The use of complementary techniques of machine learning to discover knowledge in real complex domains

Thesis · July 2002

CITATIONS
0

READS
81

1 author:

David F. Nettleton
Innovació i Recerca Industrial i Sostenible / Universitat Pompeu Fabra
117 PUBLICATIONS **579** CITATIONS

SEE PROFILE

Departament de Llenguatges i Sistemes Informatics
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Doctoral Thesis Dissertation

"The use of complementary techniques of machine learning to discover knowledge in real complex domains"

Presented for the Doctorate of Artificial Intelligence

Doctorand: David F. Nettleton

Programa de doctorat en Intel.ligència Artificial
Departament de Llenguatges i Sistemes Informàtics

Instititution: Universitat Politècnica de Catalunya
Date: 2nd July 2002
Thesis Directors: Dr. Vicenc Torra (IIIA-CSIC), Dr. Juan Jacas (EA-UPC)
Thesis Tutor: Dr. Javier Bejar (LSI-UPC)

Summary

This thesis is concerned with developing and refining a collection of methods and tools which can be applied to the different steps of the Data Mining process. Data Mining is understood as the analysis of data using sophisticated tools and methods, which include aspects of data representation, data exploration, knowledge discovery, data modelling and data aggregation. Data Mining can be applied in real and complex domains, such as the domain of clinical prognosis, as well as with artificial test, or benchmark data. Medical informatics is a dynamic area where new approaches and techniques are constantly being developed, the objective being to improve current data representation, modelling and aggregation methods to achieve better diagnosis and prognosis. In this work we focus on two medical data domains: prognosis for ICU patients and diagnosis of Sleep Apnea cases, although it is proposed that the techniques have general use for any data domain. A key approach which is used for data processing and representation is that of fuzzy logic techniques. Existing techniques are benchmarked against the data, such as neural networks, tree induction and standard statistical analysis methods such as correlation, principal components and regression models.

We carry out a survey of existing techniques, authors and their approaches, in order to establish their strong and weak points, limitations, and opportunities where improvement may be achieved.

The first major area under consideration is data representation: how to define a unified scheme which encompasses different data types, such as numeric, continuous, ordered categorical, unordered categorical, binary and fuzzy; how to define membership functions; how to measure differences and similarities in the data. This is followed by a comprehensive benchmarking of existing AI and statistical algorithms on a real ICU medical dataset, comparing the 'Data Mining' results to methods proposed by the author.

We define *'fuzzy covariance'* as a value which permits the measurement of relation between two fuzzy variables. Previous fuzzy covariance work was limited to the covariance of a fuzzy cluster to its fuzzy prototype [Gustafson79]. More recent authors [Nakamori97][Wangh95][Watada94] have created specialised fuzzy covariance calculations tailored for specific applications. In this work, a general fuzzy covariance algorithm, which measures the fuzzy covariance between two fuzzy variables, has been conceived, developed and tested. The initial work based the Hartigan joining algorithm and fuzzy covariances evolves into and is contrasted with the later work on data and attribute fusion using the WOWA aggregation operator .

We consider *'aggregation operators'* as a method for modelling data for clinical diagnosis, and use 'relevance' and 'reliability' meta-data together with grades of membership to enhance the information which the aggregation operator receives in order to model the data. We also make enhancements to the WOWA operator, to enable it to process data with missing values and we develop a novel method for learning the weighting vectors.

Acknowledgements

We have collaborated with three hospitals over the 5 year investigation period (1996 to 2001): the Hospital of Sabadell, the Respiratory Disease Institute of the Hospital Clinic of Barcelona, and the Sleep Centre of the Hospital of the Santisima Trinitat, Salamanca. The author is indebted to the medical experts who have collaborated enthusiastically in the different application areas of the work: Dr. Xavier Companys (previously) of the Hospital Parc Taulí of Sabadell for collaborating in the data mining, interpretation and validation of results using the ICU data presented in Section 4.1; Dr. Lourdes Hernandez of the Respiratory Disease Institute, Hospital Clinic of Barcelona for providing the first set of Apnea case data used in Section 4.3, evaluation of reliability and relevance of the Apnea variables, and in providing clinical background information; Dr. Joaquín Muñiz , Dr. Marina Rodriguez and the staff of the Sleep Centre, Hospital de la Santisima Trinidad, Salamanca, for their participation in providing the second set of Apnea case data used in Section 4.4, evaluation of reliability and relevance of the Apnea variables, in helping define the scalar (fuzzy) questionnaire, and especially in carrying out the clinical interviews with the patients and tests in order to capture the data.

I am also indebted to my thesis directors, Dr. Vicenc Torra, of the Institut d'Investigació en Intel.ligencia Artificial (IIIA), Bellaterra, Spain, and Dr. Juan Jacas of the School of Architecture, Universitat Politècnica de Catalunya, Barcelona, Spain. I also thank Dr. Ulises Cortés, director of the Artificial Intelligence Ph.D course, Dept. Languages and Systems, Universitat Politècnica de Catalunya, and Dr. Karina Gibert of the Dept. Mathematics and Statistics of the Universitat Politècnica de Catalunya. I would like to thank them all for their patience and guidance in the work.

Index

Chapter **Page**

Index (cont.)

Chapter **Page**

Index (cont.)

List of Figures

List of Figures (cont.)

List of Tables

List of Tables (cont.)

List of Tables (cont.)

Chapter 1. Introduction - general

The thesis covers the work carried out from 1996 to 2001, and is concerned with developing and refining a collection of methods and tools which can be applied to the different steps of the Data Mining process. This first requires the consideration of how to represent and process mixed categorical, numeric and fuzzy data types using aggregation, variable clustering and fuzzy techniques. The first section of work covers the period 1996-1997, in collaboration with Dr. Karina Gibert. This is followed by the work on contrasting different data modelling techniques such as clustering, neural networks and tree induction. In a second period, there is the work with Dr. Vicenc Torra and Dr. Juan Jacas from 1997-2001, which is centred on the use of aggregation operators such as WOWA to process real medical data domains, and the solution of some of the problems of these operators, such as missing data and weight assignment.

In addition to the standard test data sets such as Iris, and those published by Hartigan, Bezdek and Torra, two medical problem domains have been used, in collaboration with three hospitals over a five year period: ICU patient data from the Parc Tauli Hospital, Sabadell, Spain; Apnea patient data from the Hospital Clinic, Barcelona, Spain, and Apnea patient data from the Sleep Clinic, Salamanca, Spain.

1.1 Introduction - detail

Data analysis and data representation are two areas which have been revolutionised by the advent of machine learning methods from the 1950's onwards. In the mid-1960's, Zadeh introduced fuzzy concepts in data analysis, which was further developed by Bezdek and the fuzzy c-Means algorithm. Other key developments were those of neural networks for supervised modelling, of which feedforward NN were the most common, one of the earliest references being [Rosenblatt59]. Rule induction came later and Quinlan introduced ID3, which became the first 'industry standard' algorithm. Neural networks lost popularity in the 1970's due to some key unresolved theoretical problems, but were to come back in the 1980's. Expert systems became popular in the 1980's but with the advent of the following decade they were absorbed into hybrid and problem specific applications. Rule based systems became a combination of expert knowledge and rules automatically induced from historical data, together with Cased Based Reasoning and other approaches such as Belief Networks. Another approach has been that of the AI data aggregators, which matured into useful tools, especially due to the consolidative work of Yager published in the late 1980's.

1.1.1 Motivation

Many aspects of data analysis and data representation are still unresolved when data does not fall into well defined categories, or cannot be represented by simple forms. Especially in medical data analysis, there is a constant search for methods which give improved diagnostic precision for positive and negative cases, and prognostic accuracy for medium and long term recovery. The debate on how to best represent and capture data, also is an active field with no best solutions. Another aspect is the requirement of many algorithms to require large data volumes to be able to work. This is despite the fact that many real medical and other domain data sets are relatively small, that is, with less than 150 cases, while being defined by a large number of variables, that is to say, more than 15. There are still many statistical and data mining techniques which resort to arbitrary type assignment of variables in order to be able to input the data into data exploration or data modelling algorithms. In the case of the standard WOWA operator, we need it to be able to process data with missing values, with a minimum loss of overall precision. Also we needed to be able to learn the weighting vectors of the WOWA operator from historical data, given the difficulty of manual weight assignment for a real data domain. The development of a method to compare fuzzy variables and 'join' them into a reduced number of most significant factors, stems from the need to explore and model a data set containing these variables types.
If we review existing commercial Data Mining toolkits, such as Clementine, Darwin or SAS Enterprise Miner, we can identify common shortcomings, such as the lack of fuzzy data processing or representation, the impossibility of defining multiple weighting vectors as input to data model, and the lack of aggregation operators and modelling algorithms which give acceptable results for datasets consisting of a small number of cases.

1.1.2 Objectives

We wish to develop a collection of methods and tools which can be applied to the different steps of the Data Mining process: data representation, data exploration and data modelling. One of the objectives of the work is to review existing techniques, applying them to real and artificial datasets, thus identifying their limitations. In this manner we may define areas susceptible to improvement and we can develop techniques which give better solutions for the given data and application domains. A review of a selection of the 'best of breed' AI supervised and unsupervised techniques

shows their strengths and weaknesses in processing real data sets. The techniques reviewed include clustering algorithms, such as Kmeans, fuzzy c-Means, Kohonen SOM. In the case of classification or predictive modelling techniques we benchmark algorithms such as Feedforward Neural Networks, C4.5 and ID3 rule induction, Linear Regression and Logistic Regression.

We study different aspects of the nature of data, for example its type, that is, numeric, categorical, binary, and so on. We study different ways of representing and analysing data, such as by clustering and classification, adding value to it by using weight criteria to indicate attribute reliability and relevance, creating models by aggregation, and eliciting underlying data structure. It follows from Section 1.1.1, that we are also interested in finding techniques of representing and processing which make it possible to extract a classification, clustering or predictive model from a relatively small number of cases.

Thus we will develop tools and methods for all steps of Data Mining, from the initial representation and definition of the data, the exploration phase which includes the study of relationships between variables which may be defined with different types, and finally the modelling phase. These tools will enable us to represent and process data in the fuzzy form, together with non-fuzzy data. In the case of the exploration phase, we use such algorithms as Hartigans 'joining algorithm' and a new fuzzy covariance distance calculation. In the case of the modelling phase, we use aggregation operators such as WOWA, to process datasets with a small number of cases. WOWA needs to be modified to process data with missing values, and a method for learning the weighting vectors used by WOWA, from historical data.

