during the merging step. Finally, LazyCL produces a domain theory with classes described by non-related descriptions. However, it is easy to detect other possible simplifications which are difficult to automate, but which experts could assess.

Thus, the domain theory was show to dermatologists who found the domain theory provided by LazyCL compatible with their knowledge and proposed several fusions of clusters. Fig. 10 shows the descriptions for clusters *clust-223* and *clust-192*. Notice that the two descriptions are different although the set of objects that satisfies them is almost the same (*obj-61* is only in cluster *clust-223* and *obj-63* is only in *clust-192*). For this reason, dermatologists decided to merge the clusters and generalize their respective descriptions. The final result is a cluster, say *clust-223** described by ((Pigment-Network = 2) (Regression-Structures = 2) (Bloches = 0) (Vessels = 0)). This generalization has been possible because the two descriptions have several attributes in common. Dermatologists have to take the final decision either to generalize the two descriptions or to form a new cluster with two descriptions. Notice that the new generalized description may also introduce a new relation with descriptions of other clusters, and so a new *merging of clusters* step should be performed.

As in any clustering method, each execution of LazyCL produces different descriptions and therefore the final domain theory is also different. However, we performed several experiments and found that some clusters were formed repeatedly in almost all the executions. Clearly, we need to conduct more experiments with LazyCL, but our goal of using it as a clustering method for knowledge discovery has been achieved. In particular dermatol-

LID SOLUTION FOR CASE: obj-49

* clust-176
((Pattern = Reticular) (Dots-and-Globules = 1) (Streaks = 0)
(Regression-Structures = 4) (Bloches = 0))

LID SOLUTION FOR CASE: obj-56

* clust-177
((Pattern = Multicomponent) (Pigment-Network = 2) (Dots-and-Globules = 2))
(Regression-Structures = 0) (Bloches = 0) (Vessels = 0))

* clust-177
((Pattern = Multicomponent) (Pigment-Network = 2) (Dots-and-Globules = 2)
(Streaks = 0) (Regression-Structures = 0) (Vessels = 0))

* clust-177
(Pattern = Multicomponent) (Pigment-Network = 2) (Dots-and-Globules = 2)
(Streaks = 0) (Regression-Structures = 0) (Vessels = 0))

* clust-177
((Pattern = Multicomponent) (Pigment-Network = 2) (Dots-and-Globules = 2)
(Regression-Structures = 0) (Bloches = 0) (Vessels = 0))

LID SOLUTION FOR CASE: obj-60

* (clust-178 clust-176)
((Pattern = Multicomponent) (Pigment -Network = 2) (Streaks = 2)
(Regression-Structures = 2) (Bloches = 0) (Vessels = 0))

* (clust-178 clust-176)
((Pattern = Multicomponent) (Pigment-Network = 2) (Streaks = 2)
(Dots-and-Globules = 2) (Regression-Structures = 2) (Bloches = 0) (Vessels (0)))

Fig. 8. Outcome given by LID when classifying the objects *obj-49*, *obj-56* and *obj-60*.

ogists liked the simplicity of the final theories, which were very useful for educational purposes. The fact that some clusters in the final theory have non-empty intersection does not seem to be a shortcoming for the experts, since they interpret such intersections as different views of the same object.

We also compared the cluster's descriptions produced by LazyCL with those we obtained in a previous study [17] with SOMEX. SOMEX is a combination of two machine learning approaches:

clustering and generalization. In a first step, using the self-organizing maps (SOM) [18] method, the skin lesions of the data sets are clustered in patterns according to their similar characteristics. In a second step, a generalization method based on the notion of anti-unification [19] is used to explain clustering results. In short, the anti-unification of a set of elements is a symbolic description composed of the attributes with values common to all the elements of the set. Thus, the cluster's descriptions are composed by the

CLUSTER : clust-191
This cluster corresponds to : (clust-178 clust-176)

* ((Pattern = Multicomponent) (Pigment-Network = 2) (Dots-and-Globules = 2)
  (Streaks = 2) (Regression-Structures = 2) (Bloches = 0) (Vessels = 0))
satisfied by 3 objects : (obj-48 obj-60 obj-75)

* ((Pattern = Multicomponent) (Pigment-Network = 2) (Streaks = 2)
  (Regression-Structures = 2) (Bloches = 0) (Vessels = 0))
satisfied by 3 objects : (obj-48 obj-60 obj-75)

Fig. 9. Cluster *clust-191* is the intersection of clusters *clust-178* and *clust-176*. There are two possible descriptions for *clust-191*.

```
CLUSTER : clust-223
This cluster corresponds to : (clust-195 clust-189 clust-192)
* ((Pigment-Network (2)) (Streaks (2)) (Regression-Structures (2))
  (Bloches (0)) (Vessels (0)))
satisfied by 4 objects : (obj-48 obj-60 obj-61 obj-75)


CLUSTER : clust-192
* ((Pattern (Multicomponent)) (Pigment-Network (2)) (Regression-Structures (2))
  (Dots-and-Globules (2)) (Bloches (0)) (Vessels (0)))
satisfied by 4 objects : (obj-48 obj-60 obj-63 obj-75)
```

