This paper is not available here, but the enclosed unpublished paper is very similar to it.

AN EMPIRICAL COMPARISON BETWEEN LEARNING DECISION TREES FROM EXAMPLES AND FROM DECISION RULES

by

I. Imam
R. S. Michalski

Proceedings of the Ninth International Symposium on Methodologies for Intelligent Systems (ISMIS-96), Zakopane, Poland, June 10-13, 1996.

LEARNING FOR DECISION MAKING:
An Approach and a Comparative Study

Ibrahim F. Imam and Ryszard S. Michalski†

Machin Learning and Inference Laboratory
George Mason University
Fairfax, VA. 22030
†Also with the Institute of Computer Science, Polish Academy of Sciences
{iimam, michalski} @aic.gmu.edu

ABSTRACT

This paper considers the issue of what is the best form in which decision-oriented knowledge should be represented and how it can be most effectively used for decision making. The proposed answer is that such knowledge should be learned and represented in a declarative form and—whenever needed for decision making— efficiently transferred to a procedural form tailored to the specific decision making situation. Such an approach combines advantages of the declarative representation, which facilitates learning and incremental knowledge modification, and the procedural representation, which facilitates decision making. This approach also allows one to determine decision structures without attributes that unavailable or difficult to measure in the given situation. Experimental investigations of the system, FRD-1, have demonstrated that decision structures obtained via the declarative route often have higher predictive accuracy and are also simpler than those learned directly from facts.

Key words: machine learning, decision tree learning, decision making, inductive rule learning, decision structures.

1 INTRODUCTION

The main step in the development of intelligent systems or agents is the creation of knowledge structures that govern the system's decision making process. An attractive route for creating such structures (at least partially) is to acquire them through learning from original facts. A powerful and effective tool for describing decision processes is a decision structure, which is an acyclic graph that specifies an order of tests to be applied to an object (or a situation) to arrive at a decision about that object. The nodes of the structure are assigned individual tests (which may correspond to a single attribute, a function of attributes, or a relation), the branches are assigned possible test outcomes (or ranges of outcomes), and the leaves are assigned one specific decision or a set of candidate decisions (with corresponding probabilities), or an undetermined decision (Imam & Michalski, 1993; Kohavi, 1994). A decision structure reduces to a familiar decision tree, when each node is assigned a single attribute and has at most one parent, the branches from each node are assigned single values of that attribute, and leaves are assigned single, definite decisions. Thus, the problem of generating a decision structure is a generalization of the problem of generating a decision tree.

A decision tree/decision structure can be an effective tool for describing a decision process, as long as all the required tests can be measured, and the decision making situations it was designed for do not change much (e.g., there is no significant change in the frequency distribution of different decisions). Problems arise when these assumptions do not hold. For example, in some situations measuring certain attributes may be difficult or costly. In such situations it is desirable to reformulate the decision structure so that the "inexpensive" attributes are evaluated first (are assigned to the nodes close to the root), and the "expensive" attributes are evaluated only if necessary (are assigned to the nodes far away from the root). If an attribute cannot be measured at all, it is useful to either modify the structure so that it does not contain that attribute, or—when this is impossible—to specify a set of alternative candidate decisions in this situation and their probabilities. A restructuring is also desirable if there is a significant change in the frequency of occurrence of different decisions.

A restructuring of a decision structure (or a tree) in order to suit new requirements (is usually quite difficult. This is because a decision structure is a procedural knowledge representation, which imposes an evaluation order on the tests. In contrast, no evaluation order is imposed by a declarative representation, such as a set of decision rules. Tests (conditions) of rules can be evaluated in any order. Thus, for a given set of rules, one can usually build a large number of logically equivalent decision structures (trees), which differ in the test ordering. Due to the lack of "order constraints," a declarative representation (rules) is much easier to modify to adapt to different situations than a procedural one (such as a decision structure or a tree). On the other hand, to apply decision rules to make a decision, one needs to decide in which order tests are evaluated, and thus, needs a decision structure. The above indicates that the form in which knowledge can be most easily learned and updated is different than the form in which it is most readily used for decision making.

The paper presents an attractive solution of the above opposite requirements. In the proposed methodology knowledge is acquired, modified and stored in a declarative form (of decision rules). Whenever it is needed for decision making, it is efficiently transformed into a task-oriented decision structure. The methodology, called FRD-1 (from Facts to Rules to Decisions), creates decision structures that are tailored for a given decision making situation. The input for this process are either decision rules obtained from a learning program or original facts (examples of situation-decisions pairs).

Such "virtual" decision structures are easy to tailor to any given decision making situation. The above idea rests upon the assumptions that learning decision rules can be done efficiently and that there can be a fast and effective algorithm for transferring rules to decision structures. The initial idea that led to the development of this methodology is described by Imam and Michalski (1993).

