ARDA - EXPERT SYSTEM FOR RELIABILITY DATA ANALYSIS

By Dr Jake Ansell and Mulhim Al-Doori
Department of Business Studies
The University of Edinburgh
William Robertson Building
50 George Square
Edinburgh
EH8 9JY
Fax No. UK 031-668-3053
Tel No. UK 031-650-3806

Abstract

The paper explores the design and implementation of ARDA, an Expert System to analyse Reliability Data. Initially the viability of the knowledge domain is explored. The philosophy of design of the system is discussed. Details of the implementation are described. There is discussion of extension of system to other statistical analyses and of using alternative inferential bases.

KEYWORDS: EXPERT SYSTEMS, RELIABILITY DATA ANALYSIS, OBJECT ORIENTATED PROGRAMMING.

Introduction

Many application of Expert Systems have been considered in Statistics. A few of these have been successful and the failures have been fruitful in highlighting the possible problems in encoding statistical knowledge into software. The problems are usually the scope of the area and its tractability. Many areas are still too large for a workable software system. In other areas there are difficulties manipulating the knowledge.

In statistics there are two main types of Expert System: those which use statistical reasoning and those which assist statistical analysis. ARDA is an Expert System to assist Reliability Data Analysis (RDA) and is a member of the second group. We are not aware of a previously Expert Systems in this area. This might suggest the area is ill defined or the reasoning process not easily implemented. In the first section we review the knowledge domain, Reliability Data Analysis (RDA).

Whilst ARDA was originally conceived as an OOP system, ambiguities arose which meant we found the such an approach too confining. Hence the system is based on the flexibility of APL with elements taken from OOP Design. The philosophy of design is discussed in the second section. The third section covers ARDA's implementation. Obviously it is not possible to outline all aspects of ARDA in detail but an overview is given. The final section of the paper is a discussion how ARDA could be extended.

Reliability Data Analysis

For an Expert System to be viable the knowledge domain needs to be well defined and sufficiently limited. Reliability has a well developed literature: texts by Ascher and Feingold [1984], Cox and Lewis [1966], Cox and Oakes [1984], Lawless [1982] and Mann , Schafer and Singpurwalla [1974]; a number of journals such as 'Reliability Engineering and Safety System' and 'IEEE Transactions on Reliability'; and relevant articles appear in a number of other journals. A significant portion of this material is devoted to Reliability Data

Analysis. There are also a number of packages, S-Plus, SAS and others including AGSS, an APL based system, see Lewis (1993), which enable the user to analyse the data using specific techniques. These packages do not give advice on selecting or applying the techniques.

For the system to be viable the boundaries need to be defined. ARDA has been designed to fit an appropriate model to data. This includes diagnosis of the stochastic processes and selecting an underlying distribution. The types of question the system may deal with are: 'What is the underlying distribution?', 'Should the component be replaced?' and 'Is the component under test better than the one currently being used?'. An issue of concern in building Expert Systems for statistical analysis is the strategies involved in the analysis. Whilst the philosophy underpinning statistics has been developed through statistical inference little attention has been paid to the practical problems of how a full analysis should be carried out. Cox and Snell [1971] do suggest some general strategies, but these still seem technique based. Others, Nelder [1990], have suggested empirical approaches. These approaches, though ultimately rewarding, would require considerable effort.

Currently the tools to build a system are relatively simple. Of the systems developed to aid analysis two types emerge: Knowledge Enhancing systems, KENS, or Knowledge Enabling systems, GLIMPSE. A Knowledge Enhancing system is a computerised text supplying the user with the required information. Knowledge Enabling system not only supply such knowledge but can implement the acquired knowledge. For a specific technique the system may suggest steps in the analysis. In GLIMPSE, for Generalised Linear Models, the system will give advice on the next step, it
does not though concern itself with the full analysis.

In ARDA the objective of the analysis is used to provide the strategy rather than just employing techniques. Suppose the objective is to decide whether to replace a component or not. The technique based approach would be to find the precise distribution whilst the objective lead approach would simply require to establish whether the distribution has an increasing failure rate or not.

