THE CONSTRUCTION OF STATISTICAL EXPERT SYSTEMS WITH APPLICATIONS.

by

Michael Kevin Williams

NEWCASTLE UNIVERSITY LIBRARY
095 50652 X
Thesis L5518

A dissertation submitted for the degree of Doctor of Philosophy of the University of Newcastle upon Tyne.

JUNE 1995

This is dedicated to Doreen, with much love
and many thanks for all her support and encouragement.

"This is indeed a mystery," I remarked. "What do you imagine that it means?"

"I have no data yet. It is a capital mistake to theorise before one has data. Insensibly one begins to twist facts to suit theories, instead of theories to suit facts."

Sir Arthur Conan Doyle, "A Scandal in Bohemia".

ACKNOWLEDGMENTS

I am grateful to Professor G. B. Wetherill, who interested me in the subject of statistical expert systems in the first place. His confidence in me at times of difficulty was a great help. Thanks are also due to Professor J N R Jeffers, who provided a great deal of initial guidance to the solutions of certain problems considered in this thesis. I am also grateful to the Department of Statistics at the University of Newcastle upon Tyne, for the use of the computing facilities there. Thanks are also due to the Science and Engineering Research Council, who provided the funding for the research.

At the last minute, the original plan to obtain the results of Chapter 8 fell through. I therefore want to say a personal word of thanks to those people who gave time to compile the grids shown there. Dr. Richard Clayton read an earlier draft of the thesis, and his helpful suggestions have been incorporated into the text. Finally, I wish to thank the secretaries of the Department of Mathematics especially Heather Bliss, for allowing me the use of word processing facilities for the typing of the thesis.

TABLE OF CONTENTS

SECTION I - INTRODUCTION

CHAPTER 1 - A SHORT INTRODUCTION 1

1.1 The Problem of Constructing Statistical Expert Systems 1
1.2 Background to the Project 2
1.3 Structure of Thesis 4

CHAPTER 2 - BACKGROUND TO STATISTICAL EXPERT SYSTEMS 6
2.1 Introduction to Expert system Terminology 6
2.1.1 What is an expert system? 6
2.1.2 Structure of expert systems 7
2.1.3 Additions for statistical expert systems 9
2.1.4 Limitations of expert systems 10
2.2 Rile Base Storage Schemes 12
2.2.1 Procedural rule bases 12
2.2.2 Frame based rules 13
2.2.3 Causal Networks and Semantic Nets 15
2.3 A Brief History of Statistical Expert Systems 17
2.3.1 The early years 17
2.3.2 More recent work 19
2.3.3 Intelligent front-ends 20
2.3.4 Advisory systems 21
2.3.5 Test construction systems 22

SECTION II - BACKGROUND TO CONSULTANCY PROBLEMS

CHAPTER 3 - MODELLING CONSULTANCY PROCEDURES 25
3.1 Introduction 25
3.2 The system as a consultant 25
3.3 Modelling statistical consultancy 27

3.3.1 The models 27
3.3.2 Proposals for overall model 29
3.3.3 General guidelines to statistical practice 31
3.4 Considering consultancy stereotypes 33
3.4.1 The Consultant 34
3.4.2 The Client 36
3.5 The Deming Philosophy 40
3.6 Conclusions 41

CHAPTER 4 - DETERMINING THE BACKGROUND SUBJECT 42
4.1 The importance of being informed 42
4.1.1 A cautionary tale 42
4.1.2 What kind of background? 43
4.1.3 Just what is the problem? 44
4.2 The 'Tree of Knowledge' 45
4.2.1 Some definitions 45
4.2.2 Constructing a tree 46
4.2.3 Using the tree 48
4.3 Developments of the tree 49
4.3.1 Depth of knowledge 50
4.3.2 Initial Plausibility values 51
4.4 Programming of the tree 52
4.5 Conclusions 53

CHAPTER 5 - ELICITING THE FACTS OF THE PROBLEM 54
5.1 Introduction 54
5.2 Obtaining the background facts 54
5.2.1 Why bother with the facts? 54
5.2.2 Which facts will be required? 56
5.2.3 Question selection and ordering 57
5.2.4 Fact checking, validation and elicitation 59
5.3 Fact elicitation for established questions 62
5.3.1 ExpCheck 62

5.3.2 The facts needed 63
5.3.3 The structure of the questions 63
5.3.4 Help facilities 67
5.3.5 Programming details 68
5.4 Forming a new network of questions 69
5.4.1 The key questions 69
5.4.2 Formulating the structure 70
5.4.3 A sample network 71
5.4.4 Programming the network 73
5.5 Conclusions 73

SECTION III - SELECTION OF ANALYSIS PROCEDURE

CHAPTER 6 - PERSONAL CONSTRUCT THEORY - BACKGROUND AND METHODOLOGY 74
6.1 Introduction 74
6.1.1 Background to the problem 74
6.1.2 Why test selection rules? 76
6.2 Alternative rule generation methods 77
6.2.1 Data-driven methods 77
6.2.2 Psychometric techniques 78
6.3 Personal Construct Analysis 80
6.3.1 A brief history 80
6.3.2 Definitions and notations 81
6.3.3 Grid construction - an example 82
6.3.4 Focusing the grid 89
6.3.5 The example continued 91

CHAPTER 7 - INTRODUCTION TO STRATEGIC SELECTION RULES 96
7.1 Introduction and definitions 96
7.1.1 What are strategic selection rules? 96
7.1.2 Some important criteria 97
7.1.3 Why personal construct analysis? 98

7.1.4 Structure of the algorithms 99
7.2 Model Characteristic Rules 100
7.2.1 Construction of model characteristic rules 100
7.2.2 Plausibility of the MCR 101
7.2.3 MCR's for the example 105
7.3 Characteristic Implication Rules 107
7.3.1 Construction of the rules 107
7.3.1.1 Finding the implications 108
7.3.1.2 The implication map 110
7.3.2 Problems with the algorithms 112
7.3.3 CIR's for the example 115
7.4 Additional utilities and programs 119
7.4.1 Multiple Grids 119
7.4.1.1 Normal merging and analysis 119
7.4.1.2 Preliminary pruning of the combined grid 120
7.4.2 Updating and amending grids 121
7.4.2.1 Slight modifications 122
7.4.2.2 Complete renewal of a grid 122
7.5 Conclusions 123

CHAPTER 8 - STRATEGIC SELECTION RULES FOR EXPERIMENTAL DESIGN 124
8.1 Introduction 124
8.2 The Initial Grids 124
8.2.1 The selected models 124
8.2.2 The initial grids 125
8.3 Merging of grids 127
8.3.1 Merging grids using CORE 128
8.3.2 Considering the merged grid 131
8.4 Analyzing the final grid 133
8.5 Rules for Experimental Design Model Selection 135
8.5.1 Model Characteristic Rules 135
8.5.2 Discussion of MCR's 137
8.5.3 Characteristic Implication Rules 138

8.5.4 Comments on the CIR's 141
8.6 Using the derived rules 141

CHAPTER 9 - AN EXPERT SYSTEM FOR MODEL SELECTION 144
9.1 Introduction 144
9.2 The overall structure 145
9.3 How the system works 146
9.4 Conclusions 148

APPENDICES

APPENDIX - CONTENTS OF SOFTWARE DISK 149

BIBLIOGRAPHY 151

THE CONSTRUCTION OF STATISTICAL EXPERT SYSTEMS WITH APPLICATIONS.

Michael Kevin Williams,
Division of Statistics,
Department of Mathematics,
University of Newcastle upon Tyne.

Abstract

Statistical expert systems have grown more numerous over the last few years. However, the tools which are used in their development are largely modifications of existing methods from other areas. This means that the unique aspects of statistical problems may be omitted in their development.

This thesis examines the way that a statistical consultant works, and proposes guidelines and tools for system development based on this. Section I provides a brief introduction to this area. Section II considers the problems faced in establishing the actual problem brought by a client. Two areas of particular interest are establishing the background subject and the facts of a problem. Two methods are proposed to tackle these areas.

Also of interest is the way in which a consultant may select a statistical test. Section III introduces a program to elicit the rules used by a consultant. A disk containing the program and the functions used is included with this thesis.

SECTION I

INTRODUCTION

CHAPTER 1

A SHORT INTRODUCTION.

1.1 THE PROBLEM OF CONSTRUCTING STATISTICAL EXPERT SYSTEMS.

Over the last few years, the field of expert system development has become of increasing interest to those involved in statistical computation. Such systems provide intelligent and useful advice on how to proceed with a problem by using an encoded form of expert knowledge of a field. The rise in interest in these systems has coincided with an increased need for statistical advice in industry and commerce. As this need has increased, so the demand for tools that can assist the statistical consultant in his work has grown.

Statistical expert systems are increasingly being seen as the way forward. There are several reasons for this development. A consultant thinking of using such systems requires not only the mathematical capability to carry out any necessary tests, but also the deductive capability that is inherent in these programs. If consulting errors of the third kind (Kimball, 1957) are to be avoided, then the knowledge required to effectively perform a particular test has to be arranged in a way that will be of most use.

There are additional problems in developing statistical expert systems. The required knowledge for specific problems of interest in a number of areas may already have been suitably coded. However, if statistical expert systems are to become fully effective then the general rules that are employed in everyday statistical work need to be set down as well. The difficulty is

there is a vast range of knowledge that can fall within that remit. Sorting out which facts are required and developing methods for obtaining them can be a difficult task.

In order to prepare and build these systems, a number of techniques have been developed to obtain the required rules. These methods vary in complexity, ease of application and required resources. If the development of these systems is to be simplified, then simple guidelines and tools have to be developed. This thesis presents such a set of techniques. Their usefulness is demonstrated by examples relating to a theoretical system for selecting a suitable experimental design.

1.2 BACKGROUND TO THE PROJECT.

In 1984, the Department of Statistics at the University of Newcastle upon Tyne and the Port Sunlight Research Laboratories of Unilever Research were awarded a grant under the terms of the Alvey Directorate to investigate the possibilities for "A Statistical I.K.B.S. for the investigation of relationships between two variables". (I.K.B.S. is an abbreviation for Intelligent Knowledge Based Systems, the correct title for an expert system.) Regretfully, the project was one of those caught in the financial problems experienced by the directorate, and the full funding was never forthcoming. This resulted in the original project having to be shelved due to lack of resources. The problem was what to do in place of the planned work.

It was decided to amend the project focus, and some work was done on the general possibilities for statistical expert systems in a 'real-world'

setting. As a result of that work, a number of useful ideas were developed and proposed, and these ideas are currently being exploited in a number of research projects within the department.

One of the proposed ideas was to attempt to construct a system for formulating efficient and practical experimental designs. However, when the time came to put some of these ideas into practice, it gradually became clear that a more fundamental problem had not been fully considered. The core of this problem has already been mentioned earlier in this chapter - there appeared to be no set way of modelling the simple, common sense decisions that statisticians make when setting up a model. More specific to the needs of the system, the way in which the models and tests that would be required were selected had never been fully investigated and considered.

This poses an interesting problem. In order for an expert system to be effective, as complete a model of the process as it is possible to formulates needs to be included. This means that as many of the relevant aspects of the problem as possible need to be included, but without making the system so large that it would be impossible for other people to use. The aim of this thesis is to present solutions to this dilemma. The following questions provide the basis for the work presented.

1) Can any techniques be considered mandatory for a statistical expert system, and is there any way of setting down these methods?

2) Greenfield (1983) states that a good consultant should know something about the background to the problem he (or she) is looking into. Is there any way of building that into a statistical expert system?

3) Are there any general methods which can be derived for modelling and formulating the general rules for test or procedure selection in a particular situation?

1.3 STRUCTURE OF THESIS.

The thesis is broken down into three sections. Chapter 2 discusses the terminology of expert systems and the types of rules that can be used. A brief history of statistical expert systems is also provided to give background.

The second section considers the problem of learning about a client's needs and difficulties. Chapter 3 examines the literature on statistical consultancy. A number of guidelines for statistical expert system development are proposed based on these writings. The first task of any consultant is to try to understand what the client's subject area is. The next chapter presents an idea for obtaining a simple description of the client's areas of interest. Chapter 5 considers the next stage of a consultancy, that of obtaining information about the problem itself. A structure for questions to obtain facts about the problem is discussed. Two examples show how these structures can be derived in practice, and one of the examples discussed leads to a system which is included in this thesis.

The third section considers the problem of selecting a model or analysis procedure for the problem. A software package to develop rules that model the way a consultant picks the appropriate model has been developed. Chapter 6 presents the theoretical background to the techniques used, while chapter 7 presents the algorithms developed for the system. A very simple example is given throughout to show how the methods work. Finally, Chapter 8 presents the results of trials the software underwent to form rules for experimental design. The rules obtained are examined and commented upon. The main proposals and ideas presented in the thesis are then summarised in Chapter 9.

The Appendix lists the software programs used in this thesis, which are also enclosed on disks contained within the thesis.

CHAPTER 2

BACKGROUND TO STATISTICAL EXPERT SYSTEMS.

2.1 INTRODUCTION TO EXPERT SYSTEM TERMINOLOGY.

In this chapter, the terminology and structure of expert systems in general is introduced and explained. Some methods for coding the rules these systems use are illustrated, and a brief history of expert systems in statistics outlined. To begin with, it needs to be clearly stated what is meant by an expert system, and what these systems consist of.

2.1.1 What is an expert system?

Various definitions of an expert system have been given in the past. Hayes-Roth et al (1983) define them as programs which investigate "methods and techniques for constructing man-made systems with specialised problem-solving techniques". Alteratively, Hahn (1985) describes an expert system as "one in which the knowledge of an expert is built into a computer program, enabling it to emulate the process that the expert follows in tackling a problem". Other definitions exist, often serving only to add to the confusion which may exist.

The easiest way to define an expert system is to think about the way conventional programs work. Most standard programs consist of a control program, a number of techniques which the system offers, and some form of data entry and reporting. The system receives the data, executes the technique and then waits until the next task is selected. The way the system operates is solely dictated by the programmer, and no other outside influence can affect the way it works. It is in this aspect that an expert system, or more properly an Intelligent Knowledge Based System (IKBS) differs most clearly from standard programs.

Rather than following a set pattern of actions, an IKBS attempts to employ the knowledge of one or more people about the process of interest. This knowledge is usually stored in the form of procedures or rules, each of which contains one piece of information about the way in which the process or analysis works. As a result, it is the rules themselves which govern the way in which the program works. One direct consequence is that expert systems tend to be advisory or selective systems rather than simple analytic programs. Another occurs because the rules do not solely depend on the data input during the run of the program. The result is that the system can apply itself to a number of possible solutions. In addition, the system can offer alternative suggestions as to the best way to continue with a particular task.

2.1.2 Structure of expert systems.

The ways in which expert systems work vary according to the type of system that is being built. However, most IKBS have three major components in common. This leads to a structure that has been referred to as the 'Expert System Tryptych' (Barr & Feigenbaum, 1981). This structure can be represented by the following diagram.

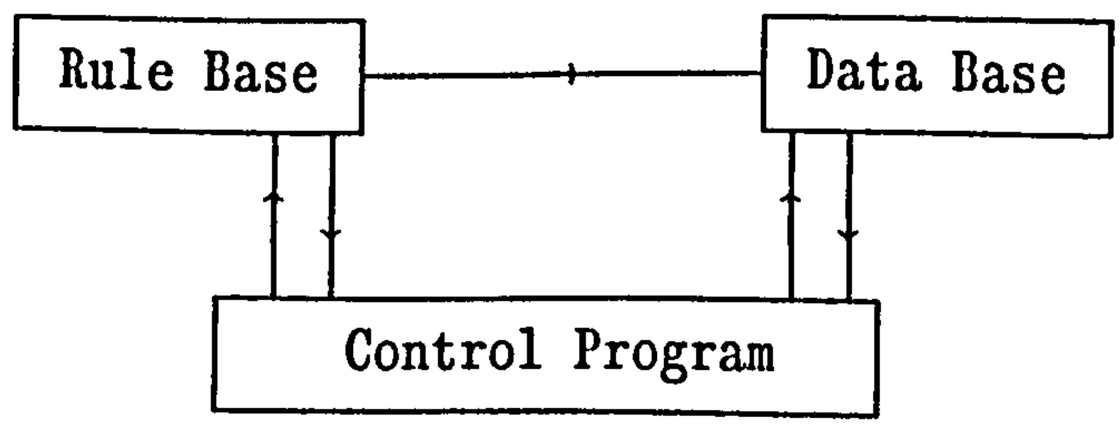


FIGURE 2.1 - THE EXPERT SYSTEM TRIPTYCH.

The first component of interest is the Rule Base. This is a collection of procedures representing the rules used by the system, codified in some suitable way. The methods used for obtaining and encoding this knowledge vary from system to system. When IKBSs were first built, the usual method was by interview with one or more experts over a period of time. As the interview progressed, the experts and the programmer would come to some form of agreement as to how the decision making process worked, and the rules that were agreed upon would be added to the system. Hand (1984) gives an example of how this process might work for a statistical expert system. Over the last few years, a number of automated knowledge elicitation procedures have been developed. Section III of this thesis deals with one of these methods

However, the rule base does not only include rules for the running of the system. It may be the case that rules which affect how the rules themselves behave are required. Known technically as meta-rules, these may also be found within the rule base, although it is more usual for them to be placed in their own rule base. These specialised rules tend to control the way that general aspects of information are considered, or else the order in which a set of rules may be considered.

Working in parallel with the rule base is the Data Base. As the name suggests, this is a storage base for any facts, figures and/or deductions that are made in the course of the running of the program. The facts themselves are obtained in a number of ways. Initial facts may come supplied with the system, for use in certain of the rules. As the system progresses, more facts are deduced or supplied by the user and these are added to the data base. In time, the data base comes to represent a complete picture of the problem that the system is trying to solve.

Obviously, communication between the two bases must be carried out in a clear and concise manner. To facilitate this, all the procedures used for communication between the user and the system, as well as any 'book-keeping' functions and utilities, are stored in the Control Program. Although this is the simplest section to understand, it can be the most difficult part of the system to design and implement. The arrows in figure 2.1 show the lines of communication between the various section of the system.

2.1.3 Additions for statistical expert systems.

The expert system triptych is useful as a basis for most expert systems. However, for statistical expert systems a few modifications have to be made. The reasons for this are made clearer by considering what a statistical expert system will be asked to do. The two main tasks are to make deductions and to give advice about some type of analysis. The structure given in Figure 2.1 will be insufficient to cope with the differing types of facts that will be available. As a result, the basic structure requires certain amendments and additions.

Many of the changes arise from the storage of data sets and matrices. While the data base appears to be the suitable place for these, the structures used will have to be modified. In addition, the rule base will have to be able to cope with the results from appropriate statistical analysis techniques. Another difference occurs because the results of any analysis of the data may need to be incorporated into the rules themselves.

Problems with data analysis lead to the most significant change to the triptych structure. Some way of storing the procedures to carry out relevant tests is required. In addition, a group of rules for use in test selection will be required, but these could be stored alongside the other rules in the

rule base. As the size and complexity of the analysis procedures would be prohibitively large, it would not make sense to store them in the rule base.

What is therefore needed is a new base of procedures, in which the analysis programs can be stored and called as required. This proposed extra base would be known as the algorithm base. When a rule in the rule base entails using an algorithm, the appropriate one would be selected from the algorithm base. The result could then be stored in the data base, to be accessed as appropriate. The structure of the statistical expert system can therefore be expanded from that shown in figure 2.1. As before, the arrowed lines indicate the flow of information from one base to another.

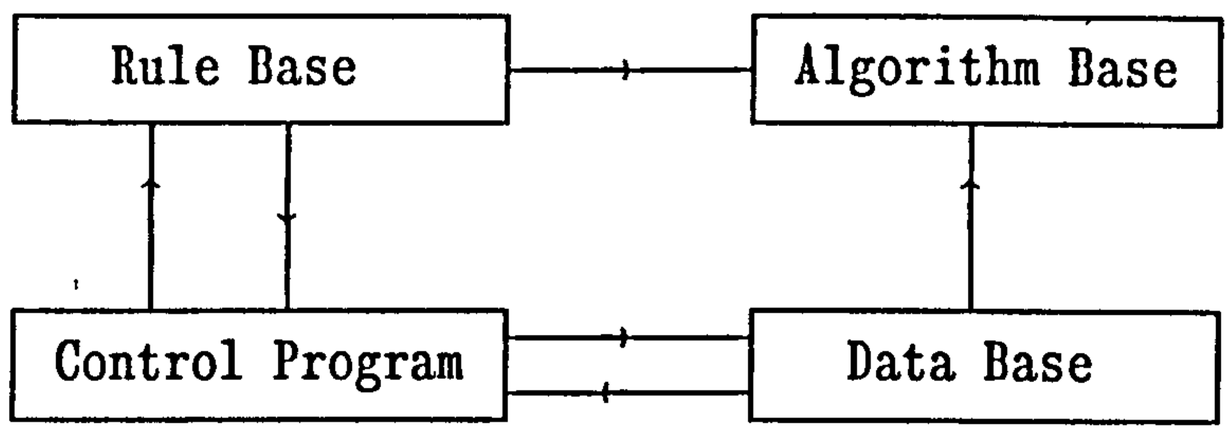


FIGURE 2.2 - STATISTICAL EXPERT SYSTEMS TRIPTYCH.

2.1.4 Limitations of expert systems.

An important point to bear in mind with expert systems is that they can only work with the rules and procedures that they are given. Any conclusions made in the course of a consultation can only be based upon what the system has been told. The result is that the accuracy of their deductions is highly dependent on the quality of the initial rule base. If the wrong information or the wrong decision making rules have been programmed, then the system will make the wrong conclusions.

It is for these reasons that IKBS are most effective when working in an advisory capacity. Most of the better known systems, such as MYCIN (Buchanan & Shortliffe, 1985) can be seen to follow this approach. Any decisions reached by the system also have to be considered in the context of actual experience. One way of ensuring this is to have some measure of the certainty of the decision included in a list of possible actions, rather than just one option. A number of measures have been proposed for gauging this measure, each with their own terminology and notation. For this thesis, we shall call this measure of certainty the plausibility of the conclusion. Later chapters will deal with this aspect as it occurs.

The above comments can be summarised in two guidelines to remember when considering statistical expert systems. Firstly, they are not infallible, and are liable to err if the rule base is not sufficiently complete. Secondly, they should not be regarded as the final arbitrator for a given problem. A list of options may be given, but if not the results have to be considered in the experience of the user.

Finally, there is another limitation to expert systems with an obvious counterpart in statistics. Kenward (1957) once wrote of the 'Statistical Error of the Third Kind', that is giving the right answer to the wrong question. This danger can also frequently occur when using expert systems. A person may have a problem which falls outside the system's area. However, he has tried to use the system and obtained an answer which seems to make sense. The result is possible confusion and waste of resources. One way of dealing with this is to establish what the problem is before any real work is done. Methods for doing this are discussed in chapter 4.

2.2 RULE BASE STORAGE SCHEMES.

Part of Volume I of Barr and Feigenbaum (1981) is devoted to an explanation of rule storage methods in an IKBS. A number of possible schemes exist, each with their own problems and structures. In this section, three possible methods for statistical expert systems will be considered, and illustrated by use of a simple example.

2.2.1 Procedural rule bases.

One of the earliest methods to be developed was the procedural rule base. One system which uses this method is MYCIN (Buchanan & Shortliffe, 1985), a diagnostic system for blood disorders. The procedural rule works by looking at the rules in turn, and checking the clauses against the known facts. If any facts asked for are unknown, the system queries the user about them. If all the relevant facts for one rule are shown to be true, then that rule is said to be fired, and a deduction of some type is made.

Each procedural rule within MYCIN has the following components:-

a) An identification number, which is used by the system for reference.

b) A set of clauses which have to be satisfied, at least in part, in order for the rule to be triggered. In MYCIN this consists of symptoms associated with a particular disease or other symptom.

c) The conclusion that is to be drawn by this rule. Each conclusion within MYCIN has a plausibility value associated with it, which is used to gauge the chances of a particular disease being present.

In order to illustrate the methods, a rule can be shown relating to a simple statistical problem. Consider the selection of a test for two samples of data. It has been decided that a comparison of means would be useful, but

the type of distribution the data comes from is unknown. A rule may be required to determine the need for a robust test, say the Aspin-Welch form of t-test (Aspin, 1948). The following is a possible procedural rule:-

RULE001

IF: 1) The sample sizes are roughly equal, and

2) A comparison of means is requested, and

3) There is evidence that the background distribution of the samples is not Normal,

THEN: There is evidence (.7) that an appropriate test would be the Aspin-Welch procedure.

(It should be pointed out that the plausibility value of .7 given above is purely arbitrary, and is included for purposes of illustration only.)

2.2.2 Frame based rules.

A more recent method to be suggested for the rule base structure is that of frames. A frame can be thought of as a notice board, with slots in which characteristics of a particular object or item of interest may be recorded. The values in these slots are used by the system for deduction of unknown facts and decision making. As new facts about the situation of interest are obtained, these are entered in the appropriate slots and used for future deductions. This means that frame representation is of most use in systems for selecting or identifying an object.

Consider the example of a rule to use in selecting the robust t-test. A general frame for tests first needs to be established, which defines the types of information which is placed in each slot. This frame is known as the Generic Frame. One possibility for the two sample test would be the following:-

Generic TEST Frame:

Self:	A TEST; a COMPARISON
Data:	
Compar:	a TYPE-OF-TEST
SpFt:	a SPECIAL-FACT
Result:	a TABLE; a FORMULA

Each slot defines a single piece of information that the system can use. 'Self' denotes what the object being described by the rule is. In this case, there are two possibilities: a test and a comparison. 'Data' describes the type of data that the test requires. 'Compar' states what the comparison is, and acts as a check to 'Self'. 'SpFt' is provided for the storage of facts pertinent to a particular test. Finally, 'Result' states what action is taken with the result. For the test, this means either consulting a table or doing a calculation.

