CHARADE : A RULE SYSTEM LEARNING SYSTEM

Jean-Gabriel GANASCIA

Laboratoire de Recherche en Informatique
Bat. 490, Universit6 Paris-Sud, U.A. 410 du C.N.R.S., 91405 Orsay FRANCE

ABSTRACT

Designed for an operational prospect, the CHARADE system automatically learns consistent rule systems from a description language, a set of axioms reflecting the language semantics and a set of examples. The technique advocated below is based on a "generate and test" mechanism where the description space is explored from the more general to the more specific descriptions. Rules and properties to be obtained are translated into exploration procedure constraints thanks to formalization of the learning set with two Boolean lattices.The underlying theoretical framework allows to both justify the heuristics conventionnaly used similarity based-learning and to introduce global properties to be satisfied by a rule system during its construction.

1. INTRODUCTION

In a pragmatic prospect, that of knowledge base construction and maintenance, the global properties of the rule system play a central part. They will assist the specialists at the time of transfer of expertise to prevent errors and possibly correct them. Yet there is a major gap between individual rule juxtaposition and a complete and efficient usable rule system. CHARADE proposes to bridge that gap. It has been designed to detect logical or statistical regularities existing in a set of examples and generate production rule systems which, reflecting such regularities, can operate on commercial inference engines. The method presented is related to the "generate and test" techniques developed by Buchanan (1) and Michalski (3), the description space being explored from the more general to the more specific descriptions. However, instead of considering only one ordering relation, as it is usually the case in learning, we shall distinguish two ordering relations, the first being linked to the inclusions of subsets of the learning set, the other to the logical implications on the description space. In fact, these are two aspects of the generalisation notion, and even if they are complementary and indissociably linked together, they cannot be reduced to one another. Each can be formalised with a Boolean lattice. Taking simultaneously into account these two aspects allows to translate the properties of rules and rule systems to be constructed into constraints for the exploration procedure, which considerably limits the number of vertices to be explored. Such a formalisation of learning with two Boolean lattices establishes a theoretical base for the learning heuristics used in classical systems and still allows the introduction of new properties. Amongst such properties, it is possible to translate the rule system semantic characteristics, as structuration and nature of rules - logical or approximate - as well as the relative relevance of descriptors. Last, completeness, consistency and minimality of the rule systems obtained can be proved. After a precise demonstration of all that differenciates a rule system from a conglomerate of individual rules, we shall study the detail of techniques used to generate rule systems and see how the rule system properties can be translated into this formalism.

2. RULE SYSTEMS

Modularity of production rule systems must not be deceptive. It is not enough for the rules, individually to make sense, they must also, as a whole, meet operational criteria as lack of redundancy and cycle, consistency, completeness, etc. To insure that such criteria are verified knowledge acquisition assistance tools are built and it is to be able to rid from such verifications that we hope to create systems to learn rules from examples. However, in most cases rule learning systems are limited to the acquisition of concept description from examples and counter-examples. To be convinced of it let us study classical similarity based learning system procedure: being a set of examples, E(cl)={el,e2,...,en) and a set of counter-examples, CE(e1)= {ce 1 ,ce2,... ,cep) of a concept *cl* a generalization is looked for: *gl* of el,e2,...,en discriminating cel,ce2,.. .,cep . When *gl* has been found, the rule *gl* —► *cl* is generated.

In spite of the difficulty to introduce a disjunction in the *gl* generalization, the same operation must be done for all the concepts to be learnt. Moreover, once defined concepts *cl* and c2, to define a concept *c3* so that *cl&c2* —► *c3* , descriptors *cl* and *c2* must be introduced in the example descriptions, which means that rule chainings can be learnt only if they are predetermined and fixed. Thus, rules are isolated from one another during acquisition.

Reversely, CHARADE does not refer to the notion of example or counter-example for separated concepts; it considers globally the set of examples, each of them being described by a logical formula, and explores the space of the elementary descriptors present in the examples conjunctions. Be d_1&d_2&-. &d_n one of those, CHARADE looks for descriptors f_1, f_2,••, fp, correlated to the set of examples covered by d_1&d_2&. -&d_n, then it generates rule d_1&d_2&...&d_n -»f_q, ..., f_m, eliminating amongst f], f2, ..., f_p descriptors f_i, already derived from d_1&d_2&. -&d_n through the rule system.

3. DETECTION OF REGULARITIES

It will be considered that from now on, an example is described by a descriptor conjunction. With an example E, we shall call d(E) description of E: d(E) = d_1&d_2&. . .& d_n

Each descriptor d_i will be originally assumed to be either an atomic proposition or the negation of an atomic proposition. The latter restriction may be removed (Cf. (2)) and it may be then possible to process any form of (<attribute><selectorxvalue>) representation, integrating the semantical properties of attributes and selectors in the learning process.

