Exocyclic

Elsevier - Biosoft
Manual

by
R. Rivers

EXPERT 4

by R. Rivers

Manual

Published by
Elsevier Science Publishers BV
P.O. Box 1527
1000 BM Amsterdam
The Netherlands

Distributors worldwide
Elsevier – BIOSOFT
68 Hills Road
Cambridge CB2 1LA
United Kingdom

Copyright © 1984, 1985 R. Rivers
PRINTED IN THE UNITED KINGDOM

EXPERT4 - Biomedical Licence Agreement

EXPERT4 is supplied to you under licence no. _____, by the biomedical distributor Elsevier Biosoft. Under the terms of this licence you can run a single copy of EXPERT4 on any machine at any one time. You cannot run multiple copies of the program simultaneously without obtaining a separate licence for each copy that is run simultaneously.

You can use EXPERT4 to specify knowledge in biomedical related domains. All rights to these knowledge domains remain with their own authors (referred to below as 'domain authors') and these biomedical domains can be sold by domain authors subject to the prevailing laws and regulations that operate in the countries of sale. All responsibilities and duties associated with the sale of knowledge domains are assumed by the domain author.

This licence specifically excludes the use of EXPERT4 for the development and marketing of any domains other than those related to biomedical subjects.

A domain author may enter into agreement with the distributor to distribute his or her biomedical domain with the EXPERT4 Run Time Module. The program author must be informed of all such agreements by either the domain author or the distributor prior to the commencement of distribution.

No part of EXPERT4 including all electronic media and both electronic and other documentation may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language in any form or by any means electronic, optical, magnetic, chemical, mechanical, manual or otherwise without the prior written permission of the author, Rod Rivers, c/o G5 EXPERT SYSTEMS, 18 Sarah House, Arabella Drive, London SW15 5LN, England, and the biomedical distributor, Elsevier Biosoft, 68 Hills Road, Cambridge, CB2 1LA, UK.

The author and distributor make no representation or warranties with respect to the program or documentation and specifically disclaim any implied warranties of merchantability or fitness for purpose. Where the program is used in conjunction with a knowledge base the author and distributor make no representations or warranties with respect to the knowledge base and specifically disclaim any consequential losses made from application of the program and domain in an advice-giving capacity or otherwise.

EXPERT4 - User Technical Documentation

Copyright Rod Rivers 1984

Table of Contents

	Page
Contents Check-list	1
Running Environment	2
Hardware	2
Files	2
Memory Considerations	2
Introduction to the program	3
Sample Applications	4
Medical	4
Expert System Development	4
Market Research	5
Education and Training	5
Research	5
Background and Technical Overview	6
Program Concepts	8
Domains and Elements	8
Attributes and Attribute Values	9
More on Attribute Values	10
Attribute Value Tables	10
Inferencing and Unknown values	12
The Expert System Development Cycle	14

Program facilities 15

Introduction to facilities

The Knowledge Base Management Routines 16

Create a New Domain 16

Load a Domain 16

See/edit/add Elements 16

See/edit/add Attributes 16

See/edit/add Attribute Values 17

See/edit Spreadsheet 17

Save the Domain 17

Exit 17

The Calculation and Display Facilities 18

See Typicality Index 18

Query by Example 20

Element Comparisons 22

Comparison of Attribute Values 23

Attribute Value Distributions 24

Attribute Cross Tabulations 26

See/edit Search Profile 27

Creating Sub-domains 29

See Attribute Statistics 29

See Probability of Immediate Classification 31

Create Sub-domains 34

User Tutorial Documentation 37

Contents Checklist

EXPERT4 is supplied on a single magnetic disk which contains the following files:

X4.EXE	The program
X4DEMO.EXE	A demonstration and training version of the program
X4DEMO.LOG	The training module
ANIMALS.X4A	The demonstration domain
X4.MSG	The messages file

The documentation for EXPERT4 bound into this manual comprises:

USER TECHNICAL DOCUMENTATION	
USER TUTORIAL DOCUMENTATION	Used in conjunction with the X4DEMO.EXE program

If you do not have the complete set of programs and documentation consult your distributor. In addition to the above your distributer may supply you with demonstration domains and accompanying documentation in your application area.

Running environment

EXPERT4 (V1.3) is supplied for running on the following minimum configuration:

An IBM PC with 128k RAM
Monochrome monitor and controller
Single floppy disk drive
PC DOS operating system

Files

EXPERT4 creates a single file for each domain created on the default drive. When saving the size of a domain must not exceed the remaining space on a disk. A typical domain may have a 5-10k file, and the number of domains that can be held on disk at one time will vary depending upon the size of the individual files.

The user enters the file name when creating a domain. The program adds the extension '.X4A' to the 8 character file name the user provides. The constraints on which characters can be used in the file name are the same as those for the operating system and the user should refer to the operating system manual if he is unsure about these. In general, user supplied names can be up to 8 characters long and can include alphabetic and numeric characters but should exclude punctuation marks like commas and semi-colons.

When loading a file from disk EXPERT4 asks for an 8 character file name. The program adds the extension '.X4A' and expects to find this when it looks for the file in the disk directory. Users can change file names using the operating system facilities, but should maintain the '.X4A' extension. Users can also save the file under a new name from within the EXPERT4 program.

Memory considerations

The complete domain is loaded into (RAM) memory at the start of a session and written back to disk whenever the user saves the domain. The RAM must therefore be big enough to take the whole domain. The size of the domain is a function of the number of elements, the number of attributes and the number of attribute values. For a particular computer configuration the maximum sizes for each of these values must be set in the source code version. No facilities exist in version 1.3, for the end user to configure the program. The amount of memory actually used will also be partly dependent upon the length of strings entered.

Domains of 48 elements and 20 attributes have successfully been run on the minimum configuration.

Introduction to the program

EXPERT4 is a program for creating, exploring and teaching about expert knowledge. The user can enter and store knowledge in the form of facts, into a knowledge base. A knowledge base is similar in some respects to a database but offers new and interesting facilities for searching, sorting, summarising and processing knowledge.

A knowledge base (or domain) is any group of objects in the same way as a database is any group of records holding information about a particular subject (e.g. employees, medical cases, products, projects etc.). With EXPERT4 you can enter information into a knowledge base and retrieve it in much the same way as you would using a database, but in addition you can process and search the knowledge in ways that database systems do not offer.

From the user's point of view EXPERT4 is half way between a database management system and a spread sheet. Knowledge is held in tables which are entered and edited using a screen based spreadsheet format similar to those found in popular spreadsheet programs used for arithmetic calculations. The difference is that with a conventional spreadsheet the cells contain numbers and formulas, in EXPERT4 the cells contain facts about the objects in the domain. The facilities provided within EXPERT4 allow you to explore the relationships between the facts held in the knowledge base, and to summarise them in a variety of ways. In this way EXPERT4 goes a step beyond being a mere repository of data or knowledge and gives both expert and novice new insight into the way knowledge in the domain is structured. In this respect EXPERT4 has a relationship to programs which calculate descriptive and predictive statistics from data.

Sample Applications

EXPERT4 can be used in a variety of ways and for a variety of different purposes. If you are intending to develop domains for resale please refer to your licence agreement which may contain restrictions on the domain subject area. Some typical applications are:

Medical

A doctor may be an expert in the diagnosis of a particular set of diseases. To further his/her understanding of the relationships between these diseases the doctor may already use the assistance of a computer for holding cases in a database, and perhaps doing some calculations to test the results of clinical trials of alternative treatments using either a spreadsheet or a statistical analysis package. EXPERT4 allows the expert to express knowledge about the domain in a more natural way than a database package will allow. It will also allow exploration of a domain in a more direct and natural way than a spreadsheet or statistics package can offer. EXPERT4 will allow the doctor to find 'typical cases' or cases that are similar to a particular case he is considering. He can produce 'prototype' diagnoses which are helpful in distinguishing between cases. Most importantly, the combination of the knowledge the expert has entered and the facilities that EXPERT4 offer provide an EXPERT SYSTEM which can be consulted for its advice, used for training novices or distributing expertise in the domain quickly and cheaply.

Expert system development

A software house or individual system developer might be interested in developing an expert systems product for resale to an identified market. They could use EXPERT4 as a development tool for knowledge harvesting and prototyping. It can also be used as the delivery vehicle. The development of knowledge bases with EXPERT4 is encouraged. Where the knowledge bases have a marketable value and EXPERT4 is the delivery software they can be resold if accompanied by an EXPERT4 run time licence agreement, the cost of which is kept low to encourage knowledge base development. Your EXPERT4 distributor may consider marketing a knowledge base on your behalf and may be able to offer you other knowledge bases that have been developed by other software houses.

Market Research

A market research organisation might be interested in comparing the characteristics of products in a particular market. They might use EXPERT4 to keep track of products in the market, compare and contrast the products with each other, and recommend the characteristics that products must have to be identified with particular markets. They can gain insights into the psychological mechanisms and reasoning processes that a consumer might be using in his perception of the products and markets. They can pass on this information as recommendations to the product design team and the public relations department of their client organisations.

Education and Training

A teacher might use EXPERT4 to illustrate the comparisons and differences between items in a domain (e.g. 20th century novels, chemical elements, historical characters, economic factors of countries, electrical circuits etc.). The teacher can convey not only the contents of the domain but also the structural relationships that hold between the elements. Students can be set the task of specifying a domain, or a domain can be supplied for the students to explore and add to. Project work can be based around the task of specifying, exploring, abstracting and writing up the product of building an expert system.

Research

A researcher might use EXPERT4 to structure aspects of the subject of the research or references to papers and books on the subject. The program can help to focus the research and provide material for the publication of research papers. By providing a format for knowledge exchange it can be used to enhance the effectiveness of communication between researchers, and as a vehicle to exploit the products of a research project.

Background and Technical Overview

This section explains some of the capabilities and limitations of EXPERT4 and puts it into perspective against other software products and the academic disciplines out of which the EXPERT4 approach emerged. For the non-technically minded the discussion is not essential reading for the operation of the program, although it will help all, at some stage, to understand the rationale for EXPERT4. Refer to the 'USER TUTORIAL DOCUMENTATION' for further instructions on running the program, or skip this section and go on to Program Concepts.

EXPERT4 is one of a new breed of programs. These are called knowledge base management systems (KBMS). The name is chosen deliberately to highlight the similarity with database management programs (DBMS). The two types of system both deal with 'things' that are stored and retrieved, and the emphasis of both is on efficient methods of doing this, involving searching and presenting summary information. It is the methods of searching and calculation of summary information that differentiates the two types of system. The data for KBMSs is 'knowledge' and may be in the form of concepts rather than the harder data usually stored in DBMSs. 'Knowledge' is processed in people by reasoning processes. These are different from the types of process used in database management.

EXPERT4 is not so much a simulation of human reasoning processes as an interface to a knowledge base which makes certain assumptions about human reasoning. Some of these assumptions are based on psychological research into categories and concepts, while others have been developed ad hoc, but with regard to a particular model of reasoning.

The design of EXPERT4 has emerged from a number of different disciplines. It draws on Logic, Multidimensional Statistics, Cognitive Psychology, and Artificial Intelligence. Its relations to psychology are discussed throughout this document and its relations to some of these other disciplines are discussed here.

The model adopted is essentially fuzzy. EXPERT4 has the capability to induce deductive rules but its emphasis is on fuzzy reasoning. It is based on a probabilistic view of reasoning. The probabilities are calculated from the instances currently stored in the knowledge base and are not input by the user. Measures of central tendency are based on 'most frequent value' logic, rather than by calculating mean or median values.