A frame for the Aspin-Welch test can now be constructed along these lines. The following may be a suitable frame:-

ASPIN-WELCH Frame:

Self:	a COMPARISON
Data:	NORMAL
Compar:	MEANS
SpFt:	ROBUST
Result:	a FORMULA

The facts in the slots are all possible values which the contents of the slots of the generic frame can take. So in the Compar slot, MEANS is a possible value for the variable TYPE-OF-TEST.

2.2.3 Causal Networks and Semantic Nets.

A third form of knowledge representation is the semantic net or causal network. Although each is related to the other, they are not equivalent. The method consists of a series of paths, each of which connects one or more nodes. It is the nodes that contain the knowledge and rules that are of interest.

For the semantic net, the nodes consist of pieces of information, rather like the slots in the frame representation. The paths are then connectors, which show the relationship between the two pieces of information. For example, for the Aspin-Welch rules it can be argued that Aspin-Welsh and Robust are both characteristics of a test. Therefore one possible path between two nodes in a semantic net would be the following:-

Robust $\xrightarrow{\text{IS-A-CHARACTERISTIC-OF}}$ Aspin-Welch

In this case, 'IS-A-CHARACTERISTIC-OF' is the name given to the path, and this represents one rule.

The beauty of semantic nets is that they can be used over more than one group of facts, with paths connecting the relevant facts from different groups. When presented graphically, this has the effect of creating 'planes of knowledge'. These depend upon the fact that the nodes can be grouped together according to some criteria, with the groups known as classes. These classes can then be grouped together, and the connectors between the members of the group arranged as if on one plane. Any connectors between the groups can then be shown and thought of as lines between planes in some geometric sense. A simple example of how this hierarchy works can be found in Table 1 of the paper by Spiegelhalter and Knill-Jones (1984).

Causal networks are slightly different, in that the nodes are not particular pieces of knowledge. Instead, the nodes represent some rules and the paths connectors between the rules. The result is a representation of the structure of knowledge of a particular area, rather than the structure of the facts of the problem. If selection of the test is seen as a causal network, then the following is one possible part of the network.

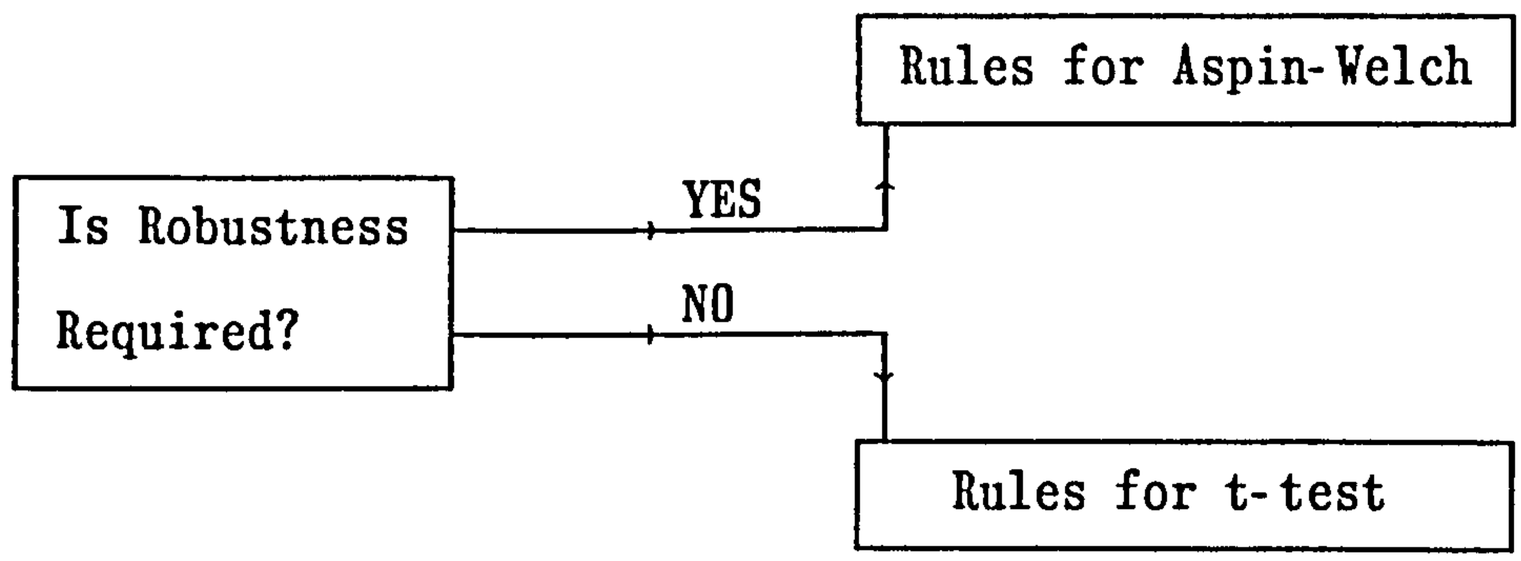


FIGURE 2.3 - SECTION OF CAUSAL NETWORK FOR TWO SAMPLE SELECTION.

2.3 A BRIEF HISTORY OF STATISTICAL EXPERT SYSTEMS.

It is only over the last decade that development of statistical expert systems has taken place. From a very simple beginning, a large number of systems have been and are continuing to be developed. The section gives a brief review of the developments in this field.

2.3.1 The early years.

It was in 1980 that Byron Jones' paper 'The computer as a statistical consultant' was published in Bulletin in Applied Statistics (Jones 1980). He discussed the ways in which a computer could be programmed to assist in statistical consulting, as well as dealing with some of the problems that do occur. This paper could be regarded as one of the first serious considerations of this area, and so the first to deal with expert system problems. The following points are taken from the paper, and their relevance to expert system development shown.

a) To be of optimum use, any program for consultancy should be interactive. Most expert systems are now either fully interactive or menu driven.

b) If a program is given the wrong information to work with, then the wrong deductions will be made. As has already been mentioned, the obvious solution is to study consultants, observe what mistakes are made and cater for them in the system. The next chapter will consider some of these problems.

c) Interpretation of the results should still lie in the hands of the user. However, an explanation of how a decision was reached should always be available. This is an essential part of these systems.

The paper was seminal in that it was the first to set out the problems that would be involved in setting up a computer system of the type envisaged. Once this had been done in a clear manner, then the problems could be tackled and possible solutions obtained. In America, William Gale and Daryl Pregibon had begun to realise the potential of such systems. They also realised that the ideas of statistical strategy discussed by Chambers (1981) would need to be incorporated within these systems. From this basis, they began work on a simple system for tackling linear regression problems (Gale & Pregibon, 1983).

One of the problems they had to overcome was that of knowledge elicitation for the systems. Their solution involved the use of examples to select the techniques which are commonly used (Gale & Pregibon, 1984). This method works for a technique like regression analysis, where examples are the method by which the techniques are learned. In time, REX (REgression eXpertise) (Pregibon & Gale, 1984) was released as a system with simple rule entry.

2.3.2 More Recent Work.

Parallel to the work of Gale and Pregibon, other researchers were considering different aspects of statistical expert systems. As was mentioned previously, a team at Newcastle were looking into aspects of data entry for statistical systems.(Williams, 1986; Wetherill & Gerson, 1986). Similarly, Hahn (1985) was considering the problems involved in forming consultation systems. As Hahn himself pointed out,

> 'Expert statistical software, if well designed, might filter some of the more mundane and less important problems and then direct the user back to the live statistician for the more challenging ones.'

One possible implication of this is that statistical expert systems would be used to tackle common problems. This would allow the statistician to work on more interesting and demanding research. However, this relegates the IKBS to the role of dumb helper, which defeats the purpose of these systems.

While this was taking place, Hand (1984;1985) was considering the structure and models for expert systems. In his paper on system attributes (Hand, 1985), he suggested a number of points to be considered when building these systems, and some useful tricks that could be incorporated. The ideas presented in chapter 3 mirror much of what is stated in this paper.

Probabilistic statisticians were not excluded from the research either. In the early years of system development, a number of methods had been proposed to handle what could be termed as 'vagueness'. This is a term to cover how certain an expert is that the deduction of a rule would be correct. This 'vagueness' was measured in a number of ways, and a number of techniques developed to handle it. A more correct title for the idea would be the concept of uncertainty. However, there is not a consistent notation or methodology for these methods, although most have a basis in Fuzzy Logic or set theory. This led to problems in handling and combining different measures of uncertainty. Bayesian inference methods solved a lot of these problems, as well as presenting new opportunities. Spiegelhalter and Knill-Jones (1984) showed how these methods could be approached from such a viewpoint.

The demand for statistical advice systems has risen sharply over the last few years, and so the number of statistical expert systems has grown. By 1987, there were over one hundred researchers in artificial intelligence in statistics (Hand, 1987a), concentrating in four areas. One of these is reasoning and uncertainty measures, but this falls outside the scope of this thesis. The others were practical system areas, and each of these will be introduced in turn.

2.3.3 Intelligent front-ends.

A front-end can be defined as a piece of software which complements an existing system by providing some sort of guidance in the use of the system. The result is that hopefully the user learns about the limitations of the system. At the same time, the advice given helps the user to improve his own techniques of analysis.

One of the major projects in this area over the last few years has been on the GLIMPSE system (Wolstenholme et al, 1988). GLIMPSE (GLIM + Prolog + Statistical Expertise) is a language to help in the selection and programming of tasks within GLIM (Generalised Interactive Linear Modelling, the system developed by National Algorithms Laboratory). The front-end also acts as a tutor in the correct use of the system. It takes the form of a semi-conversational language, with the level of help and understanding set by the user. The language is similar in structure and form to Prolog or ALGOL, which means that the language is easy to learn and use.

Another front end is the REX system (Pregibon & Gale, 1984). REX is a Lisp based program that acts as a front end to S, a specialised statistical analysis language. The system can therefore also act as a tutor and advisor on the use of S for statistical analysis.

2.3.4 Advisory systems.

One of the jobs that a statistical consultant undertakes is to advise on which test or procedure to use in some situation. It is therefore only natural that expert systems which advise on tests have become more common. Professor D.J. Hand has become one of the most respected researchers in this area. His particular interest is in systems to educate a user in the structure of various tests.

However, he does not use the terms 'expert system' or 'knowledge based system', but prefers to use the term Knowledge Enhancement System (Hand, 1987b). One example of such a system is KENS, a Knowledge Enhancement system for Non-parametric Statistics which uses a frame-based structure. The user enters a statement of the situation he wishes advice on, and KENS uses a word-matching algorithm to move through the various frames. In the course of using the system, the user is challenged to think about their situation, and what the best representation of it would be. The most appropriate test for their situation is selected, along with a list of other possible options.

Advisory systems need not have this structure to work effectively. EXPERIMENTAL DESIGN, one of the newer experimental design systems developed in America, uses a series of questions to determine the suitability of a particular design model. As such, it falls under the category of an advisory system, but works on a production rule basis. In chapter 5, a system for checking the assumptions made in setting up an experiment and a design model will be considered. ExpCheck (Williams, 1988) is also an advisory system, based upon the checklist for experimental design written by J.N.R. Jeffers (1977).

2.3.5 Procedure construction systems.

One use for statistical expert systems is to formulate a plan of statistical analysis for a problem. REX (Gale & Pregibon, 1984) was one of the first to attempt this, by helping the user to construct a strategy for analysing a regression problem. However, research over recent years has produced more viable systems. For example, GLIMPSE can be used to set up a sequence of commands to tackle the analysis of a particular problem. The result is a planned analysis of the problem along suitable lines.

Considering statistics as a subject, one area in which a system of this type would be of greatest benefit is experimental design. Hahn and Morgan (1988) list a number of commercially available systems for help in experimental design construction. One of the major projects undertaken in the UK as part of the ALVEY program was SELINA, a system to construct a range of designs. SELINA was developed by a team based at Leeds University and working jointly with Shell International. The full system constructs and analyses a simple design, usually factorial. However, it is capable of setting up completely randomised or randomised block designs. The difficulty with SELINA is the size and type of computer needed to run the system. It can also be slow to work, and requires tuition in its use.

A number of other, smaller systems have been developed specifically for factorial designs. Dr A A Greenfield has developed DEX, a system for microcomputers that constructs a fractional factorial design (Jones, 1991). His system is unique in that there is no operating manual for it. Instead, a sheet of instructions is given, and a comprehensive help facility is supplied for use as required. This allows for ease of use and access to simple, comprehensive help services. A similar system for constructing complex factorial designs is CAED, developed for ICI by P. Goldsmith (1987). This uses a specialised language peculiar to the context of chemical production. However, within its own limited field it is useful as an advisory tool.

Having outlined some of the research done over the last few years, the methods developed as part of this research can be introduced. In the next chapter, some guidelines for constructing these systems based upon consultancy papers are given, and their possible use justified.

SECTION II

BACKGROUND TO CONSULTANCY PROBLEMS

CHAPTER 3

MODELLING CONSULTANCY PROCEDURES

3.1 INTRODUCTION.

A statistical expert system can be called upon to do a number of things. Chief among these is the analysis and solution of 'real world' problems. This also encompasses the need to communicate statistical ideas in a simple manner. The solution to these problems have to be reported in a manner that is understandable to the person using the system. In this way, they reflect some of the skills required of a statistical consultant.

In this chapter, the way in which statistical consultancy takes place is examined. The result is a number of proposals for statistical expert systems, with the aim of improving their style and content. Although people may disagree with some of these, they reflect current thinking in a number of related areas.

3.2 THE SYSTEM AS A CONSULTANT.

How do we define the role of a statistical consultant? Eisenhart (1947), Greenfield (1979), and others have considered this question in some depth. Most identify two aspects as being of importance. Firstly they attempt to help a person form a clearer understanding of the problem they have, and of the possible solutions. Secondly they provide a concise solution that meets the client's needs. Expert systems have similar goals of advising and teaching people about a particular area. An examination of statistical consultancy techniques should therefore provide some pointers as to how a general statistical expert system should work.

A major problem occurs when considering an examination of the way a consultant works. As was mentioned in Chapter 2, the traditional method of obtaining models for an expert system involves long sessions formulating and checking rules and procedures with experts in a field. The problem with statistical consultancy is no two statisticians will agree on the best approach to a problem. This is inevitable given the wide areas of application and the different forms of training the experts have had. Neiss et al (1985) illustrate this dilemma by asking a number of experts how they would tackle a problem, and discussing the approaches suggested.

Another possibility is the recording of actual sessions and subsequent examination of the records. Smith (1986) describes the way in which video recordings may be used in the teaching of consultancy. The difficulty here is the question of client-consultant confidentiality. The problem being discussed may be of a sensitive nature, and many people feel uncomfortable about been recorded in any form. Studying recordings may therefore not be a realistic alternative.

An alternative approach to modelling consultancy is however available. A rich and varied selection of papers on various aspects of consultancy have appeared over the years. Examination of these could provide some proposals for statistical expert systems. Baskerville (1981) suggested a number of categories into which these consultancy papers fall. These include:-

a) Training of statistical consultants

b) The Client-Consultant Interaction

c) Model finding and Problem Solving

d) Principles and Ethics

e) Organisational Considerations.

3.3 MODELLING STATISTICAL CONSULTANCY.

The first thing to consider is if these papers lead to any ideas for the structure of a system. Figure 2.2 gives a very general picture, but only shows the basic parts of a single system. As the following models show, statistical consultancy is slightly more complex. A consideration of these models can lead to a set of proposals for the overall structure of a statistical expert system.

3.3.1 The Models.

Marquardt (1979) listed a number of aspects of effective statistical consulting, which form a model of the relationship between a consultant and his client. She suggested the following as important points.

(a) Regular meetings with the client.

(b) Obtain a basic knowledge of the main subject as well as an understanding of any technical language used.

(c) View the problem being discussed from the position of the client.

(d) Ensure that a plan of action is agreed on by everyone.

(e) Monitor any experiments and be prepared to refine the plan if appropriate.

(f) Analyse any intermediate results in a simple and straightforward way.

(g) Give an intermediate verbal report of the results, to be followed by a full written version.

(h) Ensure that any proposed changes to the plan of action are made and checked on.

Box (1976) identifies four stages to statistical consultancy. His proposals stem from the idea that the consultancy should focus upon the problem and not the solution.

(a) Use probing questions to obtain a full understanding of the client's actual problem.

(b) Define the goals of the experiment.

(c) Formulate a clear statement of the problem as well as the proposed solution.

(d) Bring together the work that has been done.

Hunter (1981b) compares the art of consultancy to that of solving a mathematical problem, involving a number of conversion and solution stages.

(a) Obtain a clear statement of the client's perception of the problem at the start.

(b) Restate the problem in a way that is acceptable to all.

(c) Translate the problem into a mathematical one.

(d) Obtain a solution to the mathematical problem.

(e) Prepare an explanation of the solution and of the actual problem.

Finally, Zahn and Isenberg (1983) identify the major stages of a long-term consultancy. Despite being similar to the Box model, these highlight different aspects of the consultancy situation.

(a) Identify the relative aspects of the problem to statistics.

(b) Clearly define the client's goals.

(c) Determine in what order any actions are to be taken.

(d) Discuss the critical aspects of the problem, and clearly state each person's responsibilities.

From these models, it is possible to identify a number of common features. These lead to proposals for a number of aspects of statistical expert systems, with particular relevance to the structure of the systems.

3.3.2 Proposals for overall model.

It should be realised that a person constructing this type of model has identified a number of important stages. The first stage of Marquardt's model is regular meetings between the clients. This forms an essential part of any long term consultation. Although a number of problems may occur, not all of them need to be handled in one session. Put another way, Boen and Zahn (1982) state that 'there is no such thing as a five minute consultation'. This can be reflected in a system by having a number of sub-systems, each of which is responsible for one aspect of the problem. This is known as a modular system. The first proposal for a statistical expert system encompasses this point.

PROPOSAL 3.1

For a statistical expert system to effectively model the way in which a consultant works, a modular system would be most suitable.

There are immediate implications of proposing a modular structure. Separate sets of rules will be required for each module, as well as some sort of communication of results between them. This can make the job of modelling a little simpler. A modular structure would divide the problems into areas to tackle separately. This also allows for the reporting at each stage of the main deductions. In this way the reporting stages of the models are catered for within a system.

If a modular structure is required, how do we divide the consultancy method into various stages? The actual structure of the systems will depend upon the class of problems that the system is dealing with. A number of stages can be identified as common to all the models stated above. These are encapsulated in the following proposal.

PROPOSAL 3.2

The following modules would be required by all systems adopting a modular structure:-

a) A preliminary module to help a user to explain his problem, and ensure that the questions asked are relevant to the problem.

b) A selection module to pick a test or method of analysis that meets the requirements of the user.

c) A report module to give as clear a summary of the results as possible.

Other modules may be added to meet the needs of the class of problem the system is advising on.

The modular structure has other advantages. By splitting the problem into several stages, it also becomes possible to emulate the multi-session method of consultancy required for inter-disciplinary work. The facts and deductions used at each stage can be stored, and all other sections requiring them can easily gain access.

3.3.3 General Guidelines to Statistical Practice.

All the stated models place emphasis on acquaintance with the background to the problem been considered, as well as the subject area. The question of how to obtain this knowledge has been considered in a number of papers. For example, Greenfield (1979) states that 'The mathematical or statistical consultant must have a wide scientific and technical education if he is to succeed in industrial research'. Other papers which have stated ideas on this area include Greenfield (1983) and Hunter (1981a). Chapter 4 considers this area in more detail.

A number of other guidelines for good statistical practice can be obtained. For example, an important part of consultancy is the way that a relationship is built up between the client and the consultant. Daniel (1969) contends that for an effective consultation the problem must be clearly stated, the client willing to work and the consultant well prepared. Social interaction also plays an important part in helping the client to discuss the problem. J.R. Boen has written extensively on the application of psychological interviewing techniques to statistical consulting (Boen & Fryd, 1978). Obviously, a computer cannot be expected to handle the complexities

of human social behaviour. However, the phrasing of questions in a suitable language would be desirable. The following proposal allows for this.

PROPOSAL 3.3

The style of the questions asked must be sufficiently simple that the client can understand what is being asked. In addition, the style of answers should be clear enough to ensure that the required information is obtained.

The presentation of results must also be considered. Chanter (1978) states that the way the consultant expresses his ideas needs to be clear and unambiguous. The situation to avoid is that which leads to the feelings reflected in the following quote .

> 'Sir Alexander Fleming didn't have the benefit of modern instrumentation, a dishwasher and a statistician to tell him what he had found. The latter, of course, could only tell him if the results were 'Significant'. I suspect Sir Alexander knew that already, don't you?'
>
> (*Science*, May 3rd, 1963)

It is necessary to plan ahead to stop this occurring. The following proposal for the reporting of results deals with this.

PROPOSAL 3.4

The way in which the system presents the results of any conclusions and analysis must be such that it is understandable to the user. In addition, it should also reflect his intelligence and experience.

Another aspect of consultancy is asking non-statistical questions about the problem in order to gain a greater understanding. Greenfield (1979) and Hunter (1981b) both note the fact that non-numeric data and facts have to be considered. The following proposal is put forward to allow for the handling of non-numerical facts.

PROPOSAL 3.5

The structure of the rule base should take into account the fact that most of the data available at any time may be non-numeric.

3.4 CONSIDERING CONSULTANCY STEREOTYPES

Every consultant has a story to tell about a situation or a client that has caused them problems. Some of the more entertaining papers in the consultancy literature have related these tales. Among these, the ones by Hunter (1981b), Boen and Zahn (1982), Box (1979), Greenfield (1979) and Greenfield in Chapter 2 of Hand and Everitt (1987) are all recommended for their insight into this area.

However, for the purposes of obtaining proposals for statistical expert systems, attention will be focused on two of the best known papers on this subject. Both of these list stereotypes of the clients and consultants and propose ways of avoiding the difficulties they cause. These can be adapted to form useful proposals for statistical expert systems.

3.4.1 The Consultant.

Leon Hyams (1971) outlined some of the problems he had experienced in his field of interest. He included a list of 'bad consultants' in his paper, to illustrate some of the common misconceptions held by statisticians. These were as follows.

(a) The Model Builder. He tries to make all situations fit the models he knows about, even if they are unsuitable for the data.

(b) The Hunter. Applies every test that they can think of to a set of data, no matter what the cost is in terms of resources and time.

(c) The Gong. Assumes that all data can be fitted to a normal distribution, and attempts to do just that.

(d) The Traditionalist. Their knowledge stops with the techniques of R.A. Fisher, and nothing else matters.

(e) The Randomphiliac. Randomisation is all that matters.

(f) The Quantophiliac. Doggedly seeks the 'p-value' and the 'F-statistic'.

(g) The More Data Yeller. No further explanation is required.

(h) The Nit Picker. They are never satisfied until just one more test is done.

These can also represent possible pitfalls with intelligent statistical programs. For example, The Hunter is a program that applies a stock set of tests to every set of data. Establishing the background to an experiment early enough can eliminate this problem. Similarly, the Quantophiliac is a program that is only interested in giving an answer without explanation. Recognising the dual nature of an expert system as assistant and tutor helps here. If taken into account, then the programmer will ensure that explanation of conclusions is always available. If the level of background information can be determined, then the plausibility of various designs can be assessed.

But what of the other types? Consider first the Model Builder, the system that fits the model built into it no matter what the situation is. However, the following proposal provides a safety net for the above assumption.

PROPOSAL 3.6

The selection of a suitable model or test by the system should NOT be made until all relevant information has been obtained, and that information has been assessed by the system.

How about the Traditionalist - the program that cannot be updated because no facility has been built in for updating? Jones (1980) suggested that one way of dealing with this was to allow for regular updating and revising of the program. Another option is open to use in expert systems. This involves updating of the rule base to take account of any new information that appears.

PROPOSAL 3.7

The rule bases used by any or all of the system modules should be constructed so that any revisions, corrections and updating can be made with the minimum of fuss and disruption to the actual program itself.

The Randomphiliac can be very easily dealt with by making randomisation the last action to be undertaken. This assumes that it is necessary for randomisation to take place. Of Hyams' stereotypes, this only leaves the Hunter, the Nit Picker and the More Data Yeller to account for. All three are found in a system that is never satisfied with just doing one task, but needs extra information or asks for more data when it is not necessary. The following proposal is made as a possible solution to this problem.

PROPOSAL 3.8

Use the knowledge of the background to the experiment, as well as the needs expressed by the client, to control the flow of questions that the system asks.

Ways of implementing this idea are discussed in chapter 5, where the use of causal networks for questioning is discussed.

3.4.2 The Client.

The above proposals will deal with some of the problems that could occur in a system acting as a consultant. But what about the people who will be using this system? Hyams also offers some thoughts on these people, listing the following stereotypes.

(a) The Probabilist. All he wants is a 'p-value' to place in his paper.

(b) The Numbers Collector. He walks into the office with large sets of data, and expects the consultant to make sense of it or 'see if there is anything in there'.

(c) The Sporadic Leech. Asks a 'quick question' at the most awkward moment, and their question rarely has a quick answer.

(d) The Amateur Statistician. He knows what the best thing for his data is, and you must do it that way.

(e) The Long Distance Runner. Rarely if ever appears himself, but sends an assistant (usually ill-informed) to talk to the consultant for him. Why is he running? To quote Hyams:-

'He has excellent reasons. Two steps behind him, scrambling as fast and furiously as he is - arms outstretched, threatening to engulf and pull him down, down, down into a pit, are a troupe of horrible demons, his Mistakes!' (Hyams, 1971, p.205)

Complimentary to these stereotypes are those suggested by Sprent (1970). His types of clients were as follows.

(a) The Timid Apologist. He knows little or no statistics, but realises he needs help even with trivial problems.

(b) The Significant Difference and Least Significant Difference Experts. Only wants a significant difference or result and not interested in explanations.

(c) The One Technique Amateur Statistician. Knows one technique well, and tries to apply it to every and all situations.

(d) The Believer in Sacred Texts or Computers. Will turn to a favorite book or program that has worked before for every problem he faces.

(e) Experimenters with Addled Statistical Ideas. Have a smattering of knowledge of statistics, but what they know is slightly inaccurate.

(f) The Non-Statistician Expert Data Handler. They have an instinctive knowledge of how to tackle the problems they have.

(g) The Statistically Informed Experimenter, a.k.a the dream client. They know the language of statistics and can grasp the techniques used.