1.1.3 Scope and Orientation

The scope in terms of data is defined as standard test datasets, and several real medical domain datasets captured specially for this work. In terms of data representation, a diversity of different data representation types are reviewed, and the case for the fuzzy form is evaluated. In terms of data processing methods, a selection of standard methods are tested against the data, such as, neural networks, rule induction and classical statistical methods. We then test complementary techniques, such as those of Hartigan, fuzzy c-Means and *aggregation operators*. Two emphases are made in the orientation of the work: (i) the use of fuzzy techniques to enhance existing data analysis and representation methods; (ii) their application to medical data for prognosis in the case of the ICU data, and diagnosis in the case of the Apnea data. In Figure 1 we can see a summary of the different methods which have been developed, together with those used for testing and benchmarking, and their relation to the steps of data mining. We note that clustering methods such as Kmeans or Kohonen SOM, are restricted to the data exploration phase, whereas classification methods such as rule induction are used both in the data exploration phase and the data modelling phase. In Chapter 2 we enter into detail of the clustering and classification methods which have been used.

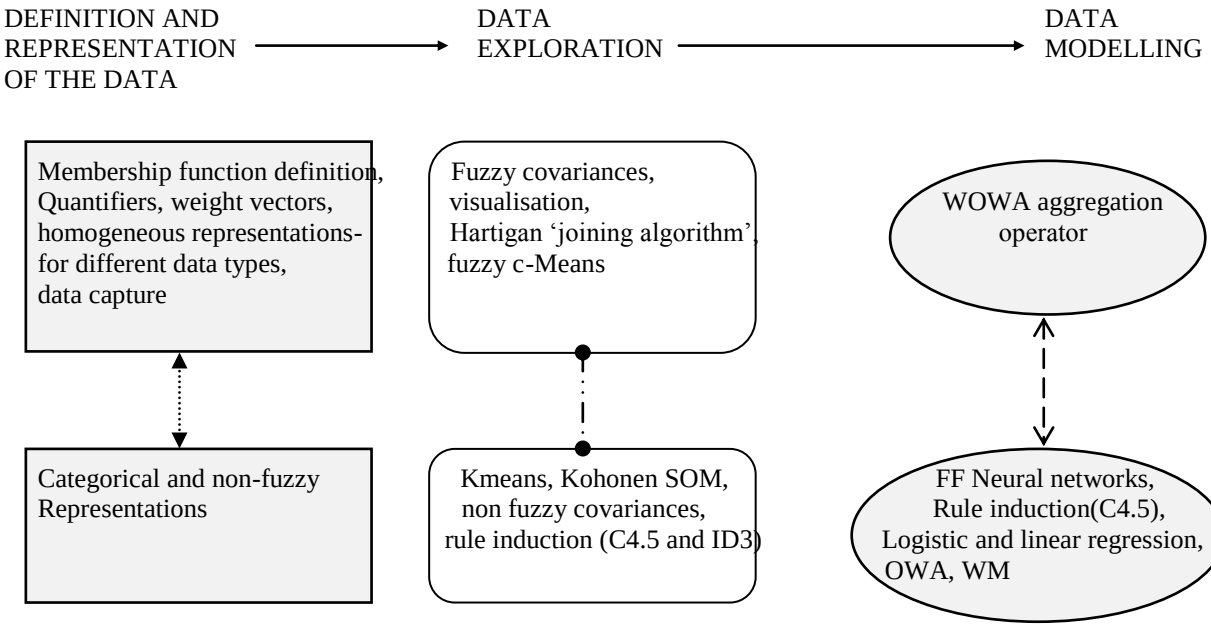


Figure 1. Tools and methods used and developed (centre row), their relation to the different steps of Data Mining (upper row), and existing methods used for benchmarking (lower row)

1.1.4 Overview of thesis organisation

The thesis is organised in the following chapters:

Chapter 1 defines the motivation, objectives, scope and orientation and organisation of the work detailed in the thesis dissertation. A brief overview of the state of the art and previous work is detailed in Chapter 1, including a summary of the main contributions which the work of the thesis gives to the field.

Chapter 2 consists of a review of preliminaries in key areas which the thesis deals with, including a detailed review of the history and evolution of AI and statistical data analysis and data modelling fields. This also includes description and discussion of key algorithms such as Hartigan's 'joining algorithm', aggregation operators (WM, OWA, WOWA, …), C4.5, ID3, Kohonen SOM and fuzzy c-Means. There is also discussion of how to best introduce fuzzy concepts into algorithms such as C4.5 and the Kohonen SOM.

Chapter 3 summarises the theoretical aspects and ideas related to the development work which has been undertaken during the four year period. This includes the evolution of ideas with respect to issues such as data fusion of variables of different types, comparison of variables of different types to generate covariances, and ideas for representation of crisp and fuzzy data attributes. The consideration of aggregation operators, provides a contrast to 'data fusion', and we consider variable and data aggregation, with application to medical diagnosis and prognosis, and developing solutions for problems such as missing values, weight assignment. We also consider a new weighting 'bias' scheme, based on a vector of 'vectors'.

Chapter 4 gives results of the application of the methods and algorithms to artificial and real data sets, with emphasis on some real medical domain problems. The real domains included are: ICU prognosis and Apnea syndrome screening. In Section 4.1 there is an extensive analysis of a real hospital ICU dataset, using first standard statistical techniques such as principal components, distribution analysis using plots and histograms, and correlation analysis. We then apply data mining techniques to the data: ID3 and C4.5 induction, back propagation neural network and Kohonen SOM. Finally we contrast these previous techniques with two approaches which are not usually included in 'Data Mining toolkits': we use the Hartigan 'joining algorithm' with crisp and fuzzy covariances as input to analyse relationships between attributes; and we use fuzzy c-Means to cluster data and give indications of relation between variables and the cluster prototypes.

In Section 4.2 we apply four variants of a novel fuzzy covariance algorithm [Nettleton98b] to artificial datasets to generate a fuzzy covariance matrix which is then given as input to the Hartigan 'joining algorithm'. The objective is to identify and rank the most significant attributes in each dataset. The benchmark results are compared with C4.5 and a Neural Network applied to the same data.

In Section 4.3 the OWA and WOWA aggregation techniques are applied to a dataset of Apnea cases from the Hospital Clinic of Barcelona, the data being captured in a crisp form, and the output being a binary valued diagnosis. Both the OWA and the WOWA operators use reliability and relevance vectors for input variable weighting which are assigned by a consensus of medical expertise and statistical analysis. We use the new weighting 'bias' scheme for the reliability weights.

In Section 4.4 we apply the WOWA aggregation operator to diagnose Apnea cases using a dataset collected by the Hospital of the Santisima Trinitat, Salamanca. In this case, the data was captured in both crisp (categorical) and fuzzy (continuous scale) form, using a specially designed questionnaire. Different types of weight assignment were tried: machine learning, medical expert assignment, machine learning and medical expert assignment. Also, the WOWA precision for diagnosing positive and negative cases was benchmarked against ID3 tree induction and a feedforward neural network. The data processing differs from the crisp Apnea data of Section 4.3, given that we also incorporate membership grade values as part of the input data. We use the techniques developed in Chapter 3, to process missing values and learn the 'relevance' weights from historical data using a genetic algorithm.

Chapter 5 summarises the work and draws together some conclusions which may be made from the results. Finally, the annexes include a selection of documents and forms used, together with a complete bibliography of references given in the text.

1.2 State of the Art and Previous Work

In this section we give an overview of some of the most recent works of investigation and innovative ideas in areas relevant to the thesis, such as the work of Takagi and Sugeno, Dubois and Nakamori and Baldwin in data modelling, representation, and factor analysis. Later, in Chapter 2, we enter into more detail of the ideas and work of the major authors such as Bezdek and Quinlan, with special reference to recent developments on their original algorithms, such as fuzzy c-Means and C4.5 rule induction, respectively.

In summarising the most important authors and papers in the fields which are relevant to the thesis work, the following areas are reviewed: fuzzy data analysis, especially the work of Zadeh and Bezdek; fuzzy data representation; aggregation of variables and data, especially the work of Yager and Torra; clustering; classification; medical diagnosis and prognosis; diagnosis of the sleep apnea syndrome. Later in Chapter 2, we enter into a greater level of detail for each of these aspects.

1.2.1 Data Mining

Data Mining is understood as data analysis with sophisticated tools, which allow processing and visualisation of multiple 'views', and the search for complex interrelations in the data. As well as presenting and manipulating known information about the Data, it allows the discovery of new information. Data Mining is characterised by the discovery of new knowledge.

Data Mining (or Knowledge Data Discovery) is also a data analysis process of an inter-disciplinary nature, whose proposal is to identify and extract high value knowledge from data. The datasets may be high or low volume, have many descriptive attributes, non evident structure, and include 'missing' values, errors and noise.

Data Mining uses diverse techniques to analyse and process data:

(a) Classical statistics: linear regression, correlation, and so on.

(b) Learning algorithms for classification and prediction: rule induction, neural networks, and so on.

(c) Data exploration using tools for graphical visualisation and manipulation.

Statistics, on its part, offers techniques such as automatic classification, discrimination, factorial methods and graphical visualization. The proposal of 'intelligent' algorithms, on the other hand, is to 'learn' from a dataset, and form a model which represents the environment, be it predictive or classicative. The techniques most often used are: neural networks to predict and classify, rule induction to explain the structure of a model and the profiles of the classifications, genetic algorithms for optimization problems, and correlation algorithms in order to identify the most relevant factors in a given problem. All these techniques are orientated towards the discovery of structure in a multidimensional dataset. The relationship between knowledge discovery in databases and classical data analysis is depicted in Figure 2.

