From: AAAI-84 Proceedings. Copyright ©1984, AAAI (www.aaai.org). All rights reserved.

LEARNING PROBLEM CLASSES BY MEANS OF EXPERIMENTATION AND GENERALIZATION

Agustín A. Araya

Departamento de Ciencia de la Computación
P. Universidad Católica de Chile
Casilla 114-D, Santiago, Chile

ABSTRACT

We discuss a method of learning by practice based on the idea of determining classes of problems that can be solved in simplified ways. A description of a class is obtained by processes that hypothesize descriptions, generate and classify problem variations, and test the hypotheses against them. The approach has been implemented in a system that learns by practice in a domain of elementary physics. The system has two main components, a Problem Solver and a Learning Agent. The Problem Solver handles the problems in the domain and the Learning Agent does the actual learning. To perform its tasks the Learning Agent utilizes algorithms, heuristics, and domain knowledge, and for this reason it can be regarded as an expert system whose expertise resides in being able to learn by experimentation and generalization.

1. INTRODUCTION

In recent years learning has been increasingly recognized as an important area for Artificial Intelligence research ([6],[11]). While human expert behavior is characterized by the ability to increase expertise by learning in the course of solving problems, current expert systems lack many important capabilities in this respect: i) They are not capable of analyzing their own solutions, and thus, they cannot determine if a solution can be "improved" (e.g., by simplifying it). ii) They don't try "mental experiments", that is, imaginary problems in which they could apply new methods or heuristics that might improve their problem solving capabilities. iii) They are not capable of remembering. If they are given a problem similar, or even identical, to one previously solved, they will not recognize this fact and will repeat the solution process.

This work was partially done at the Departments of Computer Science of the University of Texas at Austin, and the P.Universidad Católica de Chile. It was supported in part by grant 216/82 of the Dirección de Investigación, P.Universidad Católica de Chile.

We report research on a method of learning by practice that provides these capabilities to a certain degree. The method is embodied in a system called DARWIN that functions in a domain of elementary physics in which interacting ideal rigid bodies are in equilibrium. Starting with a general method of solving problems in the domain, the system learns problem classes and their corresponding specialized methods of solution by means of experimentation and generalization. This presentation is organized as follows: after characterizing the problem under study an overview of the learning approach is given, the processes used to obtain a description of a problem class are explained, and some of the limitations and difficulties encountered are discussed.

2. THE PROBLEM

In the description of the ISAAC system [10], that solves physics problems stated in natural language, it was noted that for some of the problems that it solved a human expert would have produced a simpler solution. This observation lead us to consider the following situation. Let us suppose that a system has a general method for solving problems in its domain. One of the things it can learn by practice is that there are problems that can be solved in simplified ways, that is:

IF problem belongs-to problem-classj
THEN use special-methodj

Thus, two things must be learned: a description of a problem class and its corresponding special method. One way of doing this is by applying a deductive approach. If the system had a theory of the domain it could deduce that for problems having certain characteristics a simplified solution could be obtained. In this paper, on the other hand, we take an empirical approach in which the system tries to determine descriptions of problem classes by performing experiments. Thus, a theory of the domain is not necessary and this represents an important advantage because the experimentation and generalization methods can be made relatively independent of the domain. The price one pays, however, is diminished certainty of the acquired knowledge. Since exhaustive experimentation is impossible, this knowledge represents only plausible conjectures. Research on learning by practice and discovery has been reported in [4],[6] and [9], generalization methods are presented in [2],[7] and [8], a method of experimentation based on perturbation is proposed in [3], and expertise in solving problems in physics has been analyzed in [1] and [5].