It will be noticed that Sprent's groups form a natural progression within themselves of the level of knowledge of statistics that a client may possess. Jones (1980) made the point that most software of that day dealt with those in groups (a) and (g) of Sprent, while tending to ignore the needs of the middle groups. Sprent himself recommended gentle guidance for group (a), while trying to push groups (b)-(e) towards the other extreme. It will also be noticed that the groups identified by Hyams all display a lack of understanding of statistical jargon and complexities. For example, the Numbers Collector never seems to appreciate the problems involved in just coding and storing a vast data set on computer files.

With this in mind, the construction of the rule base has to take into account the dual nature of expert systems of assistant and tutor as far as possible. The following proposed guidelines are given to allow for this.

PROPOSAL 3.9

When constructing a rule base or model for a statistical expert system, keep in mind the following points:-

a) The aim of such a system is to accurately meet the client's needs in as simple a way as possible.

b) Ensure that the explanation facilities are the finest that you can possibly create.

c) Aim to help the client understand the needs of the problem, and your view of the situation.

The difficulties that might be inherent in this proposal soon disappear, because the whole procedure is divided between a number of modules as suggested in proposal 3.1. In addition, it is worth remembering that no problem is so difficult that there isn't someone elsewhere who knows about it. This leads to the next proposal, one based on a recognition of the limitations of these systems.

PROPOSAL 3.10

If it becomes apparent that the system cannot handle a particular problem, then make it possible for the system to admit that it cannot do it.

One final idea can be learned from considering the stereotypes of clients and consultants. It concerns people like the Long-Distance Runner, the Believer in Sacred Texts or Computers, and the Sporadic Leech. If a system is been proposed that will advise on the problems in a particular area, then there will be a great temptation to just use the systems and dispose of the statisticians. However, that path is fraught with danger, not least because it assumes the system will always be right. It was made clear earlier that

these systems are not infallible. It is therefore advisable to have access to an expert statistician to consider any unusual results that these systems may produce.

3.5 THE DEMING PHILOSOPHY.

Finally, a few words need to be said about the attempts to produce a code of conduct for statistical consultants. Among the writings on this area are the appendix in the book by Boen & Zahn (1982), and the report of the A.S.A. Committee on Code of Conduct (1980). One of the most fervent speakers on this area is W.E. Deming. In three major papers (Deming, 1965, 1966, 1972), he outlined the conditions set down in any contract for work he undertook.

The contracts set out in these papers are watertight, and clearly outline the various tasks to be undertaken and who should do them. Some may say they are too rigid in their structure. They effectively remove the temptation to investigate any ideas which may arise as a result of the analysis. Having said that, they do give a clear plan of action and form a useful model in their own right for statistical expert systems. Among the points he raises are the need for experience in order to understand how to plan an experiment (Deming, 1966). The following proposals are based on two ideas from his 1972 paper.

<u>PROPOSAL 3.11</u>

The system and rule base must be constructed so that it is made clear that all deductions are only useful within the context of the experiment.

PROPOSAL 3.12

The system must have an effective security system and coding of stored results, so that the confidentiality of the consultant-client interaction can be maintained.

3.6 CONCLUSIONS.

This chapter contains a number of proposals to consider in the construction of statistical expert systems. The majority of them were used in the work that appears in the remainder of this thesis. Although there are aspects of them that people may disagree with, the results in the following chapters show that they form a firm basis for work to begin on.

CHAPTER 4

DETERMINING THE BACKGROUND SUBJECT.

4.1 THE IMPORTANCE OF BEING INFORMED.

4.1.1 A Cautionary Tale.

Hunter (1981b) tells the story of a recent recipient of a doctorate. Starting work at a certain chemical plant, he was given a large set of data and told to 'find something interesting'. After some time, he produced an impressive report. The centerpiece was a list of variables in descending order of importance determined by significance of the regression coefficients for a model of output. By his method of analysis, water appeared to have no significant effect on the output, and therefore costly monitoring of the water level could cease.

The result when he stated this in a seminar was unrestrained laughter. In fact, if any significant amount of water entered the process, the plant stood a very good chance of disappearing from the face of the earth. It was because the amount of water present was so tightly controlled that it appeared to have no effect on the process. The moral of the tale was 'Learn as much as you can about the general subject and the specific environment in which the data was collected.'

Hunter is not the only writer who emphasises the importance of background information in statistical consulting. Marquardt (1979) also states that a passing knowledge of other subject areas is useful. To quote her, ' Learn the client's subject matter and lingo [because]...it helps you to be sure you understand the real problem, and it helps you to communicate with your

clients....You should...read selected reports, journals, or books in the client's subject field.' Greenfield has also championed the need for a wide range of knowledge in statistical consulting. In one paper (Greenfield, 1979), he states that 'The mathematical or statistical consultant must have a wide scientific and technical education if he is to succeed...'. Indeed, Greenfield's proposed degree for the Polymath Consultant (Greenfield, 1983) included courses in general science, office management and report skills.

The important conclusion from these writings is that there is a need to establish the background to an client's problem. If the subject area of interest can be established, then mistakes of the type illustrated in the tale can be avoided. In this chapter, a method for establishing the background to a problem is discussed. Some results of the method are also introduced.

4.1.2 What kind of Background?

A key question to be asked is what exactly the background information is required for. One aim of a consultation system should be to select and possibly construct one of a list of tests for use. Determining the background subject will help in the efficient selection of the appropriate tests. A medical doctor will only decide on a course of treatment after a careful weighing of all the available evidence. Similarly, statistical consultants should not attempt a particular solution to a problem until all the available facts have been collected and considered. When the background is established a lot of time can be saved by removing from consideration any unsuitable methods.

A slightly less obvious purpose, but just as important, is to help the system determine some estimate of the initial plausibility of the various models. It is important that initial values are obtained for each model or procedure. In addition, the way the background is presented will help to determine how plausible a particular method is. The result is that determining how well the background is defined affects the set of plausibilities for the various options.

4.1.3 Just what is the problem?

What needs to be established is a suitable method for finding out the area of interest to the client. It may be obvious that the client is a chemist, a pharmacist or a physicist, but these are terms that cover a wide range of often complex specialties. In order to provide an effective solution to their particular problem, it is important to know exactly what their area of specialty is.

Establishing exactly what that area of interest is requires skillful questioning. As a consequence, the role of the statistical consultant has to include questioning the client's assumptions of the actual problem area. If this does not take place, then the consultant soon becomes nothing more than the scientific equivalent of a shoe salesman (Bross, 1974).

This does not mean that the computer can completely replace the consultant. A computer is a machine, and as such it cannot replace the visual aspects of the job or react to chance comments. To overcome this difficulty, it is necessary to shift the emphasis of the problem. Rather than trying to understand the client's exact problem, we need to help the

client to consider his problem against other similar ones. By helping him to recognise other problems in a related vein, we can establish the subject area of interest.

A method of questioning the client as to what problems are similar to his own would give just such a guide to the actual problem. In this chapter, a possible method for undertaking this task is introduced. For reasons that will become clear as the method is introduced, the results are referred to as a 'Tree of Knowledge.'

4.2 THE 'TREE OF KNOWLEDGE'.

4.2.1 Some Definitions.

The structure of the Tree of Knowledge comes from a consideration of semantic nets. As was discussed in section 2.2.3, these consist of a series of facts and connections between these facts. Suppose the facts were descriptions of various problems that had been encountered. If the facts were progressively more detailed and exact, then the connectors could be regarded as relationships between the facts. It is on this basis that the method proposed is developed.

In order to explain the type of work that a person is currently undertaking, they will attempt to use terms which are understandable to others. The terminology used will depend not only upon the client's depth of understanding, but also upon the level of understanding of the person he is speaking to. The speaker may choose to use precise language, or else to use analogies for work which closely match their own.

Now suppose a statistical consultant wishes to determine the subject area of interest to a client. Obviously, he will ask questions of the client in order to determine the exact problem. Therefore, it follows that a client's understanding of his problem can be ascertained by asking him to describe the work in terms the consultant will understand.

If the client uses imprecise terms, then it is probably safe to assume that they are not too sure about the problem in statistical terms, and need some help in codifying their ideas. However, if they can relate their work to some similar area, then the range of possible solutions can be drastically reduced. In addition, the clearer a definition of the problem a client can give, then the easier it will be to understand his exact needs. The method introduced here uses a semantic net to elicit just such a description, and can be represented on paper by a form of connected graph. The following definition states what the tree is composed of.

DEFINITION 4.1

A 'Tree of Knowledge' is a semantic net or connected graph, the nodes of which are phrases or descriptions of subject areas or problems, and the paths the connections between these various descriptions.

4.2.2 Constructing a tree.

There is an obvious question to ask at this point. What can be used as the nodes, i.e. how detailed a set of descriptions do we have at each stage? There needs to be a clear definition of what exactly each node consists of. The following states clearly what is required for a node in the Tree of Knowledge.

DEFINITION 4.2

A Branching Point in the Tree of Knowledge is a description of some problems, within which several more detailed descriptions can be grouped. This forms a node on the graph representing the tree, with connectors to the various more detailed descriptions within the group. The connectors then form the branches of the tree.

As would be expected, the main branches of the tree would cover the main subject areas, such as Biology, Agriculture, Industrial and so on. From this stage, the person building the system has to decide the branching points for each area. This would be a vast job for one person to undertake. However, there is no stipulation that the same person has to develop the whole tree. One person could build a tree for Agriculture, another for Industry and so on. In this way, the tree can represent knowledge from a number of experts in different fields.

But how would a person go about building a tree? Consider for a moment the subject area of Agriculture. Within this area there are a number of possible major subjects, such as produce management and pest control. The expert would select a number of major subjects, and use these as the branching points from Agriculture. He would then proceed to consider each of these groups in turn and find branching points within those groups.

For example, within produce management can be included such topics as crop yields, effects of fertilisers and milk yields. These then form the branching points for the group denoted by Produce Management. Considering milk yields, the expert may know of a number of experiments in that field. These would be the branching points for that group, and may form the final stages of the tree. It is of course possible to provide even more detailed

descriptions. The depth of detail which the descriptions provide is a question for the expert himself to decide. Similar decisions can be made for all the possible subgroups, resulting in something like the following diagram.

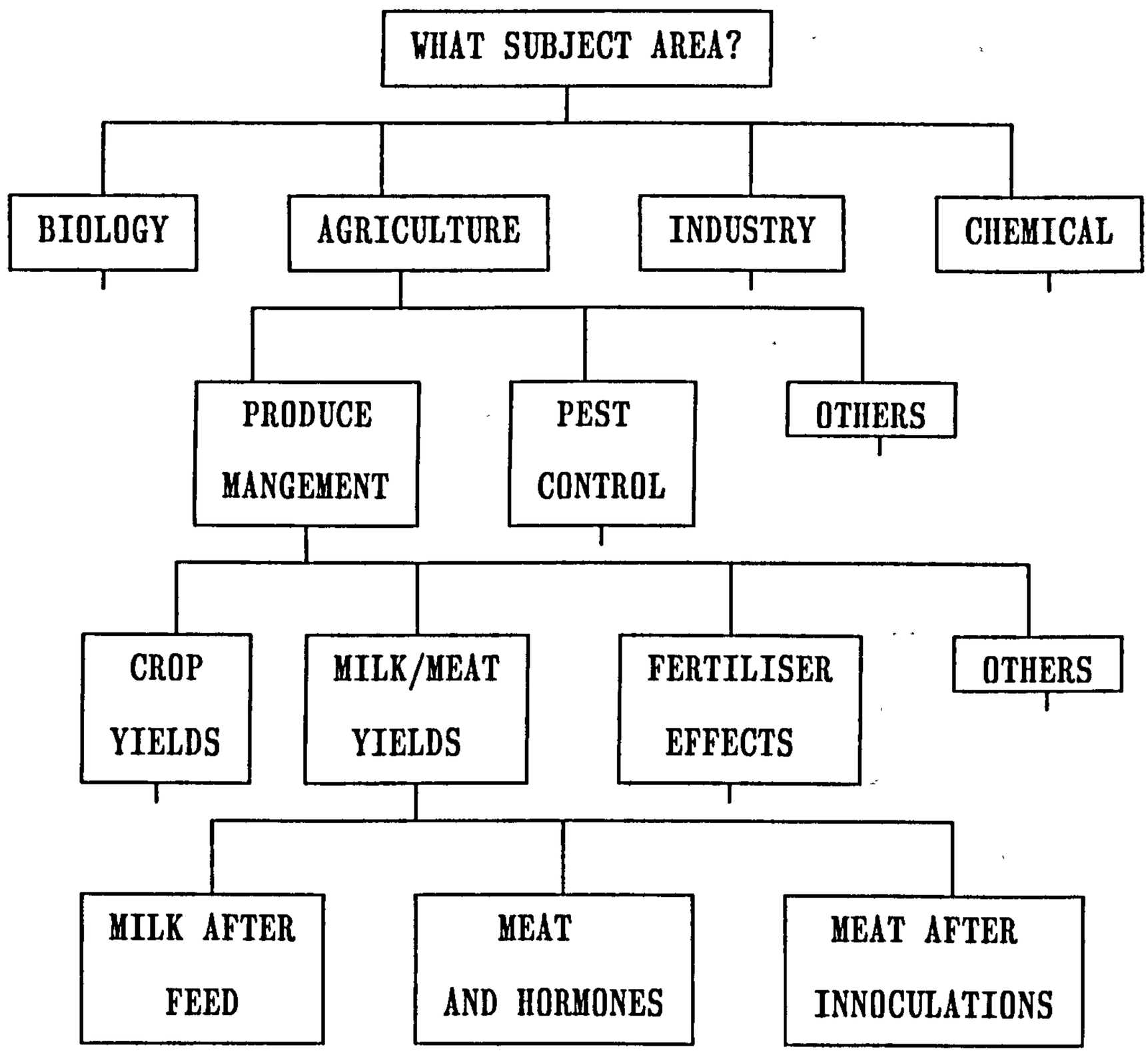


FIGURE 4.1 'TREE OF KNOWLEDGE' EXAMPLE

4.2.3 Using the tree.

Once the tree has been constructed, then it is available for use in order to determine the background subject of the experiment. Starting at the root, the user would be asked to specify which of a list of options best describes his area of interest. The list of choices at each point correspond to the

set of nodes which branch off from that particular point on the tree. Once the user has made his decision, then the system can advance down one set of branches and ask to see a more detailed set of examples.

The method of use is deliberately simple. At each node, a list of the descriptions covered by that area can be presented. For example, the first node would simply give a list consisting of Agriculture, Biology and so on. At each stage the client is asked to select the description or subject into which his problem fits most clearly. The system then follows the path selected to the next node, and the next set of descriptions presented. Once a suitable description for the client has been found, then the user can obtain a fuller description and move on to consider other aspects of the problem. The method of programming the tree will be considered later in this chapter.

When a suitable description has been selected, then the system can use that node as a starting block from which to discuss the background to the experiment. Should the tree be unable to provide a description that matches closely enough the wishes of the client, then they can be invited to provide a description of their own experiment. In that way, new experiments can be added to the tree and be used for future comparisons.

4.3 DEVELOPMENTS OF THE TREE

Obtaining a suitable description of the problem been considered is the primary function of this technique. In addition, a little thought suggests two added benefits from the structure. Both could be of value at later stages of the system, and would also help in overall system management. These benefits are given here as proposals, together with an argument for their use.

4.3.1 Depth of knowledge.

As was stated earlier, the type of language that the client uses in describing his problem reflects how much he understands the background area himself. Consequently, if he can find a description that almost exactly matches the problem, it shows that he has a firm understanding of the problem. However, if he has to use very general terms to describe his experiment, then it is likely that he does not fully understand the dynamics of his particular problem. Using this argument as a basis, the following proposal can be made.

PROPOSAL 4.1

By seeing how far the client has to progress down the tree to find a suitable description, the system is able to see how much a client understands the background to his problem.

The implication of this is that the nearer the root of the tree a person stops, the more general the assumptions that have to be made. In other words, the more questioning that may be required later to get details of the problem. If a person gets as far as selecting a very small group of problems, (i.e. gets to the end of the tree), then other benefits occur. The selected option acts as a detailed description of the type of problem being investigated. When more details of the problem emerge, they can be checked against the description to see if the facts match or not. The result is a further check on the accuracy of any assumptions made by the person using the system.

4.3.2 Initial Plausibility Values

If the tree provides a check against assumptions made on the problem, then it also provides a guide as to the suitability of the solutions. Working from this assumption, the following proposal can be made.

PROPOSAL 4.2

The initial plausibility of the available models may be determined by the depth to which the tree has been traced.

If the subject area is very general, then any number of models for solutions of a particular problem will need to be considered. However, if the subject area is known in some detail, then the number of possible solutions will be reduced. The consequence of this is that the plausibility of each solution will be known to a greater accuracy. Therefore, a measure of the plausibility for all the models at each branching point of the tree is required. The obvious question to ask at this point is how these initial values may be determined. Two possible methods are available for consideration.

Firstly, the person constructing the tree can scour the published practical and applied papers for occurrences of these models in similar situations. Empirical initial values based upon this research can then be calculated and stored. However, this approach could be far too time-consuming and wasteful of resources. A more reasonable if less accurate method is to use any experts who have designed the sub-trees for the main subject areas. They could assign values to the models for each branching point in their own sub-tree. In this way, more of the expert knowledge that they can provide is encoded in the tree structure. These values would also show how often they perceived the models were used in their main subject area.

A modification to this method would involve asking one or two other experts in that subject area to contribute their own values, based upon the branching points suggested by the main expert. When these are combined in some suitable way, say by taking a weighted average or probabilistic combination, then the resulting values can be used as the initial markers.

4.4 PROGRAMMING OF THE TREE.

Finally, the method of programming that can be used for the tree needs to be discussed. Remember that the tree is constructed by a type of branching algorithm. The branching points represent a description of the background problem to a certain degree of accuracy. What is required is a programming structure which will reflect that concept. As was stated earlier, the logical choice for this is a semantic network, where the nodes are the descriptions and the paths the connections between descriptions. If this approach is adopted, then each node will have connected with it three pieces of information.

Firstly, each node has a set of descriptions of the problem attached to it, reflecting the depth that the node is located at. These descriptions can be stored in a component file, with a unique reference to each description. The problem of passing down the tree then becomes a question of knowing which references are connected. These connections can be stored as a matrix, each row representing one node and stating the possible descriptions that follow on from that one. The vector of numbers corresponding to that row of the matrix would form the second piece of information.

Finally, there are the initial plausibilities that have been attached to each node for the models. As has already been suggested, these can be stored in a separate component file. The reference numbers for this file can tally

with those for the other pieces of information. When a suitable description has been agreed on, the relevant vector can be automatically copied from the component file to the fact storage file, ready to be used when appropriate.

The method of presentation to the user would be as discussed earlier. The user would see a progression of menus, each one offering a set of possible descriptions of their problem. These brief descriptions can also be stored as a text matrix within a component file, or else they can be the first line of the fuller descriptions. These first lines may then be removed by the program, collected together and shown as the menu. The removal of the first line is not a difficult problem to undertake. For example if the system is prepared in APL a few lines of code would be sufficient to arrange for the availability of these summaries.

4.5 CONCLUSIONS.

This chapter has outlined a possible method for establishing the background to a consultancy problem. Other more traditional methods of questioning may be used to obtain the same facts. However, this method has advantages, both in terms of simplicity of use and also in the structure of the tree. By reflecting the way in which a consultant may seek to learn of a client's subject area, this method can provide a useful model of an expert's experience in a given field. In addition, the use of different experts to cover different subject areas allows for the combining of knowledge from a number of sources.

Having established the background subject, the next task of the consultant is to determine facts about the problem being considered. In the next chapter, methods of questioning the client in order to obtain these facts are considered and introduced.

CHAPTER 5

ELICITING THE FACTS OF THE PROBLEM

5.1 INTRODUCTION.

Once the general description of the problem has been established, then some people may assume that the system can start model selection. However, this overlooks the fact that very little may be known about the actual problem itself. All that the system knows at this stage is some idea of an area that relates in some way to the actual problem. While useful as a starting point for consideration, it tells nothing of the problem itself.

In order to clarify what the client's problem is, further questioning concerning the actual problem is required. Such an approach enables the statistician to spot potential difficulties. These may include a poorly defined purpose or insufficient data to tackle the questions the client is asking. In this chapter, the way in which questions of this type may be arranged and prepared is considered. Two examples are given which show the resultant ideas in action.

5.2 OBTAINING THE BACKGROUND FACTS.

5.2.1 Why bother with the facts?

Facts are important in any sort of investigation, as is testified to by the quote from Arthur Conan-Doyle at the beginning of the thesis. One of the more interesting points of the Sherlock Holmes stories is the way the main character searches for facts before beginning to form theories. This reflects just how important facts are to the solution of a problem. The

story at the start of Chapter 4 also illustrates how important facts about a problem are to the correct solution. It is therefore vitally important that a statistical advisory system has as a main priority the gathering of facts about a problem. Indeed, for an advisory program of any type to ignore necessary information is unjustifiable.

For a statistical expert system to ignore facts would be to undermine the very reason for these systems. The form these facts take can vary, but in the main they will be either results from tests or answers to questions given by the system. Numeric facts can be deduced as and when required. Textual facts need to have questions in order to obtain them. It is for this reason that questioning of the client's assumptions and expectations is vitally important.

In addition, there is another less obvious reason why the facts about the experiment should be obtained. The final results of the system will have to be related back to the user in an understandable manner. By obtaining this information at an early stage, the facts that are obtained can be used for more efficient and effective communication. Using this argument, the following proposal can be made as the starting point for fact elicitation.

PROPOSAL 5.1

Once the initial subject area is established, then a series of questions should be asked, in order to determine more fully the background to the experiment and the wishes of the client.

5.2.2 Which facts will be required?

The real problem with deciding to obtain facts by questioning is which questions to ask. This can be made a lot easier by first determining which facts are going to be of interest. As has already been stated, the primary aim of a statistical consultant is to try and understand what the problem the client has is. Another way of putting this is to determine a number of basic facts about the problem. Once this has been achieved, then the greater understanding of the problem will lead to a more acceptable solution. The following proposal puts this more formally.

PROPOSAL 5.2

Before deciding what questions are going to be asked, the facts that the questions are to obtain should be decided upon. What these facts are will depend upon what class of problem the system is advising upon.

As the proposal states, the facts needed will depend upon the type of problem being tackled. If the system is to advise on some new investigation, then some statement of the population of interest and the questions to be answered would be required. Alternatively a system to select a procedure for analysis would need to know the structure of the data set. It also needs to be understood that the facts to be obtained are only an initial set. Other facts may become apparent as more is learnt about the problem.

5.2.3 Question selection and ordering.

Once it has been established which facts are going to be required by the system, then the questions that need to be asked can be composed and arranged into a suitable order. In some cases the types of questions that can be asked have already being established by someone. For example, Jeffers (1977) prepared a checklist of questions for agricultural experimental designs. Goldsmith (1987) prepared a similar list for industrial experimentation. Where this has occurred, then the checklists themselves can form the basis of the questions.

However, the situation where the questions have never formally been written down is more likely. If this is the case, then some form of structure is required to guide the question writing. It is in this case that the facts which have been selected as important become useful. As these facts need to be obtained, the questions should be built around the purpose of doing just that. This idea leads to the formation of Key Questions, which can be defined as follows.

DEFINITION 5.1

A Key Question can be defined as one which is instrumental in obtaining one of the facts identified as necessary to the problem.

To build the set of questions up, the key questions are the ones that should be decided upon initially. From this basis, any additional questions to fill in smaller details can be derived.

Once the set of questions to be asked has been decided upon, the order in which they are asked needs to be settled. Ordering may seem like an unnecessary task, but in fact it can ease the problem of programming the system. In an earlier paper on the subject of a system for experimental designs, the following was written.

> 'The [possible] problems themselves need to be identified.....this is accomplished by asking questions.....A logical deduction from this is that problems themselves should be identifiable <u>from the questions and the order in which they are asked</u>.' (Williams, 1988, emphasis mine)

If an order to the questions can be established, then the following proposal shows how the structure of the questions can be reflected in the system.

PROPOSAL 5.3

Once the order of questioning has been established, then the structure can be captured within the program by use of a causal network, where the nodes are the questions and/or possible actions, and the paths show which question is to be asked next, dependent upon the answer to the current question.

At this stage, it should be noted that there are some disadvantages to this method of fact elicitation. For example, it presupposes that the structure applied to the questions is a reasonable one. It would be hoped that a badly structured set of questions can be quickly discovered. It also needs to be remembered that we are talking about emulating the way in which an experienced consultant works. Provided he is competent at his work, then the sorts of questions he asks will over the years have attained some sort of reasonable structure. The questions selected should reflect that structure.

5.2.4 Fact checking, validation and alteration.

Consider the way in which a barrister may examine a witness during a legal action. At various times in the discussion, he may ask the same question in a number of different ways. This can be done for a number of reasons, possibly to confirm that what a person said earlier was true, or to show that they were lying about what did happen. If the barrister is cross-examining, then they may ask a question used by the opposing barrister in a different way for much the same reasons.

In a consultancy system, then the ability to ask the same question in different ways will simplify the problem of fact verification. As the questions have a form of structure to them, the connections between similar questions can easily be established. These connections can then be employed to compare the answers for any major differences. This is especially important if only one person is answering the questions the system poses. For this reason that the following proposal should be implemented if at all possible.

PROPOSAL 5.4

In order to allow for cross-checking of facts, the list of questions to be asked should include a number which ask the same question in different ways.

The checking of the answers to these questions will allow the system to spot possible problems of interpretation and misunderstanding before they become irrevocable. Problems will occur because of confusion over facts. Spotting and amending them at an early stage will allow the system to work more efficiently. It also provides a valuable tuition service for anyone using the system.

This principle can also be reflected in other ways. The type of answer given by a user can also reflect their level of understanding of the question. It is perfectly conceivable that a question is answered incorrectly because the user does not understand the intent of the questioner. The answers given also reflect the depth of understanding of a user in relation to the problem itself. In this way the answers provide a useful check on the results of using a system like the Tree of Knowledge introduced in the previous chapter. In any case, once a fact has been established and validated it should not be amended. The following proposal states that formally.