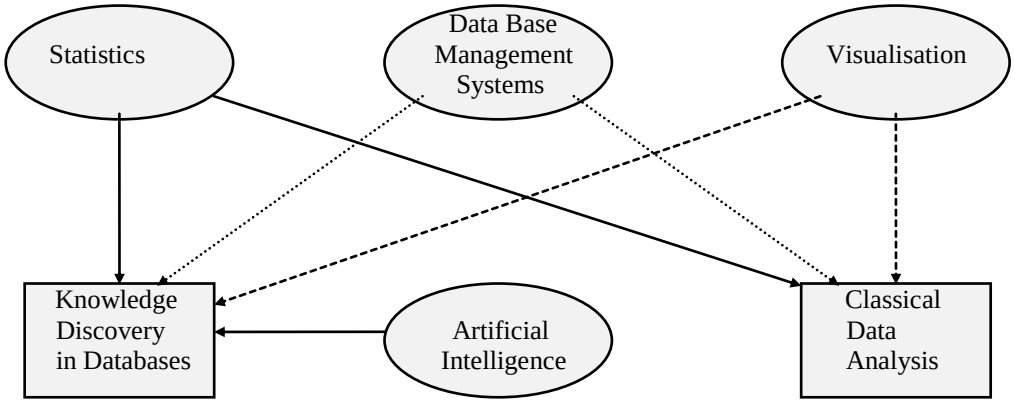


Figure 2. Knowledge Data Discovery and Classical Data Analysis understood as interdisciplinary areas

Data Mining applications are usually orientated towards knowledge discovery and the generation of models, often using techniques such as prediction, classification, segmentation, association and the discovery of sequences and time series. *Predictive* models are used, for example, to predict those male customers between 25 and 45 years of age with probability greater than 70% of contracting a pension plan. *Induction* models, on the other hand, can tell us the profiles of the 1000 most profitable customers for a product line **X**. *Association* processes extract information such as: if the customer has bought **A**, then s/he will buy **B**, in 65% of all cases. We can also use *sequence* discovery techniques to deduce, for example, that a purchase transaction of beer occurs before a purchase transaction of Coñac, for 2 out of every 5 customers. The discovery of *similar time sequences* provides us with information such as: if the customer has bought **A**, then s/he will also buy **B** in the next 3 months, in 70% of all cases. Finally, *segmentation*, or clustering, can describe underlying structures without having prior knowledge of the data. For example, we could establish common tendencies among customers in different geographical areas, and define a common offering for them.

Some of the key centres for Data Mining and investigators are: Usama Fayydad in MicroSoft Research, USA; Willi Klösgen in GMD (German National Research Centre for Information Technology) ; Heikki Mannila, previously of the University of Helsinki, Finland and now in Microsoft Research, USA; G. Nakhaeizadeh of Daimler Benz Research Centre AG,Forschungszentrum, Ulm, Germany; Gregory Piatetsky-Shapiro of GTE Laboratories, USA; Ross Quinlan, of the Centre for Advanced Computing Sciences, New South Wales Institute of Technology, Australia; Ken Totton, Data Mining Group, British Telecom, England; Barry Devlin, IBM Dublin, Ireland.

The approach of the group at Helsinki University, is based on the analysis of data sequences, and the identification of recurrent characteristics inside of sequences of events. They use Markov chains and Monte Carlo methods to examine the interdependence of events in detail. They apply clustering methods to find regularities in the data. One of the special approaches adopted by this group is based on Kohonen neural networks for unsupervised clustering.

Current focuses

At present and in the past decade, there has been much investigation with respect to neural networks, rule induction and genetic algorithms, combining these with classical statistics. There are also references to fuzzy logic concepts in Data Mining, especially for clustering, representation and the treatment of imprecision. In the area of hierarchical classification, the references tend to be related to tree induction.

Borgelt, of the University of Magdeburg, Germany, in [Borgelt97] has focussed on "Evaluation Measures for Learning Probabilistic and Possibilistic Networks", which characterises a fuzzy system with a learning capability. Borgelt has worked with the Data Mining Group at Daimler-Benz under Nakhaeizadeh. In [Borgelt97], chi-square and entropy measures are used to calculate the information gain/loss and propagate this information in a network. In Daimler-Benz the group is working on data reduction techniques for large numbers of attributes, with a reduced number of data types, and their algorithms are being tested with different data domains for benchmarking (not just fault analysis of car components and characteristics).

Dubois, of the Institut de Recherche en Informatique de Toulouse, France, in [Dubois97] has focussed on "User-Driven Summarisation of Data Based on Gradual Rules" in the context of data analysis. Some of the problems his group have encountered are: pre-process and dimension reduction and discovery of initial structure in the data. If we are interested in using Kohonen, C4.5 and c-Means to elicit the initial structure of the data, in a co-operative manner, c-means must be used with caution as it does not perform well with 'outliers', and it is necessary to define the initial number of clusters, such as in c-Medians and mixed c-Means. Alternatively, a simulated annealing (ID3) type algorithm could be used for finding a good initial solution. Also the Sugeno-Takagi model could be used as a substitute for the standard Kohonen SOM. The Kohonen SOM and c-Means may find very different partitions in the data, which would be an appropriate result in order to demonstrate contrasting techniques. [Dubois97] outlined a methodology for analysing a data set from scratch (step1, identify typical points; step2, compute cores; step3, refine rules). In the data example, there were only two attributes, because a previous pre-process of the data was assumed, in order to choose the most significant variables and the method focuses on creating rules from these variables.

In the EC (Esprit) StatLog project [StatLog94], a benchmarking was carried out of 20 principal algorithms for AI classification and classical statistics classification. The following algorithms were included: C4.5; Linear and quadratic discriminant; NewID (variant of ID3). No algorithm was included which had its basis in fuzzy logic (e.g. fuzzy c-Means).

Commercial data mining systems

There are a number of commercial data mining systems in existence for general purpose data mining. These are, for example, SPSS's Clementine, SAS Enterprise Miner, IBM's Intelligent Miner for Data, Thinking Machine's Darwin, and so on. All these systems contain algorithms which have come from investigation backgrounds. Apart from a basic set of statistical functions for data exploration and modelling, there are usually several algorithms for classification and clustering. These are typically: for prediction, feedforward neural networks, logistic and linear regression. For classification: rule induction algorithms, typically ID3 and C4.5 or similar. For clustering: Kohonen SOM Neural Network, K-means.

In the case of Intelligent Miner, for prediction there is also the Radial Basis Function (RBF), and for clustering there is the Condorcet Criterion (Demographic) model which enhances processing for symbolic type data. There are separate algorithms for association analysis, temporal sequences and sequential patterns, which are based on statistical frequency techniques and direct sequence pattern matching.

Enterprise Miner uses a data mining methodology-Sample, Explore, Modify, Model, and Assess (SEMMA). It has a 'canvas' type icon based interface which uses drag and drop to create data mining processes following the SEMMA methodology. It provides specific algorithms for associations, sequential patterns, decision trees (CHAID/CART/C4.5), neural networks, logistic regression, clustering(K-means), RBF and a wide selection of statistical techniques.

Clementine uses neural networks, regression and rule induction, with Kohonen nets for clustering and C4.5 rule induction for decision trees. Clementine makes extensive use of visualization techniques which give the user agility in manipulating the data mining process, and for viewing results through a variety of graphical representations such as plots, points, histograms or distribution tables (horizontal bar charts) and webs of relationship. It can also generate models for prediction, forecasting, estimation and classification that can be exported as C language and used in other programs. It has a 'canvas' type interface similar to Enterprise Miner.

Although these modern data mining toolkits are quite comprehensive in their data exploration and modelling capabilities, none of the mainstream systems offer fuzzy data processing or representation, or genetic algorithm based processing. Specific commercial toolkits do exist for fuzzy data processing, such as MIT GmbH's DataEngine which allows design, definition and execution of fuzzy logic rules and membership functions. In the field of genetic algorithms, Ward Systems' GeneHunter allows definition and execution of problems (dataset, modifiable genes and parameters such as mutation rate and crossover type) with a spreadsheet interface.

In terms of data aggregation, there are no explicit aggregation operators. For attribute selection and significance ranking there are usually contrasting techniques available, such as principal component analysis, neural network sensitivity analysis, decision tree pruning, and different types of correlation and covariance. What may occur is that different techniques may give different results.

1.2.2 Relevance and reliability

Consider a set of cases C_1, for example the set of low-contaminant-emission vehicles. Each vehicle V_n in the set is defined by M attributes which describe it, for example, engine horse power, type of fuel (gasoline or diesel), date of registration, length, colour, and so on. Given that we have already determined the defining concept D_{GC} for the members of the set, that is, vehicles with low contaminant emission, we can say that some attributes of the vehicle will be more relevant than others to the defining group concept D_{GC}. For example, the attribute *'date of registration'* indicates the age of the vehicle, and we know that vehicles registered before a certain date did not have to comply with current exhaust emission regulations. Also, recent innovations in engine design and the chemical composition of the gasoline itself have resulted in lower contamination. Thus we can make an initial qualitative assumption that *'date of registration'* is relevant to low contaminant emission. On the other hand, the attribute '*colour'* has no influence whatsoever on whether a vehicle pollutes more or less. In complex datasets with many attributes, a key initial problem is the quantitative determination of relevance among attributes in relation to a given concept or 'output', and the ranking of all the attributes in order of their relevance. This can lead us to eliminate attributes whose relevance is below

a certain 'threshold', and therefore achieve a reduced minimal set of the most relevant attributes. In data analysis, this is our goal in the context of relevance.

The work of [Gonzalez97] presents two contrasting approaches to the problem of obtaining the set of most relevant attributes. The first approach is to eliminate the non-relevant variables from the complete set, and the second approach is to incrementally build a set of the most relevant ones. SLAVE (Structured Learning Algorithm in Vague Environment) has as objective to accelerate the learning process, running time twice as fast with the same number of rules. Criteria for the goodness of a rule are (i) the *degree* of soft consistency and (ii) the *degree* of completeness. The datasets used for testing were the Ionosphere, Soybean & Wine datasets. Rule selection is achieved by a 2 level Genetic Algorithm, at the variable level and the value level.

Two information levels are considered, the relevance level and the level of dependence between variables. A rule has the following structure:

Rule

IF *Precedent*
THEN *Antecedent* {represented by a chromosome}

In conclusion, this work uses information about the relevance of the predictive variables to improve the resulting models.