EXPERT4 stands between deductive logic and ordinal scale statistics, and has some features of both. It deals mainly with nominal categories of data and can convert these to ordered lists. In this way it turns low grade data into higher grade data, or rather it acts to organize the data. This is an especially important area for program development in knowledge

based programming. Logic programming languages, such as PROLOG, are most suitable when the application demands deductive reasoning. They are efficient because through the use of appropriately selected high level rules, a complex knowledge domain can be represented and interrogated, in a compact and efficient way. However, PROLOG does not help us in discovering or selecting appropriate rules, and indeed, this is the biggest problem that stands in the way of creating useful knowledge based systems.

EXPERT4 is designed to help in the discovery and selection of appropriate rules. A knowledge domain is stored in the form of a matrix which relates elements to values on attributes. The elements can consist of all the elements in a domain (e.g. a list of customers, diseases, computer programs etc.) or a sample of all elements. The object of EXPERT4 is to allow the user to explore and organize this domain, with a view to assisting him in forming descriptions of it for particular purposes, which are more efficient in storage and for further processing.

Ordinal and interval scale relational statistics are really special purpose tools that can only be validly applied to specialized types of data that conform to various assumptions. Often these assumptions cannot be made about the data. EXPERT4 makes only one assumption about the data - that the elements can be sorted into into mutually exclusive classes a number of times.

By relaxing the number of assumptions that need to be made EXPERT4 makes some allowance for dissimilar elements to be mixed in a domain. Using ordinal and interval scale relational statistics, it often does not make sense to compare dissimilar elements. Knowledge tends to have disjunctive relations to other knowledge, and disjunctive relations must be dealt with at the level of classes rather than the level of ordinal or interval relations.

Storing data at this level of abstraction encourages the forming of more broadly based domains. In a database management file very specific information on similar items is held. Using a knowledge based system, a much broader domain can be formed which shows the logical (or disjunctive) relations between the data items. It is less precise but it has greater perspective.

This is not to say that the user does not have to put any effort at all into the data collection and preparation. However, it does minimise these factors. Anything from an already well established and stable taxonomy to a brainstormed first approximation of a domain can be entered. In the first instance the knowledge base can be immediately utilized to classify new objects. In the second instance the knowledge base can be expanded and modified until it reaches a satisfactory state of order to be used to classify new objects.

Program Concepts

Domains and Elements

A domain can be more or less any group of objects. For example a domain could be 'medical cases' or 'diseases'. Just as easily it could be 'trees', 'animals' , 'cars', 'Acts of Parliament', 'books' or whatever. In fact all generic nouns are possible domains. Domains can be abstract, for example, 'plans', 'scenes', or 'events'. Any group or class of objects can be a domain, and the elements of the domain would be members of the group or class.

The elements within a domain are particular examples of the domain. So an element could be a specific medical case, disease, tree etc. The elements within a domain have attributes in common and can, in general, can be compared and described in terms of these attributes.

Note that creating a domain which contains a mixture of elements from two or more different generic nouns (e.g. animals and cars) will mean that each noun will be describable using different sets of attributes. There may be some overlap between the nouns in terms of attributes (e.g. 'speed of movement' applies to both animals and cars), but often there is little overlap. EXPERT4 is recommended for domains that consist of elements from a single generic group. Of course you can create as many domains as you like, just so long as the elements in each one are all from the same group. In this respect EXPERT4 requires a top down view of what you intend to use it for. You should define a domain first, then pick elements which belong to it, rather than think of all types of elements at random then find suitable domains for each (although you could have a preliminary session with EXPERT4 to do this too, if you prefer to work from the bottom up).

Let's assume that you have chosen to work with the domain 'Animals'. You could be more or less precise about what you might mean by 'Animals'. You could mean 'the name of as many animal species as I can think of' (e.g. cat, dog etc). This would be a quite general domain. Or you might be thinking of something much more specific like the names of all the animals in your farm yard (e.g Daisy the cow, Spot the dog etc.). It is useful to think of your domain in terms of generality because the way in which you describe the elements in your domain will vary depending upon exactly what you intend to include or exclude as elements. It is possible to be imprecise about what you will include, and divide into sub-domains later those groups of elements within a domain that seem to group together.

Having decided what domain you are going to create you can now enter some elements. You don't need to enter many as you can add more later. Also you can change the name of the element later if you misspell it or think of a better name. We'll assume that you mean names of animal species and that you have entered five or six (e.g. cat, dog, horse etc.).

You can now enter some attributes. In entering attributes you are describing the domain in terms of a set of characteristics which can apply to all or at least most of the elements. For example, all the elements, dog, cat, horse etc can be described in terms of their natural habitat, what they eat and their relation to man. These then are the attributes on which the elements in the domain can be described.

Attributes and Attribute values

Exactly what an attribute is will vary from domain to domain. We have seen that in the domain 'Animals', which contains the elements 'Dog', 'Cat', 'Horse' etc. you can have attributes like 'natural habitat', 'Relation to man' etc. In the domain 'stamp collection' which has as its elements particular stamps in the collection, good attributes might be 'colour' or 'postmark'.

An attribute is a characteristic which elements can share. Elements may vary in their characteristics, so the natural habitat of a dog is not the same as that of a cat, and the colour of a particular stamp can be any of several different colours. So each attribute can take on any of a range of attribute values. The attribute 'colour' can take on any of the attribute values 'Red', 'Blue', 'Green', 'Yellow' etc.

There are several different types of attribute. The attribute 'colour' can take any of several values, while the attribute 'postmarked' can only take on two - 'marked' and 'unmarked'. It may be possible to think of some other set of values for 'postmarked' (e.g. 'clearly readable', 'postmarked before 1900' etc), but we'll assume for the moment that 'postmarked' can only take on two values. When an attribute can only take on either of two values (or any of three, in fact, because any attribute can have the value 'unknown' - see later), the way in which the attribute is named can always be changed so that the attribute values become 'yes' or 'no' (the complete set is 'unknown', 'yes', 'no'). For 'postmarked' the name could be changed to 'postmark present', which is like asking the question 'is there a postmark ?'. This type of attribute is called a 'feature'. So a feature is a type of attribute which has 'unknown', 'yes' and 'no' as its possible values.

More on Attribute Values

Attribute values can sometimes be thought of as properties. So a particular stamp has the property 'blue' on the attribute 'colour', and a particular animal species has the property 'domestic'on the attribute 'relation to man'. There can be some confusion over whether something can be regarded as an attribute or an attribute value. For example, treating 'blue' as an attribute rather than an attribute value, means that it can have its own values. These could be 'pale', 'royal', 'navy', 'dark' etc. or blue could be treated as a feature, so 'blue present?' can take the values 'yes' and 'no'. How you treat something, whether as an attribute or an attribute value, is a decision you have to make and will depend on the nature of the domain and what you are trying to do with it.

For example if you were creating the domain to compare different famous oil paintings in general terms, you might have as the attributes 'painter's name', 'period painted' 'style' and 'use of colour'. If you wanted to be more specific and compare the oil paintings in detail in terms of colour it would be useful to have all the attributes as different colours and the attribute values being more detailed specification of these - 'pale', 'dark' etc.

Attribute Value Tables

In a specific domain, elements, attributes and attribute values relate to each other in a particular way. To describe the domain fully we need to say not only what the elements, attributes and possible attribute values are but also which elements have which values on the attributes. This is like saying that 'dog' has the value 'domestic' on the attribute 'relation to man' or that 'penny black' has the value 'black' on the attribute 'colour' in the 'stamp collection' domain. There are many ways in which this type of relationship can be expressed. For example:

In natural English: dogs are domestic animals

(this form only implicitly refers to the attribute 'in relation to man' but specifies the domain 'animals')

As a formal relation:

Animal	: Dog	: Relation to Man	: Domestic
(Domain	:Element:	Attribute	: Value)

In a table: Name of table: Animals

	Relation to man
Dog	Domestic

In general:

Name of table: Domain

Element	Value

A table with many elements and attributes would look like:

	Relation to man	Where often found	No. of legs
Dog	Domestic	Dog Basket	4
Cat	Domestic	Around house	4
Horse	Domestic	Stable Yard	4
Cow	Farm	Field	4

The elements are down the side, the attributes are along the top and the entries in the table are the particular values of the attributes that those particular elements have.

In EXPERT4 once you have entered some elements and attributes and defined the possible attribute values for each attribute, you then edit a table, like the one above, which allocates values to particular elements. In addition to the values you specify, all attributes are automatically given the possible value 'unknown'. Until you specify otherwise, all elements have the value 'unknown' on all attributes. You can fill in the table any way you like. You can work through 'an element at a time' (row at a time) or 'an attribute at a time' (column at a time). Or you can jump about and gradually build the table up from one or more different starting points.

If you find that whilst editing a table you think of some more elements to include, you can add some then come back and finish off editing the table where you left off. You can also add attributes by first naming them then specifying some possible values.

If you find that when editing values for an element, the range of values available is inadequate (you cannot find a value on the attribute that appropriately describes the element) then you can add an appropriate value to the list of possible values for the attribute. For example, assume you are editing the new element 'tiger' on the attribute 'relation to man' and you have only specified 'domestic' and 'farm' as the possible values for that attribute. You can decide that neither is appropriate and add the value 'wild' to the attribute and allocate it to 'tiger'.

Inferencing and Unknown Values

You may not be able to give a value to every element on every attribute. Sometimes that value will remain 'unknown' either because you don't know which of the possible values the element has, or because it does not have a value currently in the list of possibilities and you haven't added the new possibility yet. Where the value is genuinely unknown, you can sometimes make a guess as to what it might be. In order to make this guess you use knowledge about the value which similar elements have on that attribute. You also use knowledge about the likelihood of any element having a particular attribute value, and other more implicit knowledge that is often more difficult to define.

If you have managed to give 'known' values to most entries in the table, the program can also make a guess at what an 'unknown' value might be. By specifying the rest of the table you have given it the knowledge by which it can make an intelligent guess (an inductive inference), and it can use some of the same type of strategies in making the guess as a person might. It can look to similar elements to provide an indication, and it can look at the relative likelihoods of any element having a particular attribute value.

The inferencing method used in EXPERT4 (version 1.3) adds support to values on the unknown attribute as a function of the similarity the element (with the unknown value) has with other elements. The value with the greatest support is inferred and this will be the value similar elements have on the attribute.

Inference can be very useful. If the domain were 'medical cases' and there were many attributes including 'diagnosis' and 'treatment', inferencing provides a method by which a new case can be diagnosed. Known values for the new case are allocated and the values for 'diagnosis' can be inferred. Once the diagnosis is known 'treatment' can be inferred. There are two main factors that determine the success of inferencing. The first is the accuracy and extent of specification of the rest of the domain (how good all the surrounding knowledge to the problem is). The second is the effectiveness of the inferencing strategies.

It is in its capacity to make inferences about unknown values that EXPERT4 can be regarded as an advice-giving expert system. Once a domain that accurately represents the elements has been entered EXPERT4 uses this knowledge to offer hypotheses about missing values. Adding another element and filling in the values of attributes you already know about gives the program the necessary knowledge to make inferences about the attributes that are unknown. For example, a doctor enters a domain which has 'types of cancer' as its elements, and describes these using attributes which are pertinent in distinguishing between them in terms of aetiology, symptoms, test results, site of growth, demographic factors, environmental factors, diagnosis, treatment, and prognosis. A new case is entered by filling in the known values for that case. Requesting the program to infer an unknown value for the new case is to consult it for its opinion on what that unknown value might be. Where the requested inference is for 'diagnosis' or 'prognosis' the program is being asked to provide the sort of inference that only an expert in the domain can provide. However, a word of caution is appropriate. Do not regard the inference as in any way definitive. It is easy to make the program generate unlikely inferences for any of the above reasons, so treat an inference as a hypothesis only.

