
EXPERT SYSTEM FOR DECISION-MAKING PROBLEM IN ECONOMICS

Aygun Alasgarova, Leyla Muradkhanli

Abstract: *An expert system (ES) is a class of computer programs developed by researchers in artificial intelligence. In essence, they are programs made up of a set of rules that analyze information about a specific class of problems, as well as provide analysis of the problems, and, depending upon their design, recommend a course of user action in order to implement corrections. ES are computerized tools designed to enhance the quality and availability of knowledge required by decision makers in a wide range of industries. Decision-making is important for the financial institutions involved due to the high level of risk associated with wrong decisions. The process of making decision is complex and unstructured. The existing models for decision-making do not capture the learned knowledge well enough. In this study, we analyze the beneficial aspects of using ES for decision-making process.*

Keywords: *expert system, decision-making processing*

1. Introduction

The purpose of this paper is to develop an expert system as an interactive computer program that helps a user solve a problem and make a right decision. Decision making process is inherently complex due to the various forms of risks involved. The existing models for decision-making do not capture the learned knowledge well enough [1-3,5]. In this paper, we analyze the beneficial aspects of using ES for decision-making process and organized as follows: Section 2 provides types of problems in economics solved by ES. Section 3 includes the comparison analysis. Inference rule and confidence are given in Section 4. Section 5 concludes the paper.

2. Types of Problems in economics Solved by ES

Typically, the problems to be solved are of the sort that would normally be tackled by a human "expert" – an economical or other professional, in most cases. Real experts in the problem domain are asked to provide "rules of thumb" on how they evaluate the problems, either explicitly with the aid of experienced system developers, or sometimes implicitly, by getting such experts to evaluate test cases and using computer programs to examine the test data and derive rules from that. Generally expert systems are used for problems for which there is no single "correct" solution which can be encoded in a conventional algorithm — one would not write an expert system to find shortest paths through graphs, or sort data, as there are simply easier ways to do these tasks. Simple systems use simple true/false logic to evaluate data, but more sophisticated systems are capable of performing at least some evaluation taking into account real-world uncertainties, using such methods as fuzzy logic. Such sophistication is difficult to develop and still highly imperfect.

3. ES in Comparison with traditional Problem-solving Systems

The principal distinction between expert systems and traditional problem solving programs is the way in which the problem related expertise is coded. In traditional applications, problem expertise is encoded in both program and data structures. In the expert system approach all of the problem related expertise is encoded in data structures only; none is in programs. Several benefits immediately follow from this organization. An example may help contrast the traditional problem solving program with the expert system approach. The example is the problem of tax advice. In the traditional approach data structures describe the taxpayer and tax tables, and a program in which there are statements representing an expert tax consultant's knowledge, such as statements, which relate information about the taxpayer to tax table choices. It is this representation of the tax expert's knowledge that is difficult for the tax expert to understand or modify. In the expert system approach, the information about taxpayers and tax computations is again found in data structures, but now the knowledge describing the relationships between them is encoded in data structures as well. The programs of an expert system are independent of the problem domain and serve to process the data structures without regard to the nature of the problem area they describe.

An expert system has four main architectural components that are the knowledge base, the inference engine, the knowledge acquisition module, and the user interface for input/output (Figure 1). The knowledge base contains the domain specific knowledge that is used for problem solving in the domain. Knowledge can be represented and stored in the knowledge base with a format suitable for computer manipulation. One of the most commonly used ways to represent knowledge is as rules (in the form of If-then) [2].

The inference engine is based on an inference rule and a search strategy and contains algorithms. Algorithms are used to manipulate the knowledge stored in the knowledge base in order to solve problems. An inference rule is a way to deduce new knowledge from the existing knowledge base of rules and facts. The knowledge acquisition module enables experts to store their knowledge in the knowledge base or expert system to deduce new knowledge from existing knowledge through a machine learning process. The interface for input/output is used for the expert system to interact with the user, the environment and other systems such as databases, and spreadsheets.

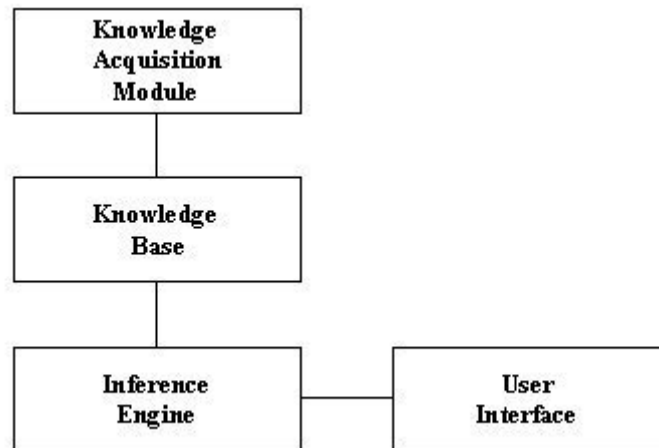


Figure 1. The Architecture of an Expert System

There are generally three individuals having an interaction with expert systems in organization. Primary among these is the end-user; the individual who uses the system for its problem solving assistance. In the building and maintenance of the system there are two other roles: the problem domain expert who builds the knowledge base, and a knowledge engineer who assists the experts in determining the representation of their knowledge and who defines the inference technique required to obtain useful problem solving activity. The end-user usually sees an expert system through an interactive dialog. Knowledge engineers are concerned with the representation chosen for the expert's knowledge declarations and with the inference engine used to process that knowledge.