Philosophy of Design

ARDA was initially conceived as an OOP system. Such systems have already been built in APL, see Alfonseca [1990] and Frey [1992]. Both these examples have extended APL to facilitate OOP design in APL. They have demonstrated the power of APL in creating features such as Abstraction, Encapsulation, Inheritance and Polymorphism. They have not indicated whether OOP is too limiting for APL design.

Obviously some features of ARDA are ideally suited to OOP. ARDA has a natural granular design. The 'System' would consist of three objects: 'Objective', 'Model' and 'Data' which conform to OOP principles. Each object would have a data structure, usually represented as a nested vector array, on which procedures would operate. There would be a hierarchy of Objects. Polymorphism is an apt description of the need to have different algorithms to deal with different forms of data producing the same output.

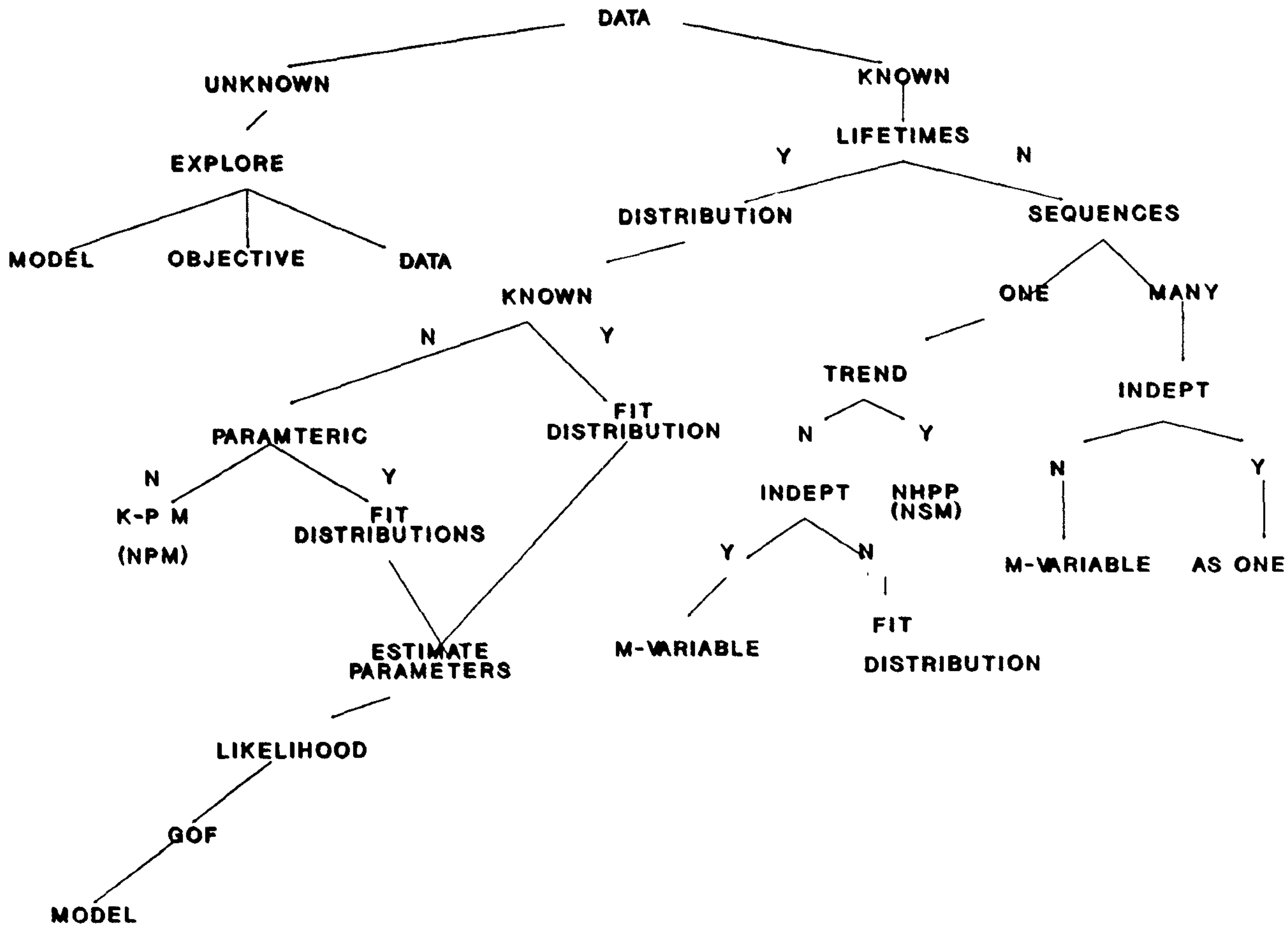
Whilst OOP has many advantages, in ARDA there was a major ambiguity in the main role of analysis. In the analysis elements from each of object 'Objective', 'Model' and 'Data' are often required simultaneously. This could be tackled at 'System' level but the supposed advantage of granularity would be lost. Alternatively one could treat 'data' ambiguously as an object and a message. In analysis 'data' would be treated as a message to the object 'model' which would be analysed in light of the elements from the object 'objective'. The hierarchical position of 'model' and 'objective' would therefore also need to be determined. Considerable effort could be involved in resolving these issues without a real return in software development.