It is possible to check the accuracy with which the program makes an inference by completing a domain with known values, then editing one of the values to 'unknown' and requesting an inference. If the program infers the original value, it has succeeded in making an accurate inference. Note that once a value is changed to 'unknown' the program has no direct knowledge about what the value should be, and must rely on the accuracy and representativeness of the other knowledge in the domain to perform the inference.

If the program infers a different value this could be for a number of reasons. The domain may contain too many other unknown values to make any other inference but 'unknown' for the missing value. Very little may be known about the new case and the inferencing strategy may not have enough information to go on. The domain may not be representative of the elements and its inference may be biased accordingly. The domain may be too small to provide enough information for the inference. The attributes may have been chosen badly and not really reflect characteristics that distinguish between the elements. The allocation of attribute values to elements may be inaccurate and not reflect a 'true' state of affairs. The basis on which the program makes inferences depends upon there being some relationship between the attributes, and it could just be that in a particular domain all the attributes are largely independent of each other. Lastly, the inferencing strategy may be inadequate. The way in which inferencing strategy relates to the structure of different domains is still a research issue and has yet to be established. At the current state of development whatever the inferencing strategy comes up with should at least be plausible.

The Expert System Development Cycle

From the above it is apparent that there are many things to get right when specifying a domain before it can be regarded as having adequate expertise. If you intend to use EXPERT4 to create a domain which has true expertise it is necessary to go through a development cycle which involves entering a domain, testing the adequacy of its inferencing, then editing it until it is representative and complete enough for the inferencing to produce 'correct' results as often as possible.

EXPERT4 is an expert system development tool, and has many advantages in terms of ease of use once a domain has been entered and it is consulted for advice. It is more flexible than the type of expert system which asks a series of questions before offering expert advice, because the spreadsheet format allows the user to volunteer information in whatever order it comes to hand. Expert systems which request the user to enter information in a sequential manner can be at best irritating, when you cannot answer a question at the time, and worse still if they do not allow you to skip the question altogether. In EXPERT4 the user is in control of the interaction rather than the program.

Expert systems developed with EXPERT4 are also likely to be more soundly based than systems based on the expert's specification of rules. In EXPERT4 inferences are induced from examples rather than from an expert's interpretation of a domain, which is notoriously difficult to capture and often unreliable.

Program Facilities

Introduction to facilities

Specifying domains and inferencing may be the only things you will ever want to use EXPERT4 for. You may be specifying domains for use in your own programs which find this a useful form in which to hold their data. You may be using the inferencing to test a theory about the way 'symptoms' relate to 'diagnoses' or the way 'artists' of different 'periods' used 'colour'.

The inferencing in EXPERT4 is good enough to test theories but cannot be relied upon to give 'correct' answers. The answers cannot assume the status of being 'correct' in a formal sense because they are induced from data rather than deduced from axioms. They can be regarded as possible or even probable answers but can never be thought of as definitive. All induced theories have this 'secondary' status even when the weight of evidence in their favour is overwhelming (e.g. the theory that the sun will rise again tomorrow morning). To be 'correct' in a formal sense a theory must be derived from axioms and can be proved. The axioms themselves need only be assumptions, and will be useful for prediction to the extent to which they represent a 'true' state of the world. Mistaken assumptions will lead to inaccurate predictions.

EXPERT4 offers many additional facilities to help in the creation and exploration of knowledge domains. These are discussed below under the headings given in the program's main menu. For the knowledge base management routines the discussion is in terms of the purpose. For the calculation and display routines the discussion covers purpose, psychological basis for the calculation, comments on the interpretation of findings, and example uses of the routines. Some aspects of the user interface are discussed under 'purpose'.

For more detailed discussion of the user interface see the USER TUTORIAL DOCUMENTATION.

The Knowledge Base Management Routines

These consist of the routines for entering, editing and displaying the contents of the knowledge base. The knowledge base consists of:

1 Elements
2 Attributes
3 Attribute values
4 A matrix relating attribute values to elements

CREATE A NEW DOMAIN

This routine is used to set up a file and collect general information for a new domain. The general information consists of a brief description of the domain, and the elements. This information is stored, along with the date of creation.

LOAD A DOMAIN

Use this routine to load a previously saved domain.

SEE/EDIT GENERAL INFORMATION

If a domain is loaded, this routine can be used to see or edit the general information associated with it.

SEE/EDIT/ADD ELEMENTS

This routine is used to see, edit or add elements to the domain. It display the elements and optionally allows you to change the name or add more elements. Whenever element names have been edited or added the domain should be re-saved so that the update can be recorded on disk.

SEE/EDIT/ADD ATTRIBUTES

This routine is used to see, edit or add attributes to the domain. It displays the attributes and optionally allows you to change the name or add more attributes. Whenever attribute names have been edited or added the domain should be re-saved so that the update can be recorded on disk.

SEE/EDIT/ADD ATTRIBUTE VALUES

First the list of attributes is displayed and the user is asked to select the attribute he wants to see the values for. Pressing return at this stage will return the user to the main menu. If an attribute is selected, the values on that attribute are displayed, and the user can edit or add attribute value names in the same way that elements and attributes are edited and added. To facilitate adding attribute values when a domain is being set up, the user is then returned to the list of attributes and can see, edit or add values for another attribute, or simply press return to go back to the main menu. Whenever attribute value names have been edited or added the domain should be re-saved so that the update can be recorded on disk.

SEE/EDIT SPREADSHEET

The spread sheet displayed the main attribute value table with elements as rows, the attributes as columns and the attribute values associated with each element and attribute in the cells of the table. The table is edited by moving around it with the cursor keys and pressing F9 to cycle through values in a cell. 'Unknown' values can be 'inferred' using F10. Whenever the table has been edited the domain should be saved in order to record the changes on disk.

SAVE THE DOMAIN

This will save the domain currently in memory onto disk and return to the main menu. It should be used at the end of any session which involved editing the elements, attributes, attribute values or the matrix.

EXIT

This checks with the user whether he wishes to save before exiting. If he does press return or answer 'N' or 'n' to the question 'Sure you want to exit?' the program will return the user to the main menu so he can save the domain before exiting.

The Calculation and Display Routines

Some of these routines mimic the processes that are thought to go on in expert judgements. An expert has a good knowledge of a domain and can call on many exemplars of a concept to help identify or categorise new elements.

Although the program is primarily designed as an interface to a knowledge base, some of the routines would have application to problems in real time pattern recognition, where a similar type of expert knowledge is often required. A database of example patterns would be stored as the elements. New elements would be classified in accordance with the procedures used in some of these routines.

SEE TYPICALITY INDEX

Purpose

To put the elements in order from the most typical to the least typical. It is often useful to know if something is typical. When adding to the knowledge base it is interesting to know whether new elements are typical of the knowledge base as a whole or whether they are unusual. An item that is typical of a class can be used to represent the class, so a typical item can be regarded as a summary of information about the class.

When classifying an element it is useful to classify it in terms of which sub-class it is most typical of.

The user is presented with a display of elements and their typicality scores, in the order of typicality score going from most to least typical.

Psychological Basis

A element is considered typical on the basis of the extent to which it has common (frequently found) values on its attributes. So, 'Sparrow' would derive some typicality in the domain 'Birds' because 'Sparrow' has the attribute value 'Can Fly' on the attribute 'Flying Ability', and 'Can Fly' is a common (frequently found) attribute value in the domain.

The most typical element is defined to be the one whose attribute values are most frequent overall. So 'Sparrow' would be typical because it has values on several attributes which are found frequently amongst 'Birds'. Whilst 'Penguin' would be low on typicality because the value of many of its attributes would not be frequently found amongst 'Birds'.

The calculation of the typicality index is based on a psychological model of 'Family Resemblance'. Research on this model has found that it correlates very highly with people's typicality ratings. The standing of this model with respect to theories of categorization and concept formation is discussed in recent literature on category and concept formation.

Typicality can be used to make inferences with respect to an element about which little is known. For example, if the values on only one or two attributes are known, a default inference can be made about the element, that it has other values which are typical of the class it is known to belong to. In the absence of further information this is the best inference that can be made. It seems likely that people often do this type of inferencing. Most decisions are made on the basis of incomplete information, and this is only possible because much unknown information can be inferred from known values and a knowledge of the underlying distribution of attributes in the domain.

Examples of this in human reasoning can be taken from perception. For example, we infer that we are looking at a dog with four legs even when we can only see its head round the door. The obscured part of the body is in fact a mystery. It might be a dog with three legs or it may have a completely different body such that we would not classify it as a dog at all. However, the inferences we can make by observing attributes of the part we can see leads us to expect to find other body attributes typically associated with those features.

Interpretation

In a homogeneous domain many elements may score highly on typicality. In some domains it is possible that all elements score relatively low on typicality reflecting a more heterogeneous domain.

A homogeneity index can be calculated by totalling the typicality indices of all elements. This can be compared across domains of different sizes, since the index takes maximum possible typicality into account.

An element with low typicality , tends to have infrequently occurring attribute values. It can be said of such elements that they are 'individual' or 'atypical'.

Example Uses

The calculation of typicality is most useful for the purpose of conveying information about a concept. A typical item represents the concept in a very compact form. The more homogeneous the class the better the typical element will represent the class. If you wished to represent the class of 'American T.V. soap operas' the phrase 'a program like Dallas' conveys more

information more briefly than a description of the class in terms of its relevant attribute and possible attribute values.

In learning and teaching it is useful to explain concepts in terms of typical elements. Rather than trying explicitly to teach the concept of dog, children learn and are taught such concepts by associating exemplars of the concept to the concept name. As well as this associative mechanism, the concept is extracted from the exemplars by some form of feature analysis, such that novel elements can be tested for class membership. Typical element are useful in teaching because they are the best exemplars of the concept, and are less likely to lead to inaccurate concept formation. For example, in teaching salesmen to distinguish between likely and unlikely sales leads, it is often useful to illustrate this by using examples which typify each case. Using atypical examples would mislead. The calculation of typicality can suggest which elements should be focussed-on for study when teaching a concept.

Checking for typicality is useful in spotting unusual elements. If the elements are candidates for a job, an atypical candidate may be more suitable for an atypical job.

If two domains are set up, for example 'genuine paintings by Picasso' and 'Picasso fakes', a new element can be tested for typicality in both domains to establish which class it is more likely to come from.

QUERY BY EXAMPLE

Purpose

To retrieve elements similar to a specified element, in order of similarity. Often in information retrieval the user's best route to the information he is looking for is by reference to similar pieces of information he already knows. So when trying to identify a disease, a doctor may consider first a disease he knows about which has similar symptoms, then search for similar diseases which have a better match to the symptoms.

The user first selects an element he wishes to compare with the others. A list of elements is presented with their similarity scores (with the selected element), in order of similarity from most similar to least similar. Pressing 'return' allows the user to select another element for comparison. If no other comparison is required pressing return again returns the user to the main menu.

Psychological Basis

The psychological basis for the calculation of similarity is based on the same model as the typicality index. Items are deemed similar by virtue of the number of attribute values they have in common.

Interpretation

The extent to which any two elements are similar is dependent upon the proportion of attributes on which they have the same value. So if an element has a score of 75 with respect to the selected element this means that the two elements have the same attribute value on three quarters of their attributes. The attributes are not weighted and each counts equally towards the calculation. For a weighted measure the attribute search routine can be used. One reason for not weighting the attributes is that, a priori, there is no reason to favour one attribute over another. Weighting is only required when you are asking a specific question about similarity such as 'are elements A and B similar with respect to a particular attribute or set of attributes?'.

