Expert Systems and Fuzzy Systems

Constantin Virgil Negoita

Expert Systems and Fuzzy Systems

Constantin Virgil Negoita

Hunter College
City University of New York

The Benjamin/Cummings Publishing Company, Inc.
Menlo Park, California • Reading, Massachusetts
London • Amsterdam • Don Mills, Ontario • Sydney

EC81/23

Sponsoring Editor: Alan Apt
Production Editor: Larry Olsen
Copy Editor: Antonio Padial
Text Designer: Nancy Benedict
Cover Designer: Gary Head

Library of Congress Cataloging in Publication Data

Negoiţă, C. V. (Constantin Virgil)
Expert systems and fuzzy systems.

Bibliography: p.
Includes index.
1. Expert systems (Computer science) 2. Fuzzy systems. I. Title.
QA76.9.E96N44 1984 001.53'5 84-12324
ISBN 0-8053-6840-X

BCDEFGHIJ-HA-8987654

The Benjamin/Cummings Publishing Company, Inc.
2727 Sand Hill Road
Menlo Park, California 94025

Preface

How machines can be made smarter is a question that has confronted the computing field since its inception. Mere repetition of steps—for instance, adding numbers, inverting matrices, or even solving equations—presupposes hard-coded instructions. Any traditional computer program directs the machine to access data, but decisions about how to process those data are invariably hard-coded in the language of the program and stored in the memory during program execution. These decisions are made by the human programmer, who has the knowledge to make them.

Knowledge engineering is a discipline devoted to integrating human knowledge in computer systems. The distinctive characteristic of any knowledge-based system is that its processes are state-driven rather than hard-coded. Decisions about how to process data are part of the knowledge of the system. In other words, an intelligent system writes its program. Knowledge is procedural, in the sense that it tells how the data concerning a problem can be manipulated to solve it. By internalizing procedural knowledge as a model of the world, the machine becomes intelligent.

An expert system is an information system that can pose and answer questions relating to information borrowed from human experts and stored in the system's knowledge base. The fact that answers are automatically extracted from the data descriptions, by a user-invisible inference procedure, results in a great degree of data independence. Not only can users represent data in a high-level, human-oriented manner, but they are also spared the effort of describing the operations used to retrieve those data.

Because the knowledge base in an expert system is put there by human experts, and because much human knowledge is vague, it is usually true that facts and rules are neither totally certain nor totally consistent. For this reason, a basic issue in the design of expert systems is how to equip them with a computational capability for evidence transmission.

To solve this problem, researchers have augmented the inference procedures with mechanisms that combine evidence degrees according to the rules of plausible reasoning. *Plausible reasoning* is simply drawing conclusions from facts that seem to be correct. In most systems, this mechanism is purely heuristic. Recently, however, some investigators have tried to make that mechanism mathematically sound.

A promising approach is based on the theory of fuzzy sets. In this case, we speak about *approximate reasoning*, which means drawing conclusions by taking the consistency of the facts into account. The treatment of fuzziness is a critical issue in knowledge representation. To say that a word is fuzzy is to say that sometimes there is no definite answer as to whether or not the word applies to something. The indeterminacy is due to an aspect of the meaning of the word rather than to the state of our knowledge. In all expert systems based on symbolic manipulation and plausible reasoning, uncertainty is supposed to reside in the state of our knowledge. In expert systems based on semantic manipulation and approximate reasoning, the emphasis is on fuzziness viewed as an intrinsic property of natural language.

An evident advantage of the fuzzy set approach is the possibility of representing numeric and linguistic variables in a uniform way and of using a sound formalism to handle them.

If we represent facts as objects and rules as morphisms, the mathematical theory of categories is a good language for describing the mechanisms of evidence combination. The difference between plausible and approximate reasoning becomes the difference between the categories on which we model the facts. In this way, an algebra of knowledge becomes available to the system designer, and knowledge diagrams become models of both production systems or declarative systems used in logic programming.

In this book, I attempt to bring together the fundamentals of approximate reasoning and to illustrate the concepts with examples wherever possible. I have tried to present the role of fuzzy systems in knowledge engineering so that it is accessible to a wide audience, including those interested in the philosophy and logic of such systems as well as those interested in their design and application. For this reason, this book is introductory, and certain simplifications have been made to ease an understanding of the most important features of knowledge-based computing. Because in real computing fuzzy sets are tables, I preferred to present them as such. For each chapter, extensive annotated Readings have been included. Although this is a textbook, it does not include sample problems to solve. The annotated Readings pose sufficient real problems, and it seemed inappropriate to add contrived ones. The reader is warned that real expert systems are tailor-made.