2 RELATED RESEARCH

The method used here for learning decision structures from decision rules is similar to learning decision trees from examples. The problem of learning decision trees from examples is the problem of generating decision trees that classify sets of given examples according to the decision classes they belong to. The essential aspect of any inductive decision tree method is the attribute selection criterion. The attribute selection criterion measures how good the attributes are for discriminating among the

given set of decision classes. The best attribute according to the selection criterion is chosen to be assigned to the root of the tree.
The attribute selection criteria can be divided into three categories. These categories are logic-based, information-based, and statistics-based. The logic-based criteria for selecting attributes use logical relationships between the attributes and the decision classes to determine the best attribute to be a node in the decision tree, such as the MAL criterion, minimizing added leaves, (Michalski, 1978). The MAL criterion uses conjunction and disjunction operators. The information-based criteria are based on information theory. These criteria measure the information conveyed by dividing the training examples into subsets. Examples of such criteria include the information measure IM, the entropy reduction measure, and the gain criteria (Quinlan, 1979, 83), the gini index of diversity (Breiman, et al., 1984), Gain-ratio measure (Quinlan, 1986), and others (Cestnik & Karalic, 1991). The statistics-based criteria measure the correlation between the decision classes and the other attributes. Such criteria use statistical distributions in determining whether or not there is a correlation. Examples of statistical criteria include the Chi-square and the G statistics (Mingers, 1989).

3 THE FRD-1 METHODOLOGY

The proposed methodology separates the function of knowledge acquisition or discovery from the function of applying knowledge to decision making. The first function is performed by an inductive learning program that searches for knowledge relevant to a given class of decisions, and stores the learned knowledge in the declarative form of decision rules. The second function is performed when the need for decision making arises in some particular situation. Such a function involves assigning a value to a decision variable based on values of attributes characterizing the decision making situation. Figure 1 shows an architecture of the proposed methodology.

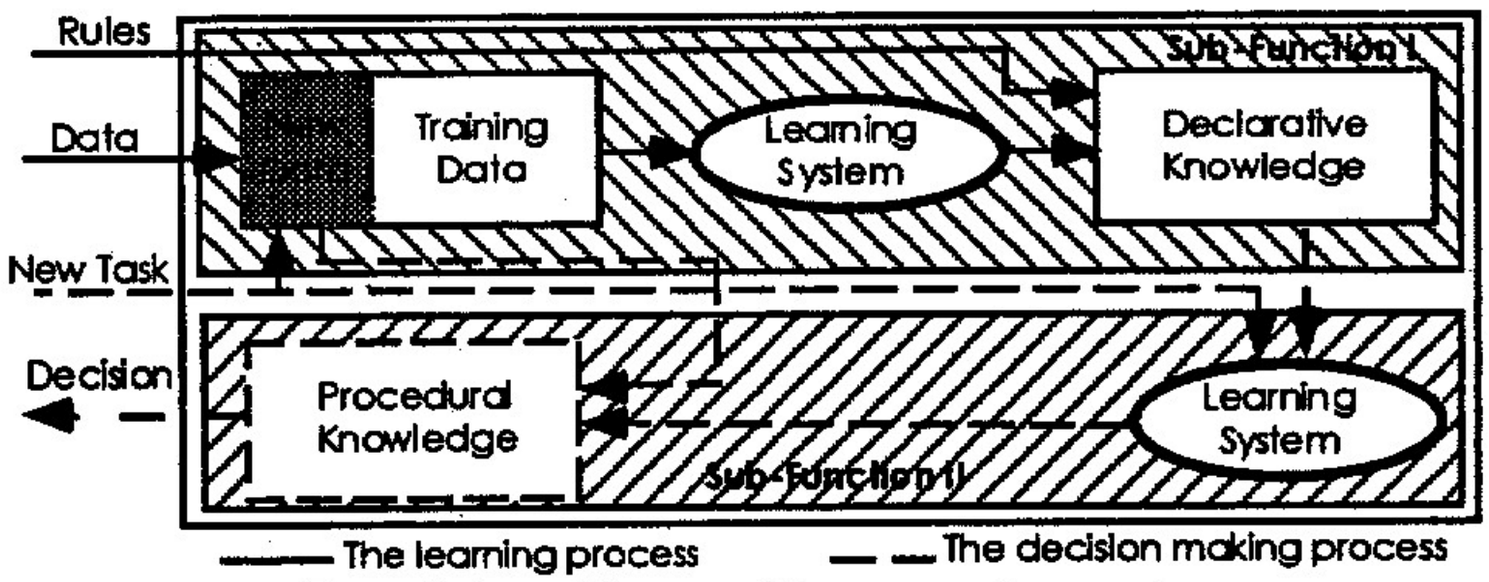


Figure 1: An architecture of the proposed approach.