[Blum97] makes several definitions of relevance, depending on the context and goals in each case. 'R*elevance to the target'*, states that a feature $\mathbf{x_i}$ is relevant to a target *concept* **c** if there exists a pair of examples **A** and **B** in the instance space such that **A** and **B** differ only in their assignment to $\mathbf{x_i}$ and $c(A) \neq c(B)$. Thus, feature $\mathbf{x_i}$ is relevant if there exists some example in the instance space for which, as a consequence of modifying its value, the classification given by the target concept will be affected. Blum also cites other relevance definitions, such as 'strong relevance to the sample/distribution', 'weak relevance to the sample/distribution', 'relevance as a complexity measure' and 'incremental usefulness'. Depending on which definition of relevance is used, different features or groups of features may be identified as relevant, as is illustrated with a simple Xor type example. With respect to pre-processing to reduce features before the classification (induction) algorithm begins, they detail a 'filter' type approach, which seems less interactive than Kohavi's approach [Kohavi97], in that a filter module first executes to completion, followed by the induction algorithm. Two examples of filter algorithms are cited. The first filter algorithm is RELIEF [Kira92], which has been used by many medical data analysis applications, and which is also referenced in Section 1.4.6 of this thesis. RELIEF assigns a 'relevance' weight to each feature, which denotes the relevance of the feature to the target concept. It then samples instances randomly from the training set and updates the relevance values based on the difference between the selected instance and the two nearest instances of the same and opposite class. The second filter algorithm is FOCUS [Almuallim91], which exhaustively examines all subsets of features, selecting the minimal subset of features that is sufficient to determine the label value for all instances in the training set.

[Kohavi97] explores the relation between optimal feature subset selection and relevance. It also develops a 'wrapper' mechanism , or FSS-Feature Subset Selection, which is embedded in rule induction algorithms C4.5 and ID3, and the Naive-Bayes algorithm. It shows an improvement in classificative accuracy for datasets such as Corral, Monk1 and Monk2-local, from the University of California at Irving repository. In some of the cases where the precision was not improved, the existing precision was equalled, but using fewer features. The justification for this approach is that many of the principal induction algorithms degrade rapidly in predictive accuracy in the presence of many features which are unnecessary for predicting the desired output. The Naive Bayes algorithm degrades for slowly under the same circumstances, but degrades rapidly when correlated features are added – that is irrelevant features with a significant correlation to other relevant features, but with low correlation to the desired output.

Kohavi states that the wrapper approach is an improvement on simply using a filter such as FOCUS or RELIEF, because it avoids the main disadvantage of the filter approach which is that it ignores the effects of the selected feature subset on the performance of the induction algorithm. In the 'wrapper' approach, the feature subset is effectively optimised for use with the induction algorithm.

[Kohavi97] also gives several definitions from the literature for 'relevance', but which are only applicable to discrete features, although Kohavi states they can be extended to continuous features. Basically it is decided that two degrees of relevance are needed, 'weak' and 'strong', in order to guarantee unique results. This is demonstrated by an example using Xor. A feature X_i is strongly relevant *iff* there exists some x_i, y and s_i for which $p(X_i = x_i, S_i=s_i) > 0$ such that $p(Y=y|X_i=x_i,S_i=s_i)!=p(Y=y|S_i=s_i)$. A feature X_i is weakly relevant *iff* it is not strongly relevant and there exists a subset of features S'_i of S_i for which there exists some x_i, y, and s'i with $p(X_i =x_i,S'_i=s'_i) != p(Y=y|S'_i =s'_i)$

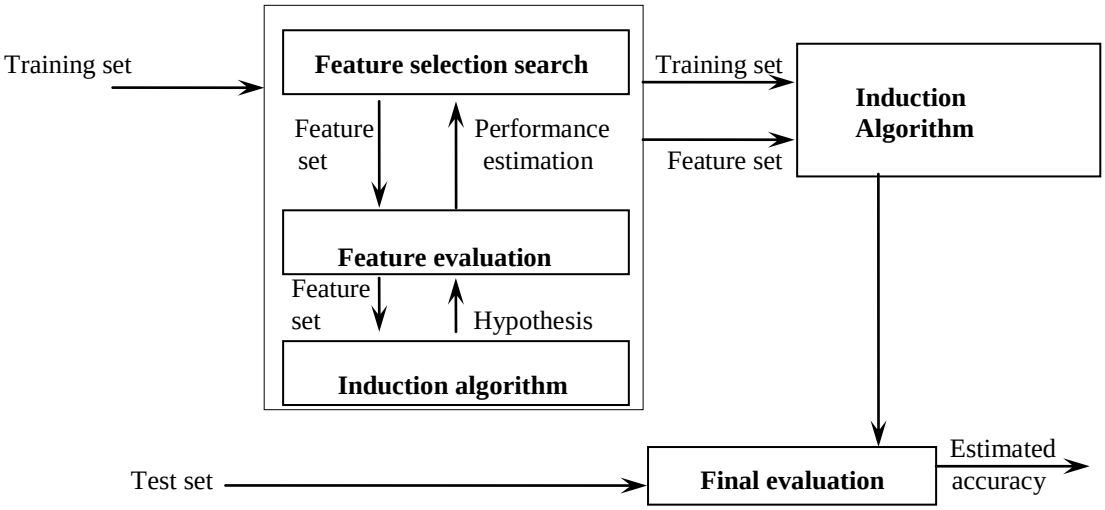


Figure 3. The wrapper approach to feature subset selection. The induction algorithm is used as a 'black box' by the subset selection algorithm

In Figure 3 we see a synthesis of the Wrapper approach [Kohavi97], in which the feature subset selection algorithm exists as a wrapper around the induction algorithm. The feature subset selection algorithm conducts a search for a good subset using the induction algorithm itself as a component of the function which evaluates the feature subsets. Hence the induction algorithm is considered as a black box, which is executed with the dataset, partitioned into internal training and test sets, for which different sets of features have been removed from the data. The feature subset with the highest evaluation is chosen as the final set on which to run the induction algorithm. The resulting classifier is then evaluated on an independent test set that was not used during the search.

Kohavi contrasts two algorithms as the engine for the feature selection search: (i) hill climbing and (ii) best first search. These are applied successively to the tests datasets, with the induction algorithms being ID3, C4.5 and Naive Bayes. The results in some cases show slight improvements in classificative accuracy, but the real improvement is the creation of a classificative model which uses significantly less input features, while demonstrating a similar predictive accuracy to the original algorithms.

Reliability

Reliability is an area which in the 1980's was an active field especially in relation to fault tolerance, and more specifically, fault tolerance of communications networks and computer CPU's and storage. One traditional, simple, but expensive solution was to replicate units, execute them in parallel, and poll the outputs for a majority vote as the outcome. Fault tolerance was also achieved by built in 'redundancy'.

In terms of the reliability of data values, this usually also means having multiple sources for the same value, for example a temperature reading, or a medical diagnosis, and polling an odd number for a majority vote as the correct output. For example: we have five temperature sensors {A,B,C,D,E} and three {A,C,E} say the temperature is between 10 and 12 Celsius, {B} says the temperature is between 10 and 15 Celsius, and {D} says the temperature is between 25 and 50 Celsius. The resulting output would be that the temperature is between 10 and 12 Celsius, by simple majority.

Note this method requires an odd number of sensors. Even though a majority of 60% of the sensors gave the same output, 40% gave different outputs. We could include this information as a further confidence level in the output.

Replicated systems and polling are of extreme importance in vital control systems such as those found in airliners, railway networks, nuclear power plants, and so on.

In the case of a numerical input variable, for example, temperature, in which we have just one sensor and one data value, we could assign a weight to the variable which indicates its reliability in general. That is, its tendency to give incorrect results, relative to some (absolute) measure. Another option is that a variable has a reliability weight for different bands of its distribution. For example, if we have the following set of temperature readings in degrees Celsius, {1,1,3,3,25}, the value 25 would be considered suspect, unlikely, or unreliable. Notwithstanding, the reliability of the values depends on the distribution in each case, thus the value 25 in the set {25, 25, 30, 30, 45} would be reliable. It follows that to each value, we could assign a reliability weight, with a value between 0 and 1, where 1 is totally reliable and 0 totally unreliable.

Later we will see how Yager and Torra have extended this idea for aggregation operators to include weights for both relevance and reliability of input variables.

1.2.3 Aggregation of variables and data

The Ordered Weighted Average (OWA) aggregation operator allows the incorporation of 'quantifiers' into an aggregation process of corresponding data cases, and was first detailed by Yager in [Yager88]. More specifically, Yager deals with the problem of aggregating multicriteria to form an overall decision function. One key property of the OWA operator is that can position its output between the "and", for which all the criteria must be satisfied, and the "or", for which at least one of the criteria have to be satisfied. This allows a closer approximation to human decision making, in which case often we require "most" or "many" or "at least half" or "more than four" of the criteria to be satisfied.

Yager's work [Yager88] considers the use of t-norms, t-conorms and the s-operator to effect a quantitative implementation of the "anding" and "oring". Although this implementation only allows the extremes "all" or "at least one", while OWA permits intermediate situations. Yager, in [Yager88], and referencing [Dubois80], understands t-norms as providing a way of quantitatively implementing the type of "anding" aggregation implied by the "all" requirement. T-conorms, a closely related set of operators, provide a way of implementing a type of "oring" operator.

The WOWA Operator: Torra in [Torra97a] described the Weighted OWA operator (WOWA), which combines advantages of the weighted mean and the OWA operator, thus solving some of the shortcomings of the latter two operators. It considers two weight vectors: ρ corresponding to the relevance of the sources (as in weighted mean), and ω corresponding to the relevance (which we interpret as 'reliability') of the values (as in OWA). One of the difficulties in using aggregation operators is the initial fixing of the associated parameters, for example the relevance weights ρ of each information source. In [Nettleton01b] different data analysis methods are contrasted for determining the weights of the aggregation function.

Choice of WOWA Operator: The WOWA operator has been chosen in order to aggregate data cases to produce a diagnosis for Apnea syndrome, as detailed later in Chapters 3 and 4. WOWA was chosen because it enabled us to include a quantification for both 'reliability' and 'relevance' into the aggregation. The operator is also adequate for processing data represented in the fuzzy form, including membership grades as part of the input. The WOWA operator has already been benchmarked, tested and compared against other operators and techniques, such as OWA, Choquet Integral, Sugeno Integral [Sugeno74], fuzzy t-Integral [Murofushi91]. One could say that the Choquet Integral or Sugeno Integral are more appropriate for processing data with grades of membership, but Torra has demonstrated in [Torra98c] that WOWA is equivalent to the Choquet Integral in given circumstances.

