RULE REFINEMENT USING THE PROBABILISTIC RULE GENERATOR

Won D. Lee and Sylvian R. Ray

Department of Computer Science, University of Illinois,
Urbana, Illinois

ABSTRACT

This work treats the case of expert-originated hypotheses which are to be modified or refined by training event data. The method accepts the hypotheses in the form of weighted VL_1 expressions and uses the probabilistic rule generator, PRG. The theory of operation, verified by experimental results, provides for any degree of hypothesis modification, ranging from minor perturbation to complete replacement according to supplied confidence weightings.

I INTRODUCTION

There are many situations where we would like to construct a knowledge base initially as a set of hypotheses, introduced by a human expert, which are later systematically modified by experimental training data events. Indeed, one might say that this ordering of the learning process is analogous to theoretical study of a problem's solution methods followed by practical experience with the problem, wherein modification or adaptation of the initial rules or hypotheses occur. Required modifications may range from small perturbations of the hypotheses through major or minor deletions and addition of new rules.

This problem has been called,"rule refinement", in which cases it was viewed as an incremental learning of *machine* generated rules. Here, we extend the idea to modifying hypotheses originated either by *human agent* or machine. Therefore, communication between human expert and machine becomes possible in the sense of human introduction of a bias or preliminary problem treatment. Our approach hinges upon a probabilistic formulation of the rule generation problem and deviates significantly from previous approaches because of this and its embodiment in the Probabilistic Rule Generator (PRG) (Lee and Ray, 1986a & b). The form of expression of the rules must also be equally convenient both for a human and for the rule generator.

First, some of the difficulties arising from the rule refinement problem are discussed. Next, related works are examined, and then, a language is presented describing the initial hypotheses appropriately to communicate to a machine. Finally, a scheme to modify initial hypotheses with the training data set is described with some practical application results.

II DIFFICULT NATURE OF THE PROBLEM

To examine the difficult nature of the problem, let us consider a simple case first. Assume that we have only one initial hypothesis V_1 for a class C_1, and one hypothesis, V_2, for a class C_2. Let F_1 and F_2 be new training event sets for C_1 and C_2, respectively, to be used in rule refinement.

If V_1 and V_2 perfectly describe classes C_1 and C_2, respectively, then all the events in F_1 will be covered by the hypothesis V_1, and likewise, there will be no events in F_2 which are not covered by V_2. But, in general, hypotheses are not perfect, hence usually not consistent with the new event sets. Therefore, hypotheses need to be modified to accommodate newly acquired facts. But, the modification process is complicated by various interactions among hypotheses and events as described below.

A hypothesis V_1 is *incomplete* if it does not cover all the events in F_1(we are still assuming that there is only one hypothesis V_1 for the class C_1).

A hypothesis is *inconsistent* if it covers events which belong to other classes.

A hypothesis V_1 *collides* with another hypothesis V_2 if their intersection is non—null. If the two hypotheses belong to different classes, then a contradiction between the hypotheses results. On the other hand, if the two hypotheses belong to the same class, then they might have to be merged together to form a new hypothesis.

Now, let us consider all of the three problems mentioned above at the same time(see Figure 1). More complications arise since when we try to resolve incompleteness by expanding a hypothesis, then inconsistency might occur during the process by covering exception events belonging to other classes.

Likewise, if we shrink a hypothesis to resolve inconsistency by not covering exception events, then incompleteness might occur during the process since some of the events included in the former hypothesis might not belong

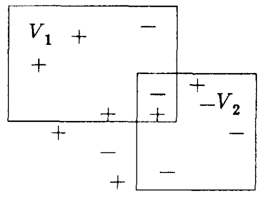

Figure 1. Simple example of a general rule refinement problem: here, "+" are events in F_1, and "–" are those in F_2. Notice the probabilistic nature of the problem.

to the modified hypothesis any more.