PROPOSAL 5.5

Once a fact has been cross-checked and validated, then that fact should be regarded as fixed, and unalterable except to be updated by the system as required.

It is also necessary to consider what happens if a person wishes to amend their answers. For example, they wish to go over a particular set of questions again to ensure that their thinking is correct. An obvious danger is that the system breaks down because of continual amendment of answers. However, it may still be necessary to allow some limited form of backtracking. Then if the user realises that a mistake has been made, he can return and rectify it without too much bother. In addition, such a facility will encourage the learning process. By realising the consequences of a particular decision, the client can discover how various aspects of their problem are connected. Bearing this in mind, the following proposal is made.

PROPOSAL 5.6

Within reason, the ability to backtrack a LIMITED way through the causal network should be built into the system.

To illustrate how these proposals may be put into practice, two examples are presented here. The first uses the checklist of Jeffers (1977) to produce a system for validating proposed experimental designs. It also illustrates how some of the proposals in Chapter 3 can be put into use. These include the modular structure and the method for storing facts. The second example shows how a set of questions for selecting an experimental design model may be constructed. This also relates to later chapters, which present a method for rule elicitation for test selection.

5.3 FACT ELICITATION FOR ESTABLISHED QUESTIONS

5.3.1 ExpCheck.

ExpCheck (Williams, 1988) was designed to ask questions of a person about an experiment they were considering undertaking. The main reasons for this were firstly to investigate the motivation behind their experiment, and also to ensure they had considered all possible problems which may occur. In this way the facts about the experiment being considered could be checked and validated. However, as ExpCheck obtained the facts of an experiment, it also provided an example of how an existing set of questions on a subject could be used for fact elicitation and verification.

The system used the checklist developed by Jeffers (1977) as the basis for the questions asked. These covered a number of aspects of experimental design, grouped under a number of headings. In its original form, the checklist listed some points to consider in experimental design. However, the final question in the checklist went as follows.

> 'If you are in doubt about the purpose of any of the questions on this checklist, should you not obtain some advice from a statistician with experience in your field of research before continuing with this experiment?' (Jeffers, 1977, Question 71.)

Considering this question, it can be reasoned that the questions themselves give pointers to problems with a design. Another way of expressing this is to reword the question along the following lines.

> 'If a series of these questions has led you to believe there is a problem with your design, would it not be wise to check with a statistician who knows your field of research for a possible remedy?'

Using this rewording as a starting point for the development, ExpCheck was constructed using the ideas expressed in the first part of this chapter. The disk marked ExpCheck that accompanies this thesis contains a working version of this system.

5.3.2 The facts needed.

If the list of questions proposed by Jeffers are to be used, then the facts that can be established need to be determined. These facts will provide information about the experiment being considered. In particular some of the details of the experimental plan can be successfully deduced. After some consideration of the questions, the following facts may be obtained from the user.

a) A clear and concise expression of the objectives of the experiment.

b) An expression of these objectives as questions to be answered.

c) A description of the materials the experiment will be carried out on.

d) The size and form of the experimental units.

e) An estimate of the optimum number of replications.

It may also be possible to obtain a description of the treatments to be used in the experiment. This was not included in the original set of facts to be deduced, but needs to be considered as well. The reasons for the exclusion of treatment structure are uncertain, but any newer versions of the system that are developed would include that within it.

5.3.3 The structure of the questions.

Having established which facts need to be obtained, the questions themselves can be considered. The questions in the checklist itself were grouped together into ten categories, with the final one being the question

quoted earlier. As some of these categories contained only one or two questions, they were rearranged into the following eight sections.

a) Introductory questions - purpose of experiment, questions to answer.
b) Selection of treatments - factorial structure, applicability to the problem.
c) Size and form of experimental unit.
d) The optimum number of replications under certain criteria.
e) Experimental layout
f) Randomisation.
g) The recording and storage of results.
h) Planning of the analysis.

For each question, the action to be taken if the answer was yes or no was considered. In this way, a network of connections between the questions was built up. If the answer indicated a possible problem, then that problem was noted and the connection made. In this way, a causal network of the problem detection capabilities of the questionnaire was built up.

A number of problems were possibly detected within each section. It was decided at an early stage that if a problem was detected, the system should give a diagnosis and a possible solution before stopping. One result of this thinking was that it was likely that a person using ExpCheck would not reach the end of a section. Indeed, it was most unlikely that a client using the system for the first time would complete the entire checklist at the first attempt. For this reason, the modular structure for a system proposed in Chapter 3 was employed. Each set of questions was programmed in a separate module, with its own causal network. The connections to the previous and

next section were clearly established and noted. In this way a complete picture of the facts of the experiment and the connections between them can be prepared.

The way in which the modules were connected together is considered in a later section. The following diagram shows the network for the section that asks questions concerning randomisation for the experiment.

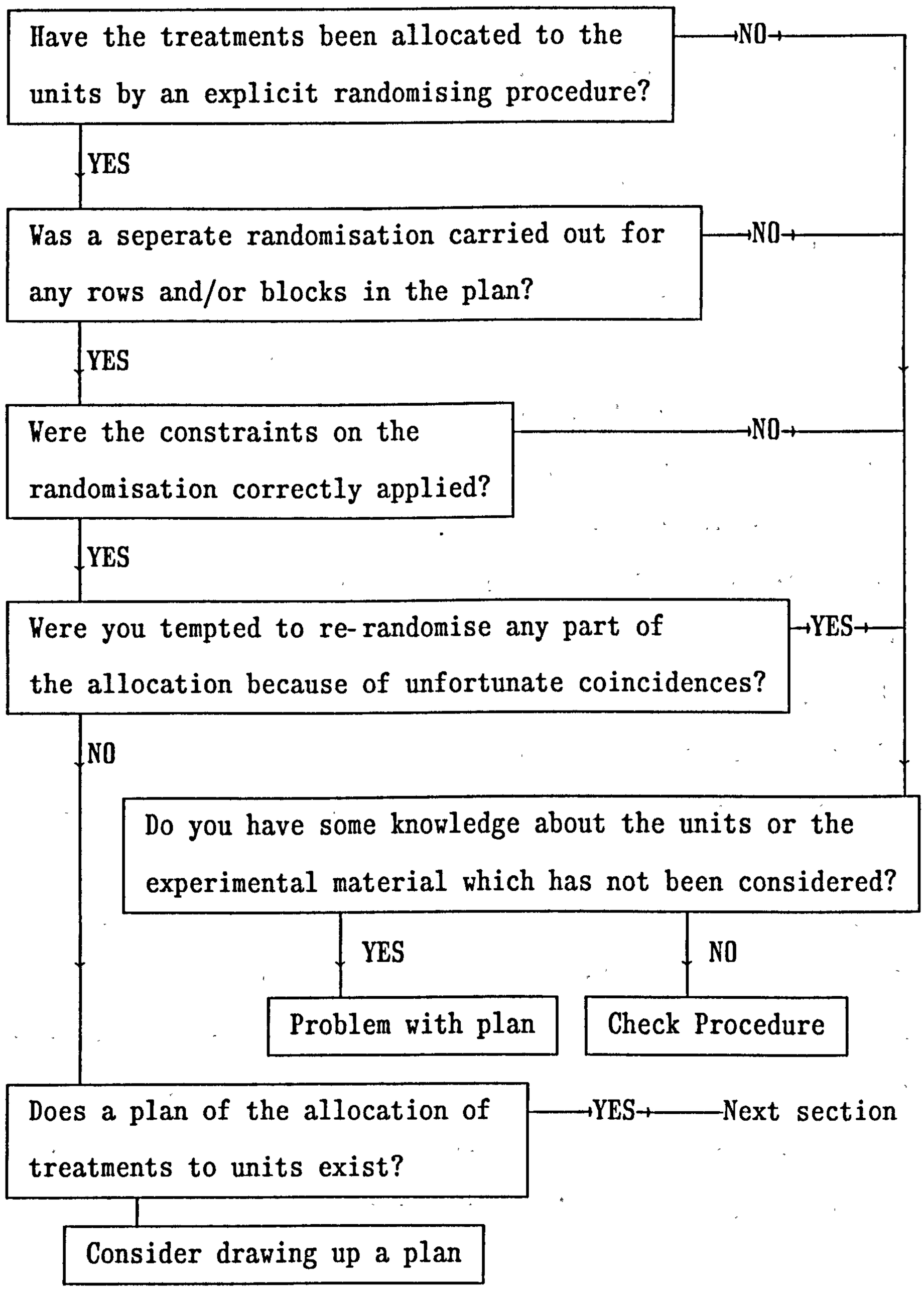


FIGURE 5.1 - CAUSAL NETWORK FOR SECTION 6 OF EXPCHECK

5.3.4 Help Facilities.

A number of help facilities and utilities were designed into the system. Firstly, a small dictionary of common terms used in experimental design was prepared. With each question asked it was possible to access this dictionary, and obtain the meaning of one or more of these terms. Should the words requested not match exactly any of the terms stored, a simple matching algorithm was used to select suitable alternatives.

The level of understanding of the questions by a client will vary greatly over the set of possible clients. As an aid to understanding, simple explanations of the questions are also available. These give the reasons for asking the questions, as well as any further considerations that may need to be taken into account. Another possible outcome of misunderstanding a question is that the system may go down a path that is incorrect, due to an inappropriate answer being given to a question. ExpCheck caters for this by allowing the client to backtrack a question and give a different answer to the previous one.

As each question is asked, a deduction of some type is made about the experiment. At the end of each set of questions, a list of the deductions made and facts about the design is given. In addition, a short report is made available at the end of each run. This gives the facts that have been obtained, the deductions and conclusions made and a list of the questions asked. All these details are stored in a file. It is possible to restart a previous run of ExpCheck at the last question to be asked, or at one of the client's choosing.

5.3.5 Programming details.

ExpCheck was programmed in APL, using a hierarchical structure incorporating the modular format of the system. A master workspace held all the necessary utilities and control programs. This in turn accessed a file containing eight functions, each of which in turn controlled the network of questions for that section. The questions for each section are stored in a file, and the control function asks each of these in turn. These questions also had access to other files containing the explanations and help facilities. The final structure is represented by the following diagram.

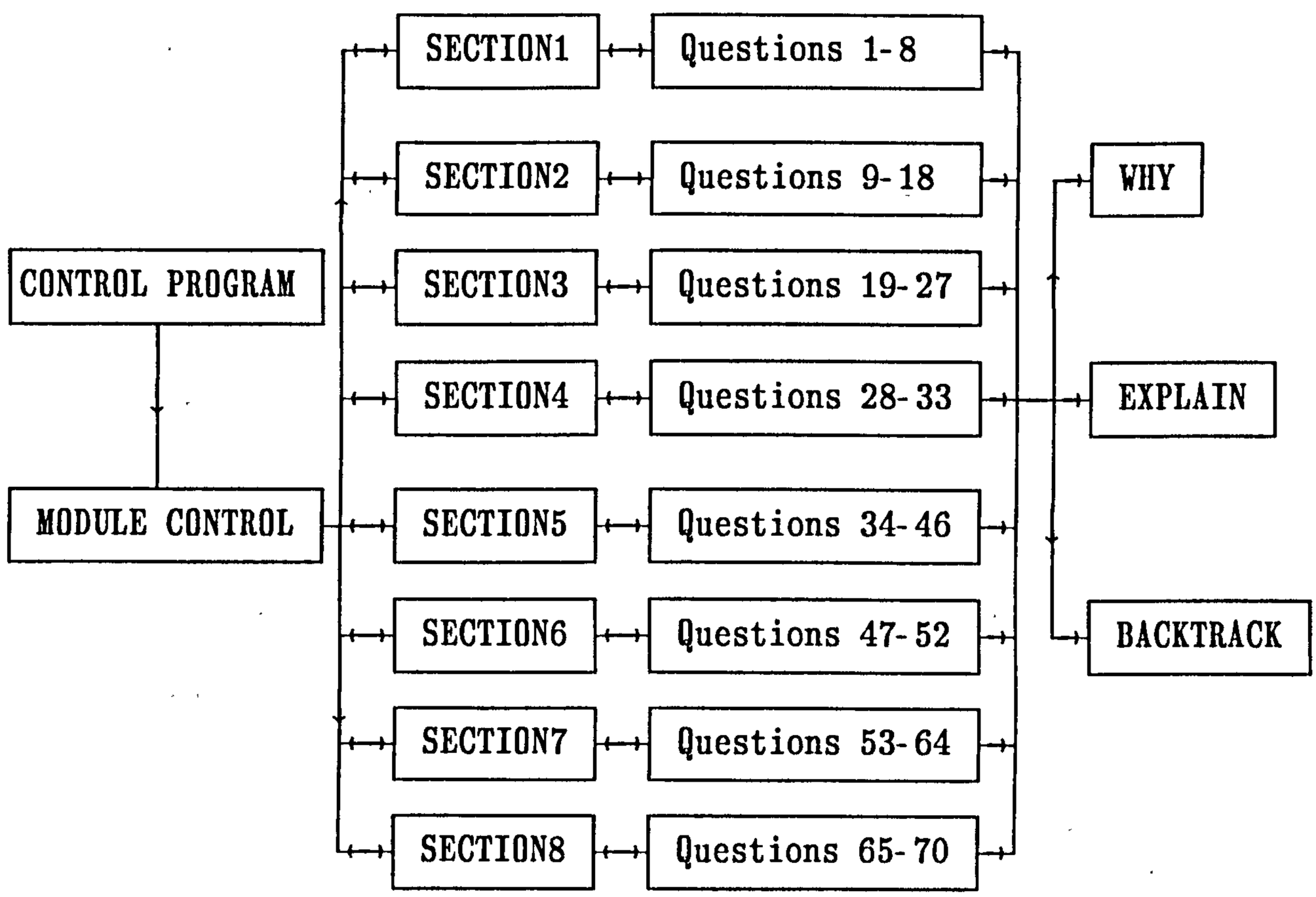


FIGURE 5.2 - STRUCTURE OF EXPCHECK SYSTEM.

The entire system can be placed on one 360K disk, and is available for use on most IBM compatible machines. The master workspace, that is the one containing all the utility and major functions, has been compiled into an executable file for security reasons, and the component files cannot be read as normal text files.

5.4 FORMING A NEW NETWORK OF QUESTIONS.

In the next section of the thesis, a method of rule elicitation is presented and illustrated using experimental design selection as an example. An alternative approach is possibly to construct a simple network of the same type as ExpCheck for model selection. The problem here is complicated by the fact that no obvious set of questions has been compiled for the selection of a design model. The closest to a set of questions appears at the beginning of John and Quenouille (1977). This presents some points to consider in model selection for a small group of designs. Working along the same lines, a network for model selection can be developed.

5.4.1 The Key Questions.

The first thing to be established are the key questions to ask when considering experimental design. The table referred to in John and Quenouille (1977) identified two aspects of an experiment to consider when selecting a model. The first was the structure of the treatments. If the structure was a factorial one, then the number of levels to each factor would virtually dictate the minimum number of experiments for analysis to be accurate. In addition, the combinations of factors that would be of interest may allow for a fractional design to be considered. On the other hand, if a factorial structure is not present then a whole group of designs need not be

considered. When this is taken into account, establishing the treatment structure is vitally important. Therefore the first key question to ask must concern the structure of the treatments.

The other key question to ask concerns the materials and area the experiment will be conducted in. If it is known that results will vary across the materials and/or area, then this must be allowed for. This is usually achieved by imposing some form of blocking structure on any experimental plan. The type of blocking used will depend very much on the number of different ways in which the results are known to be affected. The more ways there are, then the more complex the type of blocking required. For this reason, another key question to ask in model selection is the number of different sources of non-random variation. In this way the different influences on the results can be accounted for at the planning stage.

There are therefore two key questions to ask when selecting a model for an experimental design. The next stage is to establish the goals of the system, that is the models that can be selected from.

5.4.2 Formulating the structure.

There are a vast range of design models that can be considered. In this network of questions, the choice is limited to a set of simple design models. This means that they are both simple to understand and straightforward to apply. The list ranges from completely randomised designs, through randomised blocks and fractional factorials, to cyclic designs and response surfaces.

Response surfaces require different methods of design from most models, and so the early identification of problems requiring this would be an advantage. Fortunately, a consideration of the treatment structure can identify problems that are response surface ones. Similarly designs to isolate components of variance are identified at an early stage in the

network. This still leaves a set of fourteen designs, listed in figure 5.3, to build a network of questions round.

The first question to be asked should be one of the key questions. As the treatment structure can influence the choice of design at an early stage, this is the first question to be asked. If it can be established that no factorial structure is necessary, then the number of treatments can be quickly determined and a flag set to zero. On the other hand, if a factorial structure is necessary then it would be advantageous if this were as simple as possible. The number of factors and the levels to each factor are established, and these are examined and questioned. The aim would be to try to persuade the client to simplify his treatment structure to at most a selection of 2-level and 3-level factors. It is at this stage that response surface problems can be detected and treated separately. Once the treatment structure has been agreed on, then the system can calculate the number of treatment combinations and proceed.

Before considering the next key question, the system removes any clients who are interested in component of variation problems. This is to allow a separate set of designs to be used for this slightly more complex problem. If component of variation problems are not required, then the system proceeds to the next key question, that of the sources of variation. The more sources there are, then the more complicated the design required. In addition, the structure of the treatments will also affect the final choice of design model. If a blocking structure is used, then the number of units in each block will also play a vital part in the model selection.

5.4.3 A sample network.

Working along these lines, the following network can be derived. This combines all of the ideas established so far, and reflects the thoughts of one person on the selection of a design model.

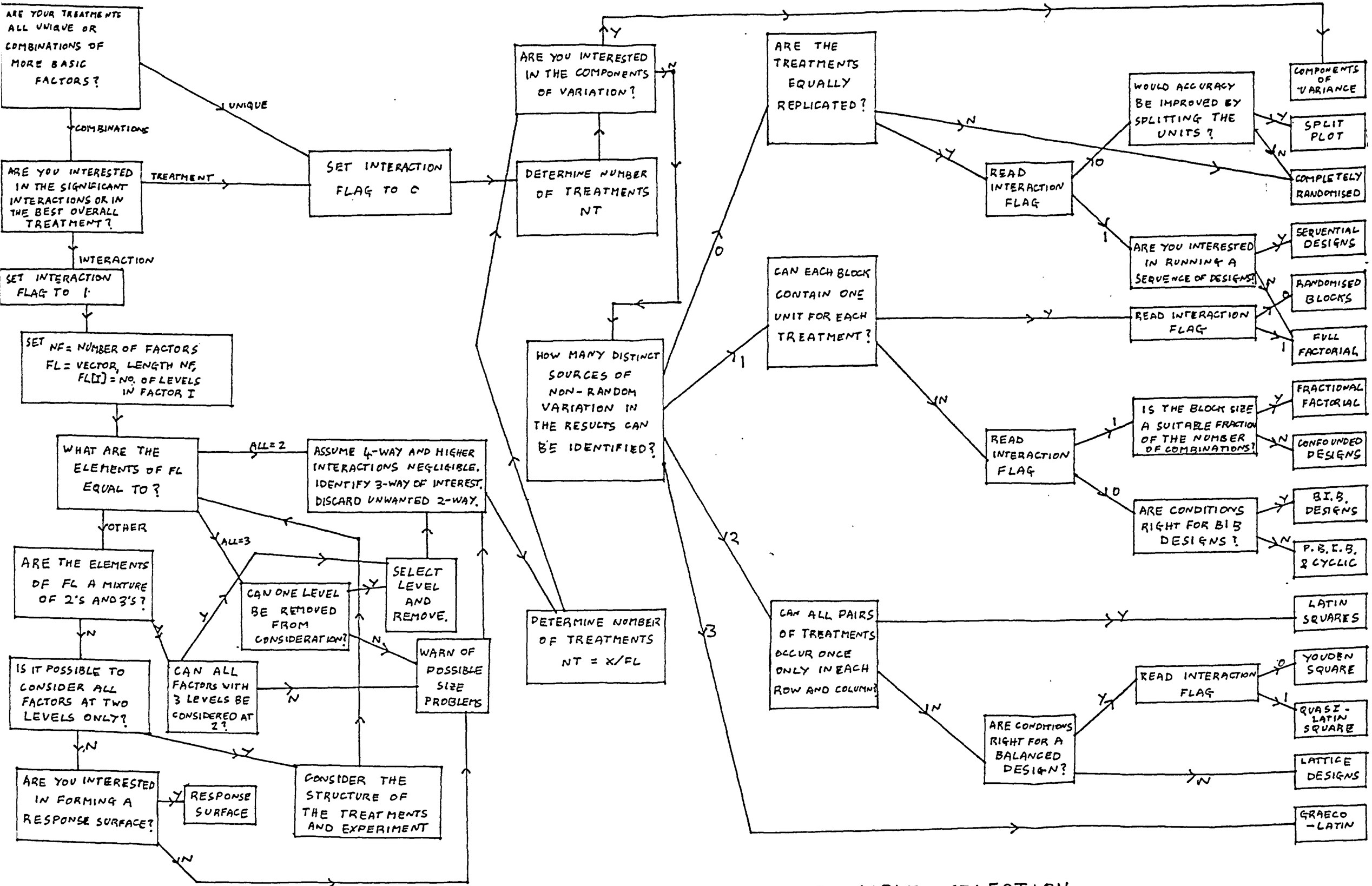


FIGURE 5.3 – NETWORK FOR MODEL SELECTION

5.4.4 Programming the network.

The network presented can easily be programmed along the same lines as ExpCheck. The utility functions and control functions are placed in an APL workspace and compiled into an executable file. All the questions are placed into a single component file, and a series of code numbers used to establish the order in which the questions are asked. Help facilities along the lines of ExpCheck and accessed in a similar way can also be prepared.

5.5 CONCLUSIONS

The discovery of the facts of a statistical problem has been discussed, and the analysis of these facts covered. A discussion of the use of causal networks for the fact elicitation shows that the method is a feasible option. This has been illustrated with two examples concerned with experimental design.

This whole section has considered the programming of a statistical expert system to emulate the statistical consultant. However, analysing the problem with a user is only half the task to be undertaken. In the next section, a method for formulating rules to use in selecting an analysis procedure is introduced.

SECTION III

SELECTION OF ANALYSIS PROCEDURE

CHAPTER 6

PERSONAL CONSTRUCT THEORY - BACKGROUND AND METHODOLOGY.

6.1 INTRODUCTION.

In this section, the problems involved in considering the structure and models used in statistical expert systems are left to one side. The problem that is now being considered involves the methods which may be used to formulate rules for selection of a suitable technique from a list of possibilities.

A number of possible methods have appeared in the last few years for automated knowledge elicitation, as the problem is normally termed. These tend to fall in one of two categories, data driven (learning from past examples), and empirical formation (modelling 'hidden' knowledge, i.e. techniques and 'tricks of the trade'.) This section is concerned primarily with the theoretical development and practical application of a method for formation of empirical rules. However, it should be noted that both methods have a place in statistical expert system development, and a brief time needs to be spent in stating which data-driven methods would be of most use.

6.1.1 Background to the problem.

As the number of statistical expert systems grows, so the ways in which rules are derived become ever more clearly defined. However, this does not mean that the problems encountered in forming the rule base have become any easier. Indeed, a number of areas still present difficulties to the system programmer.

One area that causes a great deal of problems is in test selection, either for an advisory system such as KENS (Hand, 1987b), or for a test and model construction system such as SELINA. The rules that are used in test selection need to be very carefully formed and tested to avoid bias. A great deal of time and effort has been spent in the formation of rules for the major systems, and a number of tried and tested statistical techniques used in multivariate analysis can be employed for this purpose.

This assumes, however, that the user is only interested in some form of data-driven rule selection. If we wish to derive general methods of rule elicitation, then the possibility of non-data-driven rules has to be considered. As Boen & Zahn (1982) point out throughout their book, far more can be learnt by observation than by close statistical questioning. In addition, the current techniques need a large data set or collection of examples before effective rules can be derived. However, a system builder may have little or no time for data collection, and indeed may not have the resources to run the complex software that is required.

The problem is, therefore, to develop a method for obtaining rules that model the non-statistical aspects of test selection. By implication that means some way of discovering what rules and assumptions a statistician brings to bear on a problem in order to find what he regards as a suitable solution. In addition, these rules need to be applicable to as wide a range of tests as possible, so that as many possibilities are covered as the resources available will cater for. Finally, it would help if the method used was straightforward, and did not require expensive and cumbersome computational machinery.

6.1.2 Why test selection rules?

It needs to be remembered that a proper solution to a consultancy problem can only come about with a proper understanding of the background to the problem. The difficulty with this is that different statisticians bring different perceptions to the same problem. Neiss et al (1985) illustrated that even for a fairly common situation, different people would make different deductions and use different tests to form their conclusions. In addition, there is the ever present danger of committing a statistical error of the third kind (Kimball, 1957). Even worse, there is the danger of using the wrong test merely because the criteria for this situation seem to match the previous one or it is the consultant's particular favorite at the moment - the situation called the 'Bell Ringer' by Hyams (1971).

It was stated in an earlier chapter that an expert system not only has to do the job it was built to undertake, but also to educate the user as to the reasons why it was working in a particular way. As a result, the rules have to be such that they avoid known pitfalls, as well as educate the user to avoid them and recognise them when they appear in future problems. It is therefore imperative that the way in which statistical consultants select tests is fully modelled. That includes the non-statistical aspects, such as what is learned from experience gained over the years, and the tricks they use in spotting when a particular solution is the most appropriate one.

To obtain these rules, a simple method for modelling the perceived approach needs to be determined. This poses different problems from those normally involved in classification techniques, which have been used in the past to form these rules. As has been mentioned several times already, the structure of the data available may not lend itself to multivariate classification. However, there is a science which attempts to model human perception of a situation - psychology. When these methods are considered, a

technique is available which can and has been adapted to meet the requirements. This technique is known as personal construct analysis, and has been used in management science and other areas to separate a number of options by considering ways in which small subsets of the options differ. This is developed more fully later in this chapter. It would, however, first be useful to briefly discuss other methods of rule elicitation that exist.