This attempt to provide the new or experienced knowledge engineer with a survey of the most important material on the subject necessarily leaves some gaps I have restricted myself somewhat to management-oriented applications, in which facts and rules are ill-defined and only a semantic approach seems to succeed. Perhaps the strongest implication of this approach is that it casts knowledge engineers in a new role. The problem-solving power of an expert system based on plausible reasoning and symbolic manipulation is primarily a function of the domain-specific information in the knowledge base and is only secondarily a function of the system's inference method. The problem-solving power of an expert system based on approximate reasoning and semantic manipulation is primarily a function of the inference method it employs. No longer are knowledge engineers merely intermediaries between the human expert and the developing knowledge base. In their new role, they are independent of both.

This observation deserves an explanation. Knowledge acquisition has been a long-standing bottleneck in artificial intelligence. Certainly, the most powerful knowledge systems are those that contain the most knowledge. Symbolic systems deal with description assertion and encoding of decision rules. Emphasis is on recursion and list structures, which can be treated by procedural languages.

Knowledge systems based on approximate reasoning are oriented less toward list structures and more toward logic programming. In this case, programming style bears little resemblance to the style of procedural languages, such as PL/1, Pascal, or even LISP. A conventional flowchart is of no help to those writing a logic program. The programmer must concentrate instead on the meaning of what is to be achieved, and he must express that meaning in a declarative style.

A major issue in this approach is the use of natural language and syntax. Making software user friendly will become progressively more critical in the next decade, as more powerful machines become available to a wider range of individuals than ever before. Many computer users will have little computer science training or inclination to get the training. They will want to use natural language in any dialogue with any computer. Only the fuzzy system approach makes such communication possible.

According to this book, a semantic system is software that uses fuzzy set technology to translate the meaning of a vocabulary. Once the knowledge engineer has developed the semantic system, the user can exploit it without any interface. With a semantic system, the user can encode knowledge in many forms: production rules, production systems, and verbal models. A verbal model can be viewed as a production system with mathematical operators. This book emphasizes the advantage of this approach to a semantic system: production systems can be treated as decision tables, and dynamic models can be treated with linguistic variables.

This is the kind of expertise used in management, and I hope that readers will see how knowledge systems can become a practical reality in decision making. If knowledge systems are developed responsibly, the result will be a significant improvement in both human systems management and human management systems.

C. V. Negoita

Contents

CHAPTER SEVEN

APPENDIX

CHAPTER ONE

Introduction

EXPERT COMPUTER SYSTEMS

When historians rank the technology of the present century, expert systems will almost certainly be high on the list. Expert systems—software systems that mimic the deductive or inductive reasoning of a human expert—belong to the family of information-intensive machines. For a task to qualify for knowledge engineering, there must be at least one acknowledged human expert.

The primary use of expert systems thus far has been in capital-intensive areas, such as oil drilling or exploration, where human experts are scarce and the cost of equipment lying idle is so high that the price of an expert system can be recaptured rapidly. Expert systems gained early success in medicine because a great deal of effort was spent writing down the best-known ways to solve problems. Indeed, some expert systems diagnose disease as well as the average medical practitioner does. Similarly, finance and accounting early lent themselves to expert systems, because knowledge in those areas is concrete and can be incorporated into a knowledge base relatively easily. In many fields, however, expertise is ill-defined and can be represented in an expert system only by special means.

The coming decade will to be devoted to magnifying human mental power by changing the basic design of computers so that they can carry on intelligent, natural intercourse with humans.

People speak "natural" languages, such as English or French. Computers speak "unnatural" languages, such as Pascal, FORTRAN, or BASIC. Large-scale expert systems depend on natural language as front ends to knowledge bases.

Production Systems

Expert systems require two things: a collection of facts and rules about a given field and a way of making inferences from those facts and rules. Any rule is a pattern-invoked program. Such a program is not called by other programs in the ordinary way but is instead activated whenever certain conditions hold in the data.

One pattern-invoked program of particular interest is the production rule

condition IMPLIES action

The condition is usually one or more predicates that test properties of the current state of facts. The action in turn changes the current state of the facts. Rules are frequently formulated in natural language, whose precision or vagueness reflects the human expert's knowledge. For instance, the production rule

price is low IMPLIES profit should be below normal

can be used individually or in a production system.

A production system consists primarily of a set of condition-action rules and operates in cycles. During each cycle, the conditions of each production rule are matched against the current state of facts. When rules and conditions match, actions are taken. Those actions affect the current state of facts, making new production rules match.

A promising feature of production systems is their modularity. Because each production rule is relatively independent of every other production rule, we could, in principle, construct modular systems. Production systems differ substantially from conventional computer programs because their tasks have no algorithmic solutions and because such systems must use incomplete information to make decisions.