3. OVERVIEW OF THE LEARNING APPROACH

The approach has been implemented in a system with two main components: a Problem Solver (PS) and a Learning Agent (LA). To emphasize the fact that the Learning Agent is responsible for the evolution of the problem solver by developing specialized solution methods to cope with problems in its domain, we refer to it as the DARWIN system.

```
Analyze solution of problem
Determine problem class description:
   cycle
     Hypothesize descriptions
     Generate problem variations
     Classify variations
     Test descriptions against
       variations
   end cycle
Build specialized method and integrate
it in PS
```

Fig.1: Outline of the Learning Process

The PS starts with a general method that allows it to solve problems in its elementary physics domain. The method is based on the principles of equilibrium of forces and moments. The LA acts as a supervisor of the PS and its task can be divided into the following stages (See Fig.1): 1) Determine if a given problem can be solved in a simplified way: using knowledge about the domain, the solution is analyzed to see if it has some special characteristics that could lead to simplifications (e.g., that one or more equations produced by the general method are not actually needed to solve the problem). 2) Determine a description of the problem class: when it has been found that the solution of a problem can be simplified the LA uses experimentation and generalization processes to obtain a description of the class, in such a way that all the problems belonging to it can be solved in the same simplified way. 3) Derive the specialized method and integrate it into the PS: using knowledge about the problem domain and about information consumed and produced at each step of the general solution method, the system builds the specialized solution method and adds it to the PS. The new method can be obtained by eliminating steps from the general method and by replacing some steps by simpler ones. The second stage is the most important and complex and will be analyzed in detail in the next section. Due to space limitations the other two stages will not be given further consideration.

4. DETERMINING A DESCRIPTION OF A PROBLEM CLASS

The learning situation under study is characterized by an important fact: the system has made a <u>single</u> observation that a specific problem can be solved in a simplified way. This fact is in itself useless, because the likelihood that exactly the same problem will be encountered by the system in the future is nil, so that this knowledge will most likely never be used! This implies that the system must perform some kind of generalization to determine a class of problems to which the specialized solution applies. In order for the system to obtain a generalization from only one observation it needs to perform experiments to gather additional information. Thus, the determination of the description of the problem class is carried out by two highly intertwined processes: experimentation and generalization. To perform these tasks the LA utilizes algorithms, heuristics, and domain knowledge, and for this reason it can be regarded as an expert system whose expertise resides in being able to learn by experimentation and generalization.

The description of the class will be expressed in terms of first order logic and will be based on the problem description and the background knowledge available to the system [12]. The latter kind of information can be classified as follows: i) predicates that refer to relations between positions of objects (e.g.: "symmetrically located", "at left end"); ii) predicates that refer to relations between attributes of objects (e.g.: "perpendicular", "same size"); and iii) predicates that refer to attributes of objects (e.g.: "angle of force equal to x"). To constrain the search space we have decomposed the process so that at each stage different kinds of predicates are determined in the order indicated by the previous classification. We illustrate the process with an example:

> "A horizontal lever 15 ft long is supported at the left and right ends by a pivot and rolling pivot, respectively. Two forces are applied to the lever, one which has a magnitude of 120 lb, an angle of 270 degrees, and is applied at a point 3 ft from the left end; and another which has a magnitude of 120 lb, an angle of 270 degrees, and is applied at a point 12 ft from the left end. Determine the forces exerted by the pivots so that the lever is in equilibrium." (See Fig.2a).

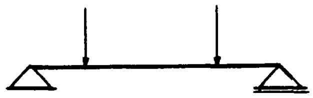

Fig. 2a: Initial Problem

Fig. 2b: Most General Problem

After preprocessing the initial description the following information is obtained:

lever:	force1:	force2:
weight=0	magn=120	magn=120
angle=0	angle=270	angle=270
length=15	pos=3	pos=12
	desunk=False	desunk=False

pivot:	rpivot:
pos=0	pos=15
desunk=True	desunk=True

(rpivot:rolling pivot; desunk:desired unknown; pos:position)

The PS produces a solution to the problem which is then analyzed by the LA. The following special characteristics are detected: the horizontal component of the force applied by the pivot is zero, the vertical components of the forces applied by the pivot and the rolling pivot have the same magnitude, and all the torques due to horizontal components of the forces are zero.