The compromise made was to used strands from OOP with standard APL usage. The 'non-authoritarian' approach of ARDA is an example of the use of APL. The phrase 'non-authoritarian' was coined by Nelder [1990] to describe the approach taken in GLIMPSE. Analysts were given the freedom to pursue their own analysis without undue restriction. An analyst may even be allowed to use techniques 'incorrectly'. In such circumstances obviously it is unlikely that advice/guidance can be given.

This 'non-authoritarian' approach is implemented by using two levels of function within the workspace. The high level function will provide the expertise of the system, while the low level function will provide the basic calculations. Obviously the users will have the freedom to chose the level of functions they use. The high level function will generally test inputs to ensure correct usage. Underpinning the use of the high level functions will be a map for the analysis, see Figure 1. The map again does not assume that an analyst will follow a specific route but that the analyst will develop paths through the map. The map links the high level functions. Further discussion of this will be provided in the implementation section.

Figure 1

The Analysis Map



Implementation of the ARDA

It is not possible to describe ARDA in detail, but we will highlight the main features in this section.

The object based system uses nested arrays as the main data structure. The 'System' will consist of five elements (slots):
'Data', 'Model', 'Objective', 'Control' and 'Knowledge'. The first three will have vector nested arrays and are discussed later. 'Knowledge' and 'Control' will consist of procedures based on the 'Status Vector' of 'System'. 'Control' will generally consist of functions checking the 'Status Vector' to see if procedures can be carried out. 'Knowledge' at the 'System' level will consist of 'HOW' functions. These will indicate how a specific function operates, what its inputs and outputs are, and the detail of the algorithm involved.

Data

This object is concerned with the entry and manipulation of data. The vector 'Data' will consist of the elements (slots):
Status, Definition, Validity, Structure and X, the actual data. The first four elements describe the data and its attributes, for example 'Status' indicates whether the data is defined or not. Associated with each element there will be a set of procedures, functions. INITIALD for example will create the array 'data' and set 'Status' to not defined.

Model

This object is more complex than 'Data', dealing with the selection/defining of an appropriate model. It also contains information about the models. There are a number of vector nested arrays in the object. The vector 'Model' contains details of the current model with elements: Status, Stochastic and Distribution. 'Stochastic' describes the stochastic processes concerned and 'Distribution' names the underlying distribution.
The other vectors contain knowledge on the distributions

and stochastic processes. There are a set of procedures which operate on these vectors. A particularly useful set of function for analysis are those which construct the distribution trees,
see MacDonald and Richards [1987].

