Improving Knowledge Discovery Using Domain Knowledge in Unsupervised Learning

Javier Béjar*

Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya.
c/ Jordi Girona 1-3. 08034 Barcelona, Spain.
Phone: 34 3 4015653, Fax: 34 3 4017014
bejar@lsi.upc.es

Abstract. Using domain knowledge in unsupervised learning has shown to be a useful strategy when the set of examples of a given domain has not an evident structure or presents some level of noise. This background knowledge can be expressed as a set of classification rules and introduced as a *semantic bias* during the learning process.
In this work we present some experiments on the use of partial domain knowledge in conceptual clustering. The domain knowledge (or domain theory) is used to select a set of examples that will be used to start the learning process, this knowledge has not to be complete neither consistent. This bias will increase the quality of the final groups and reduce the effect of the order of the examples. Some measures of stability of classification are used as evaluation method.
The improvement of the acquired concepts can be used to improve and correct the domain knowledge. A set of heuristics to revise the original domain theory has been experimented, yielding to some interesting results.

1 Introduction

The use of unsupervised learning to discover useful concepts in sets of non classified examples allow to ease the labour of data analyst in data mining tasks or any other task that involves the discovery of useful descriptions from data. Tools that help to this labour and that increase the quality of the knowledge obtained are very desirable.

In this work we present the methodology used by LINNEO$^+$ [1,2,3], that has been extended to use domain knowledge in order to *semantically bias* a conceptual clustering algorithm [7,5]. This knowledge helps to obtain more stable classifications and more meaningful concepts from unclassified observations.

It is shown, also, that little knowledge can produce considerable gain, despite of the ambiguity or the partial incorrectness of the knowledge. This ambiguity can be also solved using the improved classifications, performing specializations or generalizations that correct the domain knowledge.

* This work has been financed by UPC grup precompetitiu PR99-09

R. López de Mántaras, E. Plaza (Eds.): ECML 2000, LNAI 1810, pp. 47–54, 2000.

2 LINNEO+

LINNEO+ [1] is a tool oriented to discover probabilistic description of concepts from unclassified data. It uses an unsupervised learning strategy and incrementally discover a classification scheme from the data. The expert has to define dataset of observations to model the domain and also defines a set of attributes relevant to the classification goal intended. The expert is allowed to represent attributes by means of quantitative and qualitative attributes.

The strategy of classification in based on a distance measure and a criterion of membership. The algorithm that its used is a variation of the nearest neighbour algorithm [6] augmented by the use of probabilistic prototypes in order to describe the discovered clusters [5].

The similarity function used is a generalization of hamming distance usually used by other conceptual clustering algorithms [7]. The aggregation algorithm builds clusters of similar objects given a initial parameter that we call *radius* that selects the level of generality of the induced concepts. This radius is selected heuristically by trial and error. A detailed description of the algorithm can be found in [1,3]

This methodology has been successfully applied to some real domains as mental illnesses [11], marine sponge classification [1] and discovery and characterization of fault diagnose in wastewater treatment plants [12]

3 Using a Domain Theory

In unsupervised learning the description of the observations usually is not enough to build a set of concepts. The noise of the observations, the existence of irrelevant descriptors or the non homogeneity of the sampling of observations can deviate the learning process from a meaningful result. It is desirable, thus, a *guide* from a higher level of knowledge to assure the success of the acquisition.

In our methodology, we allow the expert to define as Domain Theory (DT) as a group of constraints guiding the inductive process. Therefore, the DT semantically *biases* the set of possible classes. This DT acts just as a guide; it does not need to be complete. It could be very interesting for the experts to play with several definitions of DT as they could model several levels of *expertise* or to obtain different classifications using different points of view or bias. The expert is allowed to express his DT in terms of rules that determine the definition of a part of the definition of classes he already knows to exist. A rule is composed by an identifier and some constraints, a set of conditions that elements must fulfill in order to belong to the defined identifier. This conditions are expressed by simple selectors including conditions as =, >, < or membership to a range of values for an specific attribute.

3.1 Biasing with a Domain Theory

If the expert is able to build a DT, it is possible to use this knowledge to *bias* the classification using the constraints as a guide to preprocessing the dataset.

Even in poor defined domains the expert knows that to ignore some attributes for certain classes can be useful, because those attributes are not relevant predicting class membership. In the same way, the expert, knows that there are other attributes, or their conjunction, that could be used, with a certain degree of confidence, to try to predict class membership. The idea is to create a partition of the dataset using the rules defined by the expert in meaningful parts, the objects with some knowledge about its relation (those described by the rules). Those objects that not fulfill none of the rules are treated as without Domain Theory.