Of course, we can always adjust hypothesis V_1 so that it includes some of the events in F_1, and does not cover any event in F_2. But, in doing so, the newly generated hypothesis might become malformed in shape as it might have to be shrunk too much to exclude all exception events. Therefore, the classical viewpoint of *strict* completeness and consistency of a concept is no longer adequate. A scheme should be flexible enough to generate an appropriate form of hypotheses according to user specification, and it is desirable to have a process with a probabilistic nature, allowing *partially* complete and *partially* consistent concepts. The point here is that a general rule refinement scheme must be able to deal with all the possible situations that the usually noisy data set might present.

As we can see, the problem of dealing with collisions between hypotheses will become increasingly complex, as the situation worsens by the addition of problems of incompleteness and inconsistency.

We have considered only a simple case so far: each class having only one initial hypothesis. But, what about more complex cases when there is more than one hypothesis for each class? A bewildering variety of interactions among the hypotheses and the event sets can occur then.

Implicit in the above discussion is that we modify the hypotheses to fit with the example events. But, if the hypotheses represent only a few cases of the facts, and if the new training data set contains a large amount of events not described by the initial hypotheses, the idea of starting with the hypotheses and adjusting them to fit the data might not be the right one. Rather, we might want to produce complexes from the data events first, and adjust those complexes according to hypotheses. An extreme case arises when the initial hypotheses do not cover any of the events in the new training data set. The idea is that we should not give any special privilege to the initial hypotheses, but treat the hypotheses equally with the event sets according to their importance of how many data events they represent.

III PREVIOUS WORK*

A. AQ Rule Refinement

The AQ rule refinement scheme was derived in (Michalski and Larson, 1978). There are some shortcomings in this scheme. First, the objective of AQ incremental rule generation is to produce a new decision rule consistent with the input hypotheses and the observed events. Initial hypotheses are not modified to make them consistent with observed events; they are rather used to find events that cause inconsistency and incompleteness to the hypotheses, and to generate a cover of an event set against those hypotheses.

Since complexes are generated around the example events only, and no attempt is made to modify the initial hypotheses to make them consistent with the example events, it is argued that the expansion strategy used in incremental rule generation in AQ should be dropped(O'Rorke, 1982).

Secondly, it has been observed that new hypotheses generated are usually overly complex compared to former hypotheses. This is because there is a lack of facility to capture complexes with some exception events in them, and therefore all the new hypotheses are formed to include strictly positive events only.

There have been attempts to remedy the problems mentioned above (Reinke and Michalski, 1985). But since those methods are based on the modified AQ star synthesis, they do not address the probabilistic nature of rule refinement fully. Some problems associated with AQ still remain in them(Lee and Ray, 1986a). The capability to capture complexes probabilistically becomes important since there will be a large number of interactions among complexes and events in the modification process.

B. ID3 iterative rule generation — methodology and discussion

A rule can be generated by ID3 iteratively by selecting a subset of training events, called a "window", at each iteration(Quinlan, 1983).

There are two methods of forming a new window in ID3. One way is to add the exceptions to the current window up to some specified number to form an enlarged window. The other one is to select "key" events in the current window and replace the rest with the exceptions, thus keeping the window size constant.

Notice that ID3 itself does not have the capability to accept initial hypotheses, but uses already existing data events to iteratively generate a decision tree. Also, ID3 itself is intended to make a decision tree, rather than to synthesize variable–valued logic expressions. Therefore, the ID3 rule refinement scheme is not intended for initial hypotheses modification.

Yet, when we relax some of the constraints in the scheme, we can extract some useful ideas from it.

First, let us consider the case of making a new window by adding some specified number of exceptions to the current window. Let us relax the requirement of "some specified number", so that we can add a large number of exceptions to the current window, if we desire. Then we observe that the ID3 rule refinement scheme will not differentiate between already available rules and the training events. In other words, the current rule can be grown in any direction. Therefore, a new rule can have any shape from almost identical to the current rule to almost entirely different from the current one.

Secondly, let us consider the idea of selecting the key events in the current window to represent the current rule. It is important to select these key events to represent the current rule faithfully as they will be mixed with "foreign" exception events.