Objectives

Currently the least developed object is 'objective'. So far only basic objectives have been included in the object, though it is hoped to extend this later. The object aims to define and clarify the objective, and hence specify the strategy. The function for clarification and manipulation of the objectives are also limited. As with the distributions in the object 'model' for each of the objectives there will be a vector. This will contain the set of conditions the objective to be satisfied. These conditions will be nodes on the analysis map, see Figure 1. The path through the map is therefore defined and strategy selected.

Analysis

Movement through the analysis path will be governed by the objective selected. This provides a control mechanism in ARDA. It will however be affected by the results obtained at each node in the map which may reroute the analysis or end it.

Each node in the map will have associated with it one or more low and high level functions. The low level function will provide the basic calculation, usually calculating a test statistic. The high level function acts as a driver for the low level function, providing guidance and advice. The general form for the function is given in Figure 2.

Figure 2

Idealised format of high level function, Test

```
Test
Check_Test
Test_Statistic
Compare_Test
Report_Test
```

The high level function will check the preconditions for test (Check_Test), whether the data exists and is in the correct form. If the preconditions are not met a report will be produced which includes details of the low level function. If the preconditions are met then the low level function will be actioned and an
appropriate comparison made (Compare_Test). Based on the result of the comparison using a Frequentist approach a report will be produced. This will provide a basic interpretation and advice/guidance on the next stage of the analysis. The user may or may not take the advice, a 'non-authoritarian' approach.

Discussion

ARDA, an APL Expert System for Reliability Data Analysis, has been described in the paper. ARDA has been built partially using OOP approach, however where this has lead to ambiguity or over elaboration we have used standard APL.

ARDA provides a model which could be used for a wide variety of data analysis. The system is innovatory in the use of objectives to guide analysis through the analysis map. To extend the system to other statistical analyses would require the definition of the set of objectives and models, and to develop an analysis map. The general framework would be usable, though, it may be also necessary to modify the 'Data' object.

Although we have described ARDA as used for data analysis it may be used to enhance the users knowledge. The links between Objectives, Models and Data can be explored either to find what data is required to establish a given objective or which models are related to which objectives. Hence it can provide a 'what-if' analysis. Such inversion reflects APL's natural flexibility.

Whilst currently using a Frequentist with hypotheses being accepted or rejected, there is no obvious impediment to
generalising the approach to take account of the probabilities of the hypotheses. Then the satisfaction of the objectives could be expressed in terms of probabilities. This could be extended to
the implementation of a Bayesian analysis. These alternatives are under consideration at the present time.

References

Alfonseca, M, [1990], 'Object Oriented Programming Tutorial', APL Quote Quad, 20, 7-8.
Acher, H, and Fiengold, H, [1984], Reliability of Reparable Systems, Marcell Dekker: New York.

Cox, DR, and Lewis, PAW, [1966], The Analysis of Series of Events, Methuen, London.

Cox, DR, and Oakes, D, [1984], Analysis of Survival Data, Methuen, London.

Cox, DR, and Snell, ES, [1971], 'Applied Statistics', Chapman Hall, London.

Frey, RJ, [1992], 'Object Oriented Extensions to APL and J', Vector, 9, 116-125.

IBM Research, (1992), A Graphical Statistical System (AGSS) Reliability Analysis, IBM Research, Yorktown Heights, New York.

Lawless, JF, [1982], Statistical Models and Methods for Lifetime Data, J Wiley & Sons, New York.

Lewis, PAW, [1993], 'A Graphical Statistical System AGSS', APL '93.

MacDonald, JB, and Richards, DO, [1987], 'Model Selection: Some generalized Distributions', Proc Reliability '87, Birmingham, UKAEA, Culcheth.

Mann, NR, Schafer, RE, and Singpurwalla, ND, [1974], Methods for the Statistical Analysis of Reliability and Life Data, J Wiley &
Sons, New York.

Nelder, J, [1990], Talk given to RSS Seminar on Future of Statistical Computing', Royal Institution , London.