The proposed methodology is called FRD (from Facts to Rules to Decisions). The decision rules are learned by either the AQ15 (Michalski, et al, 1986) or AQ17 (Bloedorn, et al, 1993) learning systems. They also can be directly edited into the system. The reasons of selecting decision rules as the form of knowledge representation are that they are modular and easy to understand. Rules do not impose any order on the evaluation of the attributes, therefore, they can be evaluated

in many different ways. This makes it possible to flexibly adapt them to different decision making situations.
A decision making problem arises when there is a case or a set of cases to which the system has to assign a decision. Each case is defined by a set of attribute values. Some attribute values may be missing or unknown. A decision structure is derived from the rules that suits the given decision making problem. The learned decision structure associates the given set of cases with the corresponding decisions.
The AQDT-2 algorithm works by determining the "best" test (here, an attribute) at each step of the process. The "best" test is determined by analyzing the decision rules. The system aims at producing decision structures with the minimum number of nodes or the minimum cost (where the "cost" is defined as the total cost of classifying examples, given the cost of measuring individual attributes and the expected probability distribution of examples of different decision classes).
The best test (attribute) is selected on the basis of its utility , which is a combination of one or more of the following elementary criteria: 1) cost (the cost of measuring the attribute or the test; if the cost of tests is unknown or irrelevant, all tests assume the same default cost 2) disjointness, which captures the effectiveness of the test in discriminating among decision rules for different decision classes, 3) importance, which determines the importance of a test in the rules, 4) value distribution, which characterizes the distribution of the test importance over its of values, and 5) dominance, which measures the test presence in the rules.
Figure 2 shows the AQDT-2 algorithm for learning decision structure from decision rules. The complexity of the AQDT-2 algorithm is estimate as:

$$Cmplx(AQDT) = O(r * m * \log l)$$

where r is the total number of rules, m is the number of attributes, and l is maximum the number of rules and the number of attributes.

4 EMPIRICAL ANALYSIS OF THE FRD-1 SYSTEM

This section presents empirical results from extensive testing of the FRD-1 system on several problems, using different amounts of training examples and applying different settings of the system's parameters. For comparison, it also presents results from applying a well-known decision tree learning system (C4.5) to the same problems. The system was applied to the following problems: EAST-WEST (or TRAINS), MONK-1, MONK-2, MONK-3, Engineering Design, Mushrooms, and Breast Cancer.
The experiments were performed 100 times with different relative sizes of the training data: 10%, 20%, ..., 90%. Specifically, from the set of all available examples for each problem, 100 randomly selected subsets of 10% of data were chosen for rules learning, then 20%, etc. The remaining examples in each case were used for testing the obtained descriptions and determining their prediction accuracy.

4.1 Learning Task-oriented Decision Structures

This subsection briefly illustrates the capabilities of the system for learning task-oriented decision structures. Experiments involved the East-West Challenge problem (Michie, et al, 1994). The East-Westbound problem is concerned with discriminating between two groups of train-like structures. Each "train" consists of several cars (two to four), each containing various loads of different shapes.
The program AQDT-2 (Imam & Michalski, 1993) accepts rules in the form of an array of attribute-value assignments. To describe the East-West Challenge problem in

the suitable format, a set of eight (8) attributes was generated that can completely describe any car in the train. Each train is described by one rule, which can be of different length. To specify the number (position) of a given car in the train, each of the eight attributes is associated with a two-digit code (i, j); the first identifies the location of the car and the second identifies the attribute itself. For example, the number 3 in the attribute "x32" refers to the third car, and the number 2 refers to the second attribute (e.g., "car shape"). Thus, attribute x32 describes the shape of the third car. Table 2 shows a summary of these data.

The AQDT-2 Algorithm

Given: A set of rules and a decision making situation.
Determined: A decision structure optimized for the given decision making situation.

Step 1: Evaluate each attribute occurring in the ruleset context using the LEF attribute ranking measure. Select the highest ranked attribute, say attribute A.
Step 2: Create a node of the tree (initially, the root; afterwards, a node attached to a branch), and assign to it the attribute A. In standard mode, create as many branches from the node as the number of legal values of the attribute A, and assign these values to the branches. In compact mode (decision structures), create as many branches as there are disjoint value sets of this attribute in the decision rules, and assign these sets to the branches.
Step 3: For each branch, associate with it a group of rules from the ruleset context that contain a condition satisfied by the value(s) assigned to this branch. Remove from the rules these conditions. If there are rules in the ruleset context that do not contain attribute A, add these rules to all branches stemming from the node assigned attribute A.
Step 4: If all the rules in a ruleset context for some branch belong to the same class, create a leaf node and assign to it that class. If all branches of the trees have leaf nodes, stop. Otherwise, repeat steps 1 to 4 for each branch that has no leaf.