Hartigan's 'joining algorithm': Hartigan's *'Joining Algorithm'* [Hartigan75] performs successive fusions of attributes (variables) using as input a covariance matrix of the attributes. One consequence of fusion is the reduction of the initial attribute set to a space of dimension 2 or 3, which simplifies, for example, the visualisation of the data. The fusion algorithm serves two objectives: the first being the reduction of attributes through their progressive unification; the second is the identification of the most significant factors and the factors between which there is the greatest relation. *Outline*: in each step, the pair of attributes with the highest covariance is *fused* to form new attributes, until the number of desired attributes is obtained, or until the binary tree of groupings is complete. It is from this tree of fused attributes

that we can select distinct descriptions of the objects being analysed; descriptions with the most convenient dimension in each case.

Choice of Hartigan's 'joining algorithm': Hartigan's book *'Clustering Algorithms'* [Hartigan75] was a landmark in the clustering algorithm community and has been used since as a source book for benchmarking algorithms and from which a wide range of variants and enhancements have emerged. Apart from providing 'tried and tested' algorithms, the book also provides and (in general) clearly explains the Fortran source code listings, although my implementation was in Borland 'C'. Hartigan's approach is based firmly in the traditional statistical field, and his algorithms are clearly 'crisp' in nature. This provides a sounding board for generalising and adapting them to enable fuzzy data processing. Later work by Hartigan includes considerations of distribution in clustering [Hartigan77][Hartigan78], consistency [Hartigan81] and more theoretical aspects [Hartigan85a][Hartigan85b].

Other comparable authors in the field of *factorial and multivariate analysis*: three other references which explain applications to 'factorial analysis' are [Mardia79][Lebart85][Kaufman90]. [Kaufman90], is especially relevant, as its analysis methods are based on the fuzzy form, and has a wide range of 10 different algorithms for attribute fusion, which are distinct to those of Hartigan.

1.2.4 Fuzzy data representation

Different techniques exist for representing data in the fuzzy form. For example, the heterogeneous representation of Hathaway and Bezdek [Hathaway96] and the 'Parmenidean Pairs' of [Aguilar91]. [Aguilar91] presents a technique called 'Parmenidean Pairs', which automatically constructs an odd number of linguistic labels from two initial antagonistic linguistic concepts. This method automatically constructs a system of 5 linguistic labels which represent the ordered values of the variable, derived from what is called a parmenidean pair, which responds to the basic opposite values which the variable may assume. This method is very apt for variables such as 'days of stay in the hospital', for which we could define the fuzzy values VERY SHORT, SHORT, MEDIUM, LONG, VERY LONG derived from the basic opposites of SHORT, LONG. The complexity and usefulness of the technique lies in the automatic calculation of the geometric properties of the membership functions: gradient, centre of mass, overlap between each linguistic labels, length of the gradients, and the resulting grade of fuzziness which these properties define.

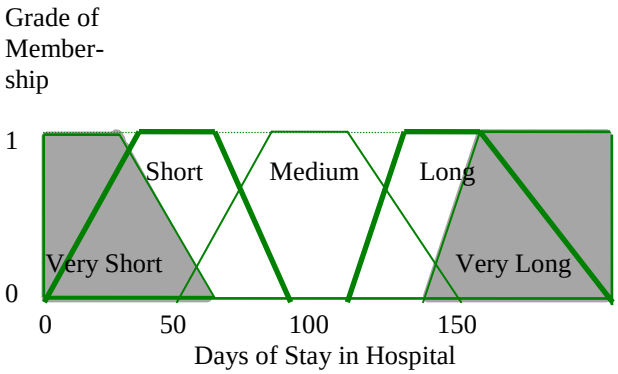


Figure 4. Representation of Lexical Variables with Trapezoidal Areas

Figure 4 shows a simple fuzzy representation for a typical questionnaire 'response' using a FLV (Fuzzy Linguistic Variable). From a semantic point of view, a FLV can be identified by 3 parameters: its relative *position* with respect to the other ones, its degree of *imprecision*, and its degree of *uncertainty*, these last can be merged into a single concept of *softness*, as opposed to *crispness*.

The trapezoidal and triangular forms can be considered as approximations of membership functions whose natural form is curved. The curved form is more complex to generate and is often represented by a parametric equation. The desired curved form has to be generated, or interpolated, from a finite number of points. In Figure 5 we see an example of a non-linear membership function, in which the five fuzzy sets defined by trapezoids in Figure 4 are now represented by smooth curves. Note that, in Figure 4 there is an area of overlap of three fuzzy sets; very short-short-medium, and medium-long-very long. This means that a point could have a non-zero membership grade to each of three possible fuzzy sets. In Figure 5, on the other hand, overlap only exists between two fuzzy sets in any one point. The ranges of the fuzzy sets over the x-axis also differs between Figures 4 and 5.

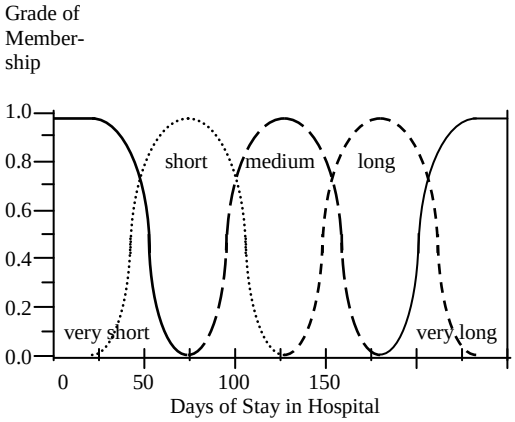


Figure 5. Example of non linear membership functions

Associated work with respect to the representation of membership functions and linguistic labels

Some recent references are as follows: in the case of trapezoidal and triangular fuzzy sets, such as [Roychowdhury97]; for complex membership functions, [Boixader97] treats of membership functions which may have irregular forms; and [Torra99c] considers the generation of membership functions from a set of observations.

1.2.5 Fuzzy data analysis

Fuzzy data analysis is covered extensively in Sections 2.2 and 3.1. This section provides a brief introduction with reference to some of the key algorithms.

Fuzzy c-means clustering: fuzzy clustering algorithms are mathematical tools for detecting similarities between members of a collection of objects. One of the most widely known algorithms is the fuzzy isostar or fuzzy c-Means algorithm developed by Dunn [Dunn74] and extended by Bezdek [Bezdek73]. The fuzzy c-Means (FCM) clustering algorithm is a set-partitioning method based on Picard iteration through necessary conditions for optimising a weighted sum of squared errors objective function (J_m). The number m is a parameter ranking from 1 to ∞; J_1 is the classical WGSS objective function which serves to define the hard (or crisp) c-means (HCM) and hard ISODATA algorithms [Duda73]. Dunn first extended J_1 to J_2 in [Dunn74], and Bezdek then generalised J_2 to J_m for $1 < m < \infty$ in [Bezdek73]. Much of the work carried out on theoretical issues related to its mathematical structure is summarised in [Bezdek81]. Later work such as [Bezdek87] and [Pal97] have introduced the *c-varieties* and *c-medians* algorithms, respectively, which do not require an *a priori* assignment of the parameter 'c' (the number of partitions), and which allow for mixed data types for input attributes.

Fuzzy covariance matrix: Gustafson and Kessel [Gustafson79] were the first to use the term 'fuzzy covariance matrix', and they generalised the fuzzy c-Means algorithm to include it, their motivation being the obtention of a more accurate clustering. The calculation itself was limited to the covariance of a fuzzy cluster with respect to its fuzzy prototype. More recent works, such as that of [Watada94][Wangh95][Nakamori97] have created specialised fuzzy covariance calculations tailored and tuned for specific applications.

Fuzzy clustering with weighting of data variables: a recent work by [Keller00] considers fuzzy clustering with weighting of data variables, in which an objective function-based fuzzy clustering technique assigns one influence parameter to each single data variable for each cluster. The distance measure determines de influence of given data attributes for each cluster, and therefore allows attributes to be identified which determine the class represented by the cluster. The influence parameter can be used to reduce the influence of one attribute on only some clusters without ignoring that attribute for the whole classification. The resulting information can be used to partition a dataset into smaller data parts with a reduced number of attributes, which can then be subject to further analysis.

Fuzzy data modelling: data modelling has as its objective the creation of a model with N inputs and M outputs, which is able to simulate the behaviour of the outputs with respect to the inputs. A typical statistical model is a regression model, which finds a best 'fit' of the outputs to the inputs. Clustering and classification are both modelling techniques,

as we will see in later sections of the thesis. If we suspect that in the nature of the data there is a 'fuzzy' component, then we can consider techniques which allow the manipulation of this type of information. In the Sugeno-Takagi fuzzy model [Takagi85], Guassians are used with the Mahalonobis distance to 'fine tune' the function. The objectives are to improve the optimisation with one of the ways being to initialise the parameters with a 'good guess' or a 'better guess'. The model is made to grow incrementally, using one, two or three initial rules and then go on adding.

Fuzzy neural modelling: neural network models attempt to simulate the functionality of the biological brain by defining an interconnected network of 'neurons' to process data inputs and produce corresponding outputs. A simple neural network model consists of an input 'layer' of neurons, a 'hidden' intermediate layer and an 'output' layer. Weights are defined for the interconnections between neurons. The weights are augmented or diminished by the stimulus of the inputs and the propagation of the data through the different layers. Thus by successive presentations of inputs the network begins to model the data and produce the most adequate outputs in each case. Fuzzy techniques can be included in different ways in a neural model, the first being in the representation of the data, including the grades of membership as input, for example. Alternatively, the internal working of the model itself may be modified to process in a fuzzy manner, for example in the internal assignment of the weights or the propagation mechanism.