6.2 ALTERNATIVE RULE GENERATION METHODS.

6.2.1 Data-driven methods.

As has already been stated, the rules that could be used in model selection can come from one of two sources. Personal construct analysis affords a method that falls into the emprical rule formation category. However, a statistical expert system should also make use of data and past results, and this is where data-driven methods come in. Others have written on the subject, ever since Gale and Pregibon proposed using examples of past analysis to guide the rule formation method. (Gale & Pregibon, 1984)

The methods are all based on examination of relevant past examples, and identification of the major aspects of the problem that led to a particular decision been made. The way in which these aspects are identified have become increasingly more sophisticated over the years, with a greater degree of automated analysis being employed. One of the main proponents of these methods is Richard Forsyth, who recently produced a system known as BEAGLE (Biological Evolutionary Algorithm Generating Logical Expressions), based on the classification tree methodology developed by L. Brieman (1984).

These methods and systems have been given a collective noun of Rule-Finder systems, defined by Forsyth himself as follows.

'A rule-finder is a program which examines a database of examples and uses machine-learning techniques to create a set of rules for classifying those examples and other examples of the same type. In short, it turns data into knowledge'(Forsyth, 1987)

The methods work by analysing a database of possible solutions to a problem, and a set of measurements for each example. By splitting the data set, using simple functions of the measurements that separate out the different solutions, rules based upon the measurements can be developed. The rules can then be tested in some way to see how effective they are, and the more effective ones selected for use. Another possibility is to conduct some form of multivariate analysis on the data set, such as principal component analysis, cluster analysis or discriminant rule analysis.

However, these methods require a fairly large data set for a small number of possible solutions. It may be that that many examples are not available, or that it is impractical to gather together a suitable number. For this reason a second method of rule generation is required which works on small sets of data for a comparatively large number of examples.

6.2.2 Psychometric techniques.

The alternative method is to employ techniques which are used in psychology, to try to gain an understanding of the underlying assumptions that occur in a consultancy situation. Most of these methods involve some sort of scaling of the tests along a number of numerical scales, and a form of analysis on the resulting matrix of values. Personal construct analysis is based upon this method, but it is by no means the only one available.

6.3 PERSONAL CONSTRUCT ANALYSIS.

6.3.1 A Brief History.

Personal construct analysis or, to give it its more recent name, personal construct technology, is a theory of how the human mind makes decisions in a given situation. It first appeared in a book by George A. Kelly (1955), where the theory is presented and a number of immediate conclusions arising from it are explained. Unfortunately, the language used is a very philosophical and technical one, but the idea itself is fairly easy to explain.

At the beginning of his book, Kelly makes the following statement:-

> 'Man looks at his world through transparent patterns or templates which he creates and then attempts to fit over the realities of which the world is composed.' (Kelly, 1955, pp8-9)

In simpler terms, as a person lives he forms a set of scales in his mind against which he assesses any new situation that he may encounter. However, rather than having the reaction to the present situation depend totally on the past experience, personal construct theory states that the person's judgment of the situation will be based upon past experience in that area. As a person encounters new situations, so the scales can be modified and adjusted accordingly.

If this theory is an acceptable one, then the decision and problem solving processes of a person can be modelled by attempting to obtain the scales that this person uses, which are known as personal constructs. Personal construct analysis is the science of obtaining these constructs and analysing them.

Shaw (1979) developed the methodology of Kelly by building and preparing a suite of computer programs to elicit the required information, not only to form the constructs but also to perform a type of analysis on them. Since that work first appeared, she and her husband Brian Gaines have developed the method further to elicit certain types of rules from the grids. However, their algorithms have never been fully published. This section of the thesis presents an attempt to emulate their programs, but in a new language and making some changes in the way that the results are presented. Before explaining the basic algorithms of personal construct theory, a few terms need to be defined.

6.3.2 Definitions and Notation.

Personal construct analysis attempts to find the ways in which a person discriminates between a set of objects. These objects may be tasks to perform, people applying for a job, services to provide, and so on. The objects themselves are known as the elements. At any one time, the elements that are currently being considered are stored in a matrix E, and the number of elements in E can be represented by EE.

The elements are separated by finding ways in which two or three differ from each other. These ways are represented by a form of scale, at one end of which is some attribute in common with one or two of the relevant elements, and the other a contrasting aspect that is true for the remaining elements. These scales are known as the constructs. The elements are graded against each construct on a scale from 1 to n, where n is some integer, usually odd. The attributes at each end of the scale are known as the poles, one corresponding to a score of 1 and the other to a score of n. At any one time, the number of constructs being considered is CC. The poles with a score of 1 are known as the left hand poles and are stored in a matrix C.

The poles on each scale which correspond to a score of n are known as the right hand poles and are stored in a matrix D.

Every element has some score against each construct, depending on how the poles relate to that particular element. The collection of scores for all the constructs over all the elements is known as the repertory grid. The rows of the grid represent the scores of an element over all constructs, and the columns the scores for constructs for all elements. The grid itself is denoted by A, and the score of the Ith element on the Jth construct by A[I;J].

Now that the relevant terms have been defined, the way in which a grid is constructed can be explained. The actual algorithms are published in Shaw (1979), so they will be illustrated by showing how a grid for selecting from a set of simple procedures can be constructed. The example presented here uses programs derived from a BASIC program of the algorithms prepared by M.L.G. Shaw.

6.3.3 Grid Construction - an example.

Suppose we want to find rules to assist in the selection from the following set of two sample procedures.

T-Test	The standard t-test statistic
F-Test	F-statistic for comparison of variances
Chi-Square	Usual measure of relationship from tables
Median	A simple non-parametric centrality measure
Corr.Coeff	The standard correlation coefficient
Spearmans	Spearman's Rank Correlation Coefficient
Aspin-Welch	A robust form of the t-test.

TABLE 6.1 - TESTS TO BE CONSIDERED IN EXAMPLE

The first six of these tests are used as the starter set of elements. The use of six tests or elements was proposed by Shaw as the minimum required to provide a basis for grid formulation. Of these six, three were felt to be required to form a construct to distinquish all the tests, and four subsets of the tests provided a sufficient number of initial constructs to sepearte the initial set of tests.

Consider the first three of this set, i.e. T-Test, F-Test and Chi-Square. A scale is needed which will give a score of 1 to two of these methods, and a suitable value of n to the third. The vale n may be any odd number greater than 3; in this case, n is set equal to 5. One possible scale is to allocate a score of 1 to those methods concerned with a parameter estimate, and 5 to those concerned with a measure of relationship. If this is used, then a score of 1 will be allocated to the first two and 5 to the third. The initial six elements are then measured on this scale and the scores recorded.

Now consider the correlation coefficient, Spearmans and Aspin-Welch. Let 1 indicate that ranking is required, and 5 no ranking. Again, this separates the three tests being considered and the initial six tests can be scored against this scale. If this is repeated for another two subsets of the tests, the following initial repertory grid can be formed.

Parameter Estimate	1	1	5	1	3	3	Relationships
Ranking	5	5	3	1	5	1	No Ranking
Normality Assumed	1	1	2	5	5	3	No Normality Assum
Parametric	1	1	3	5	2	5	Non-Parametric
	*	*	*	*	*	*	
	*	*	*	*	*	Spearmans	
	*	*	*	*	Corr.Coeff		
	*	*	*	Median			
	*	*	Chi-Square				
	*	F-Test					
	T-Test						

FIGURE 6.1 - INITIAL REPERTORY GRID.

The left hand column lists the array C, and the right hand column the array D. The data matrix represented here is the repertory grid A, and the text below the matrix the array E.

Now, for each of the tests or elements, a measure of the similarity to all the other elements is calculated, using the following formula for the similarity between elements I and J, where n is the maximum score on a construct

$$S_{I,J} = 100 - 100 \times \left(\sum_{K=1}^{CC} |A[I;K] - A[J;K]|\right) \div ((n-1) \times CC).$$

The results of these calculations lead to the following table

	ELEMENT 2				
ELEMENT1	F-Test	Chi-sq	Median	Corr.Coef	Spearmans
T-Test	100	43.75	25	56.25	25
F-Test		43.75	25	56.25	25
Chi-sq			31.25	50	56.25
Median				43.75	75
Corr.Coef					43.75

TABLE 6.2 - INITIAL SIMILARITIES FOR ELEMENTS.

The largest of these values is the measure of 100 between the F-Test and the t-test. This suggests that the two elements have an identical set of scores on the constructs and that a new construct needs to be included to separate them. In general, if the largest value in this table exceeds a certain level, then a new construct is needed. In this case, one which gives a score of 1 to a mean test and 5 to a variance test is sufficient. All the current elements are scored on this scale and the results added to the grid to give the following.

Parameter Estimate	1	1	5	1	3	3	Relationships
Ranking	5	5	3	1	5	1	No Ranking
Normality Assumed	1	1	2	5	5	3	No Normality Assum
Parametric	1	1	3	5	2	5	Non-Parametric
Mean Tests	1	5	3	1	3	3	Variance Tests
	*	*	*	*	*	*	
	*	*	*	*	*	Spearmans	
	*	*	*	*	Corr.Coeff		
	*	*	*	Median			
	*	*	Chi-Square				
	*	F-Test					
	T-Test						

FIGURE 6.2 - INTERMEDIATE REPERTORY GRID.

At this stage, EE is still equal to 6 but CC is now 5. We calculate a similar measure of similarity between the last construct to be added and all the other existing constructs, and look at the values. The formula used to calculate the similarity between constructs is slightly more complicated, in order to allow for the possibility of 'reflective similarity', i.e. elements scoring 1 on one construct and 5 on the other, and similarly for other values. Let A[K;I] be the score on the Kth construct of the Ith element, EE be the number of elements and n the maximum score on a construct. The following formula is used for the similarity between constructs I and J

$$R_{I,J} = \max\ (R_1,\ R_2),\ \text{and}$$

$$R_1 = 100 - 200 \times \left(\sum_{K=1}^{EE} |A[K;I] - A[K;J]|\right) \div ((n-1) \times EE)$$

$$R_2 = 100 - 200 \times \left(\sum_{K=1}^{EE} |A[K;I] - ((n+1) - A[K;J])|\right) \div ((n-1) \times EE)$$

The following are the similarities for the existing constructs in our example against the most recently entered construct.

LHP OF CONSTRUCT	SIMILARITY
Parameter Estimate	50
Ranking	33.33
Normality Assumed	8.33
Parametric	8.33

TABLE 6.3 - CONSTRUCT SIMILARITY SCORES.

If the largest of these values exceed an agreed limit, then an element is requested which will take opposite scores on the two constructs so identified. In this example, the highest score does not exceed the limit, and no element is asked for.

These steps are repeated, with the addition of extra elements and/or constructs at any time if wanted, until a repertory grid has been formed that satisfies the person as a model of their problem. In the example being considered here, this occurs after all seven elements have been included and eight constructs formed. The result is the following grid.

Parameter Estimate	1	1	5	1	3	3	1	Relationships
Ranking	5	5	3	1	5	1	5	No Ranking
Normality Assumed	1	1	2	5	5	3	2	No Normality Ass.
Parametric	1	1	3	5	2	5	1	Non-Parametric
Mean Tests	1	5	3	1	3	3	1	Variance Tests
Robust	5	5	3	1	4	1	1	Non-Robust
Two Groups	1	5	5	3	1	1	1	Many Groups
Original Parameters	1	1	3	1	3	3	5	Transformed Para.
	*	*	*	*	*	*	*	
	*	*	*	*	*	*	Aspin-Welch	
	*	*	*	*	*	Spearmans		
	*	*	*	*	Corr.Coeff			
	*	*	*	Median				
	*	*	Chi-Square					
	*	F-Test						
	T-Test							

FIGURE 6.3 - FINAL REPERTORY GRID.

For the final grid, A is the matrix shown, EE = 7, CC = 8, C is the left hand vertical column, D the right hand column and E the staggered list. The similarity measures that are calculated are based upon the City Block metric, that is the sum of the absolute differences between corresponding elements of the two vectors scaled in a suitable way. As such, it provides an estimate of the difference between the two vectors of scores obtained from the Grid. The result is then scaled and reversed to give a measure of similarity that lies between 0 and 100.

Having obtained the grid, the next question is how to use the results. Rule formation is considered in the next chapter, but it would be useful to consider first a type of preliminary analysis which can be conducted. This is known as focusing the grid.

6.3.4 Focusing the grid.

The purpose of focusing a grid is to show how the final set of elements and constructs group together to form a complete model of the situation being considered. In essence, it is a form of multivariate analysis, but because the variables are non-continuous measurements on a finite integer scale, it is impossible to use the more conventional methods.

The following is the algorithm for focusing the repertory grid in terms of the constructs. The algorithm for focusing by the elements is similar, except for the similarity measure which should be replaced by that for the elements given on page 80, and C and D are replaced by E.

Algorithm 6.1 Focusing a grid by constructs.

1/ Take the repertory grid A, constructs I and J from the matrices C and D and let n be the maximum score on a construct, i.e. that associated with the members of D. CC is the number of constructs, and EE the number of elements. Form an upper triangular matrix G, where

$$G[I;J] = \max\ (R_1,\ R_2),\ \text{and}$$

$$R_1 = 100 - 200 \times \left(\sum_{K=1}^{EE} |A[K;I] - A[K;J]|\right) \div ((n-1) \times EE)$$

$$R_2 = 100 - 200 \times \left(\sum_{K=1}^{EE} |A[K;I] - ((n+1) - A[K;J])|\right) \div ((n-1) \times EE)$$

$1 \leq I \neq J \leq CC$, and

$$G[I;I] = 1000 + I. \text{ (from page 81)}$$

$1 \leq I \leq CC$.

G[I,I] acts as a check on which constructs have been paired, while G[I;J] is the similarity between constructs I and J.

2/ Let U, V and W be arrays of size 2×CC for storing the results of the analysis. In addition, let Z=CC+1.

3/ Let T be the largest off-diagonal element of G, i.e. the two most closely matched constructs. Let X and Y be such that G[X;Y] = T, and this is the first occurrence of T in G. Let R=X and S=Y

4/ If G[X;X] = G[Y;Y], go to stage 8/.

5/ If (G[X;X] < 1000) and (G[Y;Y] < 1000), then let

$$G[X;X] = G[Y;Y] = Z.$$

and go to stage 7/. This notes the two constructs have been matched.

6/ Find those diagonal elements equal to G[X;X]. Set all of them equal to Z, and let R=G[X;X]. Similarly, find those diagonal elements equal to G[Y;Y], set them equal to Z, and let S=G[Y;Y]. This matches all constructs currently paired to one or other of the current constructs to the new grouping.

7/ Let U[Z] = R, V[Z] = S and W[Z] = T.

8/ Let G[X;Y]=-1, and Z=Z+1. Compare G[1,1] to the other diagonal elements. If one of them is not equal to G[1;1], go to stage 3/.

The arrays U, V and W contain information which can now be used to construct a tree diagram, illustrating the relationships between a pair of constructs and the closeness of that relationship in terms of the similarity score. In particular, U and V state which sets of constructs have been merged, and W the similarity of the sets been merged. Obviously, the earlier that a pair of constructs are grouped together, then the more closely related

that that pair of constructs will be. However, the aim is to find a method for generating rules that show the deductions made by a statistician in test selection. As a result, the focusing of a repertory grid by the constructs helps us to see where the possible rules are most likely to lie.

In a similar manner, a diagram can be constructed which shows the relationships between the elements. In order to facilitate the analysis, a small change is made to step 4 of the algorithm, replacing the formula stated with a calculation of G of the following form:-

$$G[I;J] = 100 - 100 \times \left(\sum_{K=1}^{CC} |A[I;K] - A[J;K]|\right) \div ((n-1) \times CC) \text{ (from page 80)}$$

$$1 \leq I \neq J \leq EE, \text{ and}$$

$$G[I;I] = 1000 + I.$$

$$1 \leq I \leq EE.$$

6.3.5 The example continued.

The technique can be illustrated by use of the example started earlier. Applying Algorithm 6.1, the following is the matrix G which is generated.

Parameter Estimate	1001	29	29	36	57	7	29	57
Ranking		1002	43	93	14	64	14	14
Normality Assumed			1003	50	29	36	14	29
Parametric				1004	21	71	7	21
Mean Tests					1005	50	43	43
Robust						1006	21	50
Two Groups							1007	14
Original Parameters								1008

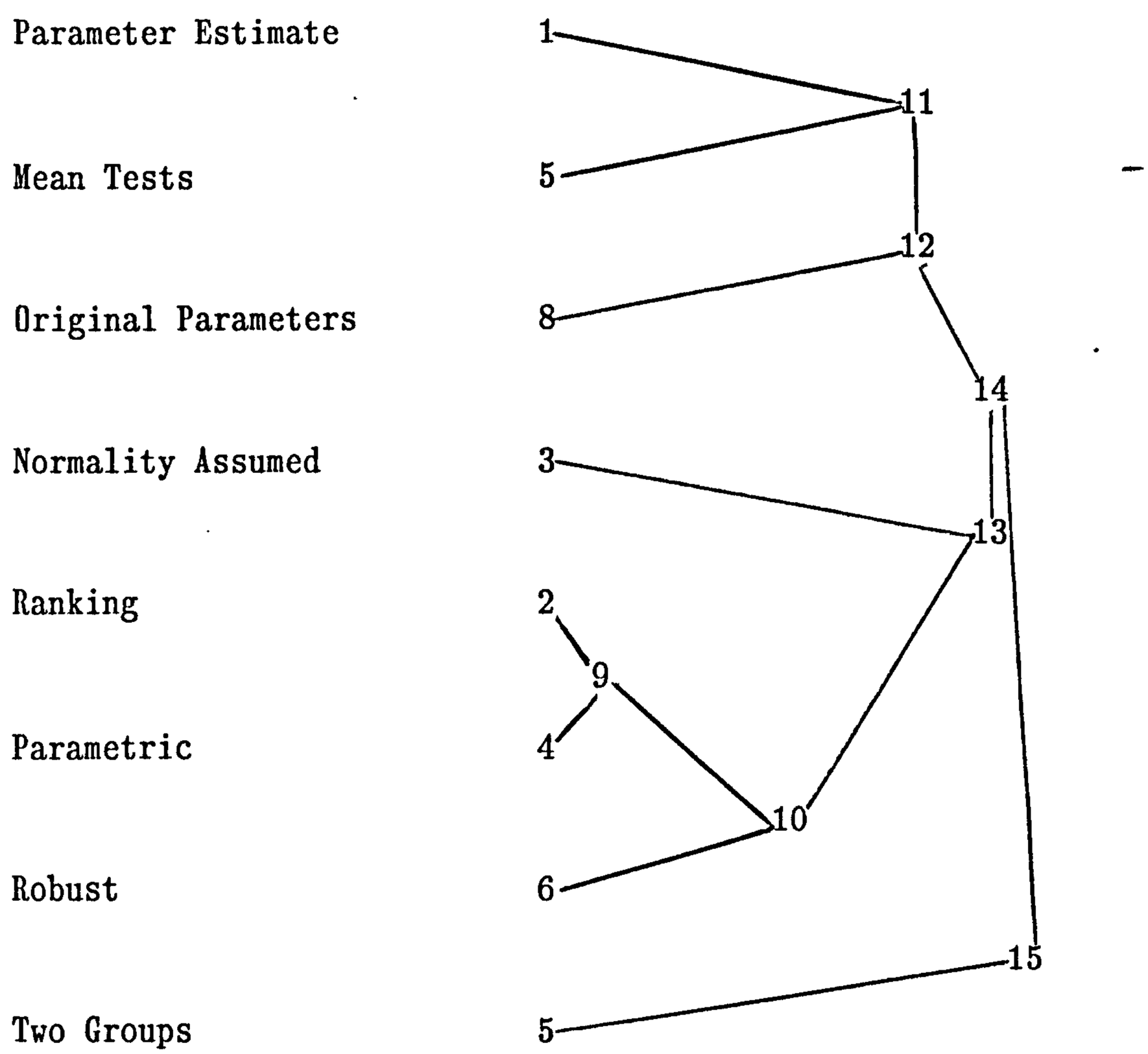
From this figure, the largest off-diagonal value can clearly be seen to be 93, which occurs between constructs 2 and 4 i.e. Ranking and Parametric. The algorithm makes note of this, adjusts the vectors U, V and W accordingly by setting U[9] = 2, V[9] = 4, and W[9] = 93, and replaces the value of 93 with one of -1, while adjusting the appropriate diagonal elements at the same time. The matrix G now looks like this

Parameter Estimate	1001	29	29	36	57	7	29	57
Ranking		9	43	-1	14	64	14	14
Normality Assumed			1003	50	29	36	14	29
Parametric				9	21	71	7	21
Mean Tests					1005	50	43	43
Robust						1006	21	50
Two Groups							1007	14
Original Parameters								1008

Continuing with the algorithm, the next largest off-diagonal element can be seen to be 71, which is between constructs 4 and 6. As 2 and 4 have already been paired, they are now grouped with construct number 6, U, V and W adjusted by letting U[10] = 9, V[10] = 6, W[10] = 71 and the matrix altered to look as follows:-

Parameter Estimate	1001	29	29	36	57	72	29	57
Ranking		10	43	-1	14	64	14	14
Normality Assumed			1003	50	29	36	14	29
Parametric				10	21	-1	7	21
Mean Tests					1005	50	43	43
Robust						10	21	50
Two Groups							1007	14
Original Parameters								1008

The algorithm continues until all of the constructs have been paired together, and the vectors U, V and W may then be used to produce a tree diagram similar to the one reproduced below.

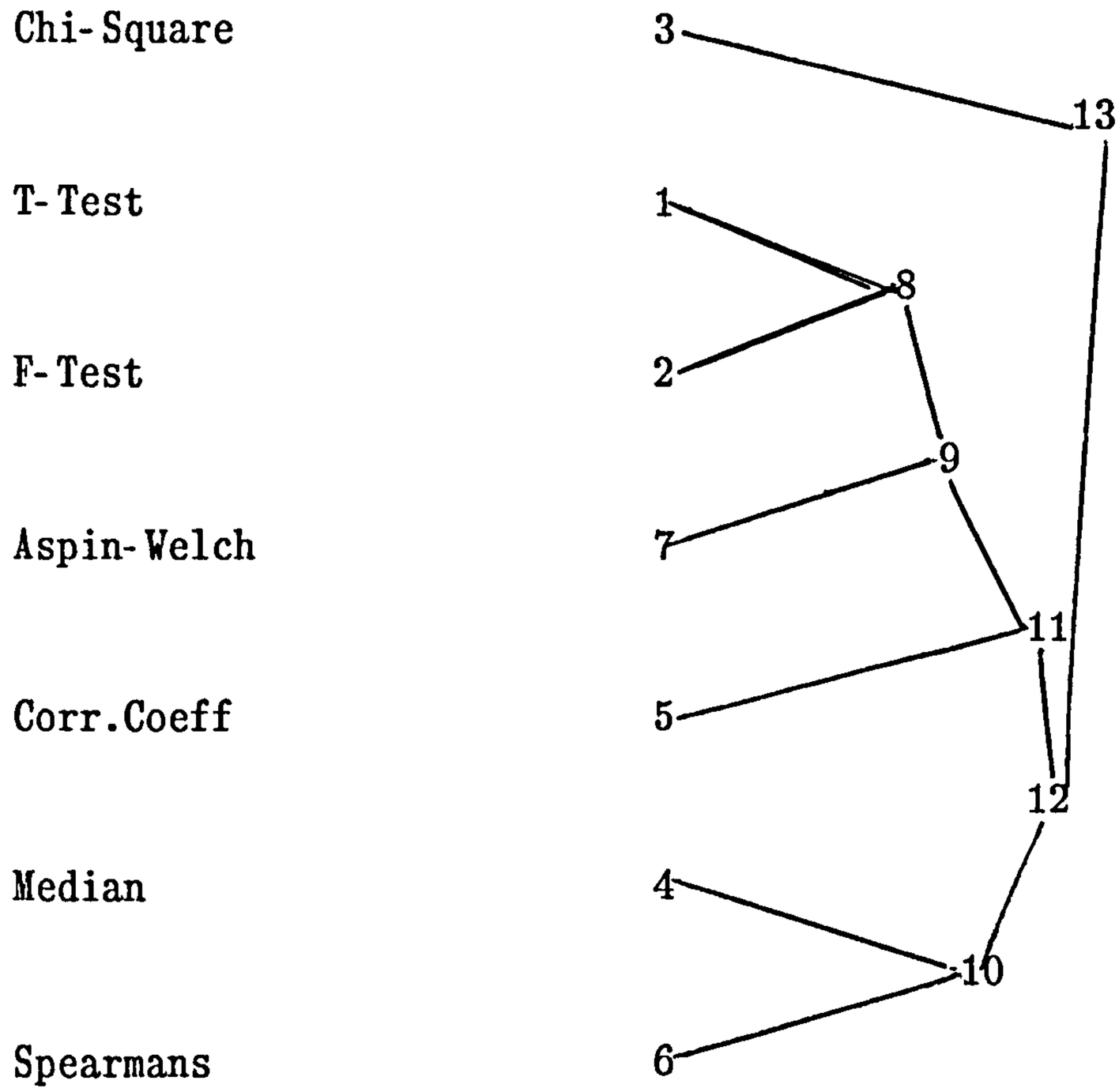


As can be seen, the relationships between the constructs are fairly clearly outlined. However, there is no real detail as to the structure of the relationships. In particular, there are no indications given to show what the relationships between the poles of the constructs actually are. In the next chapter, algorithms are given which formulate rules based upon the relationships between the poles, not the constructs. For this reason, merely focusing the grid gives an inadequate picture of the situation.

In a similar way, the initial matrix G for the elements is as follows

T-Test	1001	75	41	44	63	38	72
F-Test		1002	53	31	50	25	47
Chi-Square			1003	41	59	59	44
Median				1004	44	69	47
Corr.Coeff					1005	63	59
Spearmans						1006	53
Aspin-Welch							1007

In a similar manner to before, the matrix is analysed, the required vectors generated, and the following diagram produced.