Fig. 10. Description of clusters *clust-223* and *clust-192*. Notice that both clusters differ in only one object.

```
The cluster is composed of the objects:
(obj-69 obj-70 obj-28 obj-31 obj-56
 obj-15 obj-10 obj-11 obj-37 obj-44)

The explanation is the following

((Age 28 43 49 50 54 65 66 68)
 (Max_Diam 5 6 7 8 9 18)
 (Site Leg Arm Lower-Extr Upper_Extr Trunk Back)
 (Millia_Like_Cyst 0)
 (Pattern Globular Reticular Unspecific Multicomponent)
 (Bloches 0)
 (Regression_Structures 0)
 (Dots_And_Globules 2)
 (Streaks 0)
 (Diagnosis Mnevus P_Ltg Pl_M Nonc Ltg_M))
```

a

```
CLUSTER  :   clust-193

*  ((Dots_And_Globules (2)) (Vessels (1)))

*  ((Pigment_Network (2)) (Vessels (1)))

satisfied by 4 objects :

    (obj-11 obj-15 obj-41 obj-58)
```

b

Fig. 11. Symbolic descriptions produced by: (a) SOMEX, (b) LazyCL.

melanoma attributes that take common values in all the elements of the cluster. In a second version of SOMEX, the cluster's descriptions contain all the possible values that the common attributes can take. This change was made at the request of the dermatologists. Dermatologists tend to focus their attention on common attributes with common values, although they also want to analyze the different values that a common attribute can take. The use of symbolic descriptions to describing clusters allows dermatologists to visually recognize a melanoma. For instance, Fig. 11(a) shows a cluster description produced by SOMEX. Notice that in this description the only relevant attribute seems to be the presence of atypical dots and globules (i.e., Dots-and-Globules = 2) according to the values shown in Table 1) whereas other attributes such as Pigment-Network and Vessels are not present. The absence of these attributes in the description means that they are not relevant since they hold none of the possible values in the examples included in the cluster. Fig. 11(b) shows the description of the cluster *clust-193* produced by LazyCL. This cluster has two descriptions: one of them focuses on the presence of both atypical dots and globules and typical vessels, and the other on the presence of an atypical pigment network and typical vessels. Fig. 12 shows the melanoma corresponding to the *obj-11*. This picture was provided by dermatologists on seeing the descriptions of both SOMEX and LazyCL.

4. Experimenting with LazyCL on standard data sets

To analyze the feasibility of LazyCL, we used several data sets from the UCI repository [15] (Table 2). Most of them have attributes with numeric values, therefore we discresized them to obtain nominal values. The goal of the experiments was twofold. Firstly, we wanted to analyze whether or not the clustering is consistent with the correct classification of cases. Secondly, since the purpose of LazyCL is for knowledge discovery, we also wanted to analyze the descriptions of the clusters.

Fig. 12. Melanoma image for *obj-11*. This melanoma presents under dermoscopy atypical dots and globules, atypical pigment network and typical vessels.

For each data set two different kinds of experiments have been carried out. The first one consisted on generating a random number of clusters as indicated in Section 2. The second kind of experiments consisted on generating the same number of clusters as classes has the data set. For instance, the *hepatitis* data set has

Table 2
Description of the characteristics of datasets used on the experiments.

Dataset	Attributes	Instances	Classes
Iris	4	150	3
Hepatitis	19	155	2
Glass	9	214	7
Ionosphere	34	351	2
Bal	4	625	3

CLUSTER : clust-669 (73 objects)

  * (Left-Weight > 2.5) and (Left-Distance > 2.5) and (Right-Weight > 2.5) and (Right-Distance > 2.5)

CLUSTER : clust-668 (51 objects)

  * (Left-Weight > 2.5) and (Left-Distance > 2.5) and (Right-Weight > 2.5) and (Right-Distance < 2.5)

CLUSTER : clust-667 (49 objects)

  * ((Left-Weight > 2.5) and (Left-Distance > 2.5) and (Right-Weight < 2.5) and (Right-Distance > 2.5)

CLUSTER : clust-666 (33 objects)

  * ((Left-Weight > 2.5) and (Left-Distance > 2.5) and (Right-Weight < 2.5) and (Right-Distance < 2.5)

CLUSTER : clust-665 (47 objects)

  * ((Left-Weight > 2.5) and (Left-Distance < 2.5) and (Right-Weight > 2.5) and (Right-Distance > 2.5)

CLUSTER : clust-664 (33 objects)

