Knowledge Acquisition by Inductive Learning

Eng. Mihaela M. OPREA PhD.
Department of Computer Science, University of Ploiesti

Abstract: *Inductive learning is one of the most effective approaches used to automate the knowledge acquisition of an expert system. In this paper we present an analysis of three inductive learning algorithms, ID3, C4.5 and ILA applied to rules extraction.*
Keywords*: Knowledge Acquisition, Inductive Learning, Rules Extraction, Expert System.*

1 Introduction

Knowledge acquisition is recognized as one of the major bottlenecks in developping an expert system [10]. Knowledge elicitation from domain experts and machine learning are two distinct approaches to knowledge acquisition. Usually, elicitation of the right knowledge can be both time consuming and expensive. The other method, machine learning which is an automated one, it is recommended as being the most effective and more efficient. As rules are an elegant, expressive, straightforward, and flexible means of expressing knowledge in many application domains [2], we shall concentrate in this paper on rules extraction from the expert knowledge by inductive learning algorithms. Decision tree learning [3], neural network learning, inductive logic programming, and genetic algorithms (see e.g. [6]) are examples of inductive learning methods that operate starting from a set of training examples that represents the history of previous decisions. Inductive learning can generalize from observed training examples by identifying the attributes that empirically distinguish positive from negative training examples.

In this paper we are making an analysis of three inductive learning algorithms used for rule extraction, ID3, C4.5 and ILA. In the next section we briefly present the inductive learning problem and the learning algorithms that were used. In section 3 we discuss the experimental results obtained so far. In the last section we draw some conclusion and see the future work.

2. Inductive learning

2.1. The inductive learning problem

The formulation of the inductive learning problem as stated in [3] is the following. The learner is given a complete set of training examples $D=\{<x_1, f(x_1)>, ..., <x_n, f(x_n)>\}$, where $f(x_i)$ is the target value for the instance x_i, and a hypothesis space H from which it must select an output hypothesis. The desired output of the learner is a hypothesis h from H that is consistent with these training examples. So, extracting a rule means being able to describe a large number of cases in a concise way.

2.2. Inductive learning algorithms

The approach of inductive learning is often used by forming a decision tree from the set of training examples. Decision tree based methods are preferred mainly because they are efficient and can deal with a large number of training examples [1]. However, this kind of approaches do not always produce the most general production rules. Therefore, there are other classes of algorithms which do not employ decision trees (e.g. ILA [8], RITIO [9]). The most known algorithms that take a set of examples as input and produce a decision tree which is consistent with the examples are ID3 [3] and C4.5 [7].

The ID3-like algorithms divide the training set into homogeneous subsets without reference to the class of the subset. ID3 is concerned with finding the attribute which is most relevant overall, even though some values of that attribute may be irrelevant. The algorithm makes use of the entropy measure as a means of constraining the hypothesis search space. ID3 is a greedy algorithm that grows the tree top-down, at each node selecting the attribute that best classifies the local training examples. The best attribute is the one with highest *information gain*. If we define the entropy as a measure of the impurity in a collection of training examples, we can define a measure of the effectiveness of an attribute A in classifying the training data. The measure named *information gain* is the expected reduction in entropy caused by partitioning the examples according to this attribute.

$$Gain(D,A) = Entropy(D) - \sum_{v \in Values(A)} \frac{|Dv|}{|D|} Entropy(Dv)$$

where *Values(A)* is the set of all possible values for attribute A and D_v is the subset of D for which attribute A has value v. In the ID3 case, H is the set of possible decision trees. The algorithm performs a simple-to-complex hill-climbing search, beginning with the empty tree, then considering progressively more elaborate hypotheses in search of a decision tree that correctly classifies the training data.

C4.5 is an extension of ID3 that handles uncertain data at the expense of increasing the classification rate. Initially produces a decision tree, then it prunes this tree and generates a simplified decision tree in which all unnecessary conditions are eliminated. It then generates the rules from the simplified decision tree. In each class, the rules are revised again to discard rules that do not contribute to the accuracy of the rules, trying in this process to maintain the same degree of accuracy of the original decision tree for classifying the rules in the training set from which all the rules are generated. This leads to significantly fewer production rules, but at the expense that these rules may fail to classify all the examples in the training set from which they have been generated, i.e. the error rate in the classification process may be zero or more, while in the ID3 and ILA the aim is to keep this rate at zero throughout.