The treatment of the dataset previous to the classification is as follows:

- All the objects that satisfies a rule (R_i) are grouped together ($\mathcal{S}_{R_i}$) (the expert could give more than one rule for the same class).
- If the rules are too general, two or more rules could select the same object, in this case a *special* set is created for this objects. The objects are tagged with the conflicting rules.
- All the objects that do not accomplish any rule are grouped in a *residual* set.

After this process at maximum,it is generated $r + 2$ sets of objects, where r is the number of sets that the expert has constrained.

Each one of these sets, except for the *special* and the *residual*, is classified separately and, eventually, it is created at least a class for each. Then a new classification process begins with the centers of these classes as seeds of the new classification and the rest of objects. In this process new classes can be formed corresponding to classes not described by the rules.

The *bias* is obtained by the reordering and previous grouping of the observations in a meaningful scheme, rather than the random order of the unbiased process. This yields a more meaningful set of classes, more in the idea that the expert has of his domain structure. This avoids also the instability induced by the ordering of the observations.

4 Experiments with a Domain Theory

In order to test the effect of a domain theory in the process of classification, we have written a small set of rules for the *Soya bean* domain [3] (11 rules, see table 1 for example rules) to bias the resulting classes. These rules have been built by hand, inspecting the prototypes of the classes of a unbiased classification, extracting the attributes more relevant. This set of rules is neither complete nor consistent, because we just want to show that only a small piece of domain knowledge is enough to improve the stability, and therefore the quality, of a classification. These rules select 130 observations from a total of 307.

The experiment was carried out by comparing two sets of 20 random ordered classification using **LINNEO**$^+$ of the *Soya bean* dataset [8] obtained from the UCI Repository of Machine Learning Databases and Domain Theories [9]. The

Table 1. A Soya Bean Domain Theory

((= (diseased) fruit-pods) (= (colored) fruit-spots) (= (norm) seed) -> frog-eye-leaf-spot)	((= (norm) fruit-pods) (= (tan) canker-lesion) (= (lt-norm norm) precip) -> charcoal-and-brown-stem-rot)
((= (abnorm) seed) (= (tan) canker-lesion) -> purple-seed-stain)	((= (lower-surf) leaf-mild) -> downy-mildew)

first without use of domain theory, the second using our set of rules as domain theory.

In order to compare the resulting classifications we have developed an algorithm that provides a measure of the differences between two classifications [1,2,4]. This measure, that we call *structural coincidence*, is used to provide a value for the stability of each set of classifications as the mean of the difference of each pair of classifications in the set. Among these differences, it is taken in account the coincidence of objects in the same group and the number of classes of each classification.

Another measure of stability that is used in the comparison is based on the coincidence of the pairs of associations of observations between two partitions described in [4], this measure decreases with the similarity.

The stability of a classification of the Soya Bean dataset without the DT is 77.6% for the first measure and -1013.4 for the second. The stability using the DT increases to 91% for the first measure and -4285.6 for the second. A cross comparison between the two sets of classification yields a value of 79.9% for the structural coincidence. This value has been calculated comparing each class resulting from each method with all the others and averaging. The interpretation of this value is that the classifications using the domain theory are similar to those created without using a *bias* but much more stable. Applying this technique to other datasets yields similar results [3].

In the light of these results, we can say that the use of domain knowledge in unsupervised learning reduces the problem of obtaining meaningless groupings and also reduces the instability induced by an improper input order.

A similar technique for biasing an unsupervised algorithm has been applied in order to build concepts hierarchies successfully [13]. This encourages to apply a similar strategy to bias other incremental conceptual clustering algorithms.

5 Domain Theory Revision

Due to that the domain theory that the expert gives for the *biasing* process could be inconsistent or incomplete, it is worth to improve it in some automatic way. Some EBL systems try to improve incomplete or incorrect domain theories using labeled examples in order to fix the errors [10]. Our system is unsupervised, so we have to trust the classes formed during the classification process and the

source of detected errors only can be from the use of the DT previous to the classification.

We have been experimenting with some heuristics for theory revision. These heuristics are very conservative, they only try to discover the minimum set of changes that improve the selectivity of the rule and maintaining the consistency with the obtained clusters. The heuristics only can revise the clauses applied to one attribute with the operators $=$, $\neq$, $>$, $<$ and `range`.