  * ((Left-Weight > 2.5) and (Left-Distance < 2.5) and (Right-Weight > 2.5) and (Right-Distance < 2.5)

Fig. 13. A sketch of cluster descriptions produced by LazyCL on the *bal* data set.

cases of two classes, *die* and *live*, therefore we imposed the generation of only two random clusters. Thus, for this data set, the cardinality of each cluster has to be approximately the same, i.e., Card(Cl_i) = Card(C)/2, where C is the cardinality of the training set. The idea of this second experiment was to deal with high entropy of clusters, since the situation is to have the correct number of clusters containing elements of all the classes. For both kinds of experiments we carried out several executions. Notice that each execution use different initial clusters since they have been randomly created. In the following sections we show some interesting results corresponding to three different situations detected from the analysis of the final clusters.

4.1. Example 1: clusters with empty intersection

The *bal* dataset is composed by 625 objects of three classes. The random creation of clusters produced more than 150 clusters in all the experiments. However the result of LazyCL always has been a set of 16 clusters (see some of these clusters in Fig. 13). These clusters have discriminatory descriptions and there is no cases belonging to more than one cluster.

We also performed experiments fixing the number of initial clusters. Thus, because the *bal* data set has objects of three classes, LazyCL initially generates three clusters. At the end, LazyCL produced only one cluster but this cluster was described by 16 descriptions, the same ones obtained in previous experiments with random number of initial clusters. We compared these descriptions with the rules obtained from a decision tree built by ID3. This decision tree has 16 branches described by exactly the same rules produced by LazyCL.

Similar results have been obtained using LazyCL on the *iris* data set, although in this data set the attributes involved in the descriptions of clusters are not exactly the same than the ones of the decision tree. Fig. 14 shows both the theory from ID3 and the theory from LazyCL for *iris*. Notice that the ID3 theory contains several paths that do not classify any example (those which leaf is *null*). The theory from LazyCL has two rules for classifying *iris-setosa*. One of these rules (number 2) is the same one built by ID3. Concerning the rule 1 of LazyCL, by checking the available examples of the data set, we confirmed that this rule is correct

since there are 50 examples satisying it. So, LazyCL discovered a new useful rule for *iris-setosa*. Concerning the rules for the class *iris-versicolor*, rules 1 and 2 from ID3 correspond to rules 1 and 2 from LazyCL, respectively. Notice that rule 3 of LazyCL is an specialization of rule 1, so a possible future work could be to analyze relations between the explanations forming the lazy domain theory. Similarly, for classifying *iris-viginica*, rules 1 and 2 from LazyCL correspond to rules 1 and 2 from ID3 and rule 3 does not correspond to any rule of LazyCL. Sumarizing, for the *iris* dataset, both ID3 and LazyCL theories are formed by almost the same rules and these rules have similar specialization degree.

4.2. Example 2: clusters with descriptions satisfied by the same subset of objects

The same subset of cases can satisfy descriptions of different clusters. This is not a contradiction since it can be interpreted as two views of the same set of objects. An example of this situation can be seen in cluster *clust-109* from the *ionosphere* data set (Fig. 15). The two descriptions of this cluster are satisfied by the same subset of examples, the difference between both descriptions is that *D*1 has the attribute (A11 ≤ 0.583) and the description *D*2 has the attribute (A31 ∈ [0.212, 0.986]). This is an opportunity for knowledge discovery, since domain experts may analyze the two descriptions and decide that, for instance, the attributes A11 and A31 are irrelevant. Thus, the description of *clust-109* could be the following:

(A03 ≥ 0.287) and (A05 ∈ [0.23, 0.999]) and (A09 ≤ 0.071) and (A14 ≤ 0.583) and (A21 ≥ 0.1854) and (A30 ≤ 0.930)

This is a common situation when using LazyCL on the melanomas domain. In fact, this kind of situations when describing melanomas has given to the expert interesting knowledge about how to classify them.

4.3. Example 3: clusters with non-empty intersection

When clusters have non-empty intersection, the situation is similar to the one described in previous section. Given two clusters Cl_i and Cl_j described respectively by d_i and d_j, let S_i be the set of cases

CLASS: IRIS–SETOSA
1) Petallength = (–inf, 2.45]
2) Petalwidth = (–inf, 0.8]

CLASS: IRIS–VERSICOLOR
1) Sepallength = (6.15, inf) and
    Petallength = (2.45, 4.75] and
    Petalwidth = (0.8, 175]
2) Sepalwidth = (2.95, 3.35] and
    Petallength = (2.45, 4.75] and
    Petalwidth = (0.8, 1.75]
3) Sepallength = (6.15, inf) and
    Sepalwidth = (–inf, 2.95] and
    Petallength = (2.45, 4.75] and
    Petalwidth = (0.8, 1.75]