Figure 2: The AQDT-2 algorithm

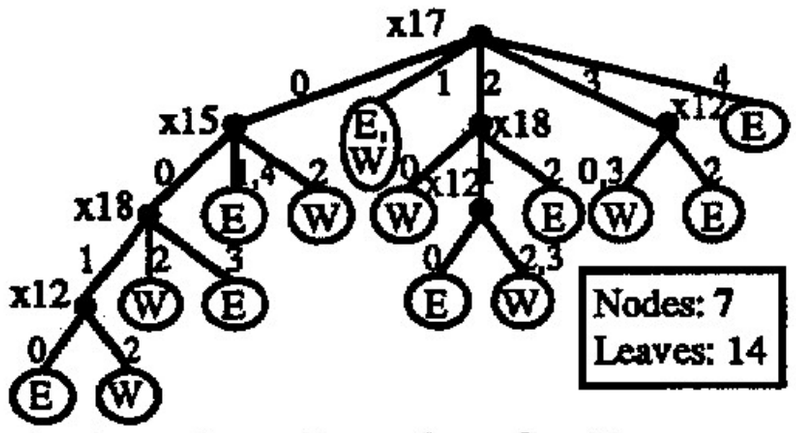
Table 2: The set of attributes used in the experiments

Name	Attribute	Values					
		0	1	2	3	4	5
x,1	Car_top	open	closed				
x,2	Car_shape	rectangle	hexagon	bucket	u_shaped	ellipse	
x,3	Car_length	short	long				
x,4	Car_frame	not_double	double				
x,5	Top_shape	none	peaked	flat	arc	jagged	
x,6	No_of_wheels	two	three				
x,7	Load_shape	rectangle	hexagon	circle	triangle	utriangle	diamond
x,8	No_of_loads	no_loads	one	two	three		

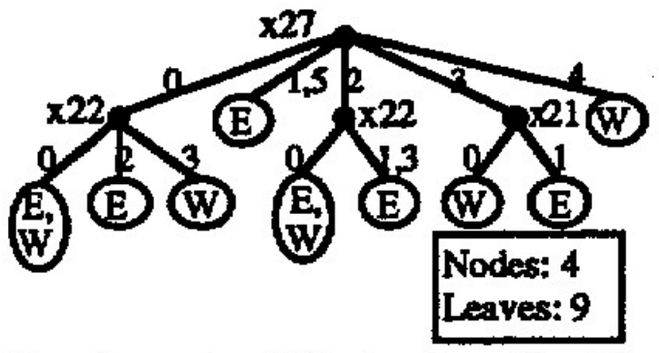
i= 1, 2, 3, 4 and stands for the car number.

To demonstrate the capability of the system to learn decision structures under different constrains regarding what attributes can be easily measured, the experiments involved different "admissible" sets of attributes. For example, Figure

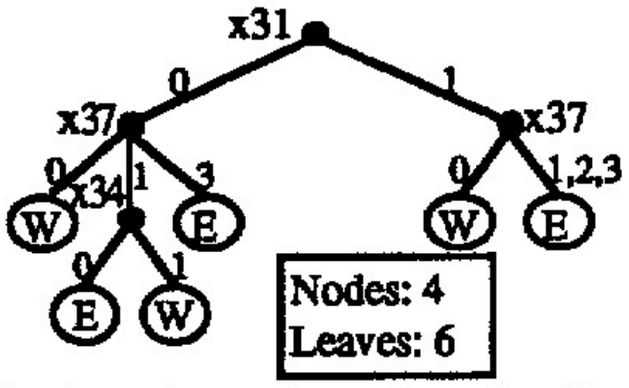
3a, shows a decision structure learned using only attributes describing the first car. This decision structure classifies correctly 19 trains (out of 20). Figure 3b shows a decision structure learned using only attributes describing the second car. It classifies correctly 18 trains. Both decision structures have leaves with multiple decisions, which means that there is identical first or second car in the two decision classes (sets of trains). Figure 3c shows a decision structure learned using attributes describing the third car only. It classifies correctly all 20 trains with three cars or more(14). In Figure 3d, attributes x37 and x34 were given lower cost than attribute x31. The last decision structure classifies correctly any train with three or more cars correctly.



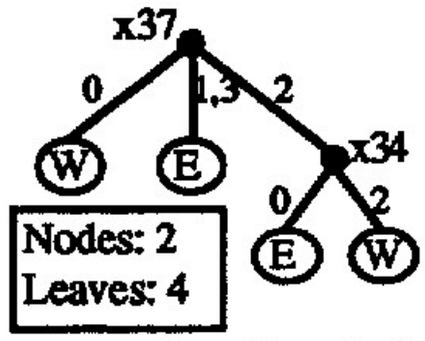
a) using only attributes from Car #1



b) using only attributes from Car #2.



c) using only attributes from Car #3



d) with lower costs for x34 and x37.

Figure 3: Task-oriented decision structures learned by AQDT-2 with different costs.