The ILA algorithm works in an iterative way, each iteration searching for a rule that covers a large number of training examples of a single class. Having found a rule, the ILA removes those examples it covers from the training set by marking them and appends a rule at the end of its rule set. So, ILA works on a rule-per-class basis. For each class, rules are induced to separate examples in that class from examples in all the remaining classes. This produced an ordered list of rules rather than a decision tree. The ILA algorithm is quite unlike ID3 and C4.5 in many respects. ILA does not employ an information theoretic approach and concentrates on finding only relevant values of attributes, mainly by eliminating the unnecessary conditions.

3. Experimental results

We have made a preliminary analysis of three inductive learning algorithms, ID3, C4.5 and ILA and we have experimented these algorithms on different sets of traininig examples. Some experiments had the role of extracting the rules for the knowledge based system developed for environmental protection that is presented in [4]. In this section we will focus on the experiments made mainly on simple sets of training examples for the investment projects analysis problem. We have to note that the analysis is a simple one, and involve only four parameters that can influence the decision regarding the acceptance or the rejection of an investment project. In table 1 it is presented the set of investment training examples.

Table 1. Investment training examples

Ex.	Global_risk	Profitableness	Return_time	Investment_level	Class
1.	high	important	long	high	N
2.	high	important	long	low	N
3.	low	important	long	high	Y
4.	medium	medium	long	high	Y
5.	medium	small	short	high	Y
6.	medium	small	short	low	N
7.	low	small	short	low	Y
8.	high	medium	long	high	N
9.	high	small	short	high	Y
10.	medium	medium	short	high	Y
11.	high	medium	short	low	Y
12.	low	medium	long	low	Y
13.	low	important	short	high	Y
14.	medium	medium	long	low	N

Class = Y - investment project accepted
Class = N - investment project rejected

After applying the ILA on this training set we have obtained the following rules:
R1:IF Global_risk = low THEN Class = Y.
R2:IF Global_risk = high AND Return_time = long THEN Class = N.
R3:IF Global_risk = medium AND Investment_level = low THEN Class = N.
R4:IF Global_risk = medium AND Investment_level = high THEN Class = Y.
R5:IF Global_risk = high AND Return_time = short THEN Class = N.
These five rules are identical with those obtained when we have applied the ID3 and C4.5 algorithms. We can see that the extracted rules do not contain any unnecessary conditions.
Another set of training examples is presented in table 2. This set is making a classification of the investment projects that are analysed. Again, we have simplified the problem by taking into account only three parameters. The rules obtained after applying the three algorithms, ID3, C4.5 and ILA are presented in table 3. All algorithms have generated the same number of rules (5), but rule 3 generated by ILA and C4.5 is simpler than that generated by ID3. This happened because ILA and C4.5 eliminates all the unnecessary conditions, such as ‘Global_risk = medium’.

Table 2. Investment project classification training examples

Ex.	Investment_level	Return_time	Global_risk	Class
1.	high	short	medium	IP1
2.	high	medium	low	IP2
3.	medium	medium	low	IP2

4.	low	medium	low	IP2
5.	high	medium	medium	IP1
6.	high	long	high	IP0
7.	high	long	medium	IP3
8.	low	long	medium	IP3
9.	low	long	high	IP0
10.	medium	long	low	IP2
11.	low	short	medium	IP1

IP0 - investment project rejected; IP1 - Short-Medium-Term-Medium-Risk project
IP2 - Medium-Long-Term-Small-Risk project; IP3 - Long-Term-Medium-Risk project

Table 3. Rules extracted by ID3, C4.5 and ILA

Algorithm	Rule number	Rule
ID3	1.	IF Global_risk = low THEN Class = IP2.
C4.5		IF Global_risk = low THEN Class = IP2.
ILA		IF Global_risk = low THEN Class = IP2.
ID3	2.	IF Global_risk = high THEN Class = IP0.
C4.5		IF Global_risk = high THEN Class = IP0.
ILA		IF Global_risk = high THEN Class = IP0.
ID3	3.	IF Return_time = short AND Global_risk = medium THEN Class = IP1.
C4.5		IF Return_time = short THEN Class = IP1.
ILA		IF Return_time = short THEN Class = IP1.
ID3	4.	IF Return_time = medium AND Global_risk = medium THEN Class = IP1.
C4.5		IF Return_time = medium AND Global_risk = medium THEN Class = IP1.
ILA		IF Return_time = medium AND Global_risk = medium THEN Class = IP1.
ID3	5.	IF Return_time = long AND Global_risk = medium THEN Class = IP3.
C4.5		IF Return_time = long AND Global_risk = medium THEN Class = IP3.
ILA		IF Return_time = long AND Global_risk = medium THEN Class = IP3.

