Mathematical and Computational Applications, Vol. 10, No. 1, pp. 113-120, 2005.

PRUNING DECISION TREES USING RULES3 INDUCTIVE LEARNING ALGORITHM

Mehmet Sabih Aksoy
King Saud University, College of Computer and Information Sciences
P.O.Box 51178 Riyadh 11543 Saudi Arabia. msaksoy@ccis.ksu.edu.sa

Abstract - One important disadvantage of decision tree based inductive learning algorithms is that they use some irrelevant values to establish the decision tree. This causes the final rule set to be less general. To overcome with this problem the tree has to be pruned. In this article using the recently developed RULES inductive learning algorithm, pruning of a decision tree is explained. The decision tree is extracted for an example problem using the ID3 algorithm and then is pruned using RULES. The results obtained before and after pruning are compared. This shows that the pruned decision tree is more general.

Key Words- Pruning, Decision Trees, Inductive Learning.

1. INTRODUCTION

The process of acquiring knowledge through interaction with an expert consists of a prolonged series of intense, systematic interviews, usually extending over a long period [Waterman, 1986]. The main problem encountered in this process is that, human experts are capable of using their knowledge in their daily work, but they usually cannot summarize and generalize their knowledge explicitly in a form which is sufficiently systematic, correct and complete for machine representation and application [Liu and White, 1991]. Experts have a tendency to state their conclusions and the reasoning behind them in general terms that are too broad for effective machine analysis. The pieces of basic knowledge are assumed and combined so quickly that it is difficult for them to describe the underlying thought process.

Expert systems require large amounts of knowledge to achieve high levels of performance, yet the acquisition of knowledge is slow and expensive [Quinlan, 1988]. The shortage of trained knowledge engineers to interview experts and capture their knowledge is another problem of knowledge acquisition [Weiss and Kulikowski, 1991].

The aforementioned problems are not just difficulties of the early days of the technology, but are still acknowledged today as paramount problems. Knowledge acquisition (and in particular machine learning) has become a major area of concern for expert systems research [Quinlan, 1988; Williams, 1988].

An alternative method of knowledge acquisition exists in which knowledge is learned, or induced, from examples. While it is very difficult for an expert to articulate his knowledge, it is relatively easy to document case studies of the expert's skills at work [Quinlan, 1988]. Instead of asking an expert to summarize and articulate his knowledge, the main idea of automatic induction is to have him provide a basic structure of his discipline. The knowledge itself will be induced from examples expressed in this structure. Michie [Michie, 1987] states that *"this blocked channel can thus be circumvented if and only if a means can be found for moving the rules from the expert's head to machine memory via the language of examples rather than via the language of explicit articulation. For that we require effective algorithms for inductive inference".* Michie's view is also supported by other AI researchers [Liu and White, 1991; Quinlan, 1987]. Recent developments have proved that this method of knowledge acquisition is entirely possible.

A number of induction algorithms have been proposed. Well-known induction algorithms include CLS [Hunt et al., 1966], ID3 [Quinlan, 1983], ID4 [Schlimmer and Fisher, 1986], ID5 [Utgoff, 1988], CN2 [Clark and Boswell, 1991], BCT [Chan, 1989] C4.5 [Quinlan, 1993], AQ [Michalski and Larson, 1978; Cervone, Panait and Michalski, 2001], and RULES [Aksoy, 1994].

In this paper we explain how to prune a decision tree obtained by a decision tree based algorithm using RULES3 inductive learning system. For ease of referencing, the paper will first summarize the rule extraction process of RULES3. It then describes the pruning process using an example problem. Finally, the paper ends with the conclusion.

2. RULES-3

RULES-3 [Pham and Aksoy, 1993] is a simple algorithm for extracting a set of classification rules from a collection of examples for objects belonging to one of a number of known classes. An object must be described in terms of a fixed set of attributes, each with its own range of possible values which could be nominal or numerical. For example, attribute "length" might have nominal values {short, medium, long} or numerical values in the range [-10, 10].

An attribute-value pair constitutes a condition in a rule. If the number of attributes is N_a , a rule may contain between one and N_a conditions. Only conjunction of conditions is permitted in a rule and therefore the attributes must all be different if the rule comprises more than one condition.