4.2 Comparative Study

This subsection presents a comparison between the decision trees obtained by FRD-1 and C4.5, a well-known program for learning decision trees (Quinlan, 1990). The experiments were performed on 6 different data sets. Both systems were set to their default parameters. The experiments were divided into two parts. The first part is concerned with designed problems, the MONKs (Thrun, Mitchell & Cheng, 1991), and the second part was concerned with real-world problems: Wind bracing for tall buildings (Arciszewski, et al, 1992), Mushrooms classification, Breast cancer diagnosis.

All the results reported here are the average of 100 runs. For each data set, we reported the predictive accuracy, the complexity of the learned decision trees, and the time taken for learning. Figure 4 shows the results from comparative study on the three MONKs problems. Figure 5 shows the results obtained when using the wind bracing, mushroom, and the breast cancer problems.

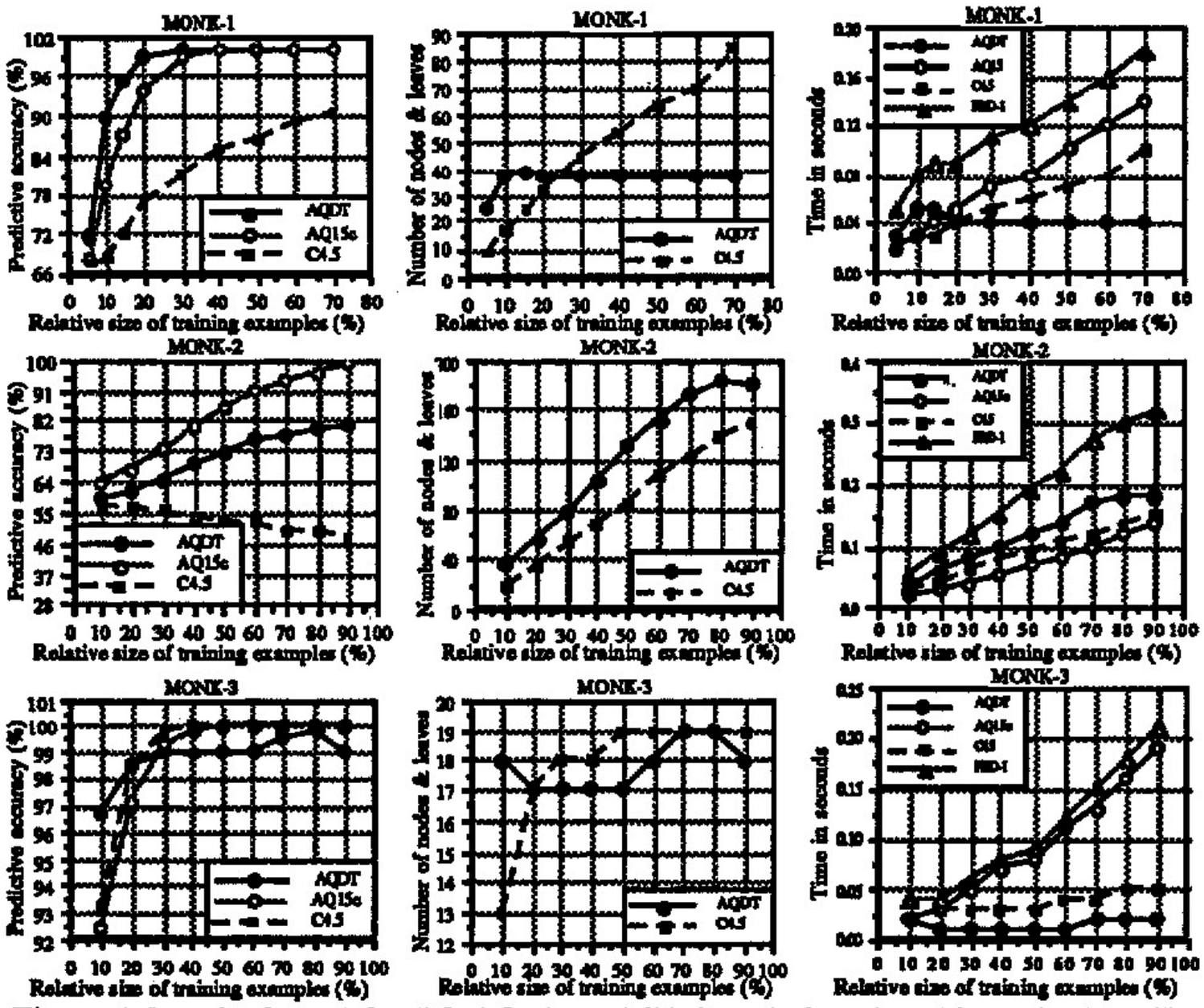


Figure 4: Results from AQDT-2, AQ15c and C4.5 on designed problems (MONKS).