The formulae and methods derived so far are all taken from Shaw (1979) and are based on a BASIC program derived by her. In later work (Shaw, 1988), she presents a method for deriving the similarities between constructs and using these similarities for rule formation. In the next chapter, an alternative method of rule formation from a repertory grid is produced, which allows for formation of rules based on both the constructs and the elements.

CHAPTER 7

INTRODUCTION TO STRATEGIC SELECTION RULES.

7.1 INTRODUCTION AND DEFINITIONS

7.1.1 What are strategic selection rules?

As already discussed in previous chapters, a variety of types of rules are required for a statistical expert system. In this chapter, a technique is presented for getting the most difficult class of rule to obtain - that which models the assumptions made by a consultant about the tests that can be applied to a given situation. In a sense, the rules model the strategy that is employed by a consultant in the selection of one test or procedure from a list of possibilities that could be applied to a problem. For this reason, the rules which are generated by this technique are called Strategic Selection Rules. The method of derivation presented in this chapter is original, and shows an alternative method to that developed by Shaw (1988). The work the rules undertake can be found in the following definition.

Definition 7.1

Strategic Selection Rules are the rules which model the hidden strategy used by a statistical consultant when selecting a model or test for use with a problem of interest.

However, there is a very real danger of oversimplifying the problem by not allowing enough structure in the rules that are generated under this definition. The rules can be said to have a two-fold purpose. Firstly, they identify the aspects of a problem that lead to a particular procedure being considered at any time. Secondly, in order to model the situation more fully, they also need to show how these aspects of the problem can relate to each other. For example, the shape of a graph of data can relate to the type of distribution that the data comes from. To allow for this, strategic selection rules can be split into two categories, defined as follows.

Definition 7.2

If a list of the attributes considered important in a particular test are given as the conditional parts of a production rule, and the test as the conclusion, then the resulting rule is known as a Model Characteristic Rule.

Definition 7.3

If any relationships can be found between the characteristics being considered by a statistical consultant in test selection, then the rules which model those relationships are known as Characteristic Implication Rules.

7.1.2 Some important criteria.

The way in which the rules are formed needs to meet certain criteria as well. For example, it would be useful if the rules could be formulated with only the information that is to hand, without the need for further questioning of the expert. If this were possible, then a great deal of time and effort could be saved, at least in the initial development stage. Naturally, any rules that are generated need to be checked for suitability at this stage of development. In the case of the procedures introduced here, the initial

information is a repertory grid produced using the methods described in the previous chapter. In addition, it would be useful if some measure of the plausibility of the rules was obtained. The plausibility simply means how reasonable the deduction would be in the experience of the consultant. As a useful starting point for the generation of these values, it would be helpful if the measures calculated meet the following criteria.

Definition 7.4

A plausibility measure for a strategic selection rule, of either type, would be expected to follow the following conditions:-

1. The value lies between 0 and 1.
2. The more that is known for certain, the higher the plausibility.
3. No further information beyond what is already known needs to be obtained.

7.1.3 Why personal construct analysis?

A reasonable question to ask at this point is how personal construct analysis can be used as a basis on which to develop algorithms to find these rules. To answer this question, it must be remembered that one purpose of the method is to find ways in which people discriminate between a set of objects, either for psychological or for management reasons. Using this as a starting point for argument, it can be seen that the resulting grid contains in itself a model of how an expert views a particular situation, in terms of the options available for action, and various aspects of the problem itself.

With this in mind, a method which acknowledges this fact and uses it as the basis for rule generation would be extremely useful. If the elements are a set of tests or models, then the constructs might represent ways in which a consultant separates the tests from each other. As a result, the repertory grid could be used to model the way in which a consultant picks a test for use. It is on this basis that the algorithms and methods presented in this chapter were developed.

7.1.4 Structure of the algorithms.

The algorithms to be presented here have been programmed in APL, and a master program developed by the author to do most of the necessary calculations. This program, known as KEIRS (Knowledge Elicitation In Rule Structures), has a number of modules to tackle the various parts of the development process. PEGASUS is a shortened version of the original personal construct analysis program developed by Shaw (1979), while FOCUS is a version of another Shaw program to undertake the initial analysis described in the previous chapter. The results in the previous chapter were prepared using these programs. A third, REVIEW, simply goes over the results of PEGASUS, i.e. a developed repertory grid, and allows changes to be made to the model by altering scores of elements on constructs.

The algorithms discussed in the remainder of this chapter are the basis for the remaining programs. A technical note describing some problems encountered in the earlier stages of development has appeared (Williams, 1989). In particular, the module SEARCH employs the following algorithms for rule development from a single grid.

7.2 MODEL CHARACTERISTIC RULES.

7.2.1 Construction of Model Characteristic Rules.

Definition 7.2 gives the basis upon which these rules can be defined. For a method to meet that definition, it would have to isolate those things which are known for certain to be true for a particular test or model. In addition, whatever information is used would have to come from what is known already, that is from the repertory grid. The result of this is that the method of obtaining these rules must isolate the items of interest from the repertory grid.

There is a further point to consider for the rules. Definition 7.4 states the plausibility of the rule should be related to what is known for certain. The method has to reflect this fact. This is considered more fully in the next section, but does have a bearing on the method described.

Now, the constructs which are used to separate the models reflect those areas which the expert considers to be important. Therefore the constructs, or more exactly the poles of the constructs, represent the important characteristics to be considered in the selection of the model. It is on this basis that the following algorithm for the construction of model characteristic rules, or MCRs for short, can be used.

Algorithm 7.1 MCR Formation

Let A be a repertory grid, with EE elements and CC constructs. C is the matrix containing the poles corresponding to a score of 1 on a construct, and D those poles corresponding to a score of $n \leq 5$.

1/ Let I be such that $1 \leq I \leq EE$.

2/ Let X be the vector of scores for model I over the set of constructs, i.e. $X = A[I;]$.

3/ Now consider each member of X, X_J, $1 \leq J \leq CC$.
If $X_J = 1$, then C[J] is a characteristic of model I.
If $X_J = n$, then D[J] is a characteristic of model I.

4/ The MCR is formed by stating model I as the conclusion part of a standard production rule, and listing the full set of selected characteristics as the clauses.

7.2.2 Plausibility of the MCR.

It has already been stated what is required of a formula to calculate the plausibility of these rules. As described above, we want the plausibility to reflect what is known for certain. To put this another way, we want the score to reflect how far the actual scores are from the extremes for the scales. Bearing this in mind, the following is proposed for the plausibility of the MCR's.

Definition 7.5

For $1 \leq J \leq CC$, if $X_J = n$ then $S_J = 1$, else $S_J = X_J$. Then let

$$R_1 = 1 - \left(\left(\sum_{I=1}^{CC} |S_I - 1|\right) \div ((n - 1) \times CC)\right)$$

Now, for $1 \leq J \leq CC$, if $X_J = 1$ then $S_J = n$, else $S_J = X_J$. Let

$$R_2 = 1 - \left(\left(\sum_{I=1}^{CC} |S_I - n|\right) \div ((n - 1) \times CC)\right)$$

We define the plausibility of the deduction from an MCR to be

$$R = \max(R_1, R_2)$$

Remember that there are three definite criteria to meet. No further information beyond what is contained in the grid is needed, and it is simple to show that the value of this formula will always lie between 0 and 1. The only other question that remains to be answered is that of how the value changes. Is there a relationship between the plausibility calculated and what is known for certain for a particular model? The following theory shows this by proving a slightly different proposition.

Theorem 7.1

The plausibility of the deduction obtained from an MCR only depends upon what is not known for certain about a particular model or test.

Proof.

Consider the following.

$$E[R_1] = E[1 - ((\sum_{I=1}^{CC} |S_I - 1|) \div ((n-1) \times CC))]$$

$$= 1 - E[((\sum_{I=1}^{CC} |S_I - 1|) \div ((n-1) \times CC))]$$

$$= 1 - \frac{1}{(n-1) \times CC} \times \sum_{I=1}^{CC} E[|S_I - 1|] \qquad (1)$$

The problem is now one of finding the expectation of a modulus. Recall from definition 7.5 that if $X_I = n$ then $S_I = 1$. Therefore,

$$|S_I - 1| = S_I - 1 \qquad (2)$$

As a result, from (1) we have

$$E[R_1] = 1 - \frac{1}{(n-1) \times CC} \times \sum_{I=1}^{CC} E[S_I - 1]$$

$$= 1 - \frac{1}{(n-1) \times CC} \times ((\sum_{I=1}^{CC} E[S_I]) - CC) \qquad (3)$$

As can now be seen, the expected value of R_1 depends mainly upon the expected value of the individual X_j. Let

$$p_1 = P(S_I = 1)$$

and similarly for $p_2, \ldots, p_n$. Also assume that the value of each S_I is not dependent upon the other S_I. Then

$$p_n = 1 - \sum_{J=1}^{n-1} p_J \qquad (4)$$

It is also true that $1 \leq X_j \leq n-1$, and

$$P(S_I = 1) = p_1^* = p_1 + p_n$$

$$= p_1 + 1 - \sum_{J=1}^{n-1} p_J \qquad \text{from (4)}$$

$$= 1 - (p_2 + p_3 + \ldots + p_{n-1})$$

Therefore

$$E[S_I] = \sum_{I=1}^{n-1} I p_I$$

$$= p_1^* + 2p_2 + 3p_3 + \ldots + (n-1)p_{n-1}$$

$$= 1 + p_2 + 2p_3 + (n-2)p_{n-1}$$

As can now be seen, the expected value of S_I depends only upon the probability of not been certain of some attribute. Putting this result back into (3), we have

$$E[R_1] = 1 - \frac{1}{(n-1) \times CC} \times \left(\left(\sum_{I=1}^{CC} (1 + p_2 + 2p_3 + \ldots + (n-2)p_{n-1})\right) - CC\right)$$

$$= 1 - \frac{1}{(n-1) \times CC} \times (CC(p_2 + 2p_3 + \ldots + (n-2)p_{n-1}))$$

$$= 1 - (p_2 + 2p_3 + \ldots + (n-2)p_{n-1})/(n-1)$$

Hence the expected value of R_1 is only dependent upon the probability of not knowing something. A similar argument can be used to show that this is also the case for R_2, and thus for R.

The only assumption that was made in the theorem was that of local independence, i.e. that the probability of scoring some value on the scale was independent of the score on any other scale. While not necessarily a normal assumption to make given the background to this technique, it can be shown that it is a reasonable assumption to make. The scales are constructed with differing sets of the elements, therefore the construction of one scale does not necessarily depend on what the other scales are composed from. The probability of an element being given one score on a scale is therefore different from the score it may obtain on any other scale.

Having established an algorithm for forming the rules and calculating the plausibility of the results, it would help to illustrate the results by continuing the example begun in Chapter 6.

7.2.3 MCRs for the example.

Consider the example that was used to illustrate personal construct analysis in the previous chapter. The first procedure from the list was the t-test. The following were the scores that this procedure obtained on the various scales or constructs that the expert selected.

PROCEDURE	SCORE
Parameter Estimate - Relationships	1
Ranking - No Ranking	5
Normality Assumed - No Normality Assumpt	1
Parametric - Non-Parametric	1
Mean Tests - Variance Tests	1
Robust - Non-Robust	5
Two Groups - Many Groups	1
Original Parameters - Transformed Param.	1

TABLE 7.1 - SCORES FOR t-TEST ON REPERTORY GRID.

The figures down the right hand side form the vector X in Algorithm 7.1. Following the rest of the procedure through, it can be seen that the MCR for the t-test from this grid is :-

Model/test is t-Test if:-

Parameter Estimate

Normality Assumed

Parametric

Mean Tests

Two Groups

Original Parameters

No Ranking

Non-Robust

In this case, the calculations for the plausibility give a value of 1, so this would be included as a part of the rule. Now consider Spearman's Rank Correlation Coefficient. The following are the results from the Grid.

PROCEDURE	SCORE
Parameter Estimate - Relationships	3
Ranking - No Ranking	1
Normality Assumed - No Normality Assumpt	3
Parametric - Non-Parametric	5
Mean Tests - Variance Tests	3
Robust - Non-Robust	1
Two Groups - Many Groups	1
Original Parameters - Transformed Param.	3

TABLE 7.2 - SCORES FOR SPEARMANS ON REPERTORY GRID.

Based on this vector of scores, the following rule can be quickly formed.

Model/test is Spearmans if:-

Ranking

Robust

Two Groups

Non-Parametric

When the calculations of the plausibility are made, we get a value for R_1 of 1 - (8/28) = 20/28, and similarly for R_2. Hence the plausibility of this rule can be set at approximately 0.75.

Similar rules and values can be generated for all the tests and procedures that are considered in the example. In this way, rules are formed which help in the selection of a procedure in terms of the factors known to be associated with that procedure. But what of the relationships between the factors? The next set of rules deals with them.

7.3 CHARACTERISTIC IMPLICATION RULES.

Having shown the algorithms for the MCRs, it is possible to consider a more difficult and less obvious area. The characteristic implication rules, or CIRs for short, model the way in which the characteristics of a situation which are identified as important are connected to each other. To develop a method for obtaining these rules, it is necessary to consider in tandem two facts about what the repertory grid represents.

7.3.1 Construction of the rules.

The constructs used to form the grid represent the aspects of a problem that the consultant considers when selecting a test for use. The poles of the constructs represent the actual facts that are taken into consideration. When forming the MCRs, it is these poles which are isolated and used to form

the rules. In a similar manner, if the elements which are associated with each pole can be isolated, then a method could be developed which uses these elements to form the rules.

It will also be remembered that an indication of where the possible implications lie has already been given. When the original grid is focused along the constructs, the resultant graph shows how closely related the constructs are to each other, but not the actual form of the relationship. What will therefore be required is a method which discovers the relationships between the poles of the constructs, and not just the constructs themselves. Another fact to bear in mind is that we are only interested in a small subset of the relationships, so some way of trimming and sorting the set of possible connections needs to be included. The following algorithms seek to fulfil these needs.

7.3.1.1 Finding the implications.

The algorithm to be presented here comes in two parts. Firstly, it is necessary to isolate those relationships or implications between the poles which may be of interest.

Algorithm 7.2 Implication deduction

Once again, A is the repertory grid and C and D the poles of the constructs. There are EE elements and CC constructs.

1/ Select a value T, where T is a measure of how plausible an implication has to be for the implication to be considered useful, and $0 \leq T \leq 1$.

2/ Let I and J be such that

$$1 \leq I \neq J \leq CC$$

3/ Let X be the subset of elements for which the score on construct I is 1, i.e. those elements for which A[I,L] = 1 if the element is E[L]. Let N be the number of members of X.

4/ Calculate a reduced similarity score R based on the members of X, where

$$R = 1 - ((\sum_{K \in X} |A[I;K] - A[J;K]|) \div ((n-1) \times N))$$ (taken from page 108).

This value is based on the similarity measure used for focusing the elements in section 6.3.4.

5/ If R > T, then we say that

C[I] implies C[J] with plausibility R.

If R < 1-T, then we say that

C[I] implies D[J] with plausibility 1-R.

6/ Repeat steps 3/ and 4/ above, but this time defining X as the subset of elements with a score of n on construct I.

If R > T, we say that

D[I] implies D[J] with plausibility R.

If R < 1 - T, then

D[I] implies C[J] with plausibility 1-R.

When this algorithm has been applied to all possible pairs of constructs, the result is a list of implications which may be of use in rule formation. However, because of the way in which the list is compiled there may be too many implications to be of practical use. Indeed, the maximum number of implications that can be derived is 2CC×(CC-1) - a number which can grow very quickly as the number of constructs increases. The reason for this is a simple one of transitivity of deduction. If one pole implies another, and

the second one implies a third, then it is reasonable to assume that the first one may also imply the third to some degree. However, the algorithm as it stands does not check for this occurring.

What is needed is a method for pruning the list down, and making the rule formation a little easier. Ideally the method should convert the list of implications generated by the algorithm into a simple presentation of the main implications. A graphical method would give both a simple picture of the main implications, and a compact way of presenting the results. Using the list of implications, it is possible to construct a graph or map of the implications. The following is one possible method for doing just that.

7.3.1.2 The Implication Map.

The implication map is a form of causal network, constructed using the results of algorithm 7.2. The nodes of this network are the poles of the constructs, or the facts that are being investigated. The connections are those paths between individual poles that the algorithm has identified as of importance. If all the poles are connected to each other along these paths, then the result is a network with 2×CC paths connecting the CC nodes, where CC is the number of constructs. The following algorithm explains how the map is formed.

Algorithm 7.3 Implication map formation

1/ Let the first node be the left hand pole of the first construct.

2/ Find the first unused pole that is implied in the list generated by algorithm 7.2. If the opposite pole to this one has already been used, find the next one.

3/ Mark the new pole on the map, and a line between the two showing the implication. Mark the appropriate value of R on the line.

4/ Let the new pole be the current one. If it implies the previous pole, show this implication. If both poles imply each other with R=1 both ways, the poles are equivalent and can be collapsed into one node.

5/ Repeat steps 2/ to 4/ until one pole from each construct has been used.

6/ Repeat the above, starting with the right hand pole of the last construct to be used and using the reverse order if required.

7/ Show any possible links between the two maps by showing implications between any nodes with implications in one direction only.

Once the full map has been formed, then it is a simple process to read off the rules, as two nodes and the path that connects them. An example is given in Chapter 8. The plausibility of the rules are quite simply the value of R calculated for the pairs of constructs in Algorithm 5.4. In addition, to counter any possible contradictions that may occur through the two maps being linked, an extra set of 2×CC rules need to be added, which basically say that opposite poles of the same construct cannot be true at the same time with plausibility 1. By allowing for this extra set of rules, the unfortunate possibility of contradiction between the facts of a situation is avoided. It also cuts down on the number of rules that actually need to be called to come to a decision.

The way the algorithm works in practice is illustrated in Section 7.3.3. First, it is necessary to consider some difficulties which may arise with the above procedures.

7.3.2 Problems with the algorithms

The method described above fits the requirements of a simple and compact presentation of the results. However, there are a number of difficulties that arise from using this method. The first and most obvious problem concerns the plausibility of the CIR's formed from the implication map.

It will be recalled that in Chapter 7, defitnition 7.4 gave three criteria that any measure of plausibility would have to follow:

- The value must lie between 0 and 1
- The more that is known for certain, the greater the plausibility
- No further information beyong what is already known needs to be obtained

In addition, Section 6.3.3 gives a formula for the similarity between two constructs as

$$R_1 = 100 - 200 \times \left(\sum_{K=1}^{EE} |A[K;I] - A[K;J]| \right) \div ((n-1) \times EE)$$

$$R_2 = 100 - 200 \times \left(\sum_{K=1}^{EE} |A[K;I - ((n+1) - A[K;J])|) \right) \div ((n-1) \times EE)$$

and select $R_{I,J} = \max(R_1, R_2)$

For the plausibility of a characteristic implication rule, the plausibility value must first needs to meet the criteria of no further information required. Consider the scores of each element on one of the constructs. By selecting only those elements with a score of 1 or n, a further restriction on the scores is imposed, restricting the calculation to one based on what is known for certain at the present moment. Removing the multiplier of 100 also removes the need for calculating the second formula, as will be explained later.

Now, consider the question of whether the value lies between 0 and 1. The formulae presented above are now reduced to, for the set of elements of interest X scoring 1 on construct I, the following:-

$$R=1-((\sum_{K\epsilon X} |A[I;K] - A[J;K]|) \div ((n-1) \times N))$$

where N is the number of elements in X, A is the repertory grid, and n the maximum score on a construct. Because of the way X is selected, A[I;K] = 1, and so

$$R = 1-((\sum_{K\epsilon X} |1 - A[J;K]|) \div ((n-1) \times N))$$

Also $1 \leq A[J;K] \leq n$, implying that $|1 - A[J;K]| = A[J;K] - 1$

Therefore

$$R = 1-((\sum_{K\epsilon X} (A[J;K] - 1)) \div ((n-1) \times N)$$

$$=1-(((\sum_{K\epsilon X} A[J;K]) - 1) \div ((n-1) \times N))$$

$$=1-[((\sum_{K\epsilon X} A[J;K])-N)/((n-1) \times N)]$$

$$=1-[((\sum_{K\epsilon X} A[J;K])/(N \times (n-1))) - 1/(n-1)]$$

If A[J;K] = 1, for all $K\epsilon X$, then

$$R=1-((N/(N \times (n-1))) - 1/(n-1)) = 1$$

If A[J;K] = n. for all $K\epsilon X$, then

$$R=1-((nN/(N \times (n-1))) - 1/(n-1)) = 0$$

Therefore R must lie between 0 and 1 for all K where A[I;K]=1. A similar exercise shows that this also holds for all K where A[I;K]=n.

It has now been shown that R, as a special case of the formulae derived in Section 6.3.3, satisfies two of the three criteria stated. It has also been shown that the method of rule genration outlined in Section 7.2 follow as a natural progression from the plausibility value. Rather more difficult to show is the third criteria, that the more which is known for certain the greater the plausibility. Consider, however, that the scores for one construct are known for certain. It has also been shown that, if the scores on the second construct are also 1 or n, then the plausibility will lie between 0 and 1. It may therefore be presumed that intermediate values occur with scores between 1 and n, and so with increasing numbers of values not relating to poles of constructs, the plausibility will alter. Further work would be required to prove this more formally.

It could also be argued that the way in which the maps are constructed is difficult and wasteful of information. A number of possible modifications do exist, such as using the dendrogram produced by FOCUS to guide the order in which the constructs are added to the map. However, the point is that it simplifies the list of implications into a form that is understandable to all. Different methods of construction will lead to different sets of rules, as would be expected. Provided a check is kept on which constructs have been used, the maps take very little time to form and provide a useful summary of the results of the algorithm.

Another question to ask at this point is why the algorithm must be applied twice to each pair of constructs. Surely it would save time to just use the pair I and J, where $1 \leq I < J \leq CC$? However, it needs to be realised that it is possible by this method for one pole to imply the other, but not vice versa. In addition, it is possible that, even if the two poles do imply each other, they will do so with differing degrees of confidence. It is therefore imperative that all possible pairings, ignoring ordering, are considered.

Finally, it has already been noted that it is possible for contradictions to occur in using the rules if certain steps are not taken against them. This is due to the cyclic nature of the graphs, which reflect the different ways in which the same information can be deduced. By including rules not allowing opposite poles of constructs to be true at the same time, the possibility of a contradiction is lessened. The problem may indeed be solved by not allowing any connections between the two sets of graphs. However, this may lead to a lack of distinction between certain treatments.

For example, suppose two tests are for variance comparisons, but one is robust and the other is not. The nodes of the map representing these characteristics may appear on different graphs. If the connections between the graphs are not allowed, then the subtle difference between the two tests may not be recognised and the rule set would be incomplete. An allowance can be made for this by adding an extra set of rules which state that if one pole of a construct is true for some example, then the other cannot be true. This also reflects a certain amount of common sense about the situation.

7.3.3 CIRs for the example.

To show how the algorithms work in practice, consider the two-sample technique example. One of the scales that was used allocates a score of 1 if ranking is required and 5 if no ranking is needed. Applying Algorithm 7.2 to this construct, we see that the following elements have a score of 1, i.e. ranking is required.

Median

Spearmans

For the purposes of illustration, T will be set to the equivalent of the value used in grid formation, i.e. 0.7. Based on these two procedures, the following reduced similarities can be calculated with the other scales.

LHP of Scale	Similarity
Parameter Estimate	0.75
Normality Assumed	0.25
Parametric	0
Mean Tests	0.75
Robust	1
Two Groups	0.75
Original Parameters	0.75

TABLE 7.3 - SIMILARITIES FOR IMPLICATION SELECTION.

Based upon the information given above, the following implications can be derived for examination.

Ranking implies Parameter Estimate with plausibility 0.75
Ranking implies No Normality Assum with plausibility 0.75
Ranking implies Non-Parametric with plausibility 1
Ranking implies Mean Tests with plausibility 0.75
Ranking implies Robust with plausibility 1
Ranking implies Two Groups with plausibility 0.75
Ranking implies Original Params with plausibility 0.75

In a similar manner, the full list of implication is drawn up for all the poles of the scales in the example. The resulting list contains 69 implications, while the number of possible implications was $2 \times 8 \times 7 = 118$. As a result, approximately 60% of the possible implications are available for use. Based upon these, the following map can be drawn up using Algorithm 7.3

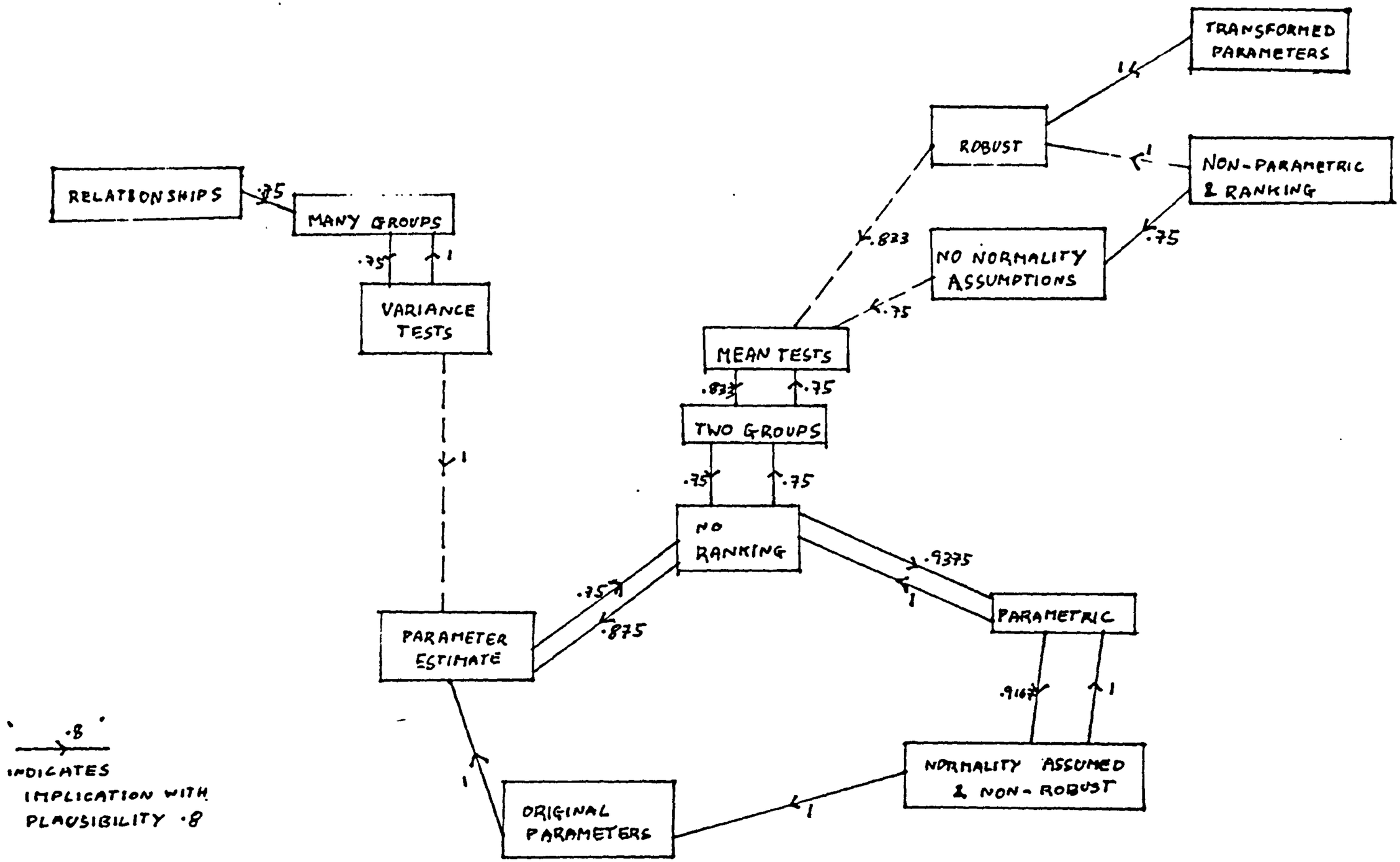


FIGURE 7.1 - IMPLICATION MAP FOR EXAMPLE

7.4 ADDITIONAL UTILITIES AND PROGRAMS.

7.4.1 Multiple Grids.

So far, the algorithms presented have been used to find rules from one grid. Suppose however that more than one person has drawn up a grid on a set of procedures. Is there any way of comparing, merging and analysing more than one grid? Two programs within the KEIRS suite allow for the merging and analysis of two grids.