CLASS: IRIS–VIRGINICA
1) Sepallength = (6.15, inf) and
    Sepalwidth = (–inf, 2.95] and
    Petalwidth = (1.75, inf)
2) Sepallength = (5.55, 6.15] and
    Sepalwidth = (–inf, 2.95] and
    Petalwidth = (1.75, inf)
3) Sepallength = (6.15, inf) and
    Sepalwidth = (3.35, inf) and
    Petalwidth = (1.75, inf)

petalwidth = '(–inf–0.8]': Iris–setosa
petalwidth = '(0.8–1.75]'
| petallength = '(–inf–2.45]': null
| petallength = '(2.45–4.75]'
| | sepallength = '(–inf–5.55]'
| | | sepalwidth = '(–inf–2.95]': Iris–versicolor
| | | sepalwidth = '(2.95–3.35]': Iris–versicolor (2)
| | | sepalwidth = '(3.35–inf)': null
| | sepallength = '(5.55–6.15]': Iris–versicolor (1)
| | sepallength = '(6.15–inf)': Iris–versicolor
| petallength = '(4.75–inf)'
| | sepallength = '(–inf–5.55]': null
| | sepallength = '(5.55–6.15]': Iris–virginica
| | sepallength = '(6.15–inf)': Iris–versicolor
petalwidth = '(1.75–inf)'
| sepallength = '(–inf–5.55]': null
| sepallength = '(5.55–6.15]'
| | sepalwidth = '(–inf–2.95]': Iris–virginica (2)
| | sepalwidth = '(2.95–3.35]': Iris–virginica
| | sepalwidth = '(3.35–inf)': null
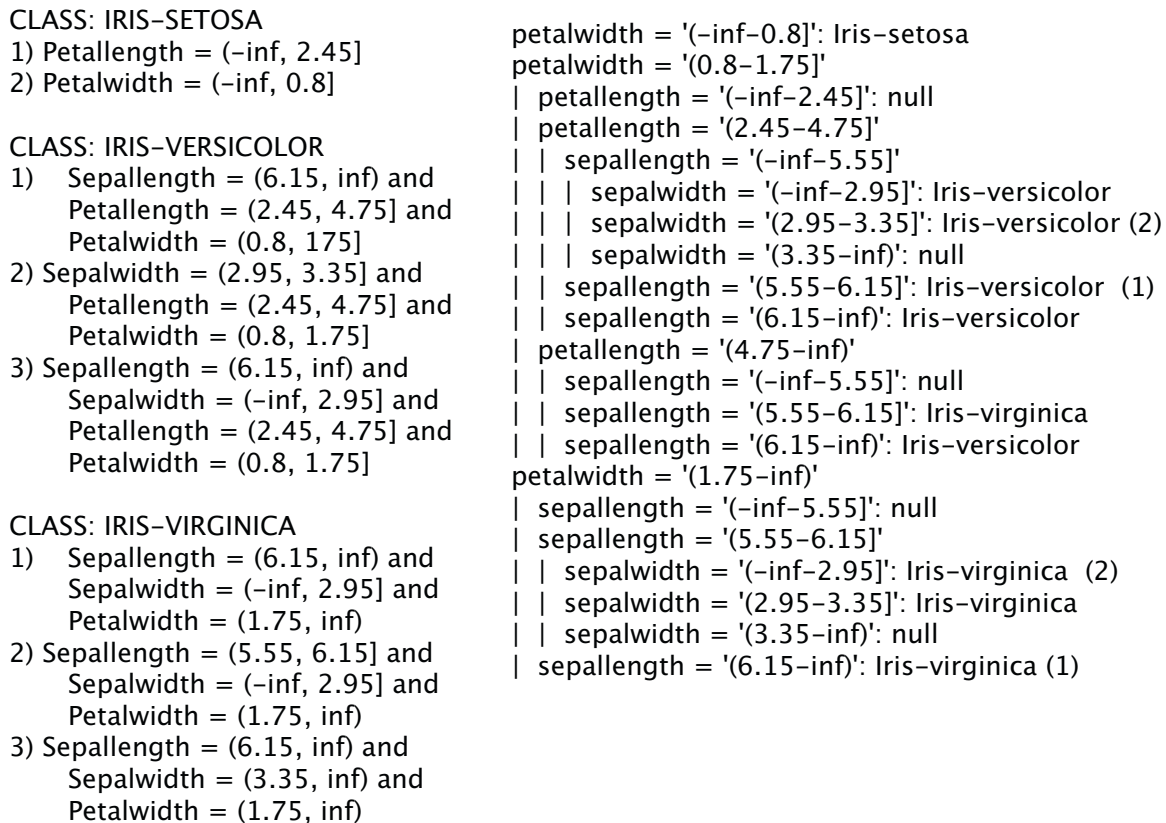| sepallength = '(6.15–inf)': Iris–virginica (1)

Fig. 14. Domain theories from LazyCL (left) and from ID3 (right) for the *iris* data set.

satisfying d_i and let S_j be the set of cases satisfying d_j. If $S_i \subset S_j$ then the expert can analyze the two descriptions to decide about the relevance of the attributes that they do not share. As before, if the differences are found irrelevant, the two clusters can be merged.

