A SYSTEM OF PROGRAMS FOR COMPUTER-AIDED INDUCTION: A SUMMARY

by

Ryszard S. Michalski

Proceedings of the Fifth International Joint Conference on Artificial Intelligence,
MIT, Cambridge, Massachusetts, pp. 319-320, August 22-25, 1977.

L/ Michalsk.

IJCAI-77

5TH INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE -1977

IJCAI-77 • PROCEEDINGS OF THE CONFERENCE

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CAMBRIDGE, MASSACHUSETTS, USA
AUGUST 22 - 25, 1977

VOLUME ONE

Sponsored by the
International Joint Conferences on Artificial Intelligence

A SYSTEM OF PROGRAMS FOR COMPUTER-AIDED INDUCTION: A SUMMARY

Ryszard S. Michalski
Department of Computer Science
University of Illinois
Urbana, IL 61801

Introduction

Inductive inference processes are not only fundamental for scientific research and discovery, but also are crucial for our everyday activities, in which we constantly have to determine relevant information in the enormous flow of information coming to us, or to make decisions based on insufficient or partial information. Despite many attempts to formalize induction, not much has yet been achieved in this direction, and the research on induction is often treated with skepticism.

One way to break a stalemate is to work on induction tailored toward solving specific problems with a hope that it will bring an insight to induction in general (e.g., work by Simon and Lea 73, Winston 70, Buchanan, Feigenbaum and Sridharan 72). Another way is to identify certain conceptually simple but basic inductive tasks, common to many inductive processes, and to find the most efficient and general solution for them (at the final stage implemented as a computer program). Having easily available programs for solving such well defined inductive tasks, a skilful user could use them as tools for attacking complex inductive problems in different practical areas.

For example, such programs could aid specialists in various applied sciences in formulating hypotheses explaining experimental data, suggesting alternative hypotheses, detecting regularities in data, discovering the simplest conditions for certain actions. They could be useful for knowledge acquisition from examples in rule-based expert systems, in research on speech and image understanding systems. Also, they could serve as building blocks for solving more complex inductive tasks. An example of the latter possibility is the fact (which was unexpectedly discovered) that inducing concepts in VL_{21} logic system (a form of the first order predicate logic with additional operators; Larson and Michalski 77a) requires at some stage a solution of a simpler problem of inducing concepts in variable-valued propositional calculus (VL_1 system; Michalski 75a) and, consequently, an already existing program was found directly applicable.

This tool building approach to induction has been followed by the author and his collaborators at the University of Illinois for the last few years. We have developed a theoretical basis, algorithms and computer programs for solving a class of inductive tasks, which we found often occuring in various practical inductive problems. The algorithms employ a number of novel ideas to combat the combinatorial complexity of the tasks. They are described in references to each program. This report gives only a brief review of the tasks we worked on and the developed programs, so that a potential user or an interested reader can be aware of their existance and availability. Some of the programs achieved a high level of maturity (e.g., AQ7, AQ11); some of them represent just an initial state. All programs are documented and can be obtained on request.

Tasks and Programs

Below is given a summary of tasks considered with a reference to programs providing a solution for them. (A more detailed review is given in Michalski 77).

1. *Determination of a relevant descriptor set.* In solving inductive problems, the first step is often a determination of a relevant set of descriptors (variables, predicates, functions, relations) which are used in the original problem description. The descriptors should be simple to measure or recognize, easy to interpret and as relevant as possible. A special case of this problem is when there is a large number of potentially useful variables (unary descriptors), and the problem is to select a small but sufficient subset consisting of the most relevant ones. Standard feature selection techniques developed in pattern recognition are often not satisfactory, because they select variables based on a measure of usefulness of each variable taken separately. We have developed an algorithm for an interactive determination of a 'best' subset of variables, taken as a group rather than separately. The developed algorithm was implemented in program AQPLUS (Forsburg 75).

2. *Determination of an optimal ('simplest') discriminant description of a collection of object classes, where objects are described by property lists (VL_1 level) or by an expression equivalent to a formula in the first order predicate logic (VL_2 level).* A discriminant description is a description which contains only necessary descriptors, in a proper relation, which distinquish a given class from all other classes. Programs AQ11 (Larson 76, Larson and Michalski 77), AQ7 (Larson and Michalski 75. Michalski 75b), AQ9 (Cuneo 75) solve various forms of this problem within the framework of the VL_1 system (Michalski 75a) and program INDUCE-1 within the framework of the VL_{21} system (Larson and Michalski 77, Larson 77).

3. *Determination of a characteristic description of a class of objects (with or without negative examples), where objects are described similarly as in 2.* The difference between this task and the previous one is that here the goal is to determine a complete description of a given class of objects, which includes all common characteristics of objects in each class. A specific subtask here is to determine the least general conjunctive description of the class, whose objects are described by a property list or by a product of predicates (or some equivalent form). Winston (Winston 70) considered this task from the viewpoint of building a class description in a form of a labeled graph, Hayes-Roth (Hayes-Roth 76) within a formalism equivalent to a quantifier-free first order predicate logic.