Example Uses

In a manufacturing environment, a shortage of one part may require that a search is made for another part that is similar enough to substitute for it.

In information retrieval of academic or technical papers, a target paper may be identified by starting the search at a paper known to be relevant, then searching papers that are similar to it.

In a criminal investigation, suspects can be ordered by similarity to the main suspect.

In matching personnel to work on assignments together, it may be useful to select people with similar temperaments but dissimilar skills.

ELEMENT COMPARISONS

Purpose

To compare directly two elements for similarities and differences in attribute values on all attributes.

The user selects two elements from the list of elements. The attribute values which are common to both are shown first, then the attribute values on which the elements differ.

Psychological Basis

This is direct information retrieval from the knowledge base, and does not do any processing on the elements.

Interpretation

The ratio between number of similarities and differences is equivalent to the similarity index between the two elements.

Example Uses

To explore in greater detail the comparison between two elements. The similarity index gives the overall similarity. The comparison shows the respects in which elements are similar and differ.

If the two elements are both typical of their own sub-classes then a comparison of the elements will provide concrete comparison of the classes in terms of exemplars.

A typically fraudulent claim for insurance can be compared with a typically genuine claim.

COMPARISON OF ATTRIBUTE VALUES

Purpose

To identify the typical characteristics of elements that have different values on a particular attribute.

The user is asked to select an attribute for the comparison. A table showing the typical attribute values for elements with each of the attribute values on the selected attribute is shown. If there are too many values in the selected attribute to fit across the screen, the user can change the screen window so that any other selected attribute values are displayed.

Psychological Basis

A typical attribute value is defined as being the most frequently occurring.

Interpretation

Where a column is constructed from values across several individuals the final profile may not match any single exemplar, even though it is considered to be typical. It is made up of all the most commonly occurring attribute values even though it is possible that no actual exemplar of this attribute value combination exists.

Example Uses

An attribute can be used to classify the domain into sub-groups. The comparison of attribute values provides a basis for comparing these subgroups in terms of their typical members.

In a banking application it may be of interest to compare typical attributes of people classified according to credit worthiness.

In catering applications it may be useful to compare typical attributes of meals in different diets.

SEE ATTRIBUTE VALUE DISTRIBUTIONS

Purpose

To present a histogram of frequency of occurrence of attribute values on a selected attribute. For some attributes in some domains, all or nearly all elements will possess a single attribute value (e.g. 'two legged' for humans). For other attributes (e.g. shoe size) the elements will be spread across several attribute values. The distribution of elements across attribute values is of interest because it shows how many elements fall into each attribute value category.

The user selects an attribute and the frequencies of elements having each value of the attribute are displayed numerically and in the form of a histogram (rotated through 90 degrees). When 'return' is pressed the user can select another attribute for display or by pressing 'return' again can return to the main menu.

The attribute value distribution shows if all values have been adaquately represented in the domain. For example where the domain is supposed to represent 'People in the U.K.', the distribution can be checked for approximately equal frequencies of males and females. If the domain represented 'mine worker' the distribution should be in accordance with the proportion of male to female 'mine workers'.

The display also shows the 'a priori' probabilities for each attribute value based on the elements entered into the domain. This is of particular interest to expert system developers because experts are known to make biased estimates of the probability of occurrence of attribute values by not taking into account 'a priori' probabilities.

Psychological Basis

This is a direct frequency count of the number of elements with different attribute values on the selected attribute. It is unlikely that human reasoning relies on direct counts such as this, but it is possible that the freqency distribution is estimated from items that are easily 'available' in memory.

Interpretation

The frequency count gives information about the distribution of elements of attribute values of a selected attribute. Many interpretations of this distribution can be made, some of which involve identifying an underlying ordinal distribution of the attribute values. These are not catered for in the program at present. For example, it can indicate how the attribute values

can be ordered from least to most frequent or can form the basis for testing for rectangular, normal, or other types of distribution. If an underlying distribution can be identified then something can be said about the relationship of attribute values to each other within a particular attribute. Knowledge of an attribute value can then be used to locate it in a non-metric or metric space on the attribute.

To illustrate this, the program does not know anything about the order of attributes. All its attribute values are treated as mutually exclusive categories. For example, although it may know of the values 'Short', 'Medium' and 'Tall' on 'Height' it does not know that they order from Short to Medium to Tall. Short, Medium and Tall are treated in the same way as the colours Red, Blue and Green might be treated, i.e. as if they had no special ordering.

A knowledge of ordering would enhance the inferencing cababilities of the program, and it is hoped to be able to include facilities for entering this information in later versions of the program.

Example Uses

To check that attribute values have been adequately represented in the domain.

To make inferences about the underlying distribution of attribute values on a particular attribute.

SEE ATTRIBUTE CROSS TABULATIONS

Purpose

To see the cross tabulation frequencies between attribute values on any two selected attributes. This can be used to generate hypotheses about the relationship between two attributes.

The user is asked to select two attributes from a list of attributes. A table of the frequency of elements in each attribute value combination is presented. The table can be displayed reversing rows and columns by entering the selected attributes in the reverse order. The user can change the window if there are so many values of the attribute running across the screen that they won't all fit, or to display any selected part of the cross tabulation matrix.

Psychological Basis

It is likely that people often do some form of cross tabulation calculation. For example most people are aware of an association between brown eyes and dark hair, and blue eyes and blond hair. The association exists because more brown eyed people have dark hair and more blue eyed people have blond hair than would be expected if eye colour and hair colour were unrelated. People are often taught about these associations but also discover them for themselves if they have a good knowledge of a domain. A experienced car mechanic will have some representation of the cross tabulation frequencies between type of car and type of fault, and uses this to achieve a more efficient and accurate fault diagnosis than the non-expert. The expert may store this representation as a heuristic rule relating attribute values on different attributes to each other.

Interpretation

Large cross tabulation tables are difficult to interpret by inspection. The tables are presented so that further statistical analysis can be done if the user requires, and it is not intended that the tables be interpretated without further analysis.

Example Uses

To test a specific hypothesis about the relationship between attribute value on two attributes.

SEE/EDIT ATTRIBUTE SEARCH PROFILE and DO ATTRIBUTE SEARCH

Purpose

To search the knowledge base for elements that are identical or similar to an attribute search profile set up by the user. It will perform either an exact or fuzzy match depending upon how the search profile has been set up.

The search profile is presented in a spreadsheet format and the user moves round this using the cursor key, setting values and weightings ('essential', 'important', 'preferable' and 'doesn't matter') by cycling through values. Once the profile is set to the user's satisfaction he presses F10 to do the search. The result of the search is displayed. Any elements not meeting essential criteria are not displayed. All other attributes are displayed in order of their match with the search profile.

Psychological Basis

People can do a limited amount of exact matching with a simple search profile. When it comes to doing complex matches a Query by Example strategy is substituted. For example, when doing a match for 'Labour politicians who have held the offices of Foreign Secretary and Home Secretary' it is likely that rather than search through all Labour politicians, or all Foreign or Home Secretaries a person is more likely to think of a politician at random or according to some other politician generation procedure that is learned, and then test these against the profile. The politicians will be tested in order of availablity according to the politician generation procedure. Where the generation procedure takes the form of associating to politicians similar to the last one thought of, the procedure matches the Query by Example paradigm. For complex essential matches the program will out-perform a person in number of items retrieved because it will exhaustively match to all elements in the domain. However, the person may out-perform the program on speed if only one or a few matches are required. On essential matches the routine is not based on any psychologically suggested principle.

On fuzzy matches, the routine compares each element with the search profile, rejects it if it does not meet the criteria of any essential matches, then calculates a matching score where matches between an element and the profile are weighted by whether the attribute value being matched is important, preferable or of no importance. People are capable of this type of fuzzy matching and it seems likely that a crude scale of 'essential', 'important', 'preferable' and 'doesn't matter' is all that can be coped with within the capacity of working memory, if several elements are being compared. For people, the order of selection of elements will be based on availability as discussed above, rather than by exhaustive matching, as in the routine.

Interpretation

The attribute search allows the user to order elements according to the extent to which they meet various criteria. The criteria are explicitly weighted on the 4 level importance score. The way the importance rating is set has different logical consequences. Settings of 'essential' cause removal of elements. Settings of 'Doesn't matter' cause certain attributes to be ignored. Adding 1 to the matching score for 'preferable' attribute values and 2 for 'important' attribute values is somewhat arbitrary but seems reasonable within this simple scheme. The exact ratios of these two scores is dependent upon the semantic relationship between 'preferable' and 'important' and this may depend on context. The 1 to 2 scoring system implies that two preferable attribute value matches are as good as one 'important' attribute value match. However, assuming that these scores approximate the relative semantic weights of 'preferable' and 'important' to a reasonable degree, the attribute search results can be regarded as forming a monotonic scale, such that elements with higher scores can genuinely be regarded as better matches, in a psychological sense, as well as by definition.

Example Uses

Any database retrieval task that would benefit from simple weighting of the attributes.

As a decision aid in selecting elements for a specific purpose. The elements could be personnel, diseases, suspects, programming languages, etc. etc.

Creating Sub-Domains

Whereas the above routines are concerned with searching and presenting summary information about the knowledge base, the following routines are mainly concerned with the forming of decision trees from the knowledge base, which can then be used as a method for classifying elements.

One use of the creation of sub-domains is to look at the typicality, and similarity index scores of sub-sets of elements. Another is to see whether associations between attributes can be established for, or hold within, sub-sets of elements. For example, sub-sets of males can be compared with sub-sets of females by creating separate sub-domains for each.

However, it is guiding the formation of decision trees that give these routines their power. If a domain is reasonably stable (i.e. the elements in it do not change), a decision tree can be constructed which can be used to classify elements. The decision tree representation will be considerably more economic on memory and search speeds than reference to the full domain.

The decision tree can potentially be used to control a dialogue which will help an inexperienced user make classifications that otherwise only an expert could make. This principle is used in other expert system development tools.

This program does not construct decision trees itself, but can be used manually to contruct them.

SEE ATTRIBUTE STATISTICS

Purpose

To display information about the number of attribute values possible and the number of attribute values actually found in a domain. The F/P*100 index shows the degree to which each attribute has all its values represented. Attributes where a low proportion of values are actually represented can be regarded as having hypothetical attribute values. The index can be regarded as a measure of the extent to which an attribute has hypothetical values.

The statistics also show the attributes with the largest and the smallest number of values found. These determine the attributes selected for the MAX and MIN search strategies, respectively.

The option is selected from the main menu. It is displayed and the menu is re-displayed when the return key it pressed.

Psychological Basis

When specifying relevant attribute and attribute values for elements in a domain, people are likely to name attribute values which in fact occur very infrequently or almost never. There is a tendency of regression towards the mean in human estimates of probability such that the estimated proportion of rare attribute values is overestimated. The program, therefore, distinguishes between the number of values specified as possible (i.e. entered as attribute values) and the number actually found in the domain. Where relevant the number actually found is used in calculations.

Interpretation

A domain can be entered that has a large number of possible values for attributes, if it is suspected that the attribute values will be useful in describing elements in the domain. Which of these values prove to be useful will depend upon whether they are used in describing actual elements. As more elements are entered the F/P*100 index is likely to increase as more of the attribute values are used.

The attribute statistics provide useful control information about the domain, suggesting when attribute values are not being used and can therefore be considered as candidates for removal from the model. It provides an overview of the size and complexity of the domain.

Example Uses

To check which attributes have more values than are being used, and to help decisions about which attribute values can be removed or substituted.

SEE PIC INDEX

Purpose

The PIC index calculates the probability that knowledge about the value an element has on each attribute, will lead to the immediate classification of the element on another selected attribute.