An example of this situation are clusters *clust-118* and *clust-103* of the *hepatitis* data set. All the cases in the cluster *clust-118* are also in *clust-103*, except *obj-3*, although there are several differences between the descriptions of both clusters (see Fig. 16). Here, the domain expert can decide whether or not the shared attributes are the relevant ones to describe a unique cluster. Another possibility is that the differences between descriptions are actually important to define two clusters. In such situation, the cases satisfying both descriptions may be interpreted as having two possible classifications depending on the attributes which are taken into account (i.e., different views of the same objects). In the case of melanomas this situation could describe several evolution states of the same kind of melanoma.

4.4. Discussion

Experimentation with LazyCL on UCI data sets took two forms: (a) experiments in which both the initial number of clusters and the number of cases inside each cluster have been chosen randomly; and (b) experiments in which the number of clusters was fixed to the number of classes of the data set (although cases inside each cluster were chosen randomly). LazyCL produced the same kind of results in both kinds of experiment and the final domain theory was the same for most domains (an exception is the *hepatitis* dataset). This is surprising, since in experiment (b) clusters have high entropy and, in consequence, we expected the clustering process to finish with non-discriminant descriptions. Nevertheless, LazyCL produced cluster descriptions that are discriminant in both kinds of experiment. Thus, for domains such as *bal* and *tao*, both experiments produce the same descriptions and, moreover, these descriptions were the same as those obtained

CLUSTER : clust-109

 * D1 : (A03 > 0.287) and (A05 in [0.23, 0.999]) and (A09 < 0.071) and (A11 < 0.583) and (A14 < 0.583) and (A21 < 0.1854) and (A30 < 0.930)

11 objects : (obj-8 obj-46 obj-64 obj-81 obj-107 obj-114 obj-155 obj-193 obj-197 obj-237 obj-273)

* D2 : (A03 > 0.287) and (A05 in [0.23, 0.999]) and (A09 < 0.071) and (A14 < 0.583) and (A21 < 0.1854) and (A30 < 0.930) and (A31 in [0.212, 0.986])

11 objects : (obj-8 obj-46 obj-64 obj-81 obj-107 obj-114 obj-155 obj-193 obj-197 obj-237 obj-273)

Fig. 15. Examples of clusters, for the *ionosphere* data set, with an inclusion relation of their associated sets.

CLUSTER : clust-118

* ((Steroid (Yes)) (Antivirals (No)) (Fatigue (No)) (Alk_Phosphate (All)) (Histology (No)))

11 objects : (obj-3 obj-4 obj-58 obj-83 obj-91 obj-92 obj-108 obj-115 obj-123 obj-125 obj-139)

CLUSTER : clust-103

* ((Steroid (Yes)) (Antivirals (No)) (Fatigue (No)) (Spiders (No))
(Albumin (More-Than)) (Histology (No)))

13 objects : (obj-4 obj-10 obj-58 obj-65 obj-83 obj-84 obj-91 obj-92 obj-108 obj-115 obj-123 obj-125 obj-139)

Fig. 16. Two clusters produced by LazyCL for the *hepatitis* data set. These clusters are almost the same but there is not an inclusion relation between them.

by a decision tree. For domains less regular than the two mentioned above, LazyCL also produced well-formed clusters although the descriptions are more specific than the ones produced by a decision tree. As we noted in [13] this result is expected because descriptions from a lazy learning method only refer to an area around the problem that is being solved.

Concerning explanations, we confirmed the feasibility of using explained case-based reasoning for knowledge discovery. Explanations of LID are easily understood by domain experts since these descriptions involve the same attributes as the ones used to represent domain objects, and they are discriminant.

5. Related work

The techniques most commonly used for knowledge discovery are clustering methods. In [20] the reader can find a survey of these methods and a classification of them. Unlike LazyCL, which deals with symbolic data, most clustering methods work better with numerical data since they have their roots in statistics. These numerical methods group objects taking into account both the similarity between the objects included in a cluster and the dissimilarity between objects in different clusters. A different approach is the one taken by *conceptual clustering* where the goal is to build a compact and understandable concept description for each cluster. This description is commonly represented by means of a conjunctive logical expression. Clearly, LazyCL can be classified as a conceptual clustering method, since cluster descriptions are designed to be understood by a human expert. One relevant difference between LazyCL and other conceptual clustering methods is that some methods begin with random seeds from which they construct a concept hierarchy, whereas LazyCL begins with a completely random clustering and the final clusters can be non-disjoint. In fact, this random initialization could be related to clustering methods such as SOM [18] or *K*-means [21] which begin with random seeds and then converge to a correct clustering. In future work, we plan to analyze the convergence of LazyCL in more detail.