Program AQ7UNI (Stepp 76) provides a solution to this task within a framework of VL_1 system (involving both conjunctive and disjunctive descriptions), and program INDUCE-1 within the framework of VL_{21} (Larson and Michalski 77a). Among novel ideas implemented in INDUCE-1 is an ability to develop new descriptors built as functions of the original ones, to involve different schemas of generalization dependent on the type of descriptors (nominal, interval or structured), to provide either descriptive or descriminant descriptions by changing a description optimality criterion, and an ability to accept user's knowledge about restrictions and relations characterizing descriptors.

4. Determination of the most representative learning examples. This task occurs when one has access to a large number of examples of each decision class, and the problem is how to select from the set the examples which are 'most instructive' to a description learning program. Program ESEL ('event selection') provides a solution to this problem (Larson and Michalski 77b).

Concluding Remarks

We have done experiments on applying programs to determine discriminant descriptions of liver diseases (Michalski 73, Larson 76), to determining filters for non-uniform texture discrimination (Michalski 73), to formulate rules for win and draw in King-Pawn versus King chess end games (Michalski, Negri 76), to determine discriminant rules describing soybean diseases. The results from testing the derived soybean disease diagnostic rules gave more than 99% of correct diagnosis on 365 test cases of soybean diseases (Chilausky, Jacobsen, Michalski 76, 77). It may be interesting to remark that expert-derived rules applied to the same data gave only 83% correct decisions. This may be due to an inadequate representation of the expert's knowledge (in the form of decision rules) or the way it was applied (an evaluation schema for the rules), or some other factors. This surprising result is a question to be explored in the future.

The above experiments have convinced us that the inductive programs of the kind we have been working on can find many useful applications.

The reported work is still at an early stage, and there is much room for further advancement in this direction.

Acknowledgments

The research described in this paper was supported in part by the National Science Foundation, Grant NSF MCS 74-03514. Author thanks Gary Kampen and James Larson for proofreading and the comments.

References

Buchanan, B. G., Feigenbaum, E. B. and Sridharan, N. S. Heuristic theory formation: data interpretation and rule formation. In Machine Intelligence 7, edt. B. Meltzer and D. Michie, John Wiley and Sons, 1972.

Chilausky, R., Jacobsen, B., and Michalski, R. S. An application of variable-valued logic to inductive learning of plant disease diagnostic rules, Proceedings of the Sixth Annual International Symposium on Multiple-Valued Logic, Utah State University, Logan, 1976.

Chilausky, R., Jacobsen, B., and Michaslki, R. S. A computer consultant for plant disease diagnosis: a pilot study involving soybean diseases (to be published in Phytopathology), 1977.

Cuneo, R. Selected problems of minimization of variable-valued logic formulas, Master's thesis, Department of Computer Science, University of Illinois, Urbana, 1975.

Forsburg, S. A user's guide for AQPLUS, an internal report, Department of Computer Science, University of Illinois, Urbana, 1975.

Hayes-Roth, F. Uniform representations of structured patterns and an algorithm for the induction of contingency-response rules, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, 1976.

Larson, J. A multi-step formation of variable-valued logic hypotheses, Sixth International Symposium on Multiple-Valued Logic, Logan, Utah, 1976.

Larson, J. A user's guide for INDUCE-1, Report No. 876. Department of Computer Science, University of Illinois, Urbana, 1977.

Larson, J. and Michalski, R. S. Inductive inference of VL decision rules, Workshop on Pattern Directed Inference Systems, Hawaii, 1977a.

Larson, J. and Michalski, R. S. ESEL and AQ11: programs for VL_1 event selection, interactive generation of VL_1 hypotheses and hypothese evaluation, Report No. 867, Department of Computer Science,University of Illinois,Urbana 1977.

Michalski, R. S., AQVAL/1--A computer implementation of a variable-valued logic system and its applications to pattern recognition, Proc. 1st I.J. Conf. on Pattern Recognition, Wash. 1973.

Michalski, R. S. Variable-valued logic and its applications to pattern recognition and machine learning. Chapter in Multiple Valued Logic and Computer Science, edt. D. Rine, North-Holland Publishers, 1975a.

Michalski, R. S. Synthesis of optimal and quasi-optimal variable-valued logic formulas, 5th International Symposium on Multiple-Valued Logic, Bloomington, Indiana, 1975b.

Michalski, R. S. Toward computer-aided induction, Report No. 874, Department of Computer Science, University of Illinois, Urbana, 1977.

Michalski, R. S. and Negri, P. An experiment on inductive learning in chess end games; the king-pawn-king case. Chapter in Machine Representation of Knowledge, edt. D. Michie and T. Elcock, NATO Advanced Study Institute, 1975.

Simon, H. A. and Lea, G. Problem solving and rule induction: a unified way, Carnegie-Mellon Complex Information Processing Working Paper 227, Pittsburgh, 1973.

Stepp, R. An investigation of a uniclass inductive program: AQ7UNI, an internal report, Department of Computer Science, University of Illinois, Urbana, 1976.

Winston, P.M., Learning structural descriptions from examples, Tech. rep. AI TR-231, MIT AI Lab., Cambridge, Mass., 1970.