7.4.1.1 Normal merging and analysis.

The first of these programs is labelled COMPARE. The program reads two repertory grids, and checks that the elements used in both grids are the same and are entered in the same order. If the order of entry is different, then one of them is rearranged so that the elements are in the same order as the other. The grids are then merged by joining them along the rows. In that way the element order is maintained, and the constructs for the second grid are added to the first. An analysis is then carried out on the combined grid.

A method for telling which constructs were used by the second person to construct a grid has to be determined, to allow for possible distinguishing at a later date. This is achieved by taking the matrices containing the labels for the constructs on the second grid and converting them into upper case letters. The matrices containing the labels attached to the poles for the constructs for both grids are then merged, leaving a complete set of constructs for the combined grid.

Once the combined grid has been formed, the analysis can proceed as for the FOCUS program and dendrograms produced to show the relationships between the two sets of constructs. The diagram can now also show which constructs/scales are identical in nature and scores, if not necessarily in

labels. The disparity in the labels could be due to one of two sets of circumstances. If the labels are similar, then the two experts have used the same scale in the same way, and one of the constructs can be safely removed. If however the labels are not similar, than the two experts have used different names for the same aspect of the problem, possibly because of their different backgrounds and training. If this is the case, then the extra information contained in the similarities will be useful, and the constructs should be included.

COMPARE allows for unwanted constructs to be deleted, and the remaining grid and matrices to be tidied up. When this has been accomplished, then the final combined grid can be used for rule formation, employing the algorithms already outlined in this chapter.

7.4.1.2 Preliminary pruning of the combined grid.

The fact that the grids are not checked for matching constructs before the rules are formed may , however, be undesirable to some people. After all, they might argue, it would be more economical if matching constructs were identified and removed before any analysis at all. Why go through all the work of analysis and tree formation when up to half the constructs may not be of any further practical use at all?

With this in mind, an alternative method of combining two grids has been developed. Based again on work of Shaw and Gaines, CORE merges two grids as before, but then conducts a simple set of tests to see which, if any, constructs can be removed. These tests operate along the following lines:-

1/ Identify the two constructs which are most similar.

2/ State which constructs have been identified, as well as the value for the similarity of these constructs.

3/ Offer the person using the program the opportunity to remove one of the constructs. If they do not wish to remove either of them, go to step 5/.

4/ Remove the specified construct/scale by deleting the appropriate column from the repertory grid, and deleting the labels from the appropriate matrices. All matrices are immediately rearranged to allow for this. Now return to step 1/

5/ Save the modified grid in a new file, along with all the other information.

In this way, a trimmed version of the grid is prepared by systematically deleting one of the two most closely matched constructs, only stopping when the expert refuses to remove one of the scales. Unlike COMPARE, this allows the expert more control over which constructs/scales are removed at a particular time. The new grid can now be passed through the programs for rule elicitation as if it were a single grid formed by one person. An illustration of how these programs work is given in the next chapter.

It has to be acknowledged that this is a very crude method of merging grids, particularly as it does not allow for merging more than two grids at the same time. A mechanisim to allow for simultaneous merging of three or more grids would be more suitable, possibly with a form of weighting to reflect the reliability of the opinion expressed. What this method does show is the possiblity of incorporating the knowledge of more than one consultant into the rule formation process.

7.4.2 Updating and amending grids.

Finally, two extra modules are available within KEIRS for the modifying of individual grids. One allows for checking of an existing grid, while the other allows for more radical modifications.

7.4.2.1 Slight modifications.

REVIEW is the module within KEIRS that allows a user to make slight modifications to an existing repertory grid. It takes each construct in turn and presents the current values the procedures take on that scale to the user. They are then prompted to make any changes that they see fit, and to save the results in an appropriate fashion. In this way, minor changes in the grid can be made simply and clearly.

Once all the changes that the user wants to make have been completed, the resulting new grid can either be saved in the original file, or else in a new file for comparison later. This program is most useful if, after an initial analysis, it is clear that one or two minor errors have crept in. If more major alterations are needed, it may be easier to form a new grid from the same set of elements. The final module of KEIRS allows for that possibility.

7.4.2.2 Complete renewal of a grid.

Each grid is stored in a component file on the computer. If a new grid is to be formed for the same purpose and using the same elements, then it is wasteful to type in the relevant information again. RESCORE is a procedure that reads the purpose and elements from an existing grid, and prompts the user to form a new grid based on these elements.

A subset of six procedures or elements is first selected at random from the set of procedures, and the remaining ones printed so that the new grid will include all of the existing procedures. This set of six, if accepted, is then used for the first stages of Algorithm 7.1, and a new grid is formed from this basis. Once completed, the new grid MUST be placed in a new file - the program will not allow the original one to be overwritten. This is unique to this module, but is necessary to allow for comparisons at a later time.

The module is also useful if a person has formed a grid based on a set of procedures some time ago, and wishes to see if their ideas of the situation have changed over any period of time. By using RESCORE, they are able to form a new grid and then compare it to the old one using one of the existing procedures. The constructs which can be deleted show which assumptions have not changed, while those remaining give an indication of how much their views on the subject have been altered over a period of time.

7.5 CONCLUSIONS.

This chapter has stated the basic algorithms required for a rule formation method to model the way in which a consultant selects a test from a list of possibilities. A number of conclusions concerning the methods and the problems involved in implementation have been discussed. All the results presented in this chapter have been prepared using the program KEIRS that has been described, and all the programs involved are available in APL. The second disk that has been included with this thesis is marked SUITE, and contains both the compiled version of the suite of programs and the source code in the form of APL2 workspaces.

CHAPTER 8

STRATEGIC SELECTION RULES FOR EXPERIMENTAL DESIGN.

8.1 INTRODUCTION

In order to illustrate how the algorithms developed for forming strategic selection rules work, it was hoped that consultant statisticians would use the program to develop rules for selecting experimental design models and test these in a small expert system. To facilitate this, an early version of KEIRS was passed on to consultants for testing and rule development. Unfortunately, the disk used would not work in their machines, and a gap of three months passed before this fact was made known. This left an insufficient length of time for the original plan to be carried out.

As a substitute plan, two colleagues in the Department of Mathematics and Statistics were asked to form their own repertory grids for selecting experimental design models. These grids are presented in this chapter, and the rules formed, after the grids are merged, presented. The way in which these rules may be rewritten for use in a design system is then considered, and some thoughts are offered on the way such a system would work.

8.2 THE INITIAL GRIDS.

8.2.1 The Selected Models.

The first person to form a grid was asked to select a set of models for both users to work from out of the choice listed in Figure 5.3. The following were his selection:

a) Randomised Blocks

b) Latin Square designs

c) Partial Balanced Incomplete Blocks (pbib design)

d) Balanced Incomplete Blocks (bib design)

e) Cyclic Designs

f) Fractional factorials

g) Confounded designs

8.2.2 The initial grids.

Having selected these models to work on, the first person then constructed a grid based around these models. The first six of the above list were used to form the initial grid, which was then built on and expanded using the methods outlined in Section 6.3.3. This grid was then focused by constructs and elements as outlined in Algorithm 6.1 to produce the following focused grid.

Square design	5	1	5	5	3	3	3	Blocks
Full design	1	1	5	5	3	5	3	Fractional
Full design	1	1	5	5	3	5	3	Fractional
Blocks	1	1	4	4	5	3	2	Cyclic arrangement
Confounding	3	3	5	4	3	1	1	Partial balance
Confounded effects	4	4	5	4	4	1	1	Partial balance
	*	*	*	*	*	*	*	
	*	*	*	*	*	*		confounded designs
	*	*	*	*	*			fractional factorial
	*	*	*	*				cyclic designs
	*	*	*					bib design
	*	*						pbib design
	*							latin squares
								randomised blocks

FIGURE 8.1 - FIRST REPERTORY GRID FOR DESIGN MODELS

As can be seen by a cursory examination of the grid, there are two pairs of constructs which may cause problems at later stages of the analysis, in that they appear to deal with similar aspects of the problem. How this is dealt with practically will be illustrated later in this chapter.

A second statistician was then asked to use the RESCORE section of the KEIRS suite to form a new grid based upon the same design models. The following repertory grid was the result of this investigation.

Factorial	5	5	1	1	4	4	3	Non-factorial
Some effects	5	5	1	1	5	5	4	All effects
Partial blocks	5	5	5	1	1	1	1	Full blocks
Some treatments	5	5	2	2	1	1	1	All treatments
Multiple blocking	1	5	4	3	4	4	4	Single blocking
Some types possible	1	5	3	3	4	4	4	All types possible
Proper fraction	3	3	5	1	4	4	4	Improper fraction
Complex to design	4	4	3	5	3	1	5	Simple to design
	*	*	*	*	*	*	*	
	*	*	*	*	*	*	cyclic designs	
	*	*	*	*	*	bib designs		
	*	*	*	*	pbib design			
	*	*	*	fractional factorial				
	*	*	confounded designs					
	*	randomised blocks						
	latin squares							

FIGURE 8.2 - SECOND REPERTORY GRID

8.3 MERGING OF GRIDS

Once the initial grids have been established, it is necessary to combine the grids together in a suitable way. To begin with, a suitable value for the similarity between two vectors needs to be established. An examination of the grids shows that one or two constructs within each grid are very similar. For example, in the second grid the constructs 'Multiple blocking - single blocking' and 'Some types possible - All types possible' can be shown to have a similarity of .92. It may therefore be necessary to remove one of these constructs, and the level at which this is done is set accordingly.

In order to illustrate the merging of the grids in this example, that level was set at 0.8. In other words, if two constructs have a similarity of more than 0.8, then one of the constructs will be removed.

8.3.1 Merging grids using CORE

The first two grids given above were merged using the CORE option in KEIRS. This allows the system to isolate those constructs which are closely matched, eliminate one of them and store the resultant gird for future analysis. The hope is that when two experts come up with similar constructs for the models, then these can be isolated and one removed to avoid confusion later.

When the two grids were merged and analysed, this was not the case. In fact, CORE succeeded in isolating some constructs that each expert used and were very similar indeed. The following table summarises the analysis and identifies those constructs which were removed.

CONSTRUCT	SIMILARITY	DECISION
Full design - fraction	1.00	Retained
Full design - fraction		Removed
Single blocking - Multiple blocking	0.93	Retained
All types - Some types		Removed
Partial balance - confounding	0.79	Retained
Partial balance - confounded effects		Retained

TABLE 8.1 - CONSTRUCTS REMOVED FROM FIRST TWO GRIDS.

Initially, neither of the final pair of constructs listed above were removed. It has already been stated that 0.8 was the agreed similarity between two constructs for one of them to be removed. As the similarity for these two was 0.79, they were retained. However, they could be said to represent virtually the same information, which would make one redundant. This paradox is dealt with at a later stage in this chapter.

The merged grid was then focused. The following is the resultant new grid, which was placed into a file for storage. Those constructs in upper case are those remaining from the second grid after the selected construct were removed.

Square design	5	1	5	5	3	3	3	Blocks
MULTIPLE BLOCKING	5	1	4	4	4	3	4	SINGLE BLOCKING
Blocks	1	1	4	4	5	3	2	Cyclic arrangement
Fractional	5	5	1	1	3	1	3	Full design
PARTIAL BLOCKS	5	5	1	1	1	1	5	FULL BLOCKS
SOME TREATMENTS	5	5	1	1	1	2	2	ALL TREATMENTS
NON-FACTORIAL	1	1	2	2	3	5	5	FACTORIAL
Confounding	3	3	5	4	3	1	1	Partial balance
Confounded effects	4	4	5	4	4	1	1	Partial balance
SOME EFFECTS	5	5	5	5	4	1	1	ALL EFFECTS
PROPER FRACTION	3	3	4	4	4	1	5	IMPROPER FRACTION
COMPLEX TO DESIGN	4	4	3	1	5	5	3	SIMPLE TO DESIGN
	*	*	*	*	*	*	*	
	*	*	*	*	*	*	confounded designs	
	*	*	*	*	*	fractional factorial		
	*	*	*	*	cyclic designs			
	*	*	*	bib design				
	*	*	pbib design					
	*	latin squares						
	randomised blocks							

FIGURE 8.3 - FIRST MERGED GRID.

8.3.2 Considering the merged grid.

The merged grid still contains the two constructs from the initial grid that seem to be concerned with the same aspects of a design. As this is a possibly unnecessary duplication of facts, it needs to be dealt with before proceeding further. It will be noticed by an examination of the grid that the same elements are given scores of 1 or 5 on the two scales. As a result, any implications that are derived for the two constructs will be identical. As this was only made clear when the merged grid was analysed using FOCUS, it was decided to rerun the two grids through CORE, but this time removing one of these constructs. The following was the resulting focused grid.

Square design	5	1	5	5	3	3	3	Blocks
MULTIPLE BLOCKING	5	1	4	4	4	3	4	SINGLE BLOCKING
Blocks	1	1	4	4	5	3	2	Cyclic arrangement
Fractional	5	5	1	1	3	1	3	Full design
PARTIAL BLOCKS	5	5	1	1	1	1	5	FULL BLOCKS
SOME TREATMENTS	5	5	1	1	1	2	2	ALL TREATMENTS
Confounding	3	3	5	4	3	1	1	Partial balance
NON-FACTORIAL	1	1	2	2	3	5	5	FACTORIAL
SOME EFFECTS	5	5	5	5	4	1	1	ALL EFFECTS
PROPER FRACTION	3	3	4	4	4	1	5	IMPROPER FRACTION
COMPLEX TO DESIGN	4	4	3	1	5	5	3	SIMPLE TO DESIGN
	*	*	*	*	*	*	*	
	*	*	*	*	*	*	confounded designs	
	*	*	*	*	*	fractional factorial		
	*	*	*	*	cyclic designs			
	*	*	*	bib design				
	*	*	pbib design					
	*	latin squares						
	randomised blocks							

FIGURE 8.4 - FINAL MERGED GRID.

From this grid, it should now be possible to form a set of rules for use in selecting a design from the given list. Before doing this, it would be useful to focus the grid to see where any possible relationships lie.

8.4 ANALYSING THE FINAL GRID

The merged grid was focused using Algorithm 6.1, and the two dendrograms showing the relationships between the constructs and between the elements formed. First, the dendrogram showing where possible relationships between the constructs is considered.

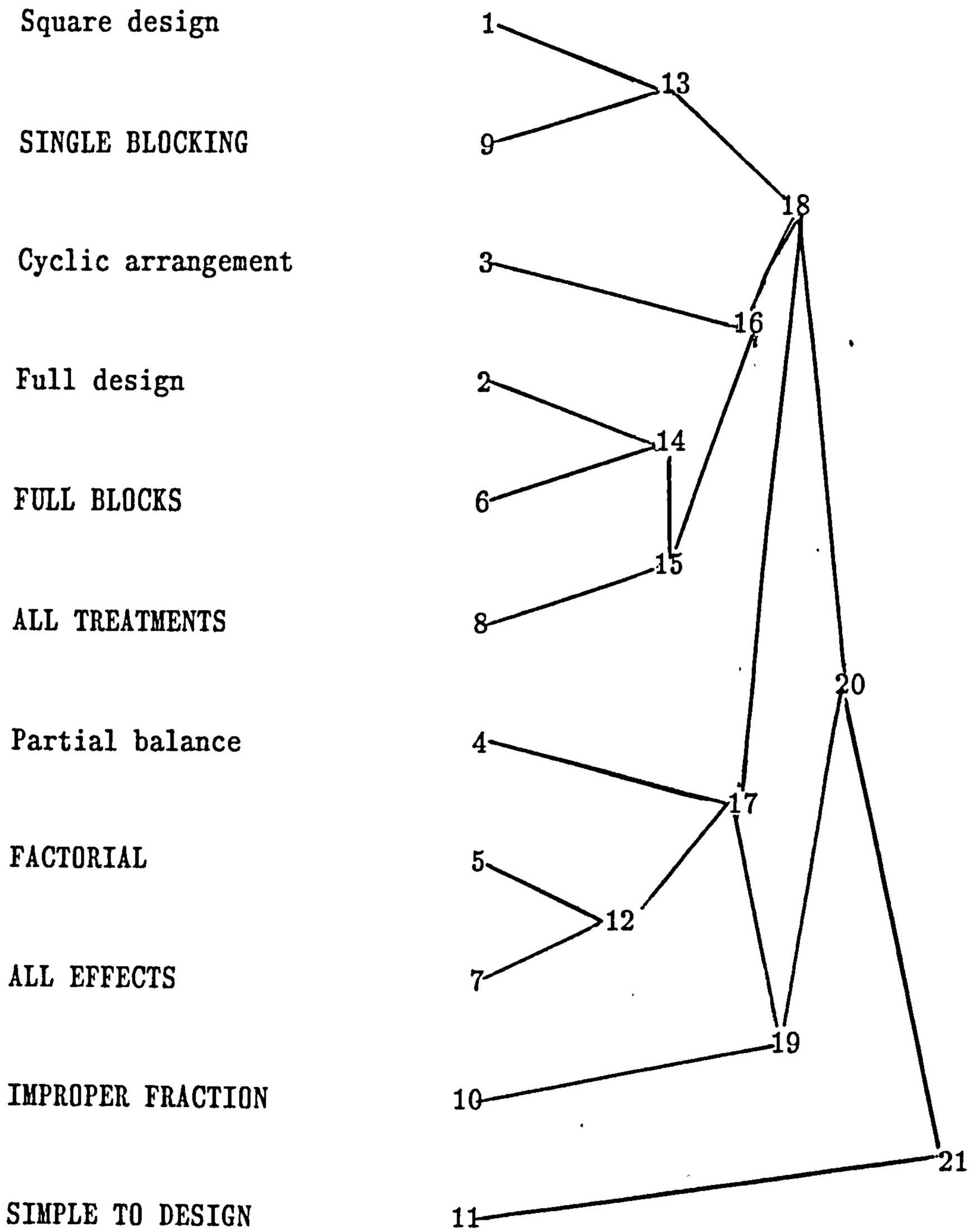


FIGURE 8.5 - DENDROGRAM FOR CONSTRUCTS AFTER FOCUSING.

As can be seen by examining the diagram, the constructs dealing with blocking structures have all been clustered together near the centre. This would indicate that blocking is an important aspect of experimental design in the view of these two people. Naturally this is to be expected, and it also vindicates the choice of blocking as a key question in the diagram produced in Chapter 5. The other constructs are arranged fairly symmetrically around the rest of the tree.

When the elements are considered, then a very interesting picture emerges. The following is the dendrogram for the elements constructed from the merged grid.

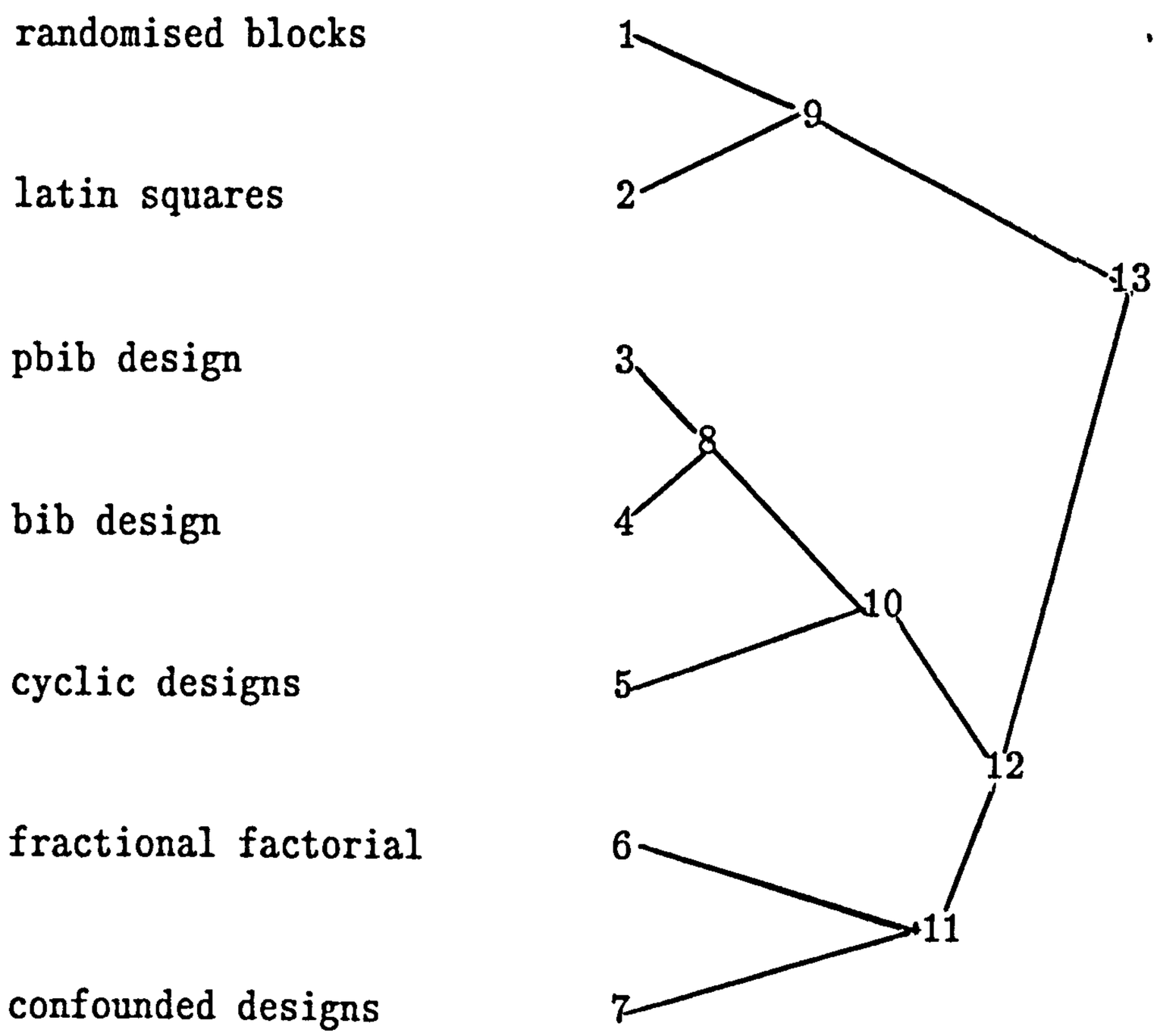


FIGURE 8.6 - DENDROGRAM FOR ELEMENTS AFTER FOCUSING

The diagram shows that the models appear to be fairly well separated. One interesting aspect that comes from an examination of the diagram is the relationship between fractional factorial and confounded designs. This is only to be expected, as confounded designs arose as a generalisation of fractional factorial models. A number of other relationships are illustrated, including the close link between bib and pbib designs and that between randomised blocks and latin squares. As such, it provides a visual check that the method is separating the models sufficiently, as well as showing that similar designs are grouped together.

8.5 RULES FOR EXPERIMENTAL DESIGN MODEL SELECTION.

It should now be possible to form a set of strategic selection rules based upon the contents of this final grid. The SEARCH module within KEIRS was used to generate the following results, with the map shown in Figure 8.3 being compiled using the techniques introduced in Algorithm 7.3

8.5.1 Model Characteristic Rules.

To begin with, the Model Characteristic Rules were generated for the seven models used as elements in the final repertory grid. The following are the rules that were generated using this information, together with the plausibility values rounded to 4 decimal places.