With the restrictive assumptions made above, it is now easy to carry out the elementary operation in the field of learning, that is generalization: E_1 and E_2 are two examples with $d(E_1) = a_1\&a_2\&...\& a_n$ and $d(E_2) = b_1\&b_2\&...\& b_m$. Let us call gen($E_1$, E_2) the least generalization of E_1 and E_2. If semantic relations between atomic descriptors present in these descriptions are not known, gen(E_1, E_2) consists of the conjunction of common descriptors of E_1 and E_2.More formally

gen(E_1,E_2) = &d such that de $\{a_i / i\in [1,n]\ \exists j\in [1,m]\ a_i=b_j\}$

Example: let us assume that:

d(E|) = blue eyes & tall &fair hair,

d(E_2) = blue eyes & short & red hair

It is then easy to calculate gen(E_1, E_2) = blue eyes

If now the learning set is represented by a function TR associating to each descriptor d the set of examples containing d in their description, to say that blue eyes is a generalisation of E_1 and E_2 is easily translated, it means thats $\{E_1, E_2\} \subseteq TR$(blue eyes). The inclusion relation introduced here translates the subsumption of a set of examples under a concept, this can be expressed by a function SUB : SUB(E) = {d such that EcTR(d)}. The two functions TR and SUB allow to go from the description space to the learning set and from the learning set to the description space. Now, combining the two functions one can associate to a description D the set of descriptors TR(SUB(D)) = $\{d_1, d_2, ..., d_n\}$ such $\forall i\in [1,n]\ D\Rightarrow d_i$ d thus, generate directly a production rule. The method that we advocate is based on this principle. As it does not refer to a preferential set of examples, the notion of example and counter-example become void of sense, the category of an example is only one descriptor among others. Moreover, this method generates all the logical correlations and thus allows to detect rule chainings. However, among those correlations, some are useless, others are accidental, the rest of this paper deals with the description of an exploration procedure limited to descriptions D likely to generate an interesting rule.

As the following statements might be very abstract, we shall illustrate them with an example drawn from (4) and modified to meet the requirements of this presentation. This example only intends to give an intuitive content to our presentation; it consists of the description of 9 individuals described by 5 attributes each: Size, Hair colour, Eye colour, Complexion and Class:

el= (size=sh<tt)&(haii*fair)&(eyes=bto
c2=(sizc=^l)&(hair=fair)&(eycs=brown)&(cx)mplexion«matt)&(class«-)
e3= (size=taU)&(hair*red)&(eyes«M
©4=(si2c=short)&(hair=brown)&(cyes=bluc)&(complcxion=matt)&(class=-)
c5= (size = tall)&(hair*brown)&(eycs=bhie)&(complcxion=pale)&(class»-)
e6=(sizes4aU)&(haii*fair)&(eyes=blue)&(c^
e7=(size=tall)&(haii^brown)&(eye^
e8=(size=short)&(haii^fair)&(cycs=brown)&(complcxion=maa)&
e9= (size=tall)&(hair=fair)&(^

The learning set can easily be represented by function TR: TR((size=tall))«(e2 c3 c5 e6 c7 c9), TR((size=short))=(el c4 c8), TR((hair=brown)=(e4 e5 e7), TR((hair=fair)Mel e2 e6 e8 e9), etc.

Thanks to this representation the set of training instances covered by a conjunction of descriptors, for instance (size=short)&(hain=fair), is automatically given by the intersection of the set of training instances covered by each term of the conjunction. Also, for disjunction, the union of sets are sufficient. So we obtain TR((size=short)&(hair=fair)=(el e8) and TR((size=short)v(hair=fair))«(cl c2 e4 e6 e8 c9)

Once function TR has been defined, it is very simple to obtain function SUB: SUB(E) = {d such that $E\subseteq TR(d)\}$.

In our example it becomes: SUB((el e4 e8))={(size»short)}, SUB((e4e5e7))={(hair=brown),(class«-)} etc...

In accordance with what we had announced above, we should generate, among others the following rules:

sizeshort -> size=short, hair=brown - hair=brown & classe=, and
hair-brown & eyes=blue -► hair=brown & eyes«blue & class=-.

However such rules comprise many redundancies which must be eliminated. To do so the Boolean lattice structure of the set of subsets of the descriptors set is used and function IMP is created to describe the set of non trivial implications derived from a descriptor:

$$IMP(D)\ -SUB(TR(D))\ -\ \bigcup_{D'<D} SUB(TR(D'))$$

We have then IMP((hair=brown)&(eyes=blue)) = Ø, as
SUP(TR((hairssbrown)))* {(hair=brown), (class=-)},and,
SUP(TR((cycs»bluc)))« {(eyes=blue)}.