If the object of an excercise is to find out which class an element falls into, the PIC index will tell us which attribute it is best to find out about in order to classify the object.

With a full domain, the PIC index can be used to find the root of a decision tree for classifying all elements. The number of nodes in the decision tree will be minimised using a one level look-ahead strategy. By creating sub-domains in turn, one for each value on the root attribute, then looking at the PIC indices for the sub-domains, the next level of the tree can be formed. The complete tree is found by repeating this procedure. The leaves of the decision tree are found whenever an attribute node is found to be a perfect classifier (i.e. have a PIC index of 100).

In a stable domain the decision tree can be used to control a dialogue with a novice to help him classify elements. Such decision trees have been found useful in minimising the cost of testing elements (for example in fault diagnosis of printed circuit boards).

In a real time environment the decision tree can be used to control the calling of attribute value testing routines which maximise the speed of classification. However, the cost of testing is not taken into account.

Psychological Basis

When selecting elements for a certain purpose it seems unlikely that people exhaustively test all attributes of each element in order to classify it. It is more likely that they test on attributes they know to be related to (i.e. are good classifiers of) the class. However, it is unlikely that people will form optimal decision trees, although a tree may converge on the optimal tree as the amount of expertise about a domain builds up. The decision tree is a preferred representation for human processing because it maximises information content while minimising storage space.

Creating Decision Trees

The user selects to see the PIC index at the Create Sub-Matrix Menu. He then selects an attribute to act as the class attribute. The PIC index will then reflect the probability of each attribute immediately classifying with the selected class attribute.

The PIC index is displayed with the attributes in diminishing order of the size of the PIC index. Pressing 'return' returns the user to the menu, where he can select to Create a Sub-domain using the MAN (manual) strategy, then form a sub-domain using the attribute with the highest PIC index value. The PIC index for the sub-domain can then be consulted and another sub-domain produced, and so on. This procedure takes the user down a particular path in the decision tree. By returning to the full domain each time a path reaches a leaf and repeating the procedure, but selecting different attribute values at each sub-domain creation, the user can trace out all paths in the tree.

By picking different attributes to act as the class attribute, different decision trees optimized for classifying within the class attribute can be constructed.

Interpretation

The PIC index indicates the probability of classifying within the class attribute from a knowledge of other attribute values. A high PIC value indicates a high probability that the attribute value on the class attribute can be predicted from a knowledge of the value on that particular attribute. The probability can be interpreted in a number of ways:

1 The attribute with the highest probability is the best attribute to select as the root of a decision tree for classification of values on the class attribute. In other words, if the exercise is to predict the value on the class attribute, a knowledge of the value on the attribute with the highest PIC will give the best chance. If the PIC index is 100, then the class attribute can be immediately predicted from the value on the attribute.

2 A PIC index of 100 suggests the induction of deductive rules of the form:

All elements with attribute value x on attribute X have value y on the class attribute Y.

There would be one rule of this sort for each value of the attribute with the PIC index of 100.

The PIC index, then, can be used as a basis of the identification of rules for a rule-based system.

3 The PIC index shows the percentage of elements that each attribute will immediately classify within the class attribute.

4 The PIC index can be regarded as an index of association between an attribute and the class attribute. A high value would indicate a high degree of association such that it would be possible to say that if an element has a value x on attribute y there is a probability p that it will have value i on attribute j.

In fact, the PIC index is constructed out of individual probabilites that are added across attribute values for each attribute, so the precise prediction cannot be made from the PIC index. However, the PIC index can be regarded as representing the individual probabilities, since it is made out of them and will, to some extent, correlate with them.

Attributes with high PIC indexes, then, can be regarded as good predictor attributes. So long as this is confined to prediction within the model (over only elements contained within the model) the PIC index need not be tested for statistical significance in order to be meaningful.

However, if a new element outside the model is to be classified, the elements in the model can only be regarded as a sample of all elements and PIC indexes below a set level of significance could be regarded as occurring by chance, since the model now is only a sample from a larger parent population.

Example Uses

To create decision trees which are used later for use in controlling expert dialogues.

To create decision trees for minimizing classification time in real time applications.

To form rules which can be used later in a rule-based system.

To generate hypotheses about the relationships between attribute values.

CREATE SUB-DOMAINS

The two routines above give the user information about which attributes to select to form decision trees, rules etc. The three routines below do the actual creation of sub-domains using three different strategies. The MAX and MIN strategies make the presumption that the elements themselves are treated as the class. They are concerned, therefore, with problems where the retrieval of a specific element is required. Decision trees can be constructed using the MAX and MIN strategies which will lead to the identification of individual elements within the domain. For example, in a medical application it is useful to identify a particular disease that is stored in the knowledge base, from its symptoms. Once the disease is identified its other attributes can be studied and if necessary tests set up to confirm the diagnosis.

CREATE SUB-DOMAIN (MAX)

Purpose

This routine can be used to form a decision tree where branching always occurs at the attribute with the largest number of alternative values. Since the class attribute consists of the elements, the attribute with the highest number of found attribute values is the attribute which will discriminate between them most successfully leaving, on average, the fewest elements for further consideration.

If the decision tree formed from always selecting MAX, is used to control an expert dialogue, each question in the dialogue will contain the maximum number of possible choices. In other words, the value of a test is determined to be a direct function of the number of different results it can give. This will minimise the number of tests that have to be performed to find the element.

Selecting MAX displays to the user the values on the attribute with the largest number of found values. Where two attributes tie for the number of found values EXPERT4 selects the first one it comes across. If you don't like this selection press return and select another attribute using the MAN option. Otherwise select a value and that value is used to create the sub-domain.

Psychological Basis

Whether there is a psychological basis for the MAX strategy will depend upon whether people ask questions, when trying to classify an element, which have a maximum number of possible responses. Does the doctor ask questions requiring 'yes/no' answers or questions that have a wider range of possible answers. The experienced doctor may prefer this strategy if it means that he will collect all the data he needs most quickly. However, the problem is not so much in formulating the question as in the

capability to answer it. It is possible that it is easier to answer questions that have fewer possible answers because fewer answers need to be considered as alternatives.

Interpretation

The sub-domain can be regarded as the class of elements that had the particular attribute value selected. This class of elements can be taken on to the element inferencing program for the identification of typical and atypical elements within the class. Also elements can be tested for similarity within the class rather than within the whole domain.

Since now only a sub-set of elements is present, the attribute inferencing program will give results specific to this class of elements.

Example Uses

To compare sub-sets of elements.

To find typical elements within different sub-sets of elements.

CREATE SUB-DOMAIN (MIN)

This is similar to the MAX strategy except that the attribute with the minimum number of found values is selected. This has the effect of lengthening the path to the element, but simplifying the decisions at each stage. The complete decision tree will have more nodes, and more levels, but fewer arcs emerging from each node.

It may be useful to use this strategy when using the decision tree to control an expert dialogue, since it presents fewer alternatives at each decision point.

CREATE SUB-DOMAIN (MAN)

This is similar to MAX and MIN except that the user selects the attribute himself, as well as the value on the attribute. The attribute can be selected on the basis of the PIC index or on any other basis that the user believes to be useful.

Since decision trees are created manually it is possible to mix all the above strategies in any combination, and so optimise a particular tree according to a variety of considerations.

EXPERT4 - User Tutorial Documentation

Copyright Rod Rivers 1984

The tutorial documentation takes you through the facilities offered by EXPERT4. It explains at the most practical level how to run the program, in terms of which keys to press when, and it describes the displays the program produces. It does not attempt to explain the rationale for the program or the meaning of the descriptive language used in modelling a domain (e.g. elements, attributes etc.). For a description of the purpose, psychological basis, interpretation and example uses of each of the facilities you should refer to the EXPERT4 TECHNICAL DOCUMENTATION.

Running EXPERT4

To run EXPERT4 load the EXPERT4 disk into any floppy disk drive and log into that drive. Type 'X4DEMO' to see a demonstration run of the program or 'X4' to run the program itself. So, if you have put the disk in drive A: , typing -

```
A>X4DEMO
```

will run the demonstration. The demo. may run for up to 15 minutes and shows most of the facilities the EXPERT4 program offers. It can be interrupted by Ctrl. Scroll Lock or a system reset Ctrl. Alt. Del. The demo. can be paused at any time by using Ctrl. Num Lock . Pressing any key will start it going again.

If you are using EXPERT4 for the first time try running the demonstration now, before going on to run the program itself.

Having seen the demonstration the tutorial will explain the operation of each of the facilities. It would be useful to run the program itself at this point, and try using the facilities while referring to this manual.

Introductory Banner

The first thing you see when the program is run is the introductory banner.

```
--------------------------------------------------------------------------

                              EXPERT4

                  Knowledge Base Manager V 1.3

              Distribution by Biosoft, Cambridge, U.K.

                        Version for IBM PC

                    Copyright R Rivers 1984

          Configured for 60 elements, 20 attributes and 10 values

--------------------------------------------------------------------------
```

Take special note of the following:

The Version Number:	V 1.3 is the 'biomedical' release, and its use is regulated under the 'Biomedical Licence Agreement'.
The distributor:	The distributor may be able to offer sample domains in your application area, and can be consulted if you are developing a domain for resale with the EXPERT4 'Run Time Module'.
The hardware version:	This shows which hardware the version you are running is supported for. It may also run on compatible hardware. Refer to the USER TECHNICAL DOCUMENTATION for more details on the hardware version.
Copyright notice:	Please respect the copyright notice and only make copies of the EXPERT4 release disk for your own use. Program output in the form of printout, if published, should reference EXPERT4 as the development tool.
Configuration:	Note the number of elements, attributes and values your version is configured for. This puts a limit on the maximum size of domain the version can cope with. There are several configurations available, and you should consult the distributor if you need a different configuration.

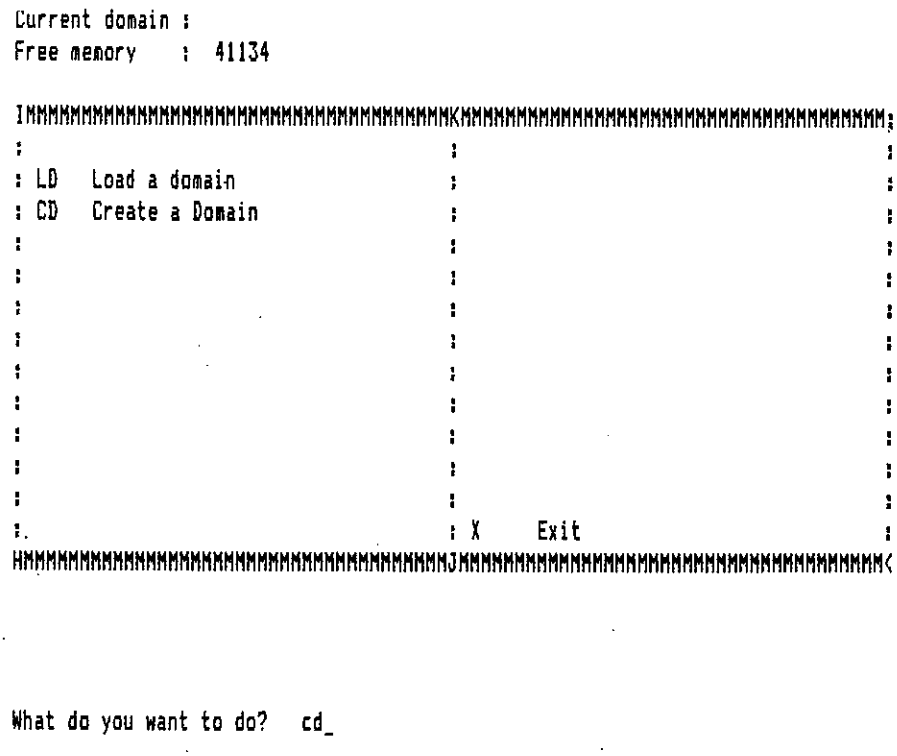
At the bottom of the screen you see the message 'Press return to continue'. Most messages in EXPERT4 pop up at the bottom of the screen, and require you to type something. In this case it's only 'return' so press it and move onto the next display.