This revision has two parts. When a dataset is classified using the domain theory two kind of rules may appear if we observe the consistency of the resulting partitions. There is a set of rules that selects a definite set of objects that no other rule selects, we call this set *non collision rules*. There is another set of rules whose sets of objects intersect among them. These are *ambiguous* rules and the multiple selected objects can not be assigned to a definite set. So, the revision can be done separately for each set of rules. First, to improve the non collision rules trying to generalize them or by deleting superfluous conditions. Second, to correct the ambiguous rules, trying to specialize them in order that no object is selected by more than one rule. A more extended description of this process can be found in [1]

5.1 Revision of Non Collision Rules

These rules can be treated separately, because all of them have their own set of examples, classified in one or more groups. The objective of this process is to fit the rules with the groups but not to select objects from other groups.

This improvement has two phases. Firstly, the phase of specializing. Some rules can have an extension so broad, or excessive disjunctive conditions, that can prevent a later generalization. So, some of these conditions can be restricted or dropped in order to be consistent with the values of the objects in the classes selected by the rules. This can be done for example eliminating modalities that do not appear in the values of a class from an equal ($=$) clause, or by changing the $<$ and $>$ clauses to the upper and lower bound of the attribute in the prototype of the class respectively.

The second phase is generalizing. Not all the objects from a class are selected by the rules that had generated it. It is desirable that the rules cover the maximum number of objects of this class in order to be more descriptive of the class. To achieve this we generalize the conditions of the class extending its ranges or dropping conjunctions, only if these changes are consistent with the rest of classes of the dataset. This generalization can be done for example by introducing more modalities in a equal ($=$) condition, modalities that appear in the class and have not been used by the expert or to change the clause to a `range` clause with the bounds of the attribute in the prototype. Also, it is possible to test the effect of eliminate each one of the conditions of the rule.

The corrected rules can help the expert to refine his knowledge.

Table 2. An ambiguous Soya Bean Domain Theory

((= (lt-normal) plant-stand) (= (severe) severity) (= (none) int-discolor) -> phytophthora-and-rhizoctonia-root-rot)	((= (lt-80%) germination) (= (norm) plant-growth) -> bacterial-pustule)
((= (no) lodging) (= (tan) canker-lesion) -> herbicide-injury)	((= (lower-surf) leaf-mild) -> downy-mildew)

5.2 Revision of Ambiguous Rules

This set corresponds to rules too general or to classes where the expert can not differentiate accurately. To treat these rules it is necessary to calculate what groups of rules are in conflict and what objects are the conflictive ones.

As information to correct those rules, it is taken in account the classes that group the conflictive objects and the rule that has formed this group. It is a logical assumption to assign the conflictive objects to the rule that has formed the class the objects belong to.

The objective is to specialize each conflictive rule using as constraints the observations that it has not to select. To do this a rule is specialized constraining its conditions or adding new conditions that exclude the conflicting observations.

The selection of the conditions is done by choosing the attributes from the classes (of the rule) that have values not present in the non desired observations. With these attributes, it is possible to construct new clauses in order to specialize the rule. If the attribute is quantitative, a clause that selects only the values between the range present in the classes can be constructed. If the attribute is qualitative, a clause that test that the modalities are only the present in the classes can be constructed.

The specialization process is done by selecting some of the candidate clauses. Each clause is tested with the clauses from the rule. The clauses selected are those that reduce the most the selection of non desirable observations and maintain the selection of correct observations.

After the specialization process a generalization can be done by testing if some of the original conditions of the rule are unnecessary because the new conditions.

6 Evaluating the Revised Domain Theory

The same dataset has been used in order to evaluate the heuristics, but and artificially ambiguous DT has been constructed [3] (see table 2 for example rules). There are, in this case, 12 rules (some of them grouping more than one category) that select 178 objects (33 in the special class) from a total of 307.