RULES-3 extracts rules by considering one example at a time. It forms an array consisting of all attribute-value pairs associated with the object in that example, the total number of elements in the array being equal to the number of attributes of the object. The rule forming procedure may require at most N_a iterations per example. In the first iteration, rules may be produced with one element from the array. Each element is examined in turn to see if, for the complete example collection, it appears only in objects belonging to one class. If so, a candidate rule is obtained with that element as the condition. In either case, the next element is taken and the examination repeated until all elements in the array have been considered. At this stage, if no rules have been formed, the second iteration begins with two elements of the array being examined at a time. Rules formed in the second iteration therefore have two conditions. The procedure continues until an iteration when one or more candidate rules can be extracted or the maximum number of iterations for the example is reached. In the latter case, the example itself is adopted as the rule. If more than one candidate rule is formed for an example, the rule that classifies the highest number of examples, is selected and used to classify objects in the collection of examples.

Examples of which objects are classified by the selected rule are removed from the collection. The next example remaining in the collection is then taken and rule extraction is carried out for that example. This procedure continues until there are no examples left in the collection and all objects have been classified. Figure 1 summarizes the steps involved in RULES-3.

Step 1. Define ranges for the attributes which have numerical values and assign labels to those ranges
Step 2. Set the minimum number of conditions (N_{cmin}) for each rule
Step 3. Take an unclassified example
Step 4. $N_c = N_{cmin} - 1$
Step 5. If $N_c < N_a$ then $N_c = N_c+1$
Step 6. Take all values or labels contained in the example
Step 7. Form objects which are combinations of N_c values or labels taken from the values or labels obtained in Step 6
Step 8. If at least one of the objects belongs to a unique class then form rules with those objects;
ELSE go to Step 5
Step 9. Select the rule which classifies the highest number of examples
Step 10. Remove examples classified by the selected rule
Step 11. If there are no more unclassified examples then STOP;
ELSE go to Step 3

Figure 1. Induction procedure in RULES-3 (N_c=number of conditions, N_a=number of attributes)

4. PRUNING DECISION TREES USING RULES3

One of the important features of RULES3 algorithm is its ability to deal with incomplete examples. In many real problems, there could be incomplete examples, that is examples in which the values of some attributes are unknown. Another feature of RULES3 is that it does not suffer from irrelevant values problem. That is, the rules produced by RULES3 contains no irrelevant attribute-values. Pruning a decision tree can be realized using these features of RULES3.

The set of examples for RULES3 is the set of rules produced from a decision tree obtained by a divide-and-conquer algorithm.

The first step for this operation is to form a set of examples for RULES3. In order to explain this operation let us consider the following example problem for which the set of examples is given in Table 1.

Table 1. Season Classification Problem [Aksoy, 1994].

No.	Weather	Trees	Temperature	Season
1	Rainy	Yellow	Average	Autumn
2	Sunny	Yellow	Average	Autumn
3	Rainy	Leafless	Average	Autumn
4	Rainy	Green	Average	Spring
5	Sunny	Green	Average	Spring
6	Sunny	Leafless	Low	Winter
7	Snowy	Leafless	Low	Winter
8	Rainy	Leafless	Low	Winter
9	Snowy	Green	Low	Winter
10	Rainy	Green	Low	Spring
11	Sunny	Green	High	Summer
12	Rainy	Green	High	Summer

Using ID3 algorithm the decision tree shown in Figure 1 is produced.

Table 2. The set of examples obtained from Decision Tree in Figure 1.

No	Weather	Trees	Temperature	Season
1	*	Leafless	Average	Autumn
2	*	Yellow	Average	Autumn
3	*	Green	Average	Spring
4	*	*	High	Summer
5	Snowy	Green	Low	Winter
6	Rainy	Green	Low	Spring
7	*	Leafless	Low	Winter

