Trans. Nat. Acad. Science & Tech. (Phils.)
1988: 10: 55-61

A CONSULTATION SYSTEM FOR EXPERIMENTAL DESIGNS

Eliezer A. Albacea
Assistant Professor of Computer Science
Computer Science Division
Institute of Mathematical Sciences and Physics
University of the Philippines at Los Banos

ABSTRACT

Lack of experts in experimental designs and the high cost of consultation with available experts constrain, if not prevent, some researchers from conducting cost-effective and more efficient experiments. To minimize the problem, a consultation system was designed and developed to assist researchers planning to conduct experiments. This paper discusses the design of the components of such a system. These components are: the consultation component which asks questions, draws conclusions, and gives recommendations regarding the design and layout of the experiment; the analysis component which performs analysis of variance on the data gathered from the experiment; and the explanation component which describes and elaborates on the layout of the experiment recommended by the system and the results of the analysis of variance.

Introduction

In the last decade, interest in expert systems grew at a rapid rate. In the field of statistics, a new area has evolved due to the developments in expert system. This new area of statistics is often refered to as the statistical expert systems. Several systems have been developed under this area. These include REX (Gale and Pregibon, 1982; Pregibon and Gale, 1984), Muse (Dambroise and Massote, 1986), Express (Carlsen and Heuch, 1986), unnamed systems described in (Berzuini *et al.*, 1986) and (Darius, 1986), Student (Gale, 1987), an experimental expert system to assembling national balances (Barna, 1987), and Rochefort (Hilhorst *et al.*, 1987). However, Gale (1987) commented that the progress in most of these systems has been limited to methodological feasibility studies.

The purpose of this paper is to present the design and development of another experimental system, quite similar in objectives to the system cited earlier, but more specific to researchers who conduct experiments.

When a researcher carries out a scientific investigation, he has to pass through several stages. These stages are: problem definition, design, experiment, analysis, interpretation. Throughout these stages, a researcher need the advice of somebody who has enough experience and knowledge in conducting a scientific investigation.

However, in a scientific investigation that requires the conduct of an experiment, the lack of experts in experimental designs and the high cost of consultation with available experts constrain some researchers from conducting cost-effective and efficient experiments. One of the main objectives of this system, therefore, is to provide low-cost consultation to the researchers from the design to the interpretation stages.

In order to achieve this objective of the system, it was designed to be composed of three parts: consultation, analysis, and explanation. The consultation component asks the user a series of questions, and from the answers provided a recommendation is issued by the system. The analysis component provides procedures for analyzing the data gathered from the experiment. The explanation component elaborates further the results from the consultation and analysis components.

A prototype of the system was built using Pascal (Albacea and Caringal, 1987). In the current version, however, the consultation component is written in Prolog.

The Consultation Component

This component aids researchers in identifying the most suitable design of the experiment to be conducted. This is achieved in the system by asking the user questions, the answers of which lead to the recommended design.

The consultation component is divided into two parts, namely: the knowledge base and the shell. The shell is further divided into two modules called the inference engine and the user interface.

Knowledge Base

The knowledge base of the system is composed of if-then rules, often called production rules. This formalism for knowledge representation is by far the most popular (Bratko, 1986) and it turns out to be a natural form of expressing knowledge in experimental designs. A simplified syntax of these rules is given in Figure 1. The symbol "::=" is read as "is defined as " and ":" signifies disjunction.

```
<rule> ::= <premise><action>
<premise> ::= ($AND <condition> . . . <condition>)
<condition> ::= <predicate>
                : ($OR <condition> . . . <condition>)
<action> ::= <conclusion> : <instruction>
```

Figure 1. A simplified syntax of the rules in the system.

A rule in this representation is defined as a premise-action par. Premises are conjunctions of conditions, which maybe disjunctions of conditions. Conditions are typically predicates which evaluate to true or false. Actions are either conclusions to be drawn or instructions to be carried out.