SUBDUE [22] is a knowledge discovery system that can deal with structured data. This structural information is represented as a graph where objects and data map to vertices and relationships and attributes map to edges. LID, the problem-solving method used by LazyCL, can also deal with structured data (see for instance [23]), so LazyCL should have no problem with data of this kind. This issue will be the focus of future experiments.

ITERATE [24] is a conceptual clustering method designed to discover interesting patterns from data. The algorithm followed by ITERATE has 3 steps: (1) classify the data using a category utility function; (2) generate an initial partition of the data from this classification; and (3) iteratively redistribute data objects to achieve maximally separable clusters. The idea is that the initial partition influences the final clustering; therefore if one starts with groups of similar objects, the clustering will be better. In LazyCL the initial partition is randomly generated, but the clusters are correctly constructed, and their descriptions are similar to those obtained by a decision tree built on labeled objects. This aspect will be the subject of future research.

One of the early methods for conceptual clustering is COBWEB [25] which defines a tree whose nodes represent concepts. Each concept is described as a set of attributes and the values of these attributes represent the probability that an object takes a value. During the classification process, COBWEB uses a heuristic measure called *category utility* [26] and categories of concepts to guide the search. The similarity between a new example and a one node description is assessed by means of a trade-off between intra-class similarity and inter-class similarity. Cluster descriptions produced by LazyCL do not form any kind of structure, mainly because our goal was to support dermatologists in finding a domain theory for classifying melanomas *in situ*. However, it would be possible to construct a graph relating the cluster descriptions in the way that COBWEB does.

ClTree [27] builds a decision tree in which leaves represent the clusters. However, to be able to build a decision tree the data must belong to some class. For this reason, prior to the construction of the tree, available data are classified as belonging to a class, say *A*. A set of random objects are created and labeled as belonging to another class *B*. The process is based on the assumption that available data (i.e., the objects in *A*) have a uniform distribution, since it is possible to define clusters on them, whereas objects in *B* are created in a non-uniform way. ClTree and LazyCL use a measure based on the information gain and both work on artificial clusters. However, initial clusters of LazyCL are randomly created without any assumption of the uniformity of data.

LazyCL was conceived as a method for knowledge discovery and, as such, it is closely related to clustering methods. Like clustering methods, LazyCL builds clusters and describes them by means of symbolic descriptions that can be seen as prototypes. In this respect, LazyCL can be related to CBR approaches such as for instance [28–30] which use clustering techniques to organize the case memory. The main difference between these CBR approaches and LazyCL is that our goal is neither to obtain an efficient retrieval of similar cases nor to do maintenance tasks, but to support experts to discover a domain theory by producing understandable descriptions. However, a great deal can be learnt from these CBR approaches.

In [28] Perner builds a hierarchy of prototypes described by the attributes that are common to the cases represented by each prototype. This kind of prototype seems to be intuitive enough to understand the structure formed by the concepts of a domain.

Because explanations built by LID are also based on attributes shared by a set of cases, the cluster' descriptions produced by LazyCL can be seen as prototypes. In the future we plan to provide a structure to the cluster' descriptions, since we believe that it will help experts understand better the relations between the domain concepts. In fact, our goal would be to use prototypes in the sense of Schmidt and Gierl [31] who use them as an intermediate level between general domain knowledge and particular cases.

The ProCaseMiner system [29] is another interesting tool. Its goal is to build a case base from biomedical literature. Although correct classes are not known in ProCaseMiner, the use of a concept ontology simplifies the construction and the abstraction of clusters. It is no possible to do something similar in the domain of melanomas *in situ* since there is no clear definition of possible classes. However, experiments performed using LazyCL on standard databases, where the classes are well defined, suggest that in domains with a clear ontology of classes it may be a useful tool.

Vague concepts are very common in medical domains. In particular the experience of dermatologists shows that it is difficult to determine the characteristics of the classes of melanomas with a great degree of accuracy; in fact, they suspect that certain attributes of melanomas may differ according to their state of development. This produces vagueness in their attempts to define a class of melanoma. Therefore a future improvement of LazyCL would be the use of fuzzy sets. Among other studies, Portinale and Montani [32] proposed the use of fuzzy sets during CBR retrieval. This is an interesting approach that could be introduced in LID during the process of constructing the explanations that will serve as the basis for the clusters. The result of using fuzzy sets will be a domain theory admitting fuzzy concepts.