If there is no further information available about the general distribution tendency of the events in each variable domain, we might want to choose events which are distributed equally over the subspace of the complex to which these events belong.

Let us further imagine the situation that, after we generate a rule, we lost all the actual data sets of events.

*Rule refinement for production rules was treated by Ginsberg et al. (Ginsberg et al., 1985).

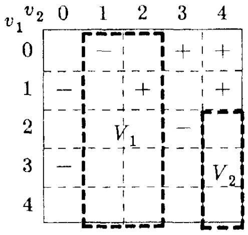

Figure 2. An example of a rule refinement problem. Hypotheses are denoted as bold dashed rectangles, and "+" and "-" are the events in the sets F_+ and F_-, respectively.

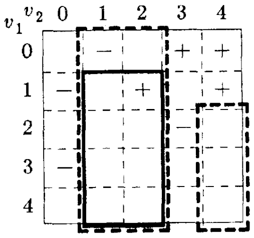

Figure 4. Case 2: notice that the major hypothesis is a specialization of the original hypothesis V_1.

Then one way to recover the rule for the next iteration is to generate events artificially, thus simulating the hypotheses.

Because the event space is generally huge, and the number of available training events is usually small, and more importantly, human conceptual knowledge may be very condensed, a concept by a human expert can play an important role. Thus, a scheme to modify an expert–driven concept systematically may be especially effective.

IV CASE STUDIES

VL_1 expressions(Michalski, 1975) have been used to describe rules generated by inductive inference machines. Consider a simple example to see whether the VL_1 expression is sufficient to convey an expert's hypotheses to a machine so that rule refinement might be done by the machine.

Let there be two classes, C_+ and C_-, and two linear variables, v_1 and v_2, describing the events, each with cardinality 5. Let there be two initial hypotheses, V_1 and V_2, both for class C_+, and new sets of example events, F_+ and F_-, belonging to C_+ and C_-, respectively(see Figure 2). Here, hypotheses V_1 and V_2 are described by the VL_1 expression as two complexes, $[v_2 = 1..2]$, and $[v_1 = 2..4][v_2 = 4]$, respectively.

Let us consider the following five cases.

Case 1

Let the hypotheses be very "light" in weight, not representing many actual events by themselves(see Figure 3). This kind of situation occurs when an expert is inexperienced and is not sure of some of his claims, or is merely guessing the rule. Therefore, the expert wants the machine to generate a new rule mainly by the new event sets F_+ and F_-. Thus, machines should be able to detect the fact that these hypotheses are indeed unimportant or light, and hence ought to generate new hypotheses by the given examples. In this case, the major cluster would emerge by capturing a cluster made of C_+ events.

Case 2

Let us consider the case when the hypothesis V_1 is heavy, but V_2 is light, compared with the actual number of C_+ example events(see Figure 4).

A major, newly generated hypothesis might be the original hypothesis except the part that a C_- event resides inside the hypothesis V_1. Thus, this example shows how a hypothesis can be *specialized* by some exception events.

Case 3

Here, let V_1 be light, while V_2 is heavy. This is the reverse of the case 2, and therefore a major new hypothesis would be created around the hypothesis V_2.

As we can see in Figure 5, because two C_+ events are in the region where V_2 can be more generalized, a new hypothesis would be formed by merging V_2 with these two C_+ events.

Therefore, this case exemplifies how a hypothesis can be merged with some example events to be more *generalized.*

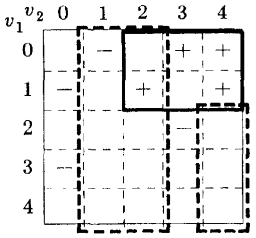

Figure 3. Case 1: here, a major hypothesis is denoted by a bold rectangle, and is created mainly from C_+ events.

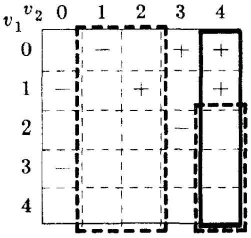

Figure 5. Case 3: notice that the major hypothesis is a generalization of the original hypothesis V_2.