The Menu

You are presented with a limited number of options in a menu. At the top of the screen 'Current domain : ' is empty. 'Free memory' shows the amount of memory space available for a new or loaded domain. It is worth noting this the first time you run the program, and whenever you start using a different version or configuration of the program.

At the bottom of the screen is a message, and space for a three character input. In the menu, to the left of each option name, is a short code for the option (e.g. LD for Load Domain'). The short code is reflected in the name by using capital letters.

The input request at the bottom of the screen expects you to give it one of the option short codes, followed by 'return'. Inputs in EXPERT4 are 'error trapped'. If you put in something that the program does not understand it will re-present the message or redisplay the screen.



CREATE A DOMAIN (CD)

Enter 'cd' in either upper or lower case and press return.

All command input in EXPERT4 is converted to upper case, so it doesn't make any difference which case you use. When entering data into EXPERT4 (elements, attributes etc.), case is preserved so things you name will appear exactly as you enter them. In version 1.3 the use of commas in input can cause problems when data are saved to disk, and use of punctuation in element, attribute and attribute value names should be avoided.

You are now asked to describe the domain, and say what the elements are. When entering this information you can use the back-space key to correct errors. The date of creation and last edit of the domain are displayed. These dates are taken from the dates set at in the operating system. The message at the bottom of the screen asks if you want to change anything. Type 'y' if you want to change something, otherwise pressing 'return' defaults to assuming that you don't want to make a change, and displays some help on how to create a domain.

```
How to create a domain


To create a domain use the folowing routines:-

        See/edit elements       - to add elements
        See/edit attributes     - to add attributes
        See/edit attrib. values - to add values to each attribute
        See/edit spreadsheet    - Use key F9 to allocate attrib. values

When the knowledge in the domain has been entered, use the other
options to explore the relationships in the domain.

Unknown (missing) values can be inferred from known knowledge
using key F10 in the see/edit spreadsheet option.
```

Pressing 'return' again will take you back to the menu. In general pressing 'return' in response to any message will get you back to the menu by the shortest route.

Note that the menu has changed since last time. The 'Load Domain' and the 'Create Domain' options have gone and some others have appeared. As far as the program is concerned you now have a domain in memory but with no elements, attributes or attribute values. You can save the domain, if you wish, but at present it only contains the general information you entered when creating the domain.

```
Current domain :  Animals (demonstration domain)
Free memory    :  41028

┌──────────────────────────────────┬──────────────────────────────────────┐
│                                  │                                      │
│                                  │                                      │
│                                  │                                      │
│ GI   See/edit General Information│                                      │
│ SE   See/edit/add Elements       │                                      │
│ SA   See/edit/add Attributes     │                                      │
│ SAV  See/edit/add Attribute Values                                      │
│                                  │                                      │
│                                  │                                      │
│                                  │ SD    Save Domain                    │
│                                  │ QD    Quit domain without saving     │
│                                  │                                      │
│                                  │ X     Exit                           │
└──────────────────────────────────┴──────────────────────────────────────┘

What do you want to do?  gi_
```

SEE/EDIT GENERAL INFORMATION (GI)

'GI' re-displays the general information you entered when creating the domain, and allows you to edit it as before.

```
See/edit general information about this domain

Domain description    Taxonomy of Animals
The elements are      Animals

Date created          01-05-1985
Date last updated     01-05-1985

Do you want to change anything  _
```

QUIT DOMAIN WITHOUT SAVING (QD)

If you 'Quit Domain without saving' a message will appear at the bottom of the screen to check with you before the domain is lost. Pressing 'return' after QD re-displays the menu without quitting the domain. You have to answer 'n' to the question before the domain is lost. If you do the program is run again from the beginning, the introductory banner and all.

EXIT (X)

If you 'Exit' a message appears to check with you before exiting to the operating system and losing the domain in memory. If no domain is in memory you exit straight away.

SEE/EDIT/ADD ELEMENTS (SE)

So, unless you've had enough already, the only things left to do are to See,edit, and add elements, attributes and attribute values. Try 'SE' you will get a new page and a message telling you that no elements have been entered, and another asking if you want to add elements. Pressing 'n' or 'return here takes you back to the menu. Press 'y' instead and you will be presented with input lines to add elements. Press 'return' after each one to go on to the next one, and press 'return' twice to finish. You can carry on adding elements up to the limit allowed by your configuration. If you do, a message will appear after the last one letting you know that you have reached the memory limit. Pressing return at this point is like pressing it twice before reaching the limit. In both cases all the entered elements are re-displayed.

If there are more than 32 elements in a domain the display of elements is paged showing you them 32 at a time. You press return each time to move onto the next page.

If there are less than 32 elements, or you are on the last page a message appears at the bottom of the screen requesting you to type a number to edit, or press 'return'. If you want to change the name of an element, you type its number and press return. The current name for the element will be displayed at the bottom of the screen and you enter the new name below it. After editing element names or if you just pressed 'return' you are asked if you want to add elements. You can do so if you wish or press 'return' to get back to the menu.

```
See/edit/add Elements
---------------------------------------------- The Elements -----

 1  Cow                    17  Giraffe
 2  Pig                    18  Elephant
 3  Chicken                19  Bird
 4  Goat                   20  Sparrow
 5  Bull                   21  Robin
 6  Cock                   22  Pelican
 7  Lamb                   23  Dog
 8  Sheep                  24  Cat
 9  Fish                   25  Lizard
10  Whale                  26  Kangaroo
11  Dolphin                27  Bear
12  Octopus                28  Mouse
13  Sea-horse              29  Squirrel
14  Lion                   30  Fox
15  Tiger                  31  Man
16  Leopard

Type number of element to edit or press return __
```

SEE/EDIT ADD ATTRIBUTES (SA)

Entering attributes is exactly the same as entering elements. Elements and attributes can be entered in either order.

SEE/EDIT/ADD ATTRIBUTE VALUES (SAV)

To enter attribute values you have to have already entered some attributes. If you try to enter attribute values first you will get a message asking you to enter attributes. Selecting 'SAV' at the menu presents a list of the attributes and asks you to select one. You enter the attribute number and press return. If no attribute values have already been entered for that attribute, you are given a message to this effect and asked if you want to add attribute values. After entering attribute values for an attribute, they are re-displayed for editing. If you press 'return' the attributes are re-displayed so that you can select another attribute to see, edit, or add its values.

As soon as elements, attributes and attribute values have been entered the menu changes again, and many more options are added.

```
See/edit/add Attributes
------------------------------------------------ The Attributes -----

 1  Relation to man
 2  Element
 3  No. of legs
 4  Wings
 5  Fins
 6  Sex
 7  Meat
 8  Killed for
 9  Respiration
 10 Distinct feature
 11 Size
 12 IQ


 Type number of attribute to edit or press return  __




 Which attribute do you want to see the value for  1_

 Attribute values for Relation to man
 The attribute values

  0 unknown
  1 Farm
  2 Domestic
  3 Wild

 Type number to edit or press return  __
```

LOAD A DOMAIN (LD)

Before looking at these other options, it is necessary to backtrack a little to explain the 'Load Domain' option. This disappeared when we used 'Create Domain' because as soon as we had created a domain, another could not be loaded without quitting the domain first, or exiting from the program. You can get back to the 'Load Domain' option now by first saving your newly created domain, then answering 'n' to the question about whether you want to continue with it.

Now, you are back to the start of the program and pressing 'return' will present the original three option menu. This time use 'Load Domain'. You are presented with the files on the logged drive, and the amount of free space left on the drive. If you decide you don't want to load a domain after all, you can press 'return' to get back to the menu. Otherwise you should enter a file name. The file name can be up to eight characters in length and is subject to the same restrictions as file names for the operating system. Check your operating system manual if you are unsure about these. You should not add any extension to the file name. The extension '.X4A' is added automatically by the program. Of the files in the disk directory, you can only load those with '.X4A' extensions. In version 1.3 you must enter the file name correctly. If the file is not found on the logged drive the operating system generates an error and control is returned to the operation system. You must start the program again from scratch. Since you never have a domain in memory when you are loading this is not a very serious problem.

Enter the file name 'ANIMALS' if you see it in the disk directory. Otherwise enter the name of any other .X4A files, including the one you saved just a while ago.

Having loaded the domain you find that the menu now has the full range of options again (excluding Load and Create).

```
Current domain :  Taxonomy of Animals
Free memory    :  5244

┌──────────────────────────────────┬──────┬──────────────────────────────┐
│                                  │ CT   See attribute Cross Tabs.      │
│                                  │ PAV  See Prototype for an Att.Value │
│                                  │ SSP  See/edit Search Profile        │
│ GI   See/edit General Information│ AVS  Attribute Value Statistics     │
│ SE   See/edit/add Elements       │ MAX  Create sub-domain (MAX strat.) │
│ SA   See/edit/add Attributes     │ MIN  Create sub-domain (MIN strat.) │
│ SAV  See/edit/add Attribute Values│ MAN  Create sub-domain (MANual)     │
│ SS   See/edit Spread Sheet       │ AS   See Attribute Statistics       │
│ TI   See Typicality Index        │ PIC  See Probab. Immediate Classif. │
│ QE   Query by Example            │ SD   Save Domain                    │
│ EC   Element Comparisons         │ QD   Quit domain without saving     │
│ CAV  Comparison of Attribute Values│                                    │
│ AVD  See Attribute Value Distrib.│ X    Exit                           │
└──────────────────────────────────┴──────┴──────────────────────────────┘
```

Main Attribute Value table (31 elements, 12 attributes)

	Relation	Element	No. of l	Wings	Fins	Sex
Cow	Farm	Land	4	No	No	Female
Pig	Farm	Land	4	No	No	Either
Chicken	Farm	Land	2	Yes	No	Female
Goat	Farm	Land	4	No	No	Female
Bull	Farm	Land	4	No	No	Male
Cock	Farm	Land	2	Yes	No	Male
Lamb	Farm	Land	4	No	No	Either
Sheep	Farm	Land	4	No	No	Female
Fish	Wild	Sea	None	No	Yes	Either
Whale	Wild	Sea	None	No	Yes	Either
Dolphin	Wild	Sea	None	No	Yes	Either
Octopus	Wild	Sea	8	No	No	Either
Sea-horse	Wild	Sea	None	No	No	Either
Lion	Wild	Land	4	No	No	Either
Tiger	Wild	Land	4	No	No	Either
Leopard	Wild	Land	4	No	No	Either

Move round with cursor keys
F1-End Edit F2-Col.Widths F7-Values F9 to edit F10-Infer

SEE/EDIT SPREADSHEET

Once you get to this point all the hard work of adding elements, attributes and values has been done (either by you or the person that created the domain). Assuming that you have loaded a supplied sample domain, the values in the main attribute value table will already be allocated.

At the top of the display you see the heading and the number of elements and attributes currently in the domain. If the domain is anything but very small, not all the elements and attributes will fit on a single screen, so it is useful to note the full extent of the table.

In the table, the elements are the rows and the attributes are the columns. The entries in the table are the attribute values that have been allocated to each element on each attribute.