6. Conclusions

In this paper we present LazyCL, a procedure for knowledge discovery based on the explanations produced by a lazy learning method. In LazyCL, a lazy learning method called LID is used to form symbolic descriptions of clusters. Our approach is based on the hypothesis that, because explanations are generalizations, they can be used as domain theory. Although this domain theory is lazy – hence it does not cover all the space of known examples – it can play the same role as domain theory obtained from eager learning methods. However, the application of eager (and, in general, supervised) learning methods requires cases to have a class label. Thus, first LazyCL randomly creates clusters and then applies LID to them with the leave-one-out method. The analysis of relationships between explanations produces an understandable clustering of the dataset.

The result of LazyCL is a set of clusters described by symbolic descriptions that the domain expert can analyze. The best situation is when clusters have empty intersection, since this means that all the known cases can be clearly separated. An interesting situation arises when some clusters have non-empty intersection, although their respective descriptions are not related. This situation can be interpreted, for example, in medical domains as a patient having a set of symptoms that are consistent with more than one disease. From the general point of view, this is an opportunity for the expert to discover relations between classes. For instance, depending on the domain, a possible result of this situation could be to decide that clusters with non-empty intersection are actually the same cluster.

Preliminary experiments of LazyCL on classification of melanomas demonstrated the potential of the approach, since experts "discovered" groups of melanomas. The explanations are discriminant and easy to understand; they completely justify the clusters

created, and they possess the attributes that the experts conjectured in advance would be the most relevant for describing groups of melanomas. Therefore our results seem to confirm the experts' theory about a possible classification of melanomas. Experiments on some UCI Machine Learning Repository data sets confirm that LazyCL produces well-formed clusters, in spite of the randomness of the process.

LazyCL does not need to choose the number of desired clusters in advance – as is the case in most of the clustering approaches – since the process seems to converge to a correct clustering independently of the initial number of clusters. This result is surprising due to LID's dependence on the correct partition, and is an interesting aspect to analyze in the future. LID uses a heuristic measure based on the information gain (the LM distance) which compares the partition induced by a particular attribute with the correct partition. Clearly, the correct partition plays a crucial role because the attribute chosen as the most relevant depends on it. Nevertheless, in LazyCL, where the clusters and the cases contained in these clusters have been randomly selected, the correct partition is not actually the "correct" one. In spite of this, the final clustering is composed of well-formed clusters (at least for the data sets we used in our experiments). We want to investigate, from a theoretical point of view, why this correct partition does not influence the final result of clustering.

Currently, LazyCL cannot be applied for data mining because the leave-one-out method is impractical for huge data sets. A way to reduce the cost and make the process scalable could be to select a subset of cases in which the leave-one-out method can be applied. This process would be similar to the divide-and-conquer approach for clustering defined in [33]. The idea would be to select one case and try to classify it. The description obtained by LID, say d_i, in classifying that case can be used as pattern to filter all cases satisfying d_i. Therefore it will not be necessary to process the whole case base.

Another direction of future work could be to experiment with lazy learning methods other than LID (for instance with Lazy Decision Trees), in order to make LazyCL independent of the problem solver.

Finally, from a theoretical point of view, we will search for a formal justification of the convergence of LazyCL to a correct clustering. The goal is to investigate why, with an initial random clustering and using generalizations and the relations between them, the process always converges to a correct clustering.

References

[1] Friedman RJ, Rigel DS, Kopf AW. Early detection of malignant melanoma: the role of physician examination and self-examination of the skin. CA-A Cancer Journal for Clinicians 1985;35:130–51.

[2] Puig S, Argenziano G, Zalaudek I, Ferrara G, Palou J, Massi D, et al. Melanomas that failed dermoscopic detection: a combined clinicodermoscopic approach for not missing melanoma. Journal of Dermatologic Surgery 2007;33(10): 1262–73.

[3] Hofmann-Wellenhof R, Blum A, Wolf IH, Zalaudek I, Piccolo D, Kerl H, et al. Dermoscopic classification of Clark's nevi (atypical melanocytic nevi). Clinics in Dermatology 2002;20(3):255–8.

[4] Argenziano G, Zalaudek I, Ferrara G, Hofmann-Wellenhof R, Soyer HP. Proposal of a new classification system for melanocytic naevi. British Journal of Dermatology 2007;157(2):217–27.

[5] Leake DB, McSherry D. Artificial Intelligence Review Special Issue on Explanation in Case-Based Reasoning 2005;24(2).

[6] Armengol E, Plaza E. Lazy induction of descriptions for relational case-based learning. In: De Reaedt L, Flach P, editors. Proceedings of the 12th European conference on machine learning 2001, number 2167 in Lecture Notes in Artificial Intelligence. Springer; 2001. p. 13–24.

[7] Armengol E. Usages of generalization in CBR. In: Weber RO, Richter MM, editors. Case-based reasoning and development. Proceedings of the international conference on case-based reasoning 2007, number 4626 in Lecture Notes in Artificial Intelligence. Springer-Verlag; 2007. p. 31–45.