An example of a rule in the system is the rule for the randomized complete block design with subsampling (RCBDs) given in Figure 2. The premises are in the form of properties of designs and the actions are in the form of recommended designs. It is worth noting that the structure of the rules in the system is similar to the structure used in the knowledge-based system for medical consultation called MYCIN.

IF

1) there is one factor in the experiment;
2) there are 15 or less treatments;
3) there is one source of variation among experimental units;
4) there is subsampling;

THEN

a possible design for this experiment is the completely randomized design with subsampling.

Figure 2. A production rule from the system.

Shell

The shell is responsible for extracting knowledge from the knowledge base (inference engine) and for providing a smooth communication between the user and the system (user interface).

The inference engine of the system is implemented by the use of Prolog's rules as representation of rules in the knowledge base. For example, the rule for randomized complete block design (Figure 2) is implemented in Prolog by making the recommendation as the goal and the premises as the subgoals (Figure 3).

```
design (rcbds) :–
        factors (1),
        treatment (less (16)),
        sourceofvar (1),
        subsampling (with).
```

Figure 3. An implementation of a rule in Prolog.

Although, the use of Prolog's rules to implement the rules in the knowledge base can answer queries about the knowledge base, this cannot be considered as simulating an expert's behaviour. It can truly simulate an expert's behaviour if an acceptable facility for interaction between the user and the system is also provided. In the system, interaction is provided by allowing the user to ask for explanations about the question being posed by the system. Moreover, after a recommendation is given, all the satisfied conditions are displayed for the user to examine. Figure 4 illustrates how explanations are implemented in the system.

```
factors (N) :–
        write ("How many factors you want to observed"), nl,
        write ("   are involved in the epxeriment? (1 to 6)"),
        get response (Response),
                (Response = ?, explain (factors), factors (N);
                 Response <> ?, str_int (Response, N) ).
```

Figure 4. An implementation of the user-interface module of the system.

Outline of the Reasoning Process in the System

Consider the system as one problem. This problem can be represented using an AND/OR graph. Specifically, AND/OR graph representation involving only the OR nodes is used. The AND/OR graph representation for the problem can be formulated as follows:

i) the nodes of the graph correspond to the questions of the system to the user and/or actions to be taken by the system; and the arcs correspond to the possible answers for these questions which leads to other questions;

ii) a successor is obtained through the answers (arcs) given by the user;

iii) the start node is the initial question which the system asks the user;

iv) a goal node is any of the recommended design.

This formulation forms a directed graph. Thus finding the most suitable design for an experiment is reduced to searching for a path in the graph from the start node to one of the goal nodes (solution path).

Given an AND/OR graph representation for the problem, the two basic search strategies, the depth-first search and the breadth-first search strategies, can be used to find a solution path. For this system, however, the appropriate strategy is the depth-first search strategy.

The Analysis Component

This component, as mentioned earlier, provides procedures for the analysis stage. The procedures provided .include those for managing data sets, for checking whether the data set gathered satisfies the assumptions of the analysis of variance, for carrying out the analysis of variance, and for doing linear comparison of means. In short, this component is no different from an ordinary statistical package. However, the procedures for checking the assumptions of the analysis of variance were included since most of the time researchers fail to include this very important step of the analysis.

The Explanation Component

The main purpose of this component is to explain further the results generated by the consultation and analysis components. It is responsible for generating the layout of the experiment and explaining the contents of this said layout to the user. It also provides explanations and interpretations of the results produced by the analysis component.

A User's View of the System

To get a broad sweeping view of how the system operates, an example session with the system is presented. The statements enclosed in "( )" are descriptions of what are supposed to be displayed, "S:" refers to the system and "U:" to the user.

***CONSULTATION**

S; How many factors you want to observed are involved in the experiment? (1 to 6)

U: I don't know.

S: I don't understand you.
Please enter "?" for explanations.

U: ?

***EXPLANATION**

Subject: Factors in the experiment
(This topic will be explained in detail, after which the system will continue with the consultation)

***CONSULTATION**

S: How many factors you want to observed are involved in the experiment? (1 to 6)

U: 1

S: How many treatments does your experiment have?