You move around the table using the cursor keys on the numeric keypad. If there are more than sixteen elements when you attempt to move down from the last one, the next lower portion of the table is displayed, consisting of the next 16 elements. Moving off to the right will display more attributes. You get back again by moving off the top or off to the left. You are stopped only at the edges of the table.

When using the spreadsheet all keys are disabled except the cursor movement keys and five of the function keys. These are used as follows:

F1 Pressing this takes you out of the spreadsheet and back to the menu. It is the only way to exit from the spreadsheet.

F2 This allows you to change the column widths of the columns in the spreadsheet. You are presented with a list of attributes and their current column widths. All column widths are 8 characters wide by default. Selecting an attribute allows you to change the '8' to any other number that is not larger than the screen table width. Pressing 'return' instead of an attribute number takes you back to the spreadsheet. If you do enter an attribute number, then press 'return' the column width will become '0'. If the column width was already '0' it becomes '8'. This allows you to toggle between displaying and not displaying a particular attribute. A column width of '0' means that the attribute is not displayed in the spreadsheet. However, it is taken into account in all calculations, even if it is hidden in this way. Press 'return' in place of an attribute number to get back to the spreadsheet.

F7 This lets you take a quick look at the possible attribute values for the attribute in the column you are currently in. You can also see these by cycling through them (see F9), but this lets you see them all at once, with their full names. Press 'return' to get back to the spreadsheet.

F9 The F9 key allows you to cycle through the attribute values for the attribute in the column you are in. You cycle through them in the order in which they were entered in the see/edit/add attribute values routine. Whatever value you leave in the cell is the value you allocate to the element in that row for the attribute in that column. If no values have been allocated the complete table will consist of the value 'unknown'. You can also re-allocate 'unknown' to values that have already been allocated. You can move around the spreadsheet as you like, allocating and changing values as you go.

F10 This is the interesting key, and in many ways it demonstrates what EXPERT4 is all about. For cells that have the value 'unknown' you can press 'F10' and the program will attempt to induce, from values already allocated, what the unknown value might be. This can be a powerful tool for making diagnoses and predictions but is totally dependent upon the quality, accuracy and representativeness of the other information in the table. There are many reason why the inference may be 'incorrect' and these are discussed in the USER TECHNICAL DOCUMENTATION. This is essential reading before making any assumptions about the quality or otherwise of the inference. Part of the object of domain construction is to make the inference 'come out right'.

The F10 key only operates on 'unknown' values. Using it on other values has no effect.

TYPICALITY INDEX (TI)

When some values have been allocated using the spreadsheet the typicality index can be used to order the elements from most to least typical of the domain. If all cells in the table are 'unknown' the elements will all be equally typical. 'Unknown' is treated like any other attribute value, and an element with 'unknown' values can be at the top of the list, if it is typical of elements in general to have 'unknown' values.

In version 1.3 there is no paging of elements in the typicality index. If there are more than 16 elements they will all be presented as one long list, the top of which may scroll off the screen. To stop the screen scrolling use Ctrl. Num Lock . To continue the display press any key. To send output to the printer use Shift PrtSc . This can be used while the screen is scrolling and will 'capture' the screen as it is at the instant in time you do it. These functions can be used at any point in the program, and will be useful for other displays when the number of elements or attributes is larger than 16.

It is worth emphasising at this stage that you may or may not agree with the ordering for typicality amongst the elements. The program simply processes the values specified in the spreadsheet according to a procedure which is thought to have some relation to human processing. Even if the process exactly mimics human processing, the program is still very much at a disadvantage because it does not have the wide range of 'world knowledge' that a person has. It only knows what you have told it. The prorams ordering of typicality may not agree with yours for much the same reasons as the inferencing might produce a result you disagree with or know to be 'wrong'. It is the job of the domain expert to describe the elements using a set of attributes that will produce the desired result. The job of the program is to alert the domain expert to problems and deficiencies in the domain specification, and it does this by presenting analyses the results of which will 'look wrong' if the domain is inadaquate. This applies to most of the calculation and display routines.

Pressing 'return' at the end of the typicality index display returns you to the menu.

```
Typicality Index

 16    Leopard                62
 14    Lion                   62
 27    Bear                   59
 28    Mouse                  58
 30    Fox                    58
 24    Cat                    57
 23    Dog                    56

  3    Chicken                42
  6    Ape                    42
 10    Whale                  38
  9    Fish                   37

Press return to continue _
```

QUERY BY EXAMPLE (QE)

Selecting Query by Example displays the elements and asks you for the number of the element you want to compare with the others. If there are fewer than 32 elements pressing 'return' at this point takes you back to the menu. If there are more elements pressing 'return' takes you on to the next page of elements (and so on to the last page). If you enter an element number (on the last page only if there is more than one page) the program will display a list of elements ordered from most similar to least similar with the selected element. As a rule of thumb you can regard elements in the first half of the list as being similar (in general) and elements in the second half as being dissimilar.

The list is not paged in version 1.3. Use Ctrl. Num Lock and Shift PrtSc to stop and print the screen, as described under Typicality Index.

Pressing 'return' takes you back to the display of elements (the first page if there is more than one page), so that you can select another element if you wish. Press 'return' (several times if there are several pages of elements) to get back to the menu.

Similarities with - Cow

5	Bull	91
4	Goat	91
8	Sheep	83
7	Lamb	75
2	Pig	75
27	Bear	66
16	Leopard	66
14	Lion	66
30	Fox	58
24	Cat	58
23	Dog	58
15	Tiger	58
3	Chicken	58
31	Man	50
28	Mouse	50
26	Kangaroo	50
18	Elephant	50
17	Giraffe	50
6	Cock	50
29	Squirrel	41
25	Lizard	41
22	Pelican	41
12	Octopus	41
19	Bird	33
11	Dolphin	33
10	Whale	33
21	Robin	25
20	Sparrow	25
13	Sea-horse	25
9	Fish	25

Press return to continue _

ELEMENT COMPARISON (EC)

Selecting element comparisons displays the elements (maybe several pages) and asks you to select an element. Pressing 'return' gets you back to the menu. If you enter an element number you are asked for a second element. If you enter 'return' here the elements are re-displayed, to allow the opportunity to correct errors on entry of the first element. If you enter the second element number a display of the similarities between the two selected elements is presented.

The left-hand column shows the attributes on which the selected elements share the same value, and the right hand column shows what that value is.

Press 'return' to see a display of the differences. The left hand column shows the remaining attributes. There are now two other columns headed by the names of the selected elements. In each are the values that the elements have been allocated in the spreadsheet. Since the similarities have been taken out the two values in any row will always be different.

In version 1.3, displays of similarities and differences are both unpaged. Press 'return' to get back to the menu.

```
Similarities between : Cow
                 and : Fish


Wings                    No
Killed for               Food
Distinct feature         unknown


Press return to see differences..... _
```

```
Differences between :

                         Cow                 Fish

Relation to man          Farm                Wild
Element                  Land                Sea
No. of legs              4                   None
Fins                     No                  Yes
Sex                      Female              Either
Meat                     Beef                Sea Food
Respiration              Air                 Water
Size                     Medium              Small
IQ                       Medium              Low

Press return to continue _
```

COMPARISON OF ATTRIBUTE VALUES (CAV)

Selecting this displays the attributes and asks you to select one. The attribute display operates like the element display, and will be paged if there are too many attributes to fit on the screen. If you select an attribute you are presented with the comparison of attribute values display for that attribute.

The display is a table of attributes (rows) by the values on the selected attribute (columns). The cells in the table are filled with the most frequent value on the attribute in the row that occurs in associated with the attribute value in the column.

Reading down a column you see the most popular values on all other attributes for a value of the selected attribute. Some cells may be blank. This means that more than one value tied for being most popular, so there is no clear most popular value. Where a complete column is blank it means that no element has been given that value on the selected attribute. If only one element has been given a particular value on the selected attribute the values in the column will be those of the single element.

Where there are too many attributes for the table to fit on the screen you can change the window on the table. In version 1.3 this is done by selecting a range of attributes to display. If you do not want to change the window pressing 'return' takes you back to the attribute display so that you can compare values on another attribute. You return from this to the menu in the usual way (pressing 'return' once or more depending on the number of pages).

Comparison of attribute values on No. of legs

	unknown	None	2	4	6	8
Relation to		Wild	Wild	Wild		Wild
Element		Sea	Land	Land		Sea
Wings		No	Yes	No		No
Fins		Yes	No	No		No
Sex		Either	Either	Either		Either
Meat			Not Eate	Not Eate		Sea Food
Killed for		Not kill	Not kill	Food		Food
Respiration			Air	Air		Water
Distinct fea		unknown	unknown	unknown		unknown
Size		Small	Small	Medium		Medium
IQ		Low	Low	Medium		Low

Do you want to change the current window _

ATTRIBUTE VALUE DISTRIBUTIONS (AVD)

Selecting this presents you with the attributes again, and this time making a selection results in the distribution of attribute value display.

This is another table, of sorts, the first column of which is the attribute value number. The second column shows all the possible values on the selected attribute in the order in which you entered them. The third column, headed 'Qty', is the number of elements that have that attribute value on the selected attribute.

The fourth column is the 'a priori' probability. This is nothing more than the proportion of attributes that have that value on the selected attribute. If you pick an element at random from the domain, and you know nothing about it, the probability of its having a particular value on the selected attribute is the value's 'a priori' probability. The probability of it not having that value is one minus the 'a priori' probability (1 - p(Vi)).

```
Distibution of values on the attribute -

No. of legs

    Value          Qty  A Priori  Profile

0   unknown          0   0.000
1   None             4   0.129    *****
2   2                8   0.258    **********
3   4               18   0.581    ************************
4   6                0   0.000
5   8                1   0.032    *

Press return to continue _
```

The last column is headed 'Profile' and draws a line the length of which is proportional to the 'a priori' probability.

Press 'return' to go back to the attribute display where you can select to see the attribute value distribution for another attribute, or press 'return' again to get back to the menu. The attribute value distribution can be displayed in a different way by using the Attribute Value Statistics (AVS) option from the menu.

CROSS TABULATIONS (CT)

If you select this you are presented with the attributes and asked to select two attributes. A table of the frequency of elements in each attribute value combination is presented. The table can be displayed reversing rows and columns by entering the selected attributes in the reverse order. You can change the window if there are so many values of the attribute running across the screen that they will not all fit, or to display any selected part of the cross tabulation matrix.

Press 'return' to get back to the attribute display, then select another or press 'return' again to re-display the menu.

Cross Tabulation

No. of legs with Fins

	unknown	Yes	No
unknown	0	0	0
None	0	3	1
2	0	0	8
4	0	0	18
6	0	0	0
8	0	0	1

Do you want to change to current window _

SEE PROTOTYPE FOR AN ATTRIBUTE VALUE (PAV)

You are presented with the attributes and asked to select the one for which you want to see the prototype. You are then presented with the values on that attribute and asked to select one. The prototype displayed is for that particular value. Pressing 'return' at any stage will move you back towards the main menu.

The table presented shows some detailed characteristics of the attribute value selected. Up to now we have been looking more generally at elements and attributes. Now we are looking in detail at a single value.

Prototype for Relation to man : Farm

Element	Land	Association Index	Distnguish-ability Index
No. of legs	4	V.Strong	Poor
Wings	No	Strong	Poor
Fins	No	Strong	Poor
Sex	Female	V.Strong	Poor
Meat	Beef	Medium	V.Good
Killed for	Food	Weak	V.Good
Respiration	Air	V.Strong	Good
Distinct featu	unknown	V.Strong	Poor
Size	Medium	V.Strong	Poor
IQ	Low	Strong	Medium
		Strong	Poor

Press return to continue

The first column lists the attributes. The second column lists the most frequent value for the row attribute which is associated with the selected value. Reading down the column can be regarded as looking at a typical member of the set of elements, although the column may not match perfectly with any particular element.