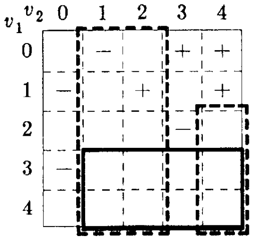

Figure 6. Case 4: notice that the major hypothesis is made from the original hypotheses V_1 and V_2.

Case 4

Next, consider the case when both hypotheses V_1 and V_2 are equally heavy(see Figure 6). If there is no way to merge these two hypotheses to make a heavier hypothesis, then new hypotheses would be created around original hypotheses.

But, in this specific example, as in the figure, if there is a way to merge these two together, and if the merged hypothesis is indeed heavier than each individual hypothesis, then a new, more important hypothesis can be made by the merge.

Case 5

Finally, let us reconsider the case 2, when hypothesis V_1 is heavy, but V_2 is not(see Figure 7). If the hypothesis V_1 is heavy enough, then a flexible rule refinement scheme should be able to ignore a small number of exception events which would otherwise make the new hypothesis shrink too much. By doing so, the new hypothesis would be general enough to be a good description of the facts. This, in turn, is the problem of a rule refinement scheme having the capability of capturing complexes probabilistically.

V COMMUNICATION LANGUAGE

We have discussed so far only some of the cases that can occur during rule refinement process. There are other cases, such as the case when two hypotheses belonging to two different classes collide with each other. Then, a decision should be made by the user whether to divide an initial hypothesis to generate new hypotheses without any exception, or to accept the initial hypothesis without change if the collision is not severe. Of course, there should be some intermediate solutions between the two extremes, according to the user specification. Again, this is most naturally treated as a probabilistic rule refinement problem. Since the data might well be contaminated, being able to generate hypotheses probabilistically becomes important.

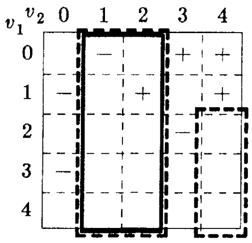

Figure 7. Case 5: notice that the major hypothesis is the same as the original hypothesis V_1, and contains one exception event.

As we have seen, if an expert wished to describe his hypotheses to communicate with the machine, he would need to describe how important each hypothesis is, compared not only with other hypotheses, but also with the sets of example events.

Let us, for instance, say that there are two hypotheses, $[v_2 = 1..2]$ having a weight of 0.8 and $[v_1 = 2..4][v_2 = 4]$ having a weight of 0.2, representing a total of 100 events. Then, we can express the initial hypotheses as

$$80[v_2 = 1..2] + 20[v_1 = 2..4][v_2 = 4].$$

Therefore, this expression contains not only the relative importance of each hypothesis, but also the relative importance of the total hypotheses to the sets of example events.

We will call this expression a *weighted* VL_1 (WVL_1) expression.*

VI PRG RULE REFINEMENT

The objective of the PRG rule refinement scheme is summarized below.

First, it should be able to consider all the interactions among hypotheses and sets of example events. In other words, it must be capable of merging hypotheses, resolving collisions between hypotheses belonging to different classes, generalizing hypotheses by adding some events, and specializing hypotheses by excluding some exception events.

Secondly, it should be flexible enough, hence be probabilistic, in generating new hypotheses according to user specification. For instance, a hypothesis can be very specific, explaining only the positive events, or be more general by ignoring some minor exceptions.

Because PRG is already probabilistic, the only problem left is to enter the WVL_1 expression into PRG to be modified.

A simple solution is used in PRG rule refinement scheme. Since a human expert's description of a concept is often probabilistic, it is suitable for simulation in PRG.

PRG rule refinement scheme is described as:

```
PRG rule refinement;
  begin
    simulate each hypothesis by a set of randomly
        generated events in the subspace described
        by the hypothesis, making the number of
        events equal to the weight of the hypothesis;
```

*Notice that this expression is different from the weighted DVL expression in (Michalski and Chilausky, 1980), since weighted DVL expressions are concerned with the weighted selectors in a VL_1 expression, rather than weighted complexes. We will call the coefficient in front of each complex, a "weight", as it represents the strength of evidence of each hypothesis.