U: 10

S: Are the experimental units homogeneous?

U: No

S: Since the expt'l units are not homogeneous, then there is/are known source(s) of variation. 1 or 2?

U: 1

S: It is assumed that you have equal number of replicates/treatment since your expt'l units are not homogeneous. Are there subsamples within an experimental unit?

U: Yes

S: From the conditions satisfied, there is enough reason to believe that one possible design for the experiment you described is the completely randomized block design with subsampling.

After the design of the experiment is determined, the system starts to generate the layout of the experiment. The researcher can then proceed to conduct the experiment using the design recommended and layout generated by the system. After the experiment has been conducted and the necessary data gathered, the system can be consulted anew for the analysis phase.

Concluding Remarks

It is not clear whether the system can really take the place of an expert in experimental designs. But most of the responses, of those who were able to observe how the system operates (most of them were statisticians), have been encouraging.

Several problems still have to be tackled before a reliable consultation system for experimental design can be developed. The most major of these problems is the development of a suitable knowledge acquisition method for this type of application.

Literature Cited

Albacea, E.A. and R.B. Caringal. 1987. "NANO: An Expert System for Experimental Designs." In the Proceedings of the Fourth National Convention on Statistics (Philippines).

Barna, G. 1987. "Experimental Expert System To Assembling National Balances in the Hungarian CSO." Invited Paper 15.3, 46th Session of the ISI, Tokyo, Japan.

Berzuini, C., G. Ross, and C. Larizza. 1986. "Developing Intelligent Software for Non-Linear Model Fitting as an Expert System." In *COMPSTAT 1986: Proceedings in Computational Statistics.* F. De Antoni, N. Lauro, and A. Rizzi, eds., Physica-Verlag, Vienna, pp. 259-264.

Bratko, I. 1986 *Prolog Programming for Artificial Intelligence.* Addison-Wesley Publishing Company, Reading, Massachusetts.

Carlsen, F. and I. Heuch. 1986. "Express – An Expert System Utilizing Standard Statistical Packages." In *COMPSTAT 1986: Proceedings in Computational Statistics,* F. De Antoni, N. Lauro, and A. Rizzi, eds., Physica-Verlag, Vienna. pp. 265-270.

Chambers, J.M. 1981. "Some Thoughts on Expert Software." In Proceedings of the 13th Symposium on the Interface, pp. 36-40.

Darius, P. 1986. "Building Expert Systems with the Help of Existing Softwares." In *COMPSTAT 1986: Proceedings in Computational Statistics,* F. De Antonii, N. Lauro, and A. Rizzi, eds., Physica-Verlag, Vienna. pp. 277-282.

Dambroise, E. and P. Massotte. 1986. "Muse: An Expert System in Statistics." In *COMPSTAT 1986: Proceedings in Computational Statistics,* F. De Antoni, N. Lauro, and A. Rizzi, eds., Physica-Verlag, Vienna. pp. 265-270.

Gale, W.A. and Pregibon, D. 1982. "An Expert System for Regression Analysis." In Proceedings of the 14th Symposium on the Interface, ed. Heiner, Sacher, and Wilkison, Springer-Verlag, New York. pp. 110-117.

Gale, W.A. 1987. "Student: A Tool for Constructing Consultation Systems in Data Analysis." Invited Paper 15.1, 46th Session of the ISI, Tokyo, Japan.

Hilhorst, R.A., *et al.* 1987. "Rochefort – Research on Creating a Human Environment for On-Line Research Tools." *Statistical Software Newsletter,* Vol. 13, No. 2, pp. 44-56.

Jackson, P. 1984. *Introduction to Expert Systems.* Addison-Wesley Publishing Company, Inc. Reading, Massachusetts.

Pregibon, D. and W.A. Gale. 1984. "REX: an Expert System for Regression Analysis." In *COMPSTAT 1984: Proceedings in Computational Statistics.* T. Havranek, Z. Zidak, and M. Novak, eds., Physica-Verlag, Vienna. pp. 242-248.