Concretely the rule number 9 has the following collisions: rule 1 (6 observations), rule 2 (2 observations), rule 5 (2 observations), rule 6 (1 observation),

Table 3. The corrected rules

((= (lt-normal) plant-stand) (= (severe) severity) (= (brown dk-brown-blk) canker-lesion) -> phytophthora-and-rhizoctonia-root-rot)	
((= (lt-80%) germination) (= (norm) plant-growth) (= (absent) leaf-mild) (= (absent brown-w/blk-specks) fruit-spots) (= (norm) seed-size) -> bacterial-pustule)	((= (no) lodging) (= (w-s-marg no-w-s-marg) leafspots-marg) (= (90-100%) germination) -> herbicide-injury)

Table 4. Number of objects selected

Rule	Before	After	Rule	Before	After	Rule	Before	After
Num 1	18	24	Num 2	17	17	Num 3	10	17
Num 4	31	31	Num 5	8	8	Num 6	5	6
Num 7	6	6	Num 8	2	2	Num 9	41	41
Num 10	11	14	Num 11	7	10	Num 12	6	10

rule 10 (5 observations), rule 11 (3 observations), rule 12 (4 observations); the rule number 8 has the following collisions: rule 3 (8 observations), rule 10 (2 observations). The total number of conflicting objects is 33. After the correcting process the number of collisions has been reduced to 7 objects, specializing 3 rules as can be seen in table 3. The number of objects selected by each rule after and before the correction can be seen in table 4.

The structural coincidence with the ambiguous rules is 88.9% With the corrected rules it has been increased slightly. The value for the structural coincidence has been increased to 89.6%. It is not expected a great increase of stability because the number of selected objects by the domain theory has not been increased, but the gain of stability is maintained.

Applying this technique to other datasets with expert build domain theories yields similar results, a slightly increase of stability is obtained, but the selectivity of the rules is increased ([1]).

7 Conclusions

It has been shown that the use of domain knowledge as semantic bias in a unsupervised learning algorithm increases the quality of the result. The domain knowledge has not to be perfect, can have some ambiguities or inconsistencies, an increase of stability of the results could still be achieved.

The approximation used to bias learning is enough algorithm independent to be exported to other incremental conceptual clustering algorithms.

The fix and revision of the domain knowledge can also be done, obtaining a benefit from the better classification. Ambiguities can be detected and corrected observing the nature of the obtained groups and generalizing and specializing the knowledge in order to fit the description of the concepts.

References

1. J. Béjar. *Adquisición de conocimiento en dominios poco estructurados.* PhD thesis, Dep. of Software. Universitat Politécnica de Catalunya, 1995. 47, 48, 50, 51, 53
2. J. Béjar, U. Cortés, and R. Sangüesa. *Experiments with Domain Knowledge in Knowledge Discovery* 1^{st} International Conference on the Practical Application of Knowledge Discovery and Data Mining, London, 1997. 47, 50
3. J. Béjar, U. Cortés. *Experiments with Domain Knowledge in Unsupervised Learning: Using and Revising Theories* . Revista Iberoamericana de Computación. Computación y Sistemas 1 (3). pags 136-144. 1998. 47, 48, 49, 50, 52
4. D. Faith, L. Belbin. *Comparison of classifications using measures intermediate between metric dissimilarity and consensus similarity. Journal of Classification*, 3:257–280, 1986. 50
5. D. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2:139–172, 1987. 47, 48
6. A. K. Jain, R. C. Dubes. *Algorithms for clustering data.* Prentice Hall, 1989. 48
7. R. Michalski, R. E. Steep. *Learning from observation: Conceptual Clustering*, *Machine Learning an A.I. Perspective*, pages 331–363. Ed. Tioga, Palo Alto, California, 1983. 47, 48
8. R. S. Michalski, R. L. Chilausky. Learning by being told and learning from examples: An experimental comparison of the two methods of knowledge acquisition in the context of developing an expert system for soybean disease diagnosis. *International Journal of Policy Analysis and Information Systems*, 4(2), 1980. 49
9. P. M. Murphy, D. W. Aha. UCI repository of machine learning databases. Irvine, University of California, Department of Information and Computer Science, 1994. 49
10. D. Ourston and R. J. Mooney. Theory refinement combining analytical and empirical methods. *Artificial Intelligence*, 1993. 50
11. E. Rojo. *Aplicación del software LINNEO a la clasificación de transtornos mentales.* PhD thesis, Facultat de Medicina. Universitat de Barcelona, 1993. 48
12. M. Sànchez, U. Cortés, J. Béjar, J. de Gracia, J. Lafuente, and M. Poch. Concept formation in WWWTP by means of classification techniques: A compared study. *Applied Intelligence* 7, pags 147-165, 1997. 48
13. L. Talavera, J. Bejar. Integrating Declarative Knowledge in Hierarchical Clustering Tasks. Third Symposium on Intelligent Data Analysis. Pages 211-222, Amsterdam, The Netherlands. LNCS vol. 1642. Springer Verlag, 1999. 50