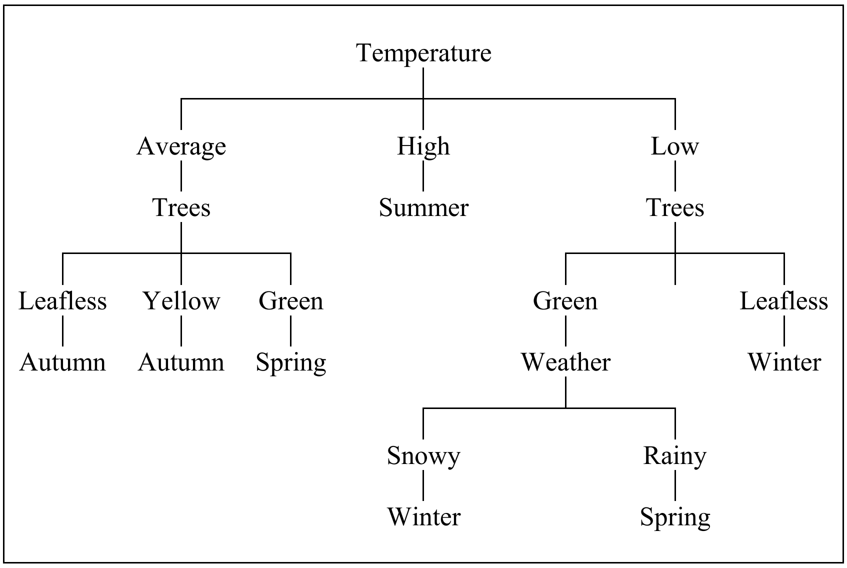

Figure 1. The Decision Tree Obtained by ID3 for Season Classification Problem

The set of examples shown in Table 2, which is going to be the input examples for RULES3 can be obtained from decision tree in Figure 1. The missing attribute-values are represented by "*". These values will not be considered during rules extraction operation by RULES3

Note that Step 1 and Step 9 must be ignored when RULES3 is used for pruning a decision tree. Because all the values must be considered as they are even if they are numerical values. Also all the rules extracted must be considered, that is there is no rule selection.

In Step 2, the minimum number of conditions is set to 1.

In Steps 3 the example 1 in Table 2 is taken for operation. The attribute-values used in this example are *{Trees: Leafless ; Temperature: Average}*. Since the minimum number of condition is one, these values are checked against the examples given in Table 1 in order to decide whether they can form a rule. The check for both of them is negative. It means none of them belongs to a unique class for the set of examples given in Table 1. The number of conditions is increased by one. There is only one combination for the two values at hand. The check for this combination is positive and the first rule can be formed as follow:

Rule 1. IF Trees are Leafless AND Temperature IS Average THEN Season IS Autumn

This rule can classify only the example 1 in Table 2, so it is removed from list of unclassified examples. The test in Step 12 for examples remaining to be classified is positive. Therefore the procedure returns to Step 3 for a new iteration.

When the same procedure is repeated for remaining unclassified examples the following rules will be extracted in six iterations:

Rule 2. IF Trees ARE Yellow THEN Season IS Autumn

Rule 3. IF Trees ARE Green AND Temperature IS Average THEN Season IS Spring

Rule 4. IF Temperature IS High THEN Season IS Summer

Rule 5. IF Weather IS Snowy THEN Season IS Winter

Rule 6. IF Weather IS Rainy AND
Trees ARE Green AND
Temperature IS Low THEN Season IS Spring

Rule 7. IF Trees ARE Leafless AND Temperature IS Low THEN Season IS Winter

Since there are no more unclassified examples, the procedure ends. The above rule induction sequence is summarized in Table 3.

Table 3. Summary of rule induction sequence for the set of examples in Table 2.

Iterati on	Example Consider ed	No. of extracted rules	Classifie d example s	Unclassified examples
1	1	1	1	2,3,4,5,6,7
2	2	1	2	3,4,5,6,7
3	3	1	3	4,5,6,7
4	4	1	4	5,6,7
5	5	1	5	6,7
6	6	1	6	7
7	7	1	7	-

It can be seen that in example 2 in Table 2, the attribute-value {Temparature = Average} is irrelevant and it is removed (that is pruned). Also in example 5, {Trees = Green and Temperature = Low} are removed meaning that they are irrelevant attribute- value pairs. The pruned attribute-values are shown in Table 4.

Table 4. The pruned attribute-values obtained from Table 2.

No	Weather	Trees	Temperature	Season
1	*	Leafless	Average	Autumn
2	*	Yellow	*	Autumn
3	*	Green	Average	Spring
4	*	*	High	Summer
5	Snowy	*	*	Winter
6	Rainy	Green	Low	Spring
7	*	Leafless	Low	Winter

The rules obtained are more general then those obtained from the decision tree before pruning because they contain no irrelevant attribute-values.