Fuzzy rule induction: rule induction is a technique whose goal is to create a set of rules from a dataset. The rule induction algorithm has no additional information apart from the data itself. The quality of the rules is a key aspect, combining precision, that is a given rule correctly classifies a high percentage of the corresponding cases, with significance. By precision, we mean that a given rule correctly classifies a high percentage of the cases which correspond to it, and by significance we mean that a significant number of cases as a proportion of the total number of cases, correspond to the given rule. Fuzzy techniques can be included in different ways to rule induction, the first being in the representation of the data, including the grades of membership as input, for example. Alternatively, the internal working of the induction process itself may be modified to process in a fuzzy manner, for example in the definition of the decisions made at each node in the tree or in the pruning and compaction phases.

A Fuzzy Projection Pursuit, called ID3* has been developed by [Miyoshi97] which references other fuzzy versions of the ID3 rule induction algorithm and more recent work by Quinlan. In his work, Miyoshi unifies the Fuzzy ID3 approach of [Umano94] and the Projection Persuit approach of [Friedman74]. [Wangc96] introduces 'FILSMR', a fuzzy inductive learning strategy for modular rules. This method chooses the best 'attribute-value' while ID3 chooses the 'best-attribute'. This indicates its greater level of 'granularity'. A 'class membership value' is considered equivalent to a 'soft instance'. The algorithm used by Wang finds relevant attribute-relation pairs, maximising the 'fuzzy information gain'. The heuristic of minimising 'entropy' is used to determine which attribute should be next selected in the decision tree, looking for good rules with truth level above a given threshold.

Fuzzy factorial analysis: factorial analysis is defined as the analysis of an initial set of input attributes, in order to identify relationships and a reduced number of factors in terms of the original values, which best represent the data. This is different to defining the most relevant attributes as seen in Section 1.2.2, because the objective in factorial analysis is create new factors in terms of the original attributes, then eliminating those original attributes. Fuzzy factorial analysis may be considered as factorial analysis, extended to treat data in a fuzzy form, or it may imply that the factorial analysis algorithm itself processes in some way in a fuzzy form.

Factor analysis for fuzzy data is also a theme tackled in [Nakamori97]. Traditional data analysis methods are initially cited, such as those of Spearman, and methods using Eigenvectors. The classification of adjectives by factor analysis is considered. Nakamori cites that one reason that factor analysis for fuzzy data has not been developed more, is the difficulty to calculate the second moment of fuzzy data given by interval fuzzy numbers. He defines a fuzzy correlation matrix and a proposal for fuzzy factor analysis based on this: (i) correlation matrix of averaged data $R = (r_{ij})$; (ii) correlation matrix $R^k = (r_{ij}^k)$ of subject k; (iii) variance of correlation $\{r_{ij}^k\} \rightarrow \sigma_{ij}^2$; (iv) fuzzy correlation matrix $R = ([r_{ij}^L , r_{ij}^R])$.

1.2.6 Clustering

Clustering may be defined as the process of dividing a data set into mutually exclusive groups such that the members of each group are as 'close' as possible to one another, and different groups are as 'far' as possible from one another, where distance is measured with respect to all available variables. We consider clustering given that it is one of the fundamental aspects of Data Mining and may be applied both in the data *exploration* phase as well as in the data *modelling* phase. [Hartigan75] defines clustering as the *grouping of similar objects*, whereas, classifying is *naming*, such as in the taxonomy of animals and plants of Aristotle, and Linnaeus (1753). Each species belongs to a series of clusters of increasing size with a decreasing number of common characteristics. For example, man belongs to the primates, the mammals, the vertebrates, the animals.

The principal classification problem in medicine is the classification of disease. The World Health Organisation produces a Manual of the International Statistical Classification of Diseases, Injuries and Causes of Death (1965). This provides a standard nomenclature which allows the compilation of health statistics, comparable across different countries and time intervals. A particular type of classification within a disease is the identification of stages of severity – for example, for renal disease (1971). Various symptoms are grouped by expert judgement to make up ordered classes of severity in three categories. Goldwyn et al (1971) use clustering techniques to stage critically ill patients.

For diseases that are caused by viruses and bacteria, the techniques of numerical taxonomy are employed, and there are many papers in the literature with respect to these techniques. For example, Goodfellow (1971) measures 241 characteristics of 281 bacteria, some being biochemical, some physiological and others nutritional. He identifies seven groups substantially conforming to previously known groups. However, the classifications of viruses of Wilner (1964) and Wildy (1971) and the classification of bacteria of Prevot (1966) are still based on picking important variables by expert judgement.

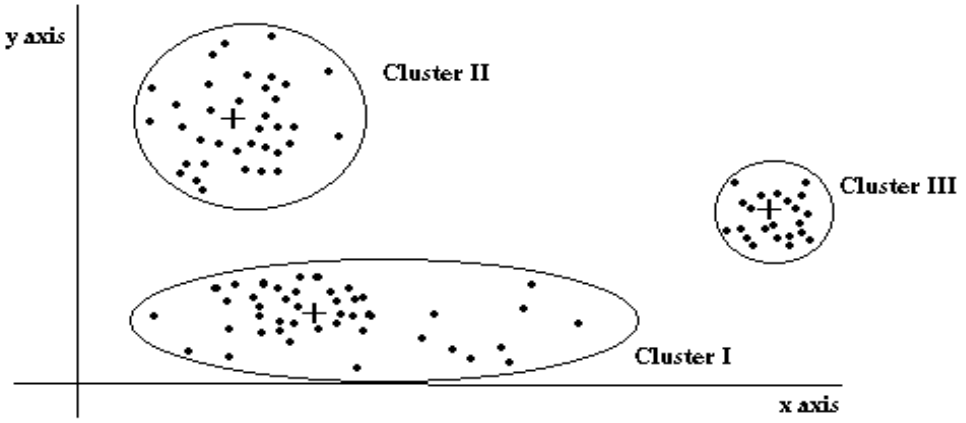


Figure 6. The objective of cluster analysis is to maximise the inter-cluster distance while minimising the intra-cluster distance

Clustering is a technique which is generally considered as unsupervised, which does not benefit from a priori structure into which the cases may be inserted. It is the job of the data analyst to later give meaning to the groupings which have been generated. For example, in Figure 6, Cluster I may correspond to young healthy patients, Cluster II to middle age overweight patients and Cluster III to middle age non-overweight patients. Notwithstanding, the clustering algorithm would have no previous information that the cases should be grouped by age and weight categories. Figure 6 shows three clusters with cluster centres indicated by a cross: in Cluster III we see the highest compactness and smallest average intra-cluster distance, whereas Cluster I demonstrates the least compactness and greatest intra-cluster density. In this sense, we could say that Cluster III has the highest 'quality rating' in terms of similarity of the cases assigned to it, and Cluster I has the lowest 'quality rating' for the same reasons. In terms of similarity of clusters, we observe that the two clusters with the minimum distance between cluster centres are Clusters I and II, whereas Clusters II and III are the most distant.

Extended details of the clustering techniques relevant to the thesis are given in Sections 2.4 and 2.5.

1.2.7 Classification

Classification may be defined as the process of dividing a data set into mutually exclusive groups such that the members of each group are as 'close' as possible to one another, and different groups are as 'far' as possible from one another, where distance is measured with respect to specific variable(s) which are trying to be predicted. For example, a typical classification problem would be to divide a database of patients with respect to a 'state of health' variable with values 'good' and 'bad'. Classification, as well as clustering, is a fundamental part of any Data Mining process, although, in contrast to clustering, its application is limited to the data *modelling* phase.

Statistics has given rise to a large number of classification methods, which are summarised in [Hunt75]. CART [Breiman84] is a well known system for building decision trees which was developed by statisticians, being based on the previous work by Friedman [Friedman77], also related to Quinlan's ID3 [Quinlan83].

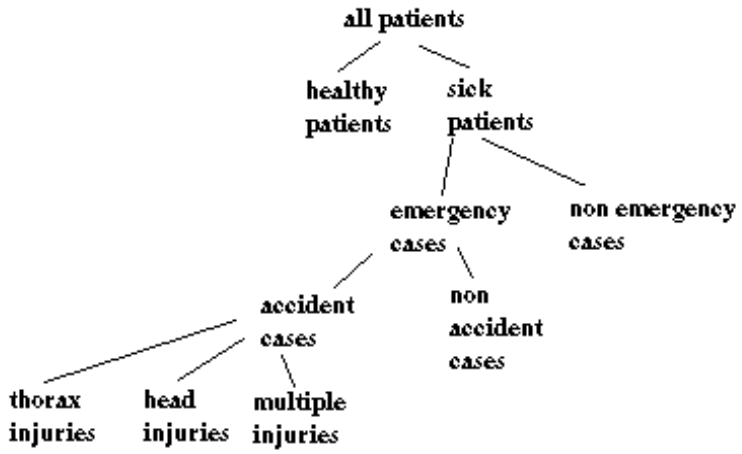


Figure 7. Example of a simple classification tree for hospital admissions

Classification distinguishes itself from *clustering* given that in the former a previous classification structure is defined, and the objective is that of successfully placing each case in the class which it best belongs, with respect to its characteristics. Classification is generally a supervised process, which may be trained, for instance, in positive and negative examples. In the above classification tree, the data would consist of different patient types, covering all classes defined in the structure: thorax injuries, head injuries, non-accident cases (e.g. heart attack) and non-emergency cases (e.g. broken arm).

Extended details of the classification techniques relevant to the thesis are given in Section 2.6.

1.2.8 Medical Diagnosis and ICU Prognosis

We distinguish *diagnosis* as the problem of establishing what category of illness or illnesses the patient has, while *prognosis* deals with the recovery prospects for a patient whose diagnosis has been previously established. Depending on the diagnosis a given treatment is prescribed, and depending on the prognosis, this treatment may be modified or adapted and a series of recovery phases planned, with the assignment of the human and clinical resources necessary for each phase.