Table 1. Result of Experiment 1.

Notes:

1. Major complex is created mainly from positive training events since hypotheses are lightly weighted.
2. The major complex comes from original hypothesis, V_1, by excluding one exception event.
3. The major complex is formed by merging the dominant hypothesis, V_2, with some positive training events.
4. A new, heavier hypothesis is created by merging hypotheses V_1 and V_2.
5. With *specificity*=0.9(which means that any subspace that has ratio of the number of positive events to the total number of events in it is greater than or equal to 0.9, is considered as an acceptable complex), a single exception event in hypothesis V_1 is ignored, permitting V_1 to be retained as the major complex.

WVL_1 Hypotheses	Refined Rule	Note
$1V_1 + 1V_2$	$4[v_1{=}0..1][v_2{=}2..4]+2[v_2{=}2]+1[v_1{=}4]$	1
$10V_1 + 1V_2$	$9[v_1{=}1..4][v_2{=}1..2]+5[v_1{=}0..1][v_2{=}2..4]+3[v_2{=}4]$	2
$1V_1 + 10V_2$	$12[v_2{=}4]+2[v_2{=}2]+4[v_1{=}0..1][v_2{=}2..4]$	3
$10V_1 + 10V_2$	$14[v_1{=}3..4][v_2{=}1..4]+12[v_2{=}4]+5[v_2{=}2]+5[v_1{=}0..1][v_2{=}2..4]$	4
$10V_1 + 1V_2$	$10[v_2{=}1..2]+3[v_2{=}4]+5[v_1{=}0..1][v_2{=}2..4]$	5

```
    add sets of training events;
    run PRG to generate new hypotheses according
      to user specification;
  end.
```

Thus, the PRG rule refinement scheme does not give any special attention to the hypotheses, but treats them equally with the example events. This makes it possible to face the complexity of the interactions among the hypotheses and the sets of example events in a uniform way, and PRG will generate new hypotheses as if there were no special hypothesis at all, and hence no partiality will take place in the rule refinement process.

VII EXPERIMENTS

Experiment 1 : In Sec. IV, we dealt with some cases that a rule refinement scheme should be able to resolve. Those cases were run by the PRG rule refinement program.

Two initial hypotheses for class C_+ were:

$$V_1 = [v_2 = 1..2] \text{ and}$$
$$V_2 = [v_1 = 2..4][v_2 = 4].$$

The two sets of training events, F_+ and F_-, used in Sec. IV, were introduced as data. Complexes generated by PRG agree with the earlier discussion(see Table 1).

Experiment 2 : Rules for five classes of sleep ("stages") were written by a human expert based on standard sleep stage scoring principles(Ray, et al., 1985) and presented as initial hypotheses to the PRG program.

A data base of 742 events from one individual's full night sleep study was introduced as new data. Sets of rules were generated by the PRG using four different relative weightings of the hypotheses and the new events in addition to the hypotheses alone. The experiment consisted of testing the accuracy of the rules in classifying the 742 events for each of the five rulesets, the results of which are shown in Table 2.

Note Class 3 where with hypotheses only ($\lambda = \infty$), the accuracy was only 19% but with $\lambda = 2$, the new events overcame the inaccuracy of the hypotheses and the resulting rules were nearly perfect(99% accuracy).

Class 2 exhibits more typical behavior, the accuracy rising monotonically (except for trivial noise fluctuations) from 71%, with hypotheses only, to 94% when only training data was used for the rules. Class 5 also exhibits accuracy growth that is monotonic, paralleling that of Class 2.

Only Class 1, which is known semantically as the noisiest, most poorly clustered class, shows strongly erratic behavior. The anomalous–looking 53% accuracy for hypotheses only is due to "overgeneralization" which becomes constrained by negative events as new data events are introduced. As hypothesis weight decreases, interaction between Class 0 and Class 1 manifests as non–monotonic accuracy increase.

