ALGORITHMS FOR LEARNING LOGICAL FORMULAS

Joel QUINQUETON and Jean SALLANTIN

INRIA
B.P. 105
78153 LE CHESNAY CEDEX (FRANCE)

ABSTRACT

We present here a learning technique which is both statistic and syntactic, by using simultaneously logical operators and counting procedures. Its modular structure makes it usable for creating the necessary redundancy for controlling the generalization of the formulas.

0. INTRODUCTION

In Data Analysis, a learning problem is generally stated as, either a discrimination problem, or a regression one. Some inductive methods [4] use metric concepts. But most of the Artificial Intelligence methods [2,3] are purely syntactic, i.e. they use concepts of Formal Logic.

Then a problem arises, which is the generalization one [5], i.e. how to apply the found logical rules to examples which are not in the training set. This problem is more and more studied, and interesting solutions have been found, based upon the idea of controlling the generalization [6,8].

We describe in this work a learning technique which builds rules to describe a given training set. Each of them can be used as an "opinion" about the training set. Then a "rule storming" is performed to complete the generalization.

1. OUTLINE OF THE METHOD

We consider a set of objects, called the training set. This set is described by several "describers", which are questions (or binary variables) with a value for *every* object.

Let us take one of these describers and call it the "variable to forecast". The aim of the method presented here is to find a combination of the others describers identical to the variable to forecast.

The method we propose consist on the iteration of an algorithm made of 3 modules : expansion, selection, compression.

Expansion step consist on combining each couple of describers with a logical operator, in order to obtain a new set of describers.

Selection consists on eliminating the describers which are not "similar" enough to the variable to forecast.

Compression step then classifies the describers according to their similarities, and summarizes each class by one(or a few) describer.

We shall now describe each of these parts.

2. EXPANSION

This step consist on combining each couple of describers. Then, the operator must be associative, so that the combinations of 3 or more describers are nothing but successive 2 by 2 combinations. On the other hand, the permutation of 0 and 1 ("true" and "false") must not disturb the result of expansion.

These constraints suggest two possible operators : the logical conjunction "and", and the logical equivalence "id".

With the conjunction, 4 describers are built for each couple of initial describers :

a and b ; (non a) and b ; a and (non b) ; (non a) and (non b).

After that, we have to check the consistency, i.e. if (d) is in on formula, then (non d) is not in another one.

For the logical equivalence, we have the following properties :

a id b = (non a) id (non b) ;
(non a) id b = a id (non b) = non (a id b).

According the remarks made at the beginning of this paragraph, it is only necessary to build "a id b".

It is easy to check the fitting of these operators with the constraints described at the beginning of this paragraph.

3. SELECTION

The aim of the selection step is to compare a describer to the variable to forecast, in order to select the describers which can be frucfully expanded again. The most natural way of comparison between 2 binary variables is to look at the list (N00 N01 N10 Nil) of the co-occurence frequencies for the different values of the variables, i.e. of the describer to be selected and the variable to forecast.

Several criteria are then possible for the selection. We consider here two kinds : overlapping criteria and information theory criteria.

An overlapping means that the describer has one value for at least a part of the objects from one

class and the other value for at most a part of the objects of the other class. The corresponding thresholds are given by the user.

The Information Theory criteria are different of the previous one, in the sense that they are not used in the same way. They are information measurements on the describers, which are then ordered according this measure, and the k best ones are selected, for a value k which is choosen by the user.

Several criteria are possible. For instance, we can use the Kullback's divergence, the Mahalanobis distance or the contingency-khi 2 criterion.

4. COMPRESSION

This step consists on summarizing the set of the selected describers, regarding their inter correlations, which are measured by a given function.

Then, we have to perform an automatic clustering of the describers into k groups.

4.1. Optimization of a clustering

In this paragraph, we state the problem of optimizing, for a given criterion, a k-class clustering. As we previously noticed, we suppose that a distance has been chosen to measure the decorrelation between the describers.

Then, the criterion to optimize is the sum of the distances of the describer i to the describers which are in the same class. Let D(i j) be the sum of distances between describer i and the describers belonging to class j. Let j (i) be the class which contains describer i. Let us state :

W(i) = D(i j(i)) - min (D(i j) j=l,k)
W(iO) = max (W(i) i=l,...)
D(iO jO) = min (D(iO j) j=l,k)

Then, the algorithm deletes iO from its class and appends it to the class jO.

This procedure is repeated until the obtained classification is invariant, i.e. W(i)=0 for all the describers. This algorithm is actually a local optimization one, because we can easily proof that the criterion W(i0) is descreasing at each step.

The initial clustering may be randomly chosen, or given by the user. We shall now use the later possibility to define a strategy for compression.