[8] Bareiss ER, Porter BW, Wier CC. Protos: an examplar-based learning apprentice. International Journal of Man-Machine Studies 1988;29(5):549–61.

[9] Bergmann R, Stahl A. Similarity measures for object-oriented case representations. In: Smyth B, Cunningham P, editors. Preedings European workshop on case-based reasoning, EWCBR-98. Springer-Verlag; 1998. p. 25–36.

[10] Friedman JH, Kohavi R, Yun Y. Lazy decision trees. In: Proceedings of the thirteenth national conference on artificial intelligence and eighth innovative applications of artificial intelligence conference. AAAI/IAAI-96, vol. 1; 1996. p. 717–24.

[11] Mitchell TM, Keller RM, Kedar-Cabelli ST. Explanation-based learning: a unifying view. Machine Learning 1986;1(1):47–80.

[12] Quinlan JR. Induction of decision trees. Machine Learning 1986;1(1):81–106.

[13] Armengol E. Building partial domain theories from explanations. Künstliche Intelligenz 2008;2/08:19–24.

[14] Frawley WJ, Piatetsky-Shapiro G, Matheus CJ. Knowledge discovery in databases—an overview. AI Magazine 1992;13:57–70.

[15] Blake CL, Merz CJ. UCI repository of machine learning databases; 1998.

[16] López de Mántaras R. A distance-based attribute selection measure for decision tree induction. Machine Learning Journal 1991;6(1):81–92.

[17] Fornells A, Armengol E, Golobardes E, Puig S, Malvehy J. Experiences using clustering and gernalization for knowledge discovery in melanomas domain. Transactions on Machine Learning and Data Mining 2008;1(2):49–65.

[18] Kohonen T, Schroeder MR, Huang TS, editors. Self-organizing maps. Secaucus, NJ, USA: Springer-Verlag New York, Inc.; 2001.

[19] Armengol E, Plaza E. Bottom-up induction of feature terms. Machine Learning 2000;41(1):259–94.

[20] Berkhin P. Survey of clustering data mining techniques. Technical report. San Jose, CA: Accrue Software; 2002.

[21] Kanungo T, Mount DM, Netanyahu NS, Piatko CD, Silverman R, Wu AY. The analysis of a simple k-means clustering algorithm. In: Symposium on computational geometry. 2000. p. 100–9.

[22] Jonyer I, Holder LB, Cook DJ. Graph-based hierarchical conceptual clustering. International Journal on Artificial Intelligence Tools 2001;10(1–2):107–35.

[23] Armengol E, Plaza E. Remembering similitude terms in case-based reasoning. In: Perner P, Rosenfeld A, editors. 3rd International conference on machine learning and data mining MLDM-03, number 2734 in Lecture Notes in Artificial Intelligence. Springer-Verlag; 2003. p. 121–30.

[24] Biswas G, Weinberg JB, Fisher DH. Iterate: a conceptual clustering algorithm for data mining. IEEE Transactions on Systems Man and Cybernetics Part C Applications and Reviews 1998;28:219–30.

[25] Fisher DH. Knowledge acquisition via incremental conceptual clustering. Machine Learning 1987;2(2):139–72.

[26] Gluck MA, Corter JE. Information, uncertainty, and the utility of categories. In: Program of the seventh annual conference of the cognitive science society. 1985. p. 283–7.

[27] Liu B, Xia Y, Yu PS. Clustering through decision tree construction. In: CIKM'00: Proceedings of the ninth international conference on information and knowledge management; 2000. p. 20–9.

[28] Perner P. Case-base maintenance by conceptual clustering of graphs. Engineering Applications of Artificial Intelligence 2006;19:381–93.

[29] Bichindaritz I. Prototypical case mining from biomedical literature for bootstrapping a case base. Applied Intelligence 2008;28(3):222–37.

[30] Bellazzi R, Montani S, Portinale L. Retrieval in a prototype-based case library: a case study in diabetes therapy revision. In: Smyth B, Cunningham P, editors. Advances in case-based reasoning, 4th European workshop, EWCBR-98. 1998. p. 64–75.

[31] Schmidt R, Gierl L. Experiences with prototype designs and retrieval methods in medical case-based reasoning systems. In: Smyth B, Cunningham P, editors. Advances in case-based reasoning, 4th European workshop, EWCBR-98. 1998. p. 370–81.

[32] Portinale L, Montani S. A fuzzy logic approach to case matching and retrieval suitable to SQL implementation. In: ICTAI'08: Proceedings of the 20th IEEE international conference on tools with artificial intelligence; 2008. p. 241–5.

[33] Jain AK, Murty MN, Flynn PJ. Data clustering: a review. ACM Computing Surveys 1999;31(3):264–323.