Logic Programming

In any production rule, knowledge is procedural in the sense that it tells how the data for a problem can be manipulated to solve a problem. Procedural knowledge can also be represented by logic programming. In logic programming, one can express knowledge as either facts or rules. The basic building units of both facts and rules are predications, that is, expressions that say simple things about the individuals in a universe. For instance, the piece of knowledge "Peter likes Sarah" can be represented as

LIKES (Peter, Sarah, *m*)

Predications are represented by a predicate name followed by a list of arguments. An argument can be the name of an individual or a measure of degree.

Rules, by contrast, have the general form

$$P_1 \text{ if } (P_2 \text{ and } P_3 \text{ and } \ldots \text{ and } P_n)$$

where P_i stands for predications. For example, the general rule that Peter and Sarah are friends can be stated as

FRIENDS (Peter, Sarah) if LIKES (Peter, Sarah) and LIKES (Sarah, Peter)

If all the conditions hold, then the conclusion holds.

Once a set of facts and rules has been defined, one can deduce information from those facts and rules. This deduction is done by writing a query, an expression of form

$$P_2 \text{ and } P_3 \text{ and } \ldots \text{ and } P_n?$$

For instance, given the world description

(1) LIKES (x, y)
(2) TALL (x)
(3) PERSON (x, y, z)

one can write the query

WHO LIKES A TALL PERSON?

Inference takes place automatically. The fact that answers are automatically extracted from the data descriptions by a user-invisible procedure results in a great degree of data independence. Not only can users represent data in a high-level, human-oriented manner (rather than in terms of bits, arrays, etc.), but they are also free from the effort of describing the operations used to retrieve the data. These operations are implicit in the inference mechanisms, which give an operational meaning to the purely descriptive fact and rule used.

Intelligence Means Internalization

Interest in the foundations of expert systems was sparked by difficulties encountered with evidence combination. Efforts to understand, formulate, and resolve the problems of inference led to some questions about the definition of artificial intelligence. Can a machine be intelligent? After all, what is intelligence? According to the expert system theorists, intelligence means the internalization of a model of the external milieu. The computer culture defined intelligence as an intentional movement.

Up to that point, intelligence had been a concern of the soft sciences. When it became a concern of the hard sciences, ontology entered the modern world. Some modern philosophers, speaking about being, define it as fulfilled in an intentional movement. They say that being possesses no more than a potentiality for its own realization. Thus, contemporary ontologists favor dynamic potentiality as the definition of being.*

Once nature is directed toward its proper end, say the philosophers, it finds fixity in an opening. A closure made for an opening is, for example, the closure of rules and its opening in language. This is the case for every external milieu that can be internalized. According to the same philosophers, to grow in being means to transform external milieux into internal ones. For humans, being is evidently the striving for and possibility of reintegration in another modality. Culture, for instance, which at the beginning represents a perfect external milieu, becomes at the end an internal milieu for those fulfilled in it. In fact, spiritual life is this movement from the external to the internal. The topic does not belong to modern ontology alone. In a much older formulation, it belongs to those theologians who said that, after the Fall, Grace acts on man from the outside.

Systems theory can formalize this perspective, and cybernetics can project it on organized reality. Cybernetics is the science of communications in and control of machines and animals. If one applies the principles of cybernetics to labor-saving machines and methods, the result is automation. A relatively long time ago, automation, with its emphasis on regulators, brought the internal model principle to the fore. According to this principle, a control system needs feedback and must incorporate in the feedback loop a suitable model of the dynamic structure of the exogenous variables. In plain English, before acting, one has to know the environment. Again, the conclusion is very old: rational means known beforehand.

Philosophers and cyberneticians are in total agreement when they link intelligence with knowledge. Lately, mathematicians have come to the same conclusion. Studying abstract structures and trying to generalize the category of sets, they focused on an intelligent structure and called it *topos* (place) because it internalized its logic. The expert system approach is a step in this direction. When knowledge is internalized, the machine becomes "beinglike" or "rationallike."

* C. Noica, "Becoming into Being," *Ed.S.E.Buc.*, 1981 (reviewed in *Kybernetes* 11 (1982): 147).

Pullback Versus Feedback

Both control theory and knowledge engineering are dominated by the internal model principle, but the difference between them is substantial. In control theory, the intelligent regulator is external to the regulated system. In knowledge engineering, the self-regulated system is intelligent. An intelligent machine is free, if freedom is defined as the possibility to act according to internalized goals. Any artificial intelligence approach, viewed from the internalization perspective, builds a disequilibrium of sequential changes, now and then exhibiting quasi-equilibria on the way.

The fundamental question is "What laws govern rationality?" Any answer emphasizes the realization of a goal. In conventional programming, the program is a series of steps controlling the machine and minimizing the distance between an actual state and a desired one. The desired state is a reference, outside the machine. We use the term *feedback* to mean that the system is controlled by the margin of error with reference to an external goal. In expert systems, the desired goal is internalized in the knowledge base, and the machine achieves the goal by resorting to (pulling back in) the structure of facts. *Pullback* means a movement governed by an internal goal. We say that the behavior of the system is controlled by pullback.