Classical Statistical Approach

The literature of statistical treatment of medical diagnosis and prognosis is very extensive. One of the key books which provides a survey of the work in the area is [Lee80], dealing with statistical methods for survival data analysis, with a comprehensive overview of survival distributions, identification of risk factors and prognostic factors, and execution of clinical trials. 'Survival time' is defined as the time to the occurrence of a given event, such as the development of a disease, injury, response to a treatment, relapse or death. 'Survival data' is defined as including variables such as survival time, response to a given treatment, and patient characteristics related to response, survival and the development of a disease or injury. If there are no censored observations, that is, those with missing data, the survivorship function is estimated as the proportion of patients surviving longer than t:

$$\hat{S}(t) = \frac{\text{number of patients surviving longer than t}}{\text{total number of patients}} \qquad (1.1)$$

where the circumflex denotes an *estimate* of the function

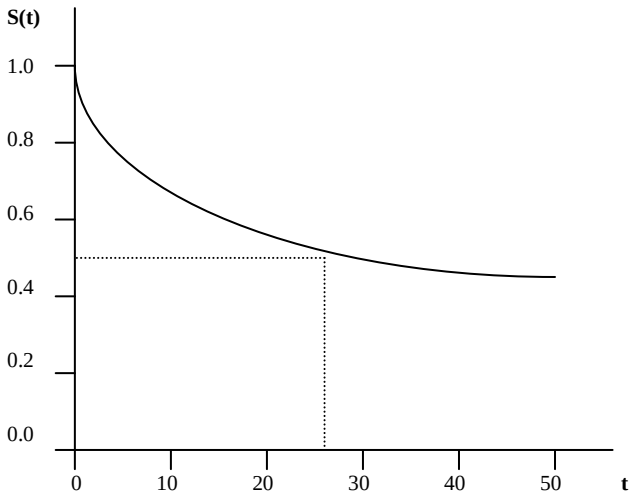


Figure 8. Example of a survival curve

Prognostic Scoring Systems in Intensive Care

Prognostic scoring systems are systems which predict patient outcome based on physiologic parameters considered to be correlated with outcome (based on statistical analysis or expert opinion). These scoring systems have been historically developed to predict outcome for populations of patients as opposed to individual patients. Prognostic scoring systems essentially allow physicians to compare observed outcome, such as mortality, with a predicted mortality for the population of patients admitted to their intensive care units (ICU).

Frequently Employed Prognostic Scoring Systems

The major scoring systems in the United States are the Acute Physiology and Chronic Health System (APACHE), the Mortality Prediction Model (MPM), and the Simplified Acute Physiology Score (SAPS).

The APACHE system first developed in 1981 by William Knaus and associates [Knaus81] is now in its third generation (APACHE III), although APACHE II is still the most widely employed system due to the high cost of APACHE III. APACHE II and III are based on four major components: diagnosis (including surgical and medical categories), physiologic derangement, chronic health, and age. Also included in the data base is patient origin. The APACHE system computes data for each ICU patient based on the first 24 hours of ICU hospitalisation, with the worst value in the 24-hour period for each variable inserted into the predictive formula.

APACHE II was developed in 1985. It was initially evaluated in a study of 5,815 patients. It is to date the most extensively used severity of disease program and as such, is the most validated of all scoring systems. Data must be hand entered into a computer by a trained technician, and from that an APACHE score is computed for each patient. The higher the score the greater the severity of disease. This data is also converted into a predicted probability of death for each patient. This data, combined with other information, is then used to compute a predicted mortality for the population of patients entered into the program. The mortality prediction equation is the following:

$$LN(R/1 - R) = -3.517 + \{(\text{APACHE II})\ (0.146) + S + D\} \qquad (1.2)$$

where

R = Risk of hospital death

S = Additional risk imposed by emergency surgery

D = Risk (+ or -) imposed by specific disease

Individual survival outcomes for each patient are entered into the program. This predicted mortality for the entire population is then compared with the observed mortality for the entire population.

APACHE III was developed in 1991. It is the first scoring system to allow for fully automated entry of data. It follows the same pattern as APACHE II. It was initially tested on 17,440 patients. Factors such as the diagnosis and patient origin play a more important role in the predictive formula in this model. An APACHE III score is tabulated based on the worst values for each parameter over the first 24-hour period. There is a decreased emphasis on the global APACHE III score, with greater emphasis placed on the APACHE III score within each diagnostic group.

Conversion of the APACHE III score to a probability of hospital mortality is achieved through logistic regression equations which are individualised for each of 79 diagnostic categories and for each of nine patient origins. Although the predictive equations have not been published by the authors of APACHE III, the authors have stated that the equations can be acquired through direct correspondence with them.

MPM was first developed in 1987 by Terres and associates. The model differs from APACHE in that it does not produce a score, but a direct probability of mortality. It is now in its second generation. This system is based on 19,000 patients from 139 ICUs. This model offers two capabilities which distinctly define itself from the APACHE model. First, it provides a probability of hospital mortality at admission. This allows a prediction to be made before any intervening ICU care can be given, which over the ensuing 24 hours could modify prognosis. In addition, those patients who die or are transferred out of the ICU before 24 hours have elapsed receive a mortality prediction at admission. Secondly, a specific probability model designed for use at 24 hours is available. This allows for a revaluation of prognosis at 24 hours based on a model designed for a 24-hour evaluation. The APACHE model can be used on a daily basis, but a specific 24-hour model does not exist.

SAPS was developed by LeGall and associates in 1984 and is now in its second generation [LeGall93]. It was developed to offer a simplified version of the original APACHE model which in turns facilitates data collection. The model relies on 13 physiology variables plus age. SAPS II employs statistical methodology to determine the range for predictor variables, to assign points to each of these ranges, and to convert a SAPS score to a probability of hospital

mortality. The SAPS score is converted to a statistical probability using a plot. SAPS uses the worst variables during the first 24 hours and does not require a diagnosis to obtain the probability of hospital mortality.

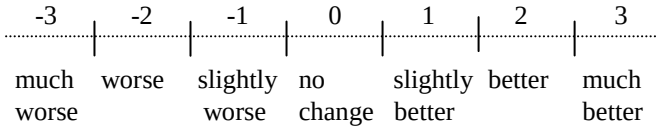
Artificial Intelligence Approaches

The SMASH project at the IIIA Institute, Catalunya, Spain [IIIA96], applies 'intelligent agent' technology to the medical environment. The objective of SMASH (Systems of Multiagents for Medical Services in Hospitals) is the definition of a rational multi-agent architecture, and the development of prototype multi-agent systems with learning capabilities that cooperate in the solution of complex problems in medical environments. The system comprises four aspects: (i) complex rational behaviour, divided into ontological, epistemic, motivational and communicational; (ii) transcription into an object-oriented environment using logic-based tools; (iii) deployment of rational general-purpose accountable software agents, that can be tailored to cooperatively solve different tasks; (iv) application to the medical and hospital-management environments for "proof of concept".

[Armengol00] is a second system developed at IIIA, which applies a CBR (Cased Based Reasoning) approach to the individual prognosis of diabetes long-term risks. The system, called DIRAS, is an application which gives support to physicians to determine the risk of complications for individual diabetic patients. The risk pattern of each diabetic patient is obtained using a Case-based Reasoning method. Case-based Reasoning is defined as technique which uses past experiences (cases) to solve new situations. For each patient, the method determines the risk of each diabetic complication according to the risk of already diagnosed patients. A description is then built which can be viewed as an explanation of the obtained risk.

[Escalada99] presents another work carried out at IIIA, which consists of a knowledge based system for real time physiopathological diagnosis in a critical care environment. In a paediatric intensive care unit (ICU), patients are monitored continuously, and many variables are gathered which indicate their physical. According to the situation in any given moment, control signals are sent to the patient so that his/her medical situation is always kept under control. The solution involves a knowledge based system appropriate for a real-time environment, and consists of specific modules which mutually interact.

[Irani95] presents a work whose objective is to elicit structure and causal relationships from a medical records database of hyperlipidemias. Some interesting concepts are used, such as a linear scale to transpose the linguistic labels onto a numeric sequence. That is:



The average agreement of the output causal relationship was calculated between the Expert Cardiologist, a Regression Equation and an Expert System. The predictive model was based on a back-propagation neural network, which, in this case gave worse results than a multiple linear regression model.

In contrast to the elicitation of causal relationships of [Irani95], the work of [McLeish95] studies information discovery, using such techniques as 'weight of evidence and belief functions'. [McLeish95] typically uses a subset of 18 key attributes as a starting point for input to the belief function. One of the conclusions of the study was that in this case, the statistically derived data outperformed the expert derived data.

One of the more interesting recent works is that of [Dreiseitl99], which presents a set of variable selection methods for diagnosis of Myocardial Infarction. This is carried out in two steps: (i) determine which inputs are deemed relevant for predicting myocardial infarction by various methods; (ii) validate and visualise these inputs using self-organising maps (SOM). The input selection methods used are: (a) logistic regression with stepwise, forward and backward variable selection, as implemented in the SAS LOGISTIC procedure; (b) feedforward neural nets with input relevance determination; (c) Bayesian neural nets with automatic relevance determination; (d) rough sets. The initial input variable set consists of 43 attributes (age, gender, smoker, ex-smoker, family history, diabetes, hypertension, ….) and from these each method must select the 8 most relevant attributes for diagnosis of Myocardial Infarction. The result was a quite good consensus between different methods, although three attributes chosen by the medical expert were not selected by any of the methods. The SOM was used to plot the distribution density of the output variable, and validate correlations.

[Demsar99] use ID3 classification trees and a Naive Bayes classifier to predict survival or not of each of 68 patients in the dataset. The initial number of features was 174, and for 78 features, data was missing for 50% or more of the cases. The data was pre-processed to categorise (discretise) the continuous features, given that the Naive Bayes technique cannot directly handle continuous features. Categorisation was carried out by using quartiles and entropy-MDL based discretisation. After categorisation, features were ranked using a system called RELIEFF [Kira92], which measures the usefulness of a feature by observing the relation between its value and the patient's outcome. After feature selected, the number of features was reduced from 174 to 12. The features were all specific clinical indicators, such as "the worst partial active thromboplastin" with categories < 78.7 and ≥ 78.7. The reasoning was that if there is a group of patients with the same of similar feature values, the observed feature is "valuable" as a predictor if it has different values on pairs of patients with different outcomes (thus distinguishing between them), but the same value on pairs with the same outcome.

Assuming the independence of predictive variables, the probability that a patient described with values of predictor variables V = (v_1, … , v_n) survives can be estimated by the naive Bayesian formula:

$$P(R / V) = P(R) \prod_{i=1}^{n} \frac{P(R|v_i)}{P(R)} \qquad (1.3)$$

where P(R) is the *a priori* probability of survival and P(R|v_i) is the conditional probability of survival if the *i*-th predictor variable has the value v_i; both are estimated from the training set of patients.