4.3 Analysis of the Results

This section includes an analysis of the results presented in section 4-2. The analysis covers the relationship between different characteristics of the input data and the learning parameters for both subfunctions of the approach. Table 3 shows the best parameter settings for learning decision rules with different databases. The information in this table is based on the predictive accuracy of decision trees learned by AQDT-2 from decision rules learned by AQ15c with different parameter settings. Some heuristics were used in driving these information. One heuristic was: if the difference in predictive accuracy between two widths of the beam search is less than 2%, then the smaller is better. Another one was: if the predictive accuracy of different types of covers is changing (i.e. for one type of covers, it is higher with some widths of beam search or with certain rule's type and lower with other types of covers), the best cover is determined according to the best width of the beam search and the best rule type.

It was clear that AQDT-2 works better with characteristics rules rather than discriminant. In most problems, when changing the width of the beam search of the AQ15c system, the changes in the predictive accuracy of decision trees learned by AQDT-2 were within $\pm$2%. Disjoint rules were better than intersected rules for learning decision trees. Generally, decision trees learned from intersected rules were slightly bigger than those learned from disjoint rules.

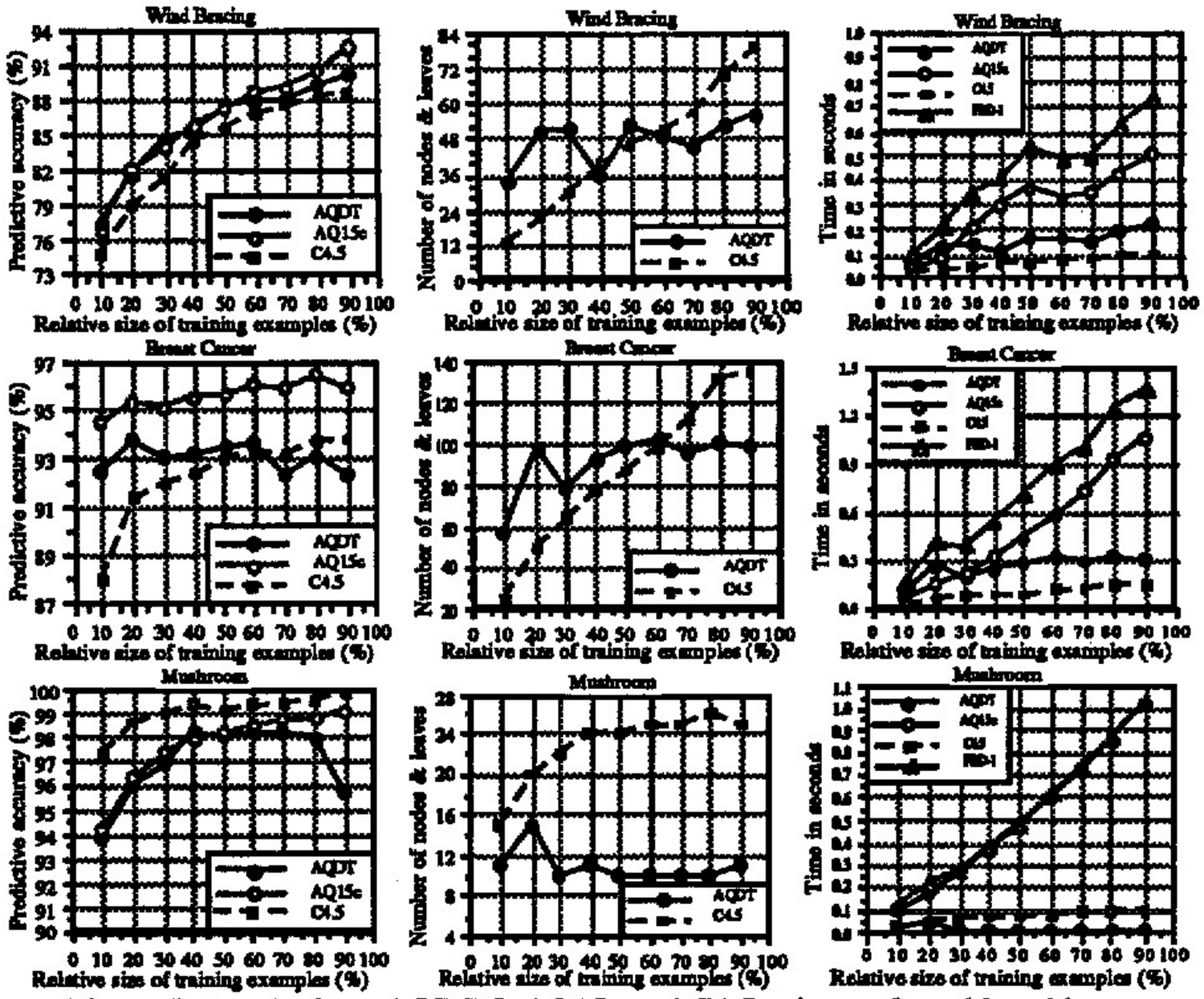


Figure 5: Results from AQDT-2, AQ15c and C4.5 using real world problems.