4.1 Determining Position and Attribute Relations

The system starts by determining which relations between positions and attributes of objects are important.

4.1.1 Hypothesize Descriptions: The system uses background knowledge and hypothesizes that the set of predicates that evaluate to True in the initial problem constitutes a description of the class. The following predicates defined in the background knowledge are true in the initial problem:

(1) symloc(pivot,rpivot) (*)
(2) atleftend(pivot)
(3) atrightend(rpivot)
(4) symloc(force1,force2)
(5) sameangles(force1,force2)
(6) perpendicular(force1,lever)
(7) perpendicular(force2,lever)
(8) samesize(force1,force2)
(9) mirrorangles(force1,force2)

(*) (symloc: symmetrically located with respect to the center of the lever)

4.1.2 Generate Problem Variations: Some of these relations may not be necessary and since we want to obtain as general as possible a description the system should try to eliminate them. This is accomplished by proposing variations of the original problem and seeing how the predicates behave with respect to them. To determine interesting problem variations the system uses a method based on what we call the "mutual support principle" (MSP). Let us assume that the description of the problem class will be expressed as a conjunction of predicates. (In general, it will consist of a disjunction of conjuncts, but the MSP can be easily restated to cover that case). The MSP simply says that conjuncts "support" each other in the sense that negative variations not rejected by one conjunct must be rejected by others, and positive variations must be accepted by all predicates (See 4.1.3, below).

Using this principle, problem variations are generated as follows: for every predicate found true in the initial problem generate, if possible, "true" and "false" variations (T-vars and F-vars, respectively). A T-var with respect to a predicate is a variation in which the predicate evaluates to true. A false variation is a variation in which the predicate evaluates to false. T-vars and F-vars are generated using functions in the background knowledge, associated with each predicate. In general there may be several ways of generating T and F-variations from a predicate. Let us consider the predicate mirrorangles(force1,force2). F-variations can be generated as follows: a) leave the angle of force1 fixed and select a random angle for force2; b) fix angle of force2 and select a random angle for force1; c) select random angles for both force1 and force2. In all of these variations the predicate evaluates to False, as desired. The other predicates are affected in different ways. For instance, perpendicular(force2,lever) will evaluate to False in a) and c), but will evaluate to True in b). We can conclude from this that in order to obtain useful variations, as many different ways of generating them as possible must be considered.

4.1.3 Classify variations: The variations generated must be classified as positive (POS) or negative (NEG) according to whether their solutions do or do not have the same special characteristics that were detected in the solution of the initial problem.

4.1.4 Test the descriptions: Finally, the predicates are tested against the sets of POSs and NEGs. In addition, since the system is trying to obtain as general as possible a description, a "most general" variation must be picked from all the POSs variations. The criterion used is to select the one that is rejected by more predicates, because that means that it satisfies fewer constraints. (In general, this criterion may give more than one "most general" variation, meaning that there are alternative most general structures. In the current implementation the system picks any one of them). Once a most general variation is selected, the predicates that reject it are eliminated, as long as all the NEGs are still rejected. Continuing with the example, it turns out that predicates (2),(3),(5),(6) and (7) can be eliminated, which means that the initial problem satisfied more constraints than needed. Thus, we obtain a new, more "general" problem, in which the forces don't have to be perpendicular to the lever, the pivots don't need to be at the ends of the lever, and whose solution has the same special characteristics as the solution of the initial problem (See Fig.2b).

4.2 Considering problems with a different number of objects

So far in our analysis the number of objects has remained constant. But in the domain we are considering there may be several forces applied to a rigid body, so that it is worth exploring such cases.