Table 2. Rule Modification Experiment Result. Entries are % of events correctly classified($\lambda{=}\infty$ is the case of hypothesis only). Here, *specificity* = 0.9, *certainty* = 0.9, and *weight* = 0.05.

Class	# New	λ=wt. of hypothesis/wt. of new events				
	Events	∞	2.0	1.0	0.25	0.0
0	100	19	59	68	54	62
1	83	53	10	39	51	53
2	385	71	85	94	93	94
3	108	19	99	99	99	96
5	66	47	47	47	67	67
Total	742	52.4	72.0	80.6	81.5	82.7

VIII CONCLUSION

A system has been developed to permit communication between expert and machine using the following principles. Weighted VL_1 expressions are used by the expert to introduce hypotheses. Training events may be appended as new data, each event having weight 1. The hypotheses are expanded into a weight-equivalent number of data events, which are joined with the actual new events. Thus, neither hypotheses nor training events have special significance except through weighting.

The superset of events is then submitted to the Probabilistic Rule Generator which is capable of capturing major complexes in spite of moderate noise.

Experiments verify that the resulting rules may range from minor refinement of the hypotheses through various reorganizations of the hypotheses and on to rules which are completely dominated by the new training events in a continuous, systematic spectrum, controlled by assigned (confidence) weighting.

REFERENCES

[1] Ginsberg, A., Weiss, S. and Politakis, P., "SEEK2: A Generalized Approach to Automatic Knowledge Base Refinement," *Proceedings of Ninth International Joint Conference on Artificial Intelligence*, University of California at Los Angeles, August 1985, pp. 367–374.

[2] Lee, W. D. and Ray, S. R., "Probabilistic Rule Generator," *Proceedings of the 1986 ACM Annual Computer Science Conference*, Cincinnati, Ohio, February 4–6, 1986.

[3] Lee, W. D. and Ray, S. R., "Probabilistic Rule Generator: A New Methodology of Variable–Valued Logic Synthesis," *Proceedings of 1986 IEEE International Symposium on Multiple–Valued Logic*, Blacksburg, Virginia, May 1986.

[4] Michalski, R. S., "Variable–Valued Logic and Its Applications to Pattern Recognition and Machine Learning," *Computer Science and Multiple–Valued Logic Theory and Applications*, Rine, D. C.(Ed.), North–Holland, 1975, pp. 506–534.

[5] Michalski, R. S. and Chilausky, R. L., "Knowledge Acquisition by Encoding Expert Rules versus Computer Induction from Examples: A Case Study Involving Soybean Pathology," *International Journal for Man–Machine Studies*, No. 12, 1980, pp. 63–87.

[6] Michalski, R. S. and Larson, J. B., "Selection of Most Representative Training Examples and Incremental Generation of VL_1 Hypotheses: The Underlying Methodology and the Description of Programs ESEL and AQ11," Technical Report 867, Department of Computer Science, University of Illinois, May 1978.

[7] O'Rorke, P., "A Comparative Study of Inductive Learning Systems AQ11P and ID–3 Using a Chess Endgame Test Problem," ISG 82–2, UIUCDCS–F–82–899, Computer Science Department, University of Illinois, 1982.

[8] Quinlan, J. R., "Learning Efficient Classification Procedures and Their Application to Chess End Games," *Machine Learning*, Michalski, R. S., Carbonell, J. G. and Mitchell, T. M.(Eds.), Palo Alto:Tioga Press, 1983.

[9] Ray, S. R., Lee, W. D., Morgan, C. D. and Airth–Kindree, W., "Computer Sleep Stage Scoring––An Expert System Approach," Technical Report 1228, Computer Science Department, University of Illinois, September 1985. Also, to Appear in *International Journal of Biomedical Computing.*

[10] Reinke, R. E. and Michalski, R. S., "Incremental Learning of Concept Descriptions," *Machine Intelligence 11*, Hayes, J. E., Michie, D. and Richards, J.(Eds.), Oxford:Oxford University Press, 1985.