Model/test is randomised blocks if:-

full design

FULL BLOCKS

ALL EFFECTS

ALL TREATMENTS

SINGLE BLOCKING

blocks

NON-FACTORIAL

with plausibility 0.8864

Model/test is latin squares if:-

square

full design

FULL BLOCKS

ALL EFFECTS

ALL TREATMENTS

blocks

NON-FACTORIAL

MULTIPLE BLOCKING

with plausibility 0.8864

Model/test is pbib design if:-

Partial balance

ALL EFFECTS

blocks

fraction

PARTIAL BLOCKS

SOME TREATMENTS

with plausibility 0.8182

Model/test is bib design if:-

ALL EFFECTS

blocks

fraction

PARTIAL BLOCKS

SOME TREATMENTS

COMPLEX TO DESIGN

with plausibility 0.8409

Model/test is cyclic designs if:-

cyclic arrangement

SIMPLE TO DESIGN

PARTIAL BLOCKS

SOME TREATMENTS

with plausibility 0.75

Model/test is frac. factorial if:-

FACTORIAL STRUCTURE

SIMPLE TO DESIGN

fraction

confounding

PARTIAL BLOCKS

SOME EFFECTS

PROPER FRACTION

with plausibility 0.8409

Model/test is confounded designs if:-

FACTORIAL STRUCTURE

FULL BLOCKS

IMPROPER FRACTION

confounding

SOME EFFECTS

with plausibility 0.75

8.5.2 Discussion of MCR's.

As can be seen, each of the model characteristic rules has identified a number of aspects of the model that are considered important. For example, the block designs all have 'blocks' as one of the characteristics associated with them. What is also noticeable is that the rules manage to capture the unique aspects of each model. It will be noticed that the incomplete block designs all have the fact that all effects can be measured as one of the clauses. By contrast, the factorial designs note that only some of the effects can be measured.

When the plausibility values for the rules are examined, the most striking feature is that none of them have been allocated a plausibility of 1. This actually reflects and reinforces a previously stated, yet often neglected aspect of experimental design. It may be that theoretically a particular model is suitable for a given situation. This does not take into account any practical difficulties in the conduct of the experiment which may preclude the use of that model. It is therefore always useful to have a number of models in consideration at the construction stage, with no one perfect model but one appearing to be a better choice than the others available. The plausibility values which are generated here reflect this fact quite neatly.

One final point to note is that the more general a particular model is, then the fewer the clauses in the rule and the lower the plausibility. This is also a useful aspect of the rules. The fact that this occurs indicates the need for as much information as possible to be at hand before a more specific design model is used. It also reflects the fact that some designs are special cases of others. For example, a pbib design can be shown to be a special case of the cyclic class of design. It will be noted that the rules concerning these two design models have a number of clauses in common, but the rule for pbib design has more clauses and a higher plausibility if these clauses are met. This naturally leads to a consideration of the characteristics themselves, and in particular the Characteristic Implication Rules that can be generated from this grid.

8.5.3 Characteristic Implication Rules.

The SEARCH module also generated a list of implications between the characteristics of the model. As there were 11 constructs in total, there was a possibility of as many as 220 implications to work from. However, the program finally generated a total of 132 implications that would be of use. For this purpose the level of the plausibility for a rule to be useful was maintained at 0.7.

From this list of selected implications, it is possible to draw up an implication map using the techniques outlined in Algorithm 7.3. Once the original map had being drawn, it was felt necessary to edit it slightly to remove one or two obvious inconsistencies within the arrangement. The following page shows the final version of the implication map for the merged grid.

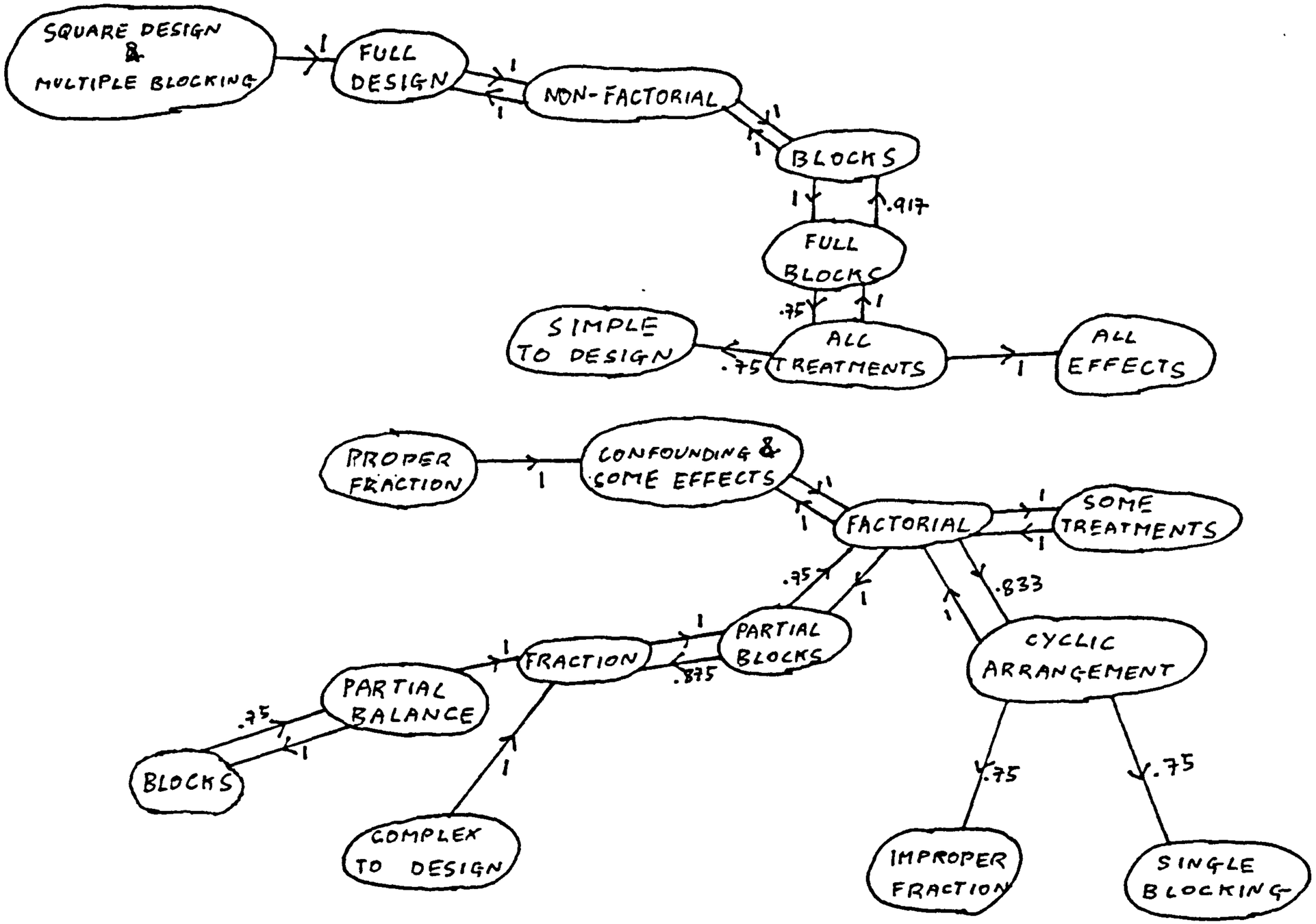


FIGURE 87 - FINAL IMPLICATION MAP

Using this implication map as a basis, it is possible to read off the following set of Characteristic Implication Rules. The numbers after the rules show the plausibility values.

Square design implies Multiple blocking (1.00)

Multiple Blocking implies Square design (1.00)

Square design & Multiple Blocking implies Full design (1.00)

Full design implies Non-Factorial (1.00)

Non-Factorial implies Full design (1.00)

Non-Factorial implies Blocks used (1.00)

Blocks used implies Non-Factorial (1.00)

Blocks used implies Full Blocks (1.00)

Full Blocks implies Blocks used (0.917)

Full Blocks implies All Treatments (0.75)

All Treatments implies Full Blocks (1.00)

All Treatments implies All Effects (1.00)

All Treatments implies Simple to Design (0.75)

Confounding implies Some Effects (1.00)

Some Effects implies Confounding (1.00)

Proper Fraction implies Confounding & Some Effects (1.00)

Confounding & Some Effects implies Factorial (1.00)

Factorial implies Confounding & Some Effects (1.00)

Blocks implies Partial Balance (0.75)

Partial Balance implies Blocks (1.00)

Partial Balance implies Fraction (1.00)

Complex to Design implies Fraction (1.00)

Fraction implies Partial Blocks (1.00)

Partial Blocks implies Fraction (0.875)

Partial Blocks implies Factorial (0.75)

Factorial implies Partial Blocks (1.00)

Factorial implies Some Treatments (1.00)

Some Treatments implies Factorial (1.00)

Factorial implies Cyclic Arrangement (0.833)

Cyclic Arrangement implies Factorial (1.00)

Cyclic Arrangement implies Improper Fraction (0.75)

Cyclic Arrangement implies Single Block (0.75)

8.5.4 Comments on the CIR's.

As stated earlier, a possible set of 220 implications was reduced to one of 132 useful ones. Out of this, an implication map was drawn which produced a set of 32 Characteristic Implication Rules - a reduction of about 85% on the number of rules to deal with. In addition, it is necessary to allow for possible contradictions in the rules by adding ones which state that opposite ends of a construct cannot be true at the same time. This will add 22 rules to this base.

The rules themselves reflect common sense decisions about design model selection. For example, the fact that confounding always leads to only some of the effects being measured is isolated. Similarly, the fact that a square design involves multiple blocking has also being successfully produced. The other rules give a picture of the types of thinking that go on when selecting a suitable model.

8.6 USING THE DERIVED RULES

Having formed our set of strategic selection rules, then some thought can be given as to how the rules can be used in an expert system for model selection. For the purposes of illustration, suppose the expert system uses production

rules in a rule base. The background facts have been established as clearly as possible, and information is available for use in model selection.

Consider first the model selection rules. For randomised blocks, the rule presented in Section 8.5.1 could be given in the following way.

RULE 001

IF: 1) A full design is required

2) Each block contains all treatments

3) All effects are to be estimated

4) All treatments are to be considered

5) Blocking is in a single direction

6) Blocks are to be used

7) The design should be non-factorial

THEN: The design required is RANDOMISED BLOCKS with plausibility 0.82.

Alternatively, the rule for cyclic designs would be

RULE 005

IF: 1) The blocks form a cyclic arrangement

2) The design is simple to construct and understand

3) Partial blocks are to be used

4) Some treatments are in each block

THEN: The design required is CYCLIC DESIGNS with plausibility 0.75.

The plausibility values quoted are initial ones, and will only be true if the facts for each design are known for certain. An earlier module of the system will have established certain facts to be true to some degree of plausibility, which is available to some extent. These facts can be used to amend the plausibility values quoted to give a starter set of partially

activated rules, i.e. some but not all of the clauses for these rules have been fired.

The initial facts that are available can then be used to ascertain how plausible it is that any other facts are true. For example, suppose it is known that a non-factorial design is required. The rule for selecting a randomised block design requires that a full design is needed. Also required is that all treatments are used in each block, but this has not been previously ascertained. The following rule would be used.

RULE 101

IF: 1) The design should be non-factorial

THEN: Full blocks will be used with plausibility 1.00

with this initial plausibility, then the following rule can also be used

RULE 110

IF: 1) Full blocks will be used

THEN: All treatments are to be considered with plausibility 0.75.

In this way, a measure of how plausible the designs under consideration are can be constructed based on the known facts A choice can then be made of the 'best' choice for the given situation

CHAPTER 9

AN EXPERT SYSTEM FOR MODEL SELECTION.

9.1 INTRODUCTION.

Now that a set of rules for selecting a design can be constructed, it remains to consider how the other ideas presented in this thesis can be brought to bear on a system for experimental design, or indeed any statistical problem. It will also help to show how the examples given in this thesis relate to these ideas.

9.2 THE OVERALL STRUCTURE

Chapter 3 presented a set of proposals to consider when setting up an expert system. How would they be related to the problems considered? Let us consider each one in turn.

3.1 For a statistical system to effectively model the way in which as consultant works, a modular system would be most suitable.

ExpCheck (Chapter 5) is an example of a system with a modular structure, each module covering one area of experimental design. For a system to construct a design, the following proposal has also to be considered.

3.2 The following modules would be required.

- A preliminary module to establish the problem.
- A selection module to choose a model.
- A report module to give the results.

The preliminary module could use the Tree of Knowledge (Section 4.2) to establish the background and any relevant facts. Having done this, either the rules outlined in the previous chapter, or else a selection network such as that illustrated in Figure 5.3, can be used in the selection module.

Reporting of the suitable design requires a knowledge of the way the client perceives the problem. This brings in the next three proposals which were made.

3.3 The style of questioning and answering must be sufficiently clear that the salient facts can be discerned.

3.4 The way the system presents the results must be understandable to the user and reflect their experience.

3.5 The rule base must be capable of handling facts as well as figures.

9.3 HOW THE SYSTEM WORKS.

So how would the system show the way in which it is working? In other words, how will the results be presented to the person using it? Let us look at some more of the proposals made in Chapter 3.

3.6 The selection of a model should not be made until all relevant information is available and processed.

There is nothing wrong in presenting a set of interim results, but the final selection of a model should not be made until no further facts can be deduced. Of course, the deductions made may show inadequacies in the rule base that may require attention. The following would then be appropriate.

3.7 The rule base must be capable of updating and correction.

In what order should the questions the system needs to address to the user be asked? This will be determined largely by the way the facts are determined, i.e.

3.8 Use what is already known to control the order of questioning.

Above all, the client's needs and views have to be recognised in some way. Proposal 3.9 stated that

- The system should aim to accurately meet the client's needs as simply as possible.
- The explanation are as clear as possible.
- Help the client to understand the problem.

ExpCheck is again an illustration of how this works in practice, with simple explanations of terms and questions available and summaries of the deductions made at various stages.

Above all, it must be remembered that no-one knows everything. This is equally true of an expert system. As proposal 3.10 says, if the system cannot handle a problem it should say so. Similar, it should be clear that deductions are only true in the context of the experiment (proposal 3.11), and arrangements are made for the results to be securely stored (proposal 3.12).

Now that the structure of a system has been established, the next problem is how to establish the background of the problem we wish to investigate. The Tree of Knowledge as introduced in Chapter 4 provides a good method for doing this, especially if constructed along the lines expressed in Figure 4.1. Remember also that the use of the tree allows estimates of the initial plausibility of the models that are being considered. The first module of a design system would be devoted to such a system, which would have initial plausibility values determined by either one expert or several in the same field. This would enable the following points to come into action.

4.1 The depth a user has to go into the tree can help determine the client's understanding of the problem.

4.2 The depth a person goes to find a suitable description enables a good estimate of the initial plausibility of the models available.

Once the background has been established, the next module of the system would build on this description to form a fuller picture of the problem. In other words:-

5.1 Once the initial description has been found, the system should build on that by discovering and determining new facts.

Of course, to know which facts are needed, one needs to know what the key relevant facts for a problem are likely to be. For experimental design, these would include the experimental units, sources of possible variation, and so on. The questions that this module should ask must therefore reflect the following points.

5.2 The facts required should be determined at the start.

5.3 Ask Key Questions (Definition 5.1) at critical points and find supporting facts to build on these.

5.4 Ask the same questions in different ways to check facts.

5.5 When a fact is established, use that to help determine other facts.

Finally, this module must be able to 'change its mind', i.e. alter facts which are shown to be incorrect. The facts that are determined should, naturally enough, be usable in the strategic selection rules, i.e. the rules are based on the facts determined here.

9.4 CONCLUSIONS

The methods and algorithms presented in this thesis form a set of tests for the development of statistical expert systems, and for the formation of rules to guide that system. There are a number of possible developments that could be made to improve and widen the scope of these methods.

The simultaneous combination of three or more grids was mentioned in Chapter 7. Methods for differentiating between the consultants, and weighting schemes to give greater emphasis to more experienced consultants, would help to provide a more useful set of rules for a particular area.

Indeed, the scope of strategic selection rules need not be confined to statistical expert systems. There is no reason why the methods could not be applied in other areas for rule generation, such as medical systems and management techniques.

Methods of actually implementing strategic selection rules within an expert system also need to be explored. Improvements should be made to both software packages (ExpCheck and KEIRS) to make them more user-friendly and easier to use. When the rules are ready, what questions could be asked to elicit the facts used by the rules, and could an initial value of plausibility values be asigned to the possible answers? These are questions that would need to be considered before any fully practical application of the rules was made possible.

Finally, professional developments within statistics may have an impact on the methods presented in this thesis. The new professional code of conduct, for example, affects the way statistical consultants work - should the guidelines be updated to reflect these changes?

One point needs to be made clear to finish. Every point raised in this thesis can be summed up in one statement:-

> Facts are the key consideration in the construction of statistical expert systems. Choose your facts, determine them, then make a decision.

So we return to the idea that facts should be determined before theories, not theories before facts. Rather like Sherlock Holmes, we work with facts and statistical expert systems should use them as constructively as possible. With this always borne in mind, the techniques presented in this thesis will help in the construction of effective, useful and working expert systems.

APPENDIX

CONTENTS OF SOFTWARE DISKS.

The following files are included on the disk attached to the cover marked as disk A.

SUITE.EXE	The program for test selection rule formation.
SUITE.AWS	The uncompiled source file for SUITE.EXE.
PEGASUS	Functions for repertory grid formation.
FOCUS	Functions for grid examination.
SEARCH	Functions for rule formation.
REVIEW	Functions for grid amendment.
RESTART	Functions for grid reformation.
COMPARE	Functions to form rules from two grids.
CORE	Functions to merge and 'prune' two grids.
SAMPLE	Grid used in Chapters 6 and 7.
GRIDA	The first grid for Chapter 8.
GRIDB	The second grid for Chapter 8.
MERGEB	The grid used for the rule formation.

Disk B contains the programms for ExpCheck as follows.

EC.EXE	The program itself.
EXPCHECK.AWS	The original workspace.
EXPQUEST	The module control functions.
SECTION 1	Questions for section 1.
SECTION 2	Questions for section 2.
SECTION 3	Questions for section 3.
SECTION 4	Questions for section 4.
SECTION 5	Questions for section 5.
SECTION 6	Questions for section 6.
SECTION 7	Questions for section 7.
SECTION 8	Questions for section 8.
EXPDEFS	Definitions.
EXPWHY	Explanations for all questions.

User's guides are available from the author if required.

BIBLIOGRAPHY.

A.S.A. COMMITTEE ON CODE OF CONDUCT (1980) *Interim Code of Conduct, American Statistical Association.* Amstat News, 69, 20-21.

ASPIN (1948) *The problem of comparing two mean values.* Biometrika, 35, 88-96

BARR, A. & FEIGENBAUM, E.A. (1981) *The Handbook of Artificial Intelligence, V.I.* HeurisTech Press, Stanford, CA.

BASKERVILLE, J.C. (1981) *A systematic study of the consulting literature as an integral part of applied training in statistics.* American Statistician, 35, 121-123.

BOEN, J.R. & FRYD, D. (1978) *Six-State Transactional Analysis in Statistical Consultation.* American Statistician, 32, 58-60.

BOEN, J.R. & ZAHN, D.A. (1982) *The Human Side of Statistical Consulting.* Lifetime Learning Publications, Belmont CA.

BOX, G.E.P. (1976) *Science and Statistics.* Journal of the American Statistical Association, 71, 791-799.

BOX, G.E.P. (1979) *Some Problems of Statistics and Everyday Life.* Journal of the American Statistical Association, 74, 1-4.

BRIEMAN, L. (1984) *Classification and Regression Trees.* Chambers, London.

BROSS, I.D.J. (1974) *The Role of the Statistician: Scientist or Shoe Clerk.* American Statistician, 28, 126-127.

BUCHANAN, B. & SHORTLIFFE, E.H. (1985) *Rule-Based Expert Systems.* Addison-Wesley, New York.

BUTLER, K.A. & CORTER, J.E. (1986) *Use of psychometric scaling methods for knowledge acquisition: A case study.* in:*Artificial Intelligence and Statistics*, ed. W.A. Gale, 295-319.

CHAMBERS, J.M. (1981) *Some Thoughts on Expert Software.* Computing Science and Statistics: Proceedings of the 13th Symposium on the Interface, 36-40.

CHANTER, D.O. (1978) *Consultancy: Communication and the Statistician.* Bulletin in Applied Statistics, 5, 49-56.

DANIEL, C. (1969) *Some General Remarks on Consultancy in Statistics.* Technometrics, 11, 241-246.

DEMING, W.E. (1965) *Principals of Professional Statistical Practice.* Annals of Mathematical Statistics, 36, 1883-1900.

DEMING, W.E. (1966) *Code of Professional Conduct.* Sankhya B, 28, 11-18.

DEMING, W.E. (1972) *Code of Professional Conduct: A Personal View.* International Statistical Review, 40, 215-219.

EISENHART, C. (1947) *The Role of a Statistical Consultant in a Research Organisation.* Proceedings of the Third International Statistics Conference, 309-313.

FORSYTH, R. (1987) *BEAGLE user's guide.* Supplied by author.

GALE, W. A. & PREGIBON, D. (1983) *An Expert System for Regression Analysis.* Computing Science and Statistics: Proceedings of the 14th Symposium on the Interface, 110-117.

GALE, W. A. & PREGIBON, D. (1984) *Constructing an Expert System for Data Analysis by Working Examples.* COMPSTAT 1984, 227-236.

GOLDSMITH, P. (1987) *CAED User's Manual.* ICI Chemicals Ltd.

GREENFIELD, A.A. (1979) *Statisticians in Industrial Research: The Role and Training of a Statistical Consultant.* The Statistician, 28, 71-82.

GREENFIELD, A.A. (1983) *The Polymath Consultant.* Proceedings of the 1st International Conference on the Teaching of Statistics, 641-645.

HAHN, G.J. (1985) *More Intelligent Statistical Software and Statistical Expert Systems: Future Directions.* American Statistician, 39, 1-16.

HAHN, G.J. & MORGAN, C.B. (1988) *Design Experiments with your computer.* ChemTech, Nov., 664-669.

HAND, D.J. (1984) *Statistical Expert Systems: Design.* The Statistician, 33, 351-369.

HAND, D.J. (1985) *Statistical Expert Systems: Necessary attributes.* Journal of Applied Statistics, 12, 19-27.

HAND, D.J. (1986) *Expert Systems in Statistics.* Knowledge Engineering Review, 1, 2-10

HAND, D.J. (1987a) *Statistical Expert Systems. A catalogue of current researchers in area.* Personal Communication.

HAND, D.J. (1987b) *A Statistical Knowledge Enhancement System.* Journal of the Royal Statistical Society (Series A), 150, 334-345.

HAND, D.J. & EVERITT, B.S. (1987) *The Statistical Consultant in Action.* Cambridge University Press, Cambridge.

HAYES-ROTH, F., WATERMAN, D.A. & LENAT, D.B. (1983) *Building Expert Systems.* Addison-Wesley,New York.

HUNTER, W.G. (1981a) *The practice of statistics: The real world is an idea whose time has come.* American Statistician, 35, 72-76.

HUNTER, W.G. (1981b) *Six Statistical Tales.* The Statistician, 30, 107-117.

HYAMS, L. (1971) *The Practical Psychology of Statistical Consultation.* Biometrics, 27, 201-212.

JEFFERS, J.N.R. (1977) *Statistical Checklist 1: Design of Experiments.* Institute of Terrestial Ecology.

JOHN, J.A. & QUENOUILLE, M.H. (1977) *Experiments: Design and Analysis.* London, Griffith.

JONES, B. (1980) *The Computer as a Statistical Consultant.* Bulletin in Applied Statistics, 7, 168-195.

JONES, B. (1991) *Statistical Software Reviews*: DEX (with response by A.A. Greenfield). Applied Statistics, 40, 491-493.

KELLY, G.A. (1955) *The Psychology of Personal Constructs*. Norton, N.Y.

KIMBALL, A.W. (1957) *Errors of the third kind in statistical consulting*. Journal of the American Statistical Association, 52, 133-142.

MARQUARDT, D.W. (1979) *Statistical Consulting in Industry*. American Statistician, 33, 102-107.

NATIONAL ALGORITHMS GROUP (1988) *NAG Fortan Library Mk.13*, Volume 6.

NEISS, A., JOECKEL, K.H., KNAPPEN, F. & NOWAK, H. (1985) *Does the result of the statistical evaluation depend on the choice of the statistical consultant*? Statistical Software Newsletter, 11, 69-75.

PREGIBON, D. & GALE, W.A. (1984) *REX: An Expert System for Regressin Analysis*. COMPSTAT 1984, 242-248.

SHAW, M.L.G. (1979) *On becoming a personal scientist*. Academic Press, London.

SHAW, M.L.G. (1988) *Interactive Elicitation and Exchange of Knowledge*. International Journal of Personal Construct Psychology,

SMITH, J.T. (1986) *Teaching statistical consulting using videotape*. A.S.A. Proceedings of Statistical Education Section, 159-162.

SPIEGELHALTER, D.J. & KNILL-JONES, R.P. (1984) *Statistical and Knowledge-based Approaches to Clinical Decision-support Systems, with an Application in Gastroenterology*. Journal of the Royal Statistical Society (Series A), 147, 35-77.

SPRENT, P. (1970) *Some Problems of Statistical Consultancy (with discussion)*. Journal of the Royal Statistical Society (Series A), 133, 139-165

WETHERILL, G.B. & GERSON, M.E. (1985) *The Quality of Data*. Internal Report, University of Newcastle upon Tyne.

WETHERILL, G.B. & GERSON, M.E. (1986) *Quality Assurance for Data Entry - an Integrated Approach.* Third International Workshop on Statistical Quality control, 123-158.

WILLIAMS, M.K. (1986) *Data Entry and Validation using Expert System Techniques.* Internal Report, University of Newcastle upon Tyne.

WILLIAMS, M.K. (1988) *EXPCHECK: An Intelligent Diagnostic System in APL.* Vector, 4.2, 83-88.

WILLIAMS, M.K. (1989) *Personal Construct Analysis and Rule Formation in APL.* Vector, 6.1, 72-85.

WOLSTENHOLME, D.E., O'BRIEN, C.M. & NELDER, J.A. (1988) *GLIMPSE: A Knowledge-Based Front End for Statistical Analysis.* Knowledge-Based Systems, 1, 173-178.

ZAHN, D.A. & ISENBERG, D.J. (1983) *Non-Statistical Aspects of Statistical Consulting.* American Statistician, 37, 297-302.