4.2.1 Hypothesize Descriptions: The approach consists of replacing the constants that appear in the predicates by variables and then of quantifying these variables so that they range over sets of objects. The system uses heuristics to constrain the number of quantified predicates that are generated. For the example above, let

```
P = [symloc(force1,force2) and
     samesize(force1,force2) and
     mirrorangles(force1,force2)].
```

That is, P represents the conjunction of all the predicates involving forces that were true in the most general problem. The system generates all predicates of the form: (quant1 f1 (quant2 f2 P)) in which quant1 and quant2 can be the quantifiers "for all", "there is" and "there is one". (Additional predicates are generated by adding the clause notequal(f1,f2) to some of those quantified predicates).

4.2.2 Generate Problem Variations: T and F-variations are generated from each hypothesis, using heuristics that depend on the quantifiers and basic predicates that appear in them. For instance, consider the predicate

```
P1 = (forall f1 (thereisone f2
                  (symloc(f1,f2) and
                   samesize(f1,f2) and
                   mirrorangles(f1,f2)) ))
```

T-variations from this predicate can be generated by adding two forces, one of them with arbitrary attribute values, and the other such that all the basic predicates are satisfied. F-variations can be obtained by adding one or two arbitrary forces.

4.2.3 Classify the variations: Similar to the process described in Section 4.1.3.

4.2.4 Test the Hypotheses: The system tries to determine a minimal set of predicates that accept all POSs and reject all NEGs. This is carried out by a process which is a modification of one proposed in [7] to obtain a disjunctive description of a concept. After performing this process for the example, predicate P1 defined above accepts all POSs and rejects all NEGs generated at this stage, so that it constitutes a partial description of the class.

4.3 Considering Values of Attributes of Objects

In the previous stages the system obtained a partial description of the problem class that takes into account relations between positions and attributes of different objects. In this last stage it is necessary to take into account the values of attributes of single objects. Additional background knowledge about "special" and "non-special" values of attributes of objects is used. For instance, the "weight" attribute of a lever has "0" as special value. Any other value is non-special. Using this knowledge new predicates are hypothesized and tested.

At the end of the whole process the following predicates were obtained for the problem class under study:

```
(forall f1 (thereisone f2
             (symloc(f1,f2) and
              samesize(f1,f2) and
              mirrorangles(f1,f2)) )),
(forall pivot (forall rpivot
              symloc(pivot,rpivot) )),
(forall lever (lever.angle = 0)),
one(pivot), one(rpivot).
```

5. DISCUSSION AND CONCLUSIONS

In a typical inductive situation a set of positive and negative instances of a concept is given. In the approach described above this information is lacking, so that the search for the description of the problem class develops in two spaces: the space of descriptions and the space of variations. The experiments we have performed show that the process of classifying the problem variations is the most expensive. For the example given above, approximately 300 problem variations had to be classified. (The exact number depends on the background knowledge available to the system). In the current (interpreted) implementation of the problem solver it takes an average of 9 seconds to solve a problem. In order to classify the proposed variations of the original problem they must first be solved. Thus, for the system to learn this new class it would need 2700 seconds plus the (substantially smaller) time required to carry out the other processes involved. To lower this cost we have followed a "mixed" approach in which the variations are classified by the instructor (who is informed by the system of the special characteristics their solutions should satisfy). If, however, the instructor is in doubt about any specific variation, he lets the system to classify it by itself. In the experiments performed, the DARWIN system has learned several problem classes that have simplified solutions. It has also learned problem classes in which the unknown(s) take special values which are remembered by the system, and problem classes in which heuristics that transform a problem into a simpler one, can be validly applied.

The classes that can be learned by the system are essentially determined by the kinds of special characteristics that it can detect in a solution and by the information available in the background knowledge. If the system is not capable of detecting some interesting characteristics of a solution then it will not even start a learning episode. On the other hand, once a learning episode is triggered, the corresponding class will be learned only if the system has the appropriate predicates in the language to express it. (The lack of a predicate will be detected by the system during the testing phase, because there will be negative variations that will not be rejected). Also, the exploration capabilities of the system are determined by the functions and heuristics that it uses to generate variations. The inability to generate certain kinds of variations may lead to learning less general descriptions (i.e., subclasses of problems) or, even, incorrect descriptions. A detailed discussion of this point, however, is beyond the scope of this paper.