So far, we have used two parameters: the number of rules generated and the average number of conditions, for the evaluation of the three algorithms. The aim is to produce the minimum number of possible rules that classify successfully the examples in the training set. A good rules extraction algorithm should produce rules that not only classify the cases in the training set, but also the unseen examples. In table 4 we present the results obtained for two sets of training examples, for the investment project classification problem and for the environmental protection problem [4]. ILA and C4.5 can produce fewer rules with fewer conditions than those generated by the ID3 algorithm. So, they can classify more unseen examples. In the case of the environmental protection training set, C4.5 gave the smallest number of rules and also the smallest average number of conditions, but at the expense that these rules may fail to classify all the examples in the training set from which they have been generated (the error rate in the classification process is 15.7%).

Table 4

Training set	Algorithm	No. of rules generated		Average no. of conditions
investment project	ID3	5		1.6
	C4.5	5 (0%)	error rate	1.4

	ILA	5	1.4
environmental	ID3	17	2.17
protection	C4.5	10 (15.7%)	error rate 1.8
	ILA	12	2.08

4. Conclusion and future work

We have presented a comparison between three inductive learning algorithms, two decision trees-based algorithm, ID3 and C4.5, and a decision trees-not based algorithm, ILA. The results obtained so far demonstrated that, in general, all the algorithms worked well and had achieved the generality of the extracted rules. However, ILA and C4.5 worked better than ID3 mainly because they do the elimination of all unnecessary conditions (as it is the case for rule 3 mentioned in table 3). In some cases, ILA worked better than C4.5 if we take into account the error rate. As a future work we will extend our analysis by including an explanation-based learning algorithm [5], and a new rule induction algorithm, RITIO, which uses the information theoretic function in a novel way in order to induce directly a set of rules. RITIO eliminates attributes in order of decreasing irrelevancy and achieves high levels of predictive accuracy, even on noisy databases.

References

[1] L. Breslow, D. Aha, Symplifying decision trees: A survey, *The Knowledge Engineering Review*, Vol.12:1, 1997, 1-40.

[2] T.E. McKee, Predicting bankruptcy via induction, *Journal of Information Technology*, 10,1995,26-36.

[3] T. Mitchell, *Machine Learning*, McGraw-Hill, 1997.

[4] M. Oprea, Toward the Development of a Knowledge-based System for Environmental Protection, *Proceedings of the 8th International Conference IPMU - Information Processing and Management of Uncertainty in Knowledge-based Systems*, Vol. II, Madrid, Spain, 3-7 July 2000, 1003-1008.

[5] M. Oprea, Considerations Regarding the Development of the Explanation Module of an Expert System, *The Romanian Journal of Informatics and Automation*, vol. 9, nr. 4, 1999, 43-48.

[6] M. Oprea, Some Remarks about Rules Learning by Genetic Algorithms Approach, *The Proceedings of the 4-th International Symposium of Economic Informatics*, Bucharest, May 1999, INFOREC Printing House, Bucharest, 722-727.

[7] J.R. Quinlan, *C4.5: Programs for Machine Learning*, Philadelphia, PA: Morgan Kaufmann, 1993.

[8] M. Tolun, S. Abu-Soud, ILA: an inductive learning algorithm for rule extraction, *Expert Systems with Applications*, 14, 1998, 361-370.

[9] X. Wu, D. Urpani, J. Sykes, Rule Induction without Decision Tree Construction, *Proceedings of the 12th European Conference on Artificial Intelligence*, John Wiley, 1996, 463-467.

[10] X. Wu, *Knowledge Acquisition from Databases*, Ablex Publishing Corp., USA, 1995.