We can also obtain: IMP((hair=brown)>={(class=-)} and IMP((eyes=blue))-0.

The mathematical properties of Boolean lattices allow to simplify this formula into :

$$IMP(D)\ - SUB(TR(D))\ -\ \bigcup_{D'<D} IMP(D')$$

Therefore we could construct rules on the pattern S → IMP(S). However, redundancies would remain. To be convinced of it let us note that

IMP((hair=fair)&(eyes-blue))= ((complexion=pale),(class=+)}, and
IMP((size=short)&(complexion=pale))= {(hair=fair),(eyes«blue),(class=+)}

so we ought to have simultaneously the two following rules:

(hair=fair)&(eyes=blue) -► (complexion=pale)&(class*+)
(size=short)&(complexion«pale) -> (hain=fair)&(eyes=blue)&(class=+)

Now this second rule is obviously redundant, as when it is trigered the first one must be trigered too. Such redundancies come from the implication transitivity. We free ourselves from it with a transformation x which demonstrably maintains all the properties of the rule system (Cf. (2)). In our example transformation x would tranform the second rule into:

(size=short)&(complexion=fair) - (hair=fair)&(eyes*blue)

After the transformation it is possible to generate, for each description, rules of the type S -> x(IMP(S)). The rule system so established reflects all the logical relations between descriptors. In that way it is complete. Moreover, one can prove that it is minimal (Cf. (2)). Each rule plays a part and, if eliminated, it diminishes the deductive power of the rule system. For instance, a few rules obtained with the last example are shown below :

If hair = red Then class = +, eyes = blue, size = tall.
If hair = fair, eyes = blue Then class = +, complexion = pale.
If size = short, eyes * brown Then complexion = matt, hair = fair.
If size = short, complexion = pale Then hair = fair, eyes = blue.
If eyes = blue, size = tall Then complexion = pale.
If hair = brown Then class = -.
If eyes = brown Then class = -.
If complexion - matt Then class = -.

The function x o IMP, that we shall name RU, is able to constitue a set of rules, this remains to be built and so for all the description space. This is the subject of the next sub-section.

4. RULE CONSTRUCTION

The first way to build RU would be to explore completely all the description space. Yet a procedure based on this principle would be cumbersome, as the calculation time would be proportional to the number of possible descriptions. But as descriptions for which function RU is not nil are the only ones of interest as being the only ones to lead to rules, it is sufficient to determine a-priori nullity criteria for that function to curb the exploration procedure.

First of all, one can note that it is enough to limit oneself to the conjunctions of descriptors, in fact: $d\in RU(D_1 \vee D_2)$ if and only if $d \in RU(D_1)$ and $d \in RU(D_2)$as $(D_1 v D_2) \Rightarrow d$ is equivalent to $D_1 \Rightarrow d$ and $D_2 \Rightarrow d$, now as all the rules are generated, if the rule $D_1 v D_2$->d was to be present, the rules $D_1 \rightarrow d$ and $D_2 \rightarrow d$ would be present too and this would make the rule $D_1 v D_2 \rightarrow d$ useless. The study of all descriptions is then the study of the descriptors conjunctions. As the set of descriptors conjunctions is isomorphic to the set of parts of the set of descriptors, it can be represented by a Boolean lattice which properties are used to prove theorems predicting the uselessness of a description D, that is the nullity of RU(D).

More precisely, the description space is explored from the more general to the more specific. Thus, if there are three descriptors, d_1, d_2 and *d_3,* the description space will be explored in the following order : {0, d_1, d_2, d3, d_1&d2» d_1&d_3, d_2&d_3, d_1&d_2&d_3}. To avoid examining all the conjunctions of descriptors, the *useless* descriptions are characterized : We shall say that a description D is *useless* if RU(D) is nil and if all descriptions D' more specific than D verify also $RU(D') = \emptyset$. If a description verifies such a property neither this description, nor the more specific descriptions derived from it will be explored; it will then be possible to considerably reduce the space to study. Now, one can *formally demonstrate* that the properties of rules and rule systems allow to characterize the *useless* descriptions. To make the presentation more simple, let us define a predicate US to characterize the uselessness of a description: US(D) ⇔ D is *useless.* As regards the proof of these properties refer to (2).

4.1. Properties of rules

Let us give first, the translation of some rule properties:

Theorem 1: For all descriptions *D_1* and D_2.