The principle of pullback is a direct consequence of any knowledge representation based on logic, whether that logic is two-valued or multivalued. Pullback is best explained when the knowledge base is modeled as a category. One goal of this book is to discuss this problem in more detail.

The Computer Revolution

Some specialists question that expert systems represent a quantum leap from conventional computer systems. They argue that expert systems represent an evolutionary rather than a revolutionary development. Computers and information systems are well-structured environments for accomplishing a task. Expert systems do something similar: they are a record of accumulated experience. The programs lead users in a logical way through alternatives that, without computers, they would have to recognize themselves. Combining in one computer program the know-how to solve a problem and accumulated data and experience is a major step forward, but no new knowledge is being created. Know-how is not knowledge, and training is not education, no matter how important know-how and training are.

From a social perspective, expert systems have much in common with mathematical models, and the credibility of both has often been questioned. If modeling computers are used by unsophisticated technical workers, intellectual insight is gained at the expense of intuition. Expert systems allow organizations to place unsophisticated staff in key analytical positions. Working knowledge does not come easily from an algorithm that is used uncritically under varying conditions. It is vital to understand the roles computing is coming to play. It is also vital not to overstate the contributions of otherwise interesting technologies. Improved literacy does not depend on better computers, and computer-based systems do not always yield better decisions. In some places, people have no access to an average practitioner, and some specialists argue that an artificial expert is better than none at all. A sure fact is that work on languages and systems for knowledge representation will significantly reduce the effort required to develop intelligent systems. These systems, in turn, will help users to focus more on the problem itself than on the implementation of the underlying program.

Expert systems evolved from machines dedicated to numeric computation. What is new is that they assess the meaning of information and understand the problem to be solved. Intelligent programming software will allow machines to take over the burden of programming, and this is a revolution.

THE FIELD AND THE BOOK

An expert system is a machine that makes inferences from internalized facts and rules. The facts and the rules are chunks of knowledge or statements about the external world. Yet, it is a well-known fact that any observer's ability to make precise but significantly certain statements about complex external worlds decreases as their complexity increases. Precision and certainty seem to be incompatible.

In the philosophy of science, this fact has been known for a long time. In the sixth century, Leontius from Byzantium observed: "Our impression of the world is general but vague, not revealing the truth; and if we attempt to particularize by division into genera and species and individuals, the general view is lost: we are heading not towards the truth but towards an infinite regress.* Fourteen centuries later, Pierre Duhem distinguished between practical facts expressed in vague, qualitative, ordinary language and theoretical facts expressed in pre-

* A. Armstrong, *The Cambridge History of Later Greek and Early Medieval Philosophy,* (Cambridge University Press, 1970) p. 490.

cise, quantitative language.* Duhem argued that confidence in the truth of a vague assertion may be justified just because of its vagueness, which makes the assertion compatible with a whole range of observed facts. The laws of physics acquire detailed precision at the expense of the fixed and absolute certainty of common-sense laws. According to Duhem, there is a balance between precision and certainty; one is increased only to the detriment of the other.

This principle explains the considerable intellectual investment required to approximate reasoning. The key idea is the representation of a fact as an evaluation and of a rule as a transformation of evaluations. Such an evaluation is a function, as is the fuzzy set. The exact relationship of a fuzzy set to an ordinary set is best perceived by recalling the definition of the characteristic function of a set. The characteristic function of an ordinary set has this form:

$$U \rightarrow \{0,1\}$$

This set maps the universe U to a set of two elements. This is a binary choice between being in or out of the set.

A fuzzy set is a function with more than two values, usually with values in the unit interval

$$U \rightarrow [0,1]$$

This function allows a continuum of possible choices. Such a function can be used to describe imprecise terms. For example, the term *old* can be defined according to the universe of human ages. Clearly, someone over seventy is old, so the degree of membership of an age seventy or greater is 1.0. It is not as certain that a sixty-year-old is old. Rather than saying that a sixty-year-old is old or not, one can say that that individual is partially old. We could evaluate the degree of oldness at age sixty as 0.7. In this way, the vagueness of the term *old* can be captured mathematically and dealt with in an algorithmic fashion.

As in ordinary set theory, the characteristic function of fuzzy sets links fuzzy set theory with logic. The degree of membership corresponds to a truth value of the statement "is a member of," which is equivalent to "is partially defined as." This correspondence has a profound implication for artificial reasoning.

Reasoning means drawing conclusions from facts. When the facts are represented as setlike objects, the meaning of the logical operators AND, OR is precisely defined by the category of these objects. The logic

* Pierre Duhem, *La théorie physique: Son objet et sa structure* (Paris: Chevalier & Rivière, 1906).