4. The Inference rule and Confidence

An understanding of the "inference rule" concept is important to understand expert systems. An inference rule is a statement that has two parts, an if-clause and a then-clause.

An expert system's rulebase is made up of many such inference rules. They are entered as separate rules and it is the inference engine that uses them together to draw conclusions. Because each rule is a unit, rules may be deleted or added without affecting other rules (though it should affect which conclusions are reached). One advantage of inference rules over traditional programming is that inference rules use reasoning which more closely resemble human reasoning. Thus, when a conclusion is drawn, it is possible to understand how this conclusion was reached. Furthermore, because the expert system uses knowledge in a form similar to the expert, it may be easier to retrieve this information from the expert.

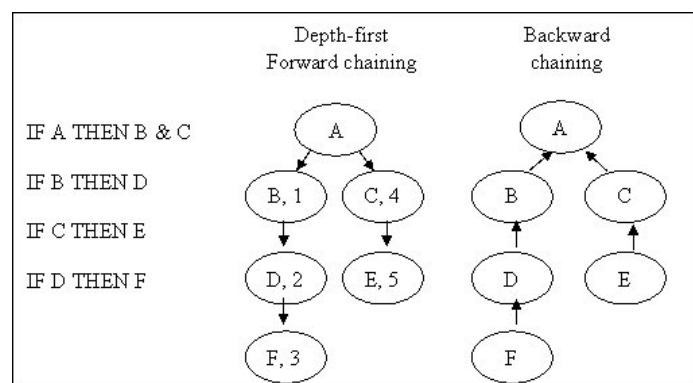


Figure 2. Complex rules

Actually there are two methods of reasoning when using inference rules: Forward chaining and backward chaining. Forward chaining starts with the available data and uses inference rules to extract more data until an

optimal goal is reached. An inference engine using forward chaining searches the inference rules until it finds one where the If clause is known to be true. When found it can conclude, or infer, the Then clause, resulting in the addition of new information to its dataset. Backward chaining starts with a list of goals (or a hypothesis) and works backwards to see if there are data available that will support any of these goals. An inference engine using backward chaining would search the inference rules until it finds one, which has a Then clause that matches a desired goal. If the If clause of that inference rule is not known to be true, then it is added to the list of goals (in order for your goal to be confirmed you must also provide data that confirms this new rule). The complex rules are demonstrated below in the Figure 2 [6]:

Another advantage of expert systems over traditional methods of programming is that they allow the use of confidences. When a human reasons he does not always conclude things with 100% confidence. The user might say, "If the rate is blue, then it is probably increasing". This type of reasoning can be imitated by using numeric values called Confidences. For example, if it is known that rate is blue, it might be concluded with 0.85 Confidence that it is increasing. These numbers are similar in nature to probabilities, but they are not the same. They are meant to imitate the Confidences humans use in reasoning rather than to follow the mathematical definitions used in calculating probabilities [4].

5. Conclusion

The following general points about ES:

A. The sequence of steps taken to reach a conclusion is dynamically synthesized with each new case. It is not explicitly programmed when the system is built.

B. ES can process multiple values for any problem parameter. This permits more than one line of reasoning to be pursued and the results of incomplete (not fully determined) reasoning to be presented.

C. Problem solving is accomplished by applying specific knowledge rather than specific technique. This is a key idea in expert systems technology. It reflects the belief that human experts do not process their knowledge differently from others, but they do possess different knowledge. With this philosophy, when one finds that their expert system does not produce the desired results, work begins to expand the knowledge base, not to re-program the procedures.

So, the ES, including knowledge base, production rule and inference engine, provide a rich and meaningful addition to the traditional methods. This approach is also most likely to be used in a real-world implementation of a decision support system. In this paper we have analyzed the beneficial aspects of using expert system for decision-making process. The kernel of the discussed system is Access DBMS and programming language Visual Basic.

Bibliography

- [1] G.J. Vachtsevanos and S.S. Kim, J.R. Echaz, V.K. Ramani "Neuro-fuzzy approaches to decision making: A comparative study with an application to check authorization". Journal of Intelligent and Fuzzy systems 6 (1998) 259-278.
- [2] Jinmu Choi "A Rule-Based Expert System Using an Interactive Question-and-Answer Sequence", Dept. of Geography University of Georgia.
- [3] K. Althoff, S. Wess, and M. Manago, "Reasoning with Cases-Theory and Practice", Proceedings of the European Conference on Artificial Intelligence, pp. 1-26. 1994
- [4] http://en.wikipedia.org/wiki/Expert_system
- [5] W. Siler, Birmingham, AL, USA. Building Fuzzy Expert systems. <http://members.aol.com/wsiler/>
- [6] L. Watson, Applying Case-Based Reasoning, Morgan Kaufmann Publishers, Inc. 1997

Authors' Information

Aygun N. Alasgarova - Khazar University, PhD student in Computer Engineering and Management; 11 Mehseti St., AZ1096 Baku, Azerbaijan, e-mail: a.alasgarova@mail.ru

Leyla G. Muradkhanli - Khazar University, Assoc. Professor. Sc. Techn.; 11 Mehseti St., AZ1096 Baku, Azerbaijan, e-mail: leyla@khazar.org