In order to apply the learning method to another domain it will be necessary to replace the background knowledge that the system currently has by the background knowledge appropriate for the new domain. This can be done without difficulty. There are certain characteristics of the current domain, however, that are more deeply embedded in the system, so that it may be necessary to make some non-trivial changes to it. An example of this is the order in which different aspects of a problem are examined. More importantly, in some domains it may be difficult or even impossible to have the system classify the variations by itself. In those cases a human instructor will have to perform that task. Also, the "well formedness" of the variations generated by the system may become an issue. By the nature of the problems in the physics domain it was very easy to generate "legal" problems, but this may not be the case for other domains so that complex rules of formation may have to be introduced.

When comparing the method proposed here with inductive methods in which sets of positive and negative variations are given by an instructor, certain advantages and disadvantages can be discerned. The main advantage is that a system implementing this method is more autonomous than other systems because instead of being a passive receptor of instances it is an active explorer of the domain, and so is less dependent on an instructor. Disadvantages are that, i) as the system assumes more of the burdens of the learning process its complexity is increased, and ii) the method may not be applicable to some domains, as was indicated above. A detailed comparison between the two kinds of approaches constitutes an important topic for research, because it may very well be that the advantages of the new method far outweigh its disadvantages, at least for certain domains.

ACKNOWLEDGMENTS

I would especially like to thank Gordon Novak for many discussions of the ideas presented here. I would also like to thank Julian Gevirtz for his comments and his assistance in editing earlier drafts of this paper.

REFERENCES

1. Chi,M.T., Feltovich,P.J. and Glaser,R. "Representation of Physics Knowledge by Experts and Novices" Tech.Rep. 2. Learning Research and Development Center, University of Pittsburgh, 1980.

2. Dietterich,T. and Michalsky,R. "Learning and Generalization of Characteristic Descriptions: Evaluation Criteria and Comparative Review of Selected Methods" In Proc. IJCAI-79, Tokyo, Japan, August 1979.

3. Kibler,D. and Porter,B. "Perturbation: A Means for Guiding Generalization" In Proc. IJCAI-83, Karlsruhe, Germany, August 1983.

4. Langley,P., Bradshaw,G.L. and Simon,H.A. "Rediscovering Chemistry with the BACON system" In Machine Learning, Michalsky, Carbonel, Mitchell (Eds), Tioga Publishing Company, 1983.

5. Larkin,J.L, McDermott,J., Simon,D.P. and Simon,H.A. "Models of Competence in Solving Physics Problems". Tech.Rep. CIP 408, Dept.of Psychology, Carnegie-Mellon University, 1979.

6. Lenat,D.B. "The Role of Heuristics in Learning by Discovery: Three Case Studies" In Machine Learning, Michalsky, Carbonell, Mitchell (Eds), Tioga Publishing Company, 1983.

7. Michalsky,R.S. "A Theory and Methodology of Inductive Learning" Artificial Intelligence 20:2 (1983) 111-161.

8. Mitchell,T.M. "Generalization as Search" Artificial Intelligence 18:2 (1982) 203-226.

9. Mitchell,T.M., Utgoff,P.E., Nudel,B. and Banerji,R. "Learning Problem Solving Heuristics through Practice" In Proc. IJCAI-81, Vancouver, August 1981.

10. Novak,G.S. "Computer Understanding of Physics Problems Stated in Natural Language" American Journal of Computational Linguistics, Microfiche 53, 1976.

11. Schank,R.C. "The Current State of AI: One Man's Opinion" AI Magazine 4:1 (1983) 3-8.

12. Vere,S.A. "Induction of Relational Productions in the Presence of Background Information" In Proc. IJCAI-77, Cambridge,Mass., August 1977.