5. CONCLUSION

In this paper pruning a decision tree using RULES3 inductive Learning Algorithm is explained. One of the important features of RULES3 algorithm is its ability to deal with incomplete examples. In many real problems, there could be incomplete examples, that is examples for which the values of some attributes are unknown. Another feature of RULES3 is that it does not suffer from irrelevant values problem. It means that, the rules produced by RULES3 contains no irrelevant attribute-values. Pruning a decision tree is realized using these features of RULES3.

This ability of RULES3 was demonstrated using a simple example problem. The results obtained showed that the rule set obtained after pruning is more general as it contains no irrelevant attribute-values. For the example considered, three irrelevant attribute-values were removed. It is obvious that if the number of attributes, possible number of values and the number of examples are bigger, more irrelevant attribute-values will became available. This will cause the set of rules to have a lower ability of classifying unseen examples.

6. REFERENCES

1. M.S. Aksoy , New Algorithms for Machine Learning, *Ph.D. thesis*, University of Wales, College of Cardiff, U.K., 1994.
2. G. Cervone , L.A. Panait and R.S. Michalski, The Development of the AQ20 Learning System and Initial Experiments, *Tenth International Symposium on Intelligent Information Systems*, Zakopane, Poland, June, 2001.
3. P.K.Chan, Inductive Learning With BCT, *Proc. Sixth Int. Workshop on Machine Learning*, Cornell Univ. Ithaca, New York, pp.104-108, 1989
4. J. Cheng et al., Improved decision trees: a generalised version of ID3, in *proc. Fifth Int. Conf. on Machine Learning*, The University of Michigan, Ann Arbor, MI, June 12-14, pp.100-106, 1988.
5. P. Clark and R. Boswell, Rule induction with CN2: Some recent improvements, In *J. Siekmann , Lecture Notes in Artificial Intelligence*, Springer-Verlag, pp.151-163,1991.
6. E.B. Hunt, J. Marin and P.J. Stone, *Experiments in induction*, Academic Press, Newyork, 1966.
7. W.Z. Liu and A.P. White, A Review of Inductive Learning, in I.M.Graham and R.W. Milne, *Research and Development in Expert Systems* VIII, Proc. of Expert Systems 91, Cambridge Univ. Press, London, pp.112-126,1991.
8. J. Ortego, G. Lee and D. Fisher, Representation and processing issues underlying machine learning from examples, proc. of *the Second International Conf. on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, V.II, June 6-9, pp.845-851,1989.
9. D.T. Pham, D.T. and M.S. Aksoy, A new algorithm for inductive learning", *Journal of Systems Eng.,* No.5 pp. 115-122, UK, 1993.
10. J.R. Quinlan, Learning efficient classification procedures and their application to chess end games, in *Machine Learning, An Artificial Intelligence Approach* edited by R.S. Michalski, J.G. Carbonell and T.M. Mitchell, Tioga, Palo Alto, CA, 1983, pp.463-482, 1983.
11. J.R. Quinlan, Inductive knowledge acquisition: a case study, in *Applications of Expert Systems,* Ed: J.R. Quinlan, Turing Institute Press, pp.157-173, 1987.
12. J.R. Quinlan, Induction, knowledge and expert systems, in *Artificial Intelligence Developments and Applications*, Eds J.S. Gero and R. Stanton, Amsterdam, North-Holland, pp.253-271, 1988.
13. J.R. Quinlan J.R., Learning logical definitions from relations, in *Machine Learning, 5*, Kluwer Publishers, Boston, pp.239-266,1990.

14. J.R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, San Mateo, California ISBN 1-55860-238-0, 1993.
15. J.C. Schlimmer and D. Fisher, A case study of incremental concept induction, *Proc. Fifth National Conference on Artificial Intelligence*, Morgan Kaufmann, Philadelphia, Pennsylvania, pp. 496 – 501, 1986.
16. S.C. Shapiro et al., *Encyclopedia of artificial intelligence* , Second edition, 1991.
17. P.E. Utgoff, ID5: An incremental ID3. *Proc. of fifth International Conference on Machine Learning*, The Univ. of Michigan, Ann Arbor, 1988, pp. 107-120, 1991.
18. D.A. Waterman , *A guide to expert systems*, Addision-Wesley, California, 1986.
19. S.M. Weiss and C.A. Kulikowski, *Computer systems that learn*, Morgan Kaufmann, San Mateo, California, 1991.