The third column is headed 'Association Index' and shows the degree of association between the value in column 2 and the selected value. The association index answers the question:

> If I know that an element has the selected value,
> what is the probability that it will have the value
> in column 2 ?

The possible values for the association index are - 'Very Weak', 'Weak', 'Medium', 'Strong', and 'Very Strong'. Note that the value in column 2 has already been pre-selected. It is the most frequent value associated with the selected value. The 'association index' is telling us how significant that association is in the direction from the seleced value to the column 2 value.

The fourth column is headed 'Distinguishability index'. This answers the same question the other way round:

> If I know that an element has the value in column 2, what is the probability that it will have the selected value ?

The possible values for the distinguishability index are 'None', 'Very Poor', 'Poor', 'Medium', 'Good' and 'Very Good'.

You may well ask whether these two questions are different. In fact they are, but it is easy to forget it. If 'all men are mortal' it does not mean that 'all mortal things are men'. The association index assesses the extent to which all A are B, while the distinguishability index assesses the extent to which all B are A.

Pressing 'return' takes you back to select another value, on the original attribute. If you choose not to and press 'return' again you are re-presented with the attributes, so you can select a new attribute. Pressing 'return' here takes you back to the menu.

SEE/EDIT SEARCH PROFILE (SSP)

The display for the search profile is like the See/edit Spreadsheet option. It too is a spreadsheeted display in which you move around with the cursor keys and allocate values to cells in a table. Like the main attribute / value table spreadsheet you can move down if there are more attributes than fit on the screen at one time. You cannot move across because there are only two columns in the table.

The attributes are listed down the screen to the left of the table. The first column of the table lists the values you want to search for. The second column sets a criterion for the search on the value in the same row. This can be 'Essential', 'Important', 'Preferable', or 'Doesn't matter'.

```
See/edit search profile and Do Search

                                  Value          Importance

Relation to man                   unknown        Doesn't matter
Element                           unknown        Doesn't matter
No. of legs                       2              Essential
Wings                             Yes            Preferable
Fins                              unknown        Doesn't matter
Sex                               unknown        Doesn't matter
Meat                              unknown        Doesn't matter
Killed for                        unknown        Doesn't matter
Respiration                       unknown        Doesn't matter
Distinct feature                  unknown        Doesn't matter
Size                              unknown        Doesn't matter
IQ                                High           Important

Move round with cursor keys
F1-End Edit  F2-Reset          F7-Values  F9 to edit  F10-Do Search
```

```
Attribute search results

Man                               66
Pelican                           33
Robin                             33
Sparrow                           33
Bird                              33
Cock                              33
Chicken                           33
Kangaroo                          0

Press return to continue
```

As with the main attribute / value table only the cursor control keys and five of the function keys are active. These are:

F1 Pressing this takes you out of the search profile and back to the menu. It is the only way to exit from the search profile (same function as in SS).

F2 This allows you to reset the search profile. In its reset state all values to be searched for are 'unknown' and all criterea are 'Doesn't matter'. (Different function from SS).

F7 This lets you take a quick look at the possible attribute values for the attribute in the row you are currently in. You can also see these by cycling through them (see F9), but this lets you see them all at once, with their full names. Press return to get back to the spreadsheet. (Same function as in SS).

F9 When you are in the first column of the table the F9 key allows you to cycle through the attribute values for the attribute in the row you are in. You cycle through them in the order in which they were entered in the see/edit/add attribute values routine. Whatever value you leave in the cell is the value you set to the attribute in that row for the search profile. If no values have been set the complete column will consist of the value 'unknown'. You can also re-set 'unknown' to values that have already been set.

When you are in the second column of the table the F9 key allows you to cycle through the choice of criteria for the search on the value specified in column 1. If you pick 'Essential' the search will only display elements that have the value. If you pick 'Important' or 'Preferable' all elements that pass the 'Essential' criteria on any other attribute value, are ordered, with higher scores being given to elements that have a value that is 'important' to those that have a 'preferable' value. The weighting for 'important', in the search, is twice that of the weighting for 'preferable'. If you pick 'Doesn't matter' the search will ignore that attribute even though you may have set a value for it in column 1.

You can move around the search profile as you like allocating and changing values as you go. The F9 key has a similar function to the use of the key in SS.

F10 At any time you can press the F10 key to do the search. The search will be carried out by going through the elements using the profile as a filter. Only elements that have 'Essential' values will pass through the filter. Once through, the elements are ordered according to the extent of the match on 'Important' and 'Preferable' attribute values. You do not have to specify any 'Essential' attributes and if you do not all the elements, in order of similarity with the weighted search profile, will be displayed. Press 'return' to get back to the search profile.

ATTRIBUTE VALUE STATISTICS (AVS)

This displays exacly the same information as the Attribute Value Distribution (AVD) routine, but cycles through the attributes each time you press 'return', until you reach the last attribute when you return to the menu. It is a useful way of seeing a quick summary of all the attribute values.

ATTRIBUTE STATISTICS (AS)

This shows some summary statistics for the attributes. A table is displayed. Column 1 is the attribute number and column 2 is the attribute name. Column 3 shows the number of possible values the attribute can have. This is the number of values entered using the see/edit/add attribute values (SAV) option. Column 4 shows the number of those values that have been allocated to elements in the main table. Not all values you think you will need for a domain are necessarily used, so where a cell in column 4 contains a smaller number than a corresponding cell in column 3, not all the values for that attribute have been used. Column 5 is an index calculated by dividing column 4 by column 3 and multiplying by 100. You can scan this column to compare the extent to which you have used all the possible values of each attribute.

At the bottom of the display the values for the attributes with the largest and smallest number of values found are shown. Where two values tie for largest or smallest number, the program arbitrarily picks the one it comes across first. These attribute values are those used by the MAX and MIN search strategies (see below), so it is worth noting what they are if you intend to use these strategies.

Pressing 'return' takes you back to the menu.

No.	Attribute Name	No. Values Possible	No. Values Found	F/P * 100 Index
1	Relation to man	4	3	75
2	Element	4	3	75
3	No. of legs	6	4	67
4	Wings	3	2	67
5	Fins	3	2	67
6	Sex	4	3	75
7	Meat	7	7	100
8	Killed for	8	6	75
9	Respiration	3	2	67
10	Distinct feature	7	7	100
11	Size	6	5	83
12	IQ	4	3	75

Largest no. of values found for - Meat
Smallest no. of values found for - Wings

Press return to continue _

SEE PROBABILITY OF IMMEDIATE CLASSIFICATION INDEX (PIC)

Selecting the PIC index displays the attributes and you select an attribute to be the 'class' attribute. The 'class' attribute is an attribute you want to find out something about. For example the class attribute might be 'diagnosis'. The PIC index answers the question:

> If I want to find out which value of the class attribute an element has, and I know nothing about the element, which other attribute should I select to find out about in order to have the best chance of classifying on the class attribute ?

This seems like a complicated question, but is in fact asked all the time in some form or another. If I want to know which shops sell bread, what chacteristics of 'shop' should I look out for? In this example the elements are 'shops', the class attribute is 'type of goods sold', 'bread' is a particular value of the class attribute, and the 'characteristics of shops' worth noting are the attributes of 'shops' that have some association to the class attribute 'type of goods sold'.

Often an expert is called on to make a classification of some sort. A doctor is asked to classify patients into diagnostic groups, or a geologist is asked to classify a particular site in terms of the likelihood of finding oil. In order to make the classification the expert has to collect data about the element (e.g. by examining the patient and carrying out tests). The expert is able to do this because he knows which attributes of elements give indications about the classification. A non-expert would not know which attributes to look for.

The PIC index shows, for any selected class attribute, which attributes are useful in order to perform the classification. The display of the index is ordered from highest to lowest, so the attribute at the top of the list will be the most useful one to find out about in order to arrive at the classification as quickly as possible.

The PIC index can be used in conjunction with the MIN strategy to manually construct a decision tree for a class attribute.

Pressing 'return' takes you back to the attributes display, and pressing 'return' again gets you to the menu.

Probability of each attribute classifying the class attribute
Fins

3	No. of legs	87
2	Element	84
7	Meat	38
1	Relation to man	36
8	Killed for	35
6	Sex	23
4	Wings	19
10	Distinct feature	18
11	Size	13
12	IQ	0
9	Respiration	0

CREATE SUB-DOMAIN (MAX)

When you select MAX you are given a warning that you should save the domain in memory (if you have not already saved it or if you have changed it since last saving it) before creating a sub-domain. When you create a sub-domain the rest of the domain is lost unless you save the original domain first.

Press 'return' to continue and you are presented with the values of the attribute which has the maximum number of values found. This is the attribute shown at the bottom of the AS display. If you do not want to use this attribute, or if you want to go back and save the original domain first, press 'return' and you will get back to the menu.

Note that the attribute chosen for MAX may be only one of several attributes that tie for having the highest number of values found. You may prefer to use another, in which case press 'return' to get back to the menu. Select AS to look at the table which shows the number of values found in the main attribute value table, and find the attribute you want. Then use MAN to select the attribute before selecting a value.

If you select one of the values, a sub-domain will be constructed. The sub-domain will consist of only elements that have the value you specify on the MAX attribute. Since all the elements in the sub-domain have only this value on the MAX attribute, the attribute no longer distinguishes between the elements, so it too is removed from the sub-domain.

You are left with a sub-domain which consists of the sub-set of elements which have the value you selected on the MAX attribute, and all the attributes except the MAX attribute. In version 1.3 it is advisable to save this sub-domain once you have created it, then to 'quit the domain' and re-load the sub-domain. You can then use all the facilities of the program on the sub-domain in the same way as if you were using the full domain.

CREATE SUB-DOMAIN (MIN)

This operates in the same way as MAX except that this time the attribute values are displayed for the attribute that has the smallest number of values found. This attribute is also shown at the bottom of the AS display.

CREATE SUB-DOMAIN (MAN)

This is similar to MAX and MIN except that it does not pre-select the attribute. The attributes are displayed and you pick one out. You are then presented with its values and a sub-domain is created for whatever value you select. As usual, pressing 'return' only at any stage will take you back towards the menu.

Creating sub-domains and decision trees

By using a combination of the AS, PIC and MAN facilities you can go through the domain manually constructing a decision tree. Version 1.3 only provides the facilities for constructing these trees on paper. The process is not automated and no record of the tree is maintained by the program. See USER TECHNICAL DOCUMENTATION for more details.

SAVE SUB-DOMAIN (SSD)

The save sub-domain option operates like the save domain (SD) option except that it will prompt you for a new file name for the sub-domain so that you don't overwrite the original domain.

Removing Elements and Attributes

Elements can be removed from the domain in two main ways. The first way is to re-name the element with a new element name, then to alter all the values in the main attribute value table using SS. This keeps the overall number of elements the same.

Using the second method you can remove several elements at once. Enter a new attribute called 'Keep Flag' and give it the single value 'Keep'. It will already have the value 'unknown'. Use SS to allocate the value 'Keep' to those elements you want to stay in the domain, then use MAN to create a sub-domain for all the 'Keep' values. All the elements with the value 'unknown' on the 'Keep Flag' attribute will be removed, along with the 'Keep Flag' attribute itself.

Like elements, attributes can be removed by changing their names, and changing the names of the attribute values associated with them. It is also possible to remove attributes by changing all the values for the attribute column in the main attribute value table to a single value (using SS), then using MAN and selecting on this value.

In version 1.3 you should save all sub-domains and re-load them before continuing.