4.2. The compression algorithm

An interesting aspect of the previous algorithm is that it works even if one class is empty. This remark suggests a strategy for compression, which needs only to input the maximum number of classes.

The algorithm starts with a one-class clustering. Obviously, the value of the criterion is zero in this case.

When the best (k-l)-class clustering has been found by the algorithm we described in the previous paragraph, an empty class is created and we look for the best k-class classification.

The algorithm stops when k is the given maximum value nmc. We can notice that, if the sum of intra-class distances is zero for k < nmc, then the (nmc-k) other classes will remain empty.

Once we have obtained the desired classificatior each class is summarized by one (or a small number) of its elements. We shall describe this point in the following paragraph.

4.3. Summary of a compression

The summary of a compression must depend upon the chosen distance. The purpose of the distance is to compare 2 describers. Then, it will be defined with the list of co-occurences of values of the describers (N00,N01,N10,N11).

In the case of the equivalence distance min (N00+N11,N01+N10), we can choose any element of each cluster, because they are supposed to be logically equivalent, except for a few objects.

In the case of the comparability distance min (N00,N01,N10,N11), the characteristic of the elements of the same cluster is to be comparable to each other, except for a small number of objects. This kind of relationship may be summarized by ordering the describers, and then using a dichotomic decision tree to compress.

Then, this kind of compression can be viewed as an "unfolding" (dimensionality reduction) of the training set, and we studied it in some previous works [7].

5. END CRITERION

As the aim is to find a formula which is, on the training set, logically comparable to the variable to forecast, there are several ways of stopping conveniently the algorithm :

- maximum number of iterations (it is careful)
- emptiness of the describers list (it may happen after a bad choice of the selection parameters)
- one (or several)of the built describers is sufficiently correlated to the variable to forecast.

These criteria are applied in the previous order. We can add some more, in order to detect the case when it is useless to continue. For instance, if the describers list remains identical after a new expansion, selection and compression ("stability" criterion), it is clear that further iterations will give the same result.

6. GENERALIZATION BY "RULE STORMING"

The generalization consists on decision making outside the training set. Then, the logical rules built with the previous algorithm can be considered as "opinions" about the training set, each onebeing related to the choice of a particular describer as variable to forecast.

The generalization itself consist on performing a "rule storming" on these opinions. Let us summarize this idea. According Michalski [5], a generalization is a filter (in the topological sense) on the space of objects.

Let w be an object and 0P(i) the i-th opinion Then, 0P(i,w) is the new object produced by applying the rule 0P(i) to w. Several cases are possible :

0P(i,w) = w (at least on the training set).
0P(i,w) = w\ then OP (i,w') = 0P(i,w)
0P(i,w) = 0 (the rule is not applicable).

Then, the filter is :

V(i) = {w} U (U 0P(i,w)
V(p) = {w} U (U 0P(i,V(p-l))).

The rule storming is then made by a vote at a given level of this filter.

An advantage of this technique is that we actually build a topology, then we need not a discrimination problem to generalize.

7. RESULTS AND DISCUSSION

This technique has been tested on real problems : learning of animal behavior, control problems on nuclear plants, forecasting earthquakes, learning meta-rules for expert systems, decision making in psychology.

In all these applications, only a few (3 to 5) iterations of the expansion-selection-compression were necessary to find the rules. For the generalization, the good decision was made at level V(l) or V(2), but never more.

8. CONCLUSION

The algorithm that we presented in this paper in a first draft of a tool for learning problems. We work now on its enhancement and integration in a complete learning system. We think that the presented results show reasonable efficiency and computing costs. The theoretical background can be found in the field of non classical Logic [1], more precisely non distributive logics.

REFERENCES

[1] J. Fargues "Contribution a l'Stude du raisonnement", These d'Etat, University of Paris 6, May 1983

[2] Y. Kodratoff and R. Loisel "Learning Complex Structural Descriptions from Examples", ICPR 1982 Munich (FR6).

[3] R. Loisel, "Apprentissage de Descriptions Structurelles Complexes", These d'Etat,Paris 6 University, Oct. 1981.

[4] R.S. Michalski "Pattern Recognition as Rule Guided Inductive Inference", IEEE Trans on PAMI, Vol. 2, n° 4$_t$ 1981.

[5] R.S. Michalski, "inductive Learning", Artificial Intelligence, 1983, PP 111-161.

[6] T. Mitchell, "Version Spaces : a Candidate Elimination Approach to Rule Learning", IJCAI 1979. Tokyo (Japon).

[7] J. Quinqueton, "Intrinsic Dimensionality of Ordinal Data", ICPR 1980, Miami Beach (USA).

18] J. Sallantin and J. Quinqueton "Expansion and compression of Binary Data to Build Features by Learning", ICPR 1982, Munich (FRG).