Table 3: Recommended parameter settings for AQ15c.

Characteristics of the Data	Cover Type	Beam Size	Rules Type
Average Size, Complex, Noise-Free	Intersecting	1	Characteristics
Small Size, Simple, Noise-Free	Disjoint	1	Characteristics
Small Size, Complex, Noise-Free	Disjoint	1	Characteristics
Small Size, Simple, Noisy	Disj. or Inter.	1	Characteristics
Large Size, Complex, Noise-Free	Disjoint	N/A	Characteristics
Large Size, Complex, Noisy	Intersecting	N/A	Characteristics

To analyze the comparative study between AQDT-2 and C4.5 for learning decision trees, a set of heuristics were used to summarize these results. These heuristics are: 1) if the difference between the average predictive accuracy of the two systems is within $\pm$ 2%, the predictive accuracy is considered to be the same. Otherwise, the predictive accuracy is considered high or low; 2) if the average learning time is within $\pm$ 0.1 seconds, the learning time is considered the same.

Table 4 shows a summary of the comparison between AQDT-2 and C4.5. The summary includes comparing the predictive accuracy, the size of learned decision trees, and the learning time. The value in each cell refers to the system which perform better. When the training data represents a small portion of the representation space, AQDT-2 produces bigger but accurate decision trees. However,

C4.5 produces smaller but less accurate decision trees. When the training data represents a very large portion of the representation space, AQDT-2 usually produces smaller decision trees with better accuracy except with noisy data. The size of decision trees learned by C4.5 relatively grow higher when the training data increases. Also, C4.5 works better than AQDT-2 with noisy data. The reasons of this are because AQDT-2 over generalizes the decision rules.

Table 4: The best performing system for different data characteristics.
≈ : means that the difference between the two systems is insignificant.

Characteristics of the Problems	Best Performing System				
(Wind Bracing, MONK-1,	Accuracy		Complexity		CPU
MONK-2, MONK-3	Training size		Training size		Time
Breast Cancer, and Mushroom)	<25%	>75%	<25%	>75%	
Average Size, Complex, Noise-Free	AQDT-2	≈	C4.5	AQDT-2	≈
Small Size, Simple, Noise-Free	AQDT-2	AQDT-2	C4.5	AQDT-2	AQDT-2
Small Size, Complex, Noise-Free	AQDT-2	AQDT-2	C4.5	C4.5	≈
Small Size, Simple, Noisy	≈	≈	≈	≈	≈
Large Size, Complex, Noise-Free	AQDT-2	≈	C4.5	AQDT-2	C4.5
Large Size, Complex, Noisy	C4.5	≈	AQDT-2	AQDT-2	≈

5. CONCLUSION

The paper presented a methodology for determining task-oriented decision structures from decision rules or examples. The preliminary experiments with an implemented system, FRD-1, have demonstrated many advantages of the presented approach such as the ability to tailor the decision structure to the decision problem at hand, and learning decision structures directly from rules or from examples. Generally, the structures obtained by the AQDT-2 system are simpler, accurate and easy to modify (due to incremental learning ability) than decision trees obtained from examples. Future research needs to investigate the best areas of applicability of this methodology, confirm these findings on other decision making tasks and to develop the ability of building more complex decision structures that those explored here.

The proposed methodology advocates storing the decision knowledge in the declarative form of decision rules, which are determined by induction from examples or by an expert. A decision structure is generated when is needed, and in the form most suitable for the given decision-making situation (i.e., a class of cases of interest). A criticism may be leveled against this methodology that in order to determine a decision structure from examples, it is necessary to go through two levels of processing, while there exist methods that produce decision trees efficiently and directly from examples. Putting aside the issue that decision structures are more general than decision trees, it is argued here that this methodology has many advantages that fully justify it. The main advantages include: 1) decision structures produced by the methods in the experiments conducted had higher predictive accuracy and were simpler (sometimes significantly so) than decision trees produced from the same data; 2) decision structures produced from rules can be easily tailored to a given decision-making situation, i.e., they can avoid measuring expensive attributes, or can put them in the lowest parts of the structure; 3) by storing decision knowledge in the declarative form of modular decision rules, the methodology makes it easy to modify decision knowledge to account for new facts or changing

conditions; 4) the process of deriving a decision structure from a set of rules is very fast and efficient, because the number of rules per class is usually much smaller than the number of examples per class; and 5) the presented method produces decision structures, whose nodes can be original attributes, or constructed attributes that extend the original knowledge representation (this is due to the application of constructive induction programs AQ17-DCI and AQ17-HCI). The price for these advantages is that the system has to generate decision rules first, and then create from them decision structures. In the AQDT-2 method, this first phase is done by an AQ algorithm-based rule learning method.