$RU(D_1) \geq D_2$ and $(D,\&D_2) > D_{\{} \Rightarrow US(D,\&D_2)$

Intuitively, this means that when the descriptor d is logically derived from description D_1 that is when $D_1 \Rightarrow d$, then it is not necessary to study the description D_2 = D,&d&.. as $RU(D_2) = \emptyset$.

Theorem 2: For all descriptions D_1 , D_2 and D_3,

$TR(D_1\&D_2) \subseteq 'TR(D_1\&D_3) \Rightarrow US(D_1\&D_2\&D_3)$

In other words, this theorem stipulates that if the set of examples covered by the description D&d_1 is included in the set of examples covered by D&d_2 then it is not necessary to study the descriptions D&d_1&d_2&.. as they will not tell anything new.

4.2. Rules systems

These theorems are fundamental. They insure the technique feasability, but they are not alone. Other theorems are related to the properties of the rule system. Here, as an example, are a few of the characteristic properties of a rule system as they may be introduced in CHARADE to define the exploration procedure parameters:

- *Goal of the rule system:* disease diagnosis, determination of remedy etc.
- *Structure of the rule system:* rules that go from the symptom to the disease, from the disease to the type of problem, the remedy and the potential danger.
- *Minimum number of examples to be covered by a rule premise.* Thanks to this coefficient it is possible to control the ill-effects due to noise: a rule that would be verified by a single example could not be generated. Thus, a parameter v is introduced so that any description D covering a number of examples smaller than v, that is such as $card(TR(D)) \leq v$, be a-priori nil for RU, and so no rule $D \rightarrow d$ could be generated.
- *Maximum number of descriptors present in a rule premise.*
- *Descriptor relevance:* this characteristic allows, heuristically, to eliminate a-priori rules which do not make sense. In fact, there are descriptors which, per se, have no meaning but which, in conjunction with symptom descriptors, limit their field. Thus, considering an expert system in agronomy, the optimum temperature or the level of humidity cannot lead by themselves to a conclusion. This would be absurd. Nevertheless, in conjunction with other descriptors, such as the colour of spots on the leaves, they can be favorably introduced in the rules.
- *Example coverage:* this is the generalisation of a heuristic used by Michalski (3). It consists in stopping the exploration as soon as a number N of rules cover the examples and conclude as to the final condition.
- *Class partition:* in the same spirit, it is possible to determine the minimal proportion of examples in the learning set covered by a class d and a rule $D \rightarrow d$. This allows to introduce disjunctions in the rules and at the same time to avoid having too specific rules covering one example only.
- etc.

All these properties give rise to a set of parameters which characterize the rule system taken as a whole and give a formal base to the various deletions made in the process of exploring the description space. Actually, one should note that all the heuristics are parameterized by the user and that the adjusment of such parameters depends as much on the properties of the system of rules to be generated as on those of the learning set. Thus, if a classification system must be created and that the number of examples is large, a high coefficient v will be introduced whereas if there is only one proptotypical example in each class, we shall necessarily have v=0.

5. CONCLUSION

CHARADE has been implemented on Macintosh Plus. Programmed in Le„Lisp, it was tested in several fields, from tomato pathology to first call at bridge, classification of archaeological objects and galaxy recognition. Running time is reasonable (approximately 45 minutes in the interpreted version for 43 examples and 116 Boolean descriptors) and the space explored is sufficiently limited to leave room for expectations. Moreover, as regards classification, the system returns the examples that it has not been able to classify, which allows a feed back on the description language. Also, the rule construction technique allows to take into account noise in data to generate approximate rules using certainty factors (Cf. (2)).

REFERENCES

1. Buchanan B. G., Mitchell T., *Model-directed learning of production rules,* in Pattern-Directed Inference Systems, Waterman D. and Hayes-Roth F. eds., Academic Press, New-York, 1978.

2. Ganascia J.-G., *AGAPE et CHARADE : deux techniques d'apprentissage symbolique appliquies a la construction de bases de connaissance.* These d'etat, Universite Paris-Sud, May 1987.

3. Michalski R. S., *A theory and methodology of inductive learning,* in Machine Learning : an artificial intelligence approach , Eds. R. S. Michalski, J. G. Carbonell and T. M. Mitchell, Pub. TIOGA publishing company, Palo Alto, California, 1983, pp.83-134.

4. Quinlan R. J.,f *Learning Efficient Classification Procedure and their Application to Chess End Game,* in Machine Learning : an artificial intelligence approach, Eds. R. S. Michalski, J. G. Carbonell and T. M. Mitchell, Pub. TIOGA publishing company, Palo Alto, California, 1983, pp.463-482.