ACKNOWLEDGMENTS

The authors thank the UCI repository of machine learning databases for the data used in most of the presented experiments. They also thank M. Mustafa and T. Arciszewski for the data used in the "Wind bracing" experiments.

This research was done in the Machine Learning and Inference Laboratory at George Mason University. The Laboratory's activities are supported in part by the Advanced Research Projects Agency under grant No. N00014-91-J-1854 administered by the Office of Naval Research, in part by the Advanced Research Projects Agency under grants F49620-92-J-0549 and F49620-95-1-0462 administered by the Air Force Office of Scientific Research, in part by the Office of Naval Research under grant N00014-91-J-1351, and in part by the National Science Foundation under grants DMI-9496192 and IRI-9020266.

REFERENCES

Arciszewski, T, Bloedorn, E., Michalski, R., Mustafa, M., and Wnek, J., "Constructive Induction in Structural Design", Reports of Machine Learning and Inference Laboratory, MLI-92-7, George Mason University, 1992.

Bloedorn, E., Wnek, J., Michalski, R.S., and Kaufman, K., "AQ17: A Multistrategy Learning System: The Method and User's Guide", Reports of Machine Learning and Inference Laboratory, MLI-93-12, George Mason University, 1993.

Bratko, I. & Lavrac, N., (Eds.), Progress in Machine Learning, Sigma Wilmslow, England, Press, 1987.

Breiman, L., Friedman, J.H., Olshen, R.A. & Stone, C.J., "Classification and Regression Structures", Belmont, California: Wadsworth Int. Group, 1984.

Clark, P. & Niblett, T., "Induction in Noisy Domains" in I. Bratko and N. Lavrac, (Eds.), Progress in Machine Learning, Sigma Press, Wilmslow, 1987.

Cestnik, B. & Karalic, A., "The Estimation of Probabilities in Attribute Selection Measures for Decision Structure Induction" in Proceeding of the European Summer School on Machine Learning, July 22-31, Priory Corsendonk, Belgium, 1991.

Imam, I.F. and Michalski, R.S., "Learning Decision Structures from Decision Rules: A method and initial results from a comparative study", in Journal of Intelligent Information Systems JIIS, Vol. 2, No. 3, pp. 279-304, Kerschberg, L., Ras, Z., & Zemankova, M. (Eds.), Kluwer Academic Pub., MA, 1993.

Kohavi, R., "Bottom-Up Induction of Oblivious Read-Once Decision-Graphs: Strengths and Limitations", Proceedings of AAAI-94, pp. 613-18, Seattle, 1994.

Michie, D., Muggleton, S., Page, D. and Srinivasan, A., "International East-West Challenge", Oxford University, UK, 1994.

Michalski, R.S., "Designing Extended Entry Decision Tables and Optimal Decision Trees Using Decision Diagrams", Technical Report No.898, Urbana: University of Illinois, March, 1978.
Michalski, R.S., "A Theory and Methodology of Inductive Learning", Artificial Intelligence, Vol. 20, (pp. 111-116), 1983.
Michalski, R.S., Mozetic, I., Hong, J. and Lavrac, N., "The Multi-Purpose Incremental Learning System AQ15 and Its Testing Application to Three Medical Domains", Proceedings of AAAI-86, (pp. 1041-1045), Philadelphia, PA., 1986.
Mingers, J., "An Empirical Comparison of selection Measures for Decision-Structure Induction", Machine Learning, Vol. 3, No. 3, (pp. 319-342), Kluwer Academic Publishers, 1989.
Niblett, T. and Bratko, I., "Learning decision rules in noisy domains" Proceeding Expert Systems 86, Brighton, Cambridge: Cambridge University Press, 1986.
Quinlan, J.R., "Discovering Rules By Induction from Large Collections of Examples", in D. Michie (Edr), Expert Systems in the Microelectronic Age, Edinburgh University Press, 1979.
Quinlan, J.R., "Learning efficient classification procedures and their application to chess end games" in R.S. Michalski, J.G. Carbonell and T.M. Mitchell, (Eds.), Machine Learning: An Artificial Intelligence Approach. Los Altos: Morgan Kaufmann, 1983.
Quinlan, J. R., "Probabilistic decision structures," in Y. Kodratoff and R.S. Michalski (Eds.), Machine Learning: An Artificial Intelligence Approach, Vol. III, San Mateo, CA, Morgan Kaufmann Publishers, (pp. 63-111), June, 1990.
Thrun, S.B., Mitchell, T. and Cheng, J., (Eds.) "The MONK's Problems: A Performance Comparison of Different Learning Algorithms", Technical Report, Carnegie Mellon University, October, 1991.