[Pessi95] is an application of SOM nets to patient grouping. Patient grouping is normally based on 'period of stay', 'intensive care needs', 'different service needs', and so on. A correct classification is important to enable a good planning of resource allocation, and cost optimisation. Diagnosis codes were used as part of the input to a SOM in order to group 8000 patient cases. The diagnostic codes are complex, having a binary tree structure down to 4 levels. The best accuracy was a 75% correct classification, which compares favourably with conventional assignment methods.

[Khang99] on the other hand, applies the 'hedge algebras', originally defined in [Zadeh73] (see Section 2.2) to extend a MYCIN type rule based medical diagnosis system. For example, the linguistic variable FEVER = {*high_feverish, low_feverish, ,medium_feverish, continuous_feverish, fitful_feverish, fever_in_afternoon, fever_at_night, fever_with_sweats, ...*}. This is analysed in terms of a 'symmetrical extended hedge algebra' which is developed in the paper: FEVER_DEGREE: high_feverish, medium_feverish, … . FEVER_TYPE: continuous_feverish, periodic_feverish, … . FEVER_TIME: morning, afternoon, night_feverish, … . FEVER_DAYS: one_day, two_days, some_days_feverish, … . WITH_HEADACHE: with_headache, without_headache, … . WITH_COLD_TREMBLE: with_, without_coldtremble, … . WITH_SWEATS: with_sweats, without_sweats, … .

From the previous definition, [Khang99] can develop rules of the following type, based on the 'aggregate hedge algebra' for the objective label of FEVER_OF_HEPATITIS:

Rule 1: If FEVER_DEGREE="low_feverish"
And FEVER_TYPE="not_fitful_feverish"
And FEVER_DAYS="7-10 days"
And WITH_HEADACHE="with_headache"
Then FEVER_OF_HEPATITIS="very_specific"

Rule 2: If FEVER_DEGREE="medium_feverish"
And FEVER_TYPE="not_fitful_feverish"
And FEVER_DAYS="7-10 days"
And WITH_HEADACHE="with_headache"
Then FEVER_OF_HEPATITIS="little_specific"

Rule 3: If FEVER_DEGREE="high_feverish"
And FEVER_TYPE="fitful_feverish"
Then FEVER_OF_HEPATITIS="very_unspecific"

UCI Machine Learning Group Data Base Repository

This is a key source for test data sets for investigators which wish to test new algorithms and techniques on standard datasets, for which there exists previous benchmarks. This makes it easier for investigators working in the field to cross compare their results with other techniques. In the medical domain, the following four datasets can be highlighted: (i) Echocardiogram Database from the Reed Institute, Miami. This has a reasonable level of documentation, consisting of 13 attributes with numerical values, and a binary classification: patient alive or dead after a given 'survival' period. (ii) ICU Data from Serdar Uckun (AIM '94), which is a data set of treatment of patients in the ICU who have 'Adult respiratory distress syndrome (ARDS). It is one of the more complex datasets. (iii) Post operative patient data base from Jserzy W. Grzymala-Busse, which consists of 3 classes, 90 instances and 8 attributes, one of which is numeric with missing values.
(iv) Coronary disease data base which comes with extensive documentation. It consists of 4 data bases: Cleveland, Hungary, Switzerland and VA Long Beach. 13 of the 75 attributes are used for prediction in two separate test, each of which achieved a classification precision of 75-80%. All 13 chosen attributes are continuous values, and it includes clinical cost data which is useful for studies whose objectives are the minimisation of operating costs.

1.2.9 Diagnosis of the Sleep Apnea Syndrome

The Sleep Apnea Syndrome

The Sleep Apnea Syndrome is a frequent problem, which to a greater or lesser extent affects between 2 and 4% of the adult population in the developed countries[Duran96][Olson95]. It is characterized by complete (apnea) or partial (hypopnea) interruption of respiration during sleep. The presence of this syndrome has been associated with excessive somnolence, with consequences such as traffic accidents and the reduction in quality of life and professional development[Lavie84]. It has also been linked to cardiovascular illnesses, there being a greater prevalence of hypertension, cardiac arrhythmia's, cardiopathic isquemica and cerebral-vascular accidents (stroke) in these patients.

The Obstructive Sleep Apnea Syndrome (OSAS) is a set of secondary clinical manifestations relating to the ceasing (apnea) or reduction (hypopnea) of air flow during sleep, caused by a partial or total collapse of the upper air way at the faring level. The severity of the OSAS is defined by the *apnea hypopnea index* , or AHI, (also known as RDI, *Respiratory Disorder Index*) which is the number of apneas plus the number of hypopneas per hour during sleep. Generally an AHI $\exists$10-15 is considered pathological. Patients with low AHI's, that is, less than 5 apneas, do not tend to have clinical consequences. *Light cases*, between 5 and 20, have slight consequences while *moderate cases*, between 20 and 40 usually show clinical manifestations. *Severe cases*, with an index above 40, show the most evident symptoms and present an increase in illnesses and death[Lugaresi83][Partinen88].

Clinical presentation

There are diverse symptoms associated with OSAS. They often become introduced insidiously during a certain period of time and are often overlooked in clinics and even by the patients themselves, due to their lack of specificity. The snore is one of the principal symptoms. The long snoring history which refer to patients with OSAS reflects the increase of resistance of the upper air tract during sleep. The presence of respiratory pauses witnessed by the room partner is another important data referenced in the literature, and tends to be a good symptom predictor.

Other clinical manifestations of OSAS seem to be due to the de-structuring of sleep, by the multiple transitory micro-awakenings, the loss of deep sleep levels, and to recurrent episodes of arterial hypoxemia. Among these symptoms we can highlight daytime hypersomnolence, alterations of personality, loss of memory and of concentration, which can gravely alter the daily life of these people.

Prevalence

The prevalence of OSAS oscillates between 1-9% according to studies. This difference in the percentages obtained reflects the diversity of methods and criteria used to diagnose OSAS and the possible differences in the populations that have been studied. The study of reference is that realised in the population of Wisconsin is [Young94], where the prevalence obtained reached 2% for females and 4% for males, showing minimum symptoms. When we extrapolate these results to the general population, 9% of women and 24% of men would present sleep related respiratory

alterations. This elevated prevalence in adults is considered to be a significant problem for public health. The studies realised in mixed sex populations are limited, but it is estimated that the proportion of men/women is 3/1.

Morbidity and mortality

Daytime hypersomnolence has been related to a reduction of physical and mental effectiveness, in the daily activity of the individual, including the work environment, and the ability to drive automobiles (drive worse and have greater risk of suffering traffic accidents). As well as daytime hypersomnolence, a certain relation has been identified between OSAS and systemic arterial hypertension. The patient with OSAS tends to present an elevated sympatic activity, which can cause an increase in the daytime blood pressure.

Some studies with patients suffering from high blood pressure, indicate that a third of them suffer from OSAS. Other studies indicate that snoring and OSAS increase the risk of suffering encefalovascular and cardiovascular accidents. [Guilleminault92] has found that the cardiovascular morbidity and mortality are lower, in statistically significant level, in treated patients compared to the patients in the control group, independently of age, body mass index (BMI) and previous severity index. It is also known that OSAS can contribute to the development of respiratory insufficiency, pulmonary hypertension and failure of the right ventricular. The presence of chronic limitation of airflow, daytime hypoxemia, hypercapnia and profound nocturnal hypoxemia are factors related to this fact.

The causes of mortality are variable and include cardiovascular complications derived from systemic arterial and pulmonary hypertension, episodes of arterial hypoxemia and those derived from excessive daytime hypersomnolence, such as accidents in the workplace and traffic accidents.

Diagnosis

The diagnosis of Sleep Apnea Syndrome, and the categorization of its seriousness (light, moderate and severe) is achieved by the evaluation of a combination of clinical manifestations and data derived from a polysomnogram. The polysomnogram consists of a continuous recording, during night-time, of numerous physiological variables, including electroencephalogram, electrooculogram, electromyogram, leg movement, oral-nasal airflow, snoring, thoracic and abdominal respiratory effort, electrocardiogram, body position and haemoglobin oxygen saturation. Other biological signs can also be used if considered necessary.

Due to the high cost of this type of clinical study, and the shortage of adequate centers, a series of more limited tests have been devised, which can be used for 'screening' in diagnosis. In general, the tests consist of a reduced number of variables (for example, only the pulsioximetry), which allow non-supervised studies to be made in the patients own home[Martin85].

One of the most interesting tools available for diagnosis, due to its simplicity and low cost, are self-administered or supervised questionnaires. Having identified a set of variables with high predictive value for sleep apnea syndrome, diverse questionnaires have been developed, with combinations of questions and clinical variables. Unfortunately, this method has not found great acceptance in clinical use, due to its low predictive accuracy and the numerous false negative and positive diagnosis that it produces[Kushida97].

Table 1. Multiple linear regression models for diagnosing sleep apnea

Study	n	Diagnostic criterion	Predictive variables	r^2
Stradling (1991)	1001	ID4%>5	Neck circumference, alcohol consumption, age, obesity	0.14
Davies (1992)	150	ID4%	Sleep when inactive Neck circumference	0.13 0.35
Hoffstein (1993)	594	AHI>10	BMI,age, sex, snoring, exploration of ORL	0.36
Flemons (1994)	180	AHI>10	Neck circumference, HTA, snoring, observed apneas	0.34
Deegan (1994)	250	AHI∃15	BMI, age, alcohol consumption	0.19

ID4%: index of desaturation with fall of 4%. AHI: apnea-hypopnea index. r^2: regression coefficient. BMI: body mass index. ORL: otorrinolaringologic exploration. HTA: arterial hypertension.

The predictive value of the clinical data in OSAS diagnosis is low. Hoffstein [Hoffstein93] published results that indicated that clinical data explains 36% of the variability of the AHI (apnea hypopnea) and Katz [Katz90] reported a figure of 39%, other authors report lower figures (Table 1). The subjective clinical evaluation of the interviewer has also been evaluated and tends to have a low sensibility and specificity, in the order of 55%-65% respectively, for correctly classifying the sick. On the other hand, The predictive models for AHI based in clinical data have a higher sensibility of up to 90%. Their specificity, in the best of cases, does not reach 70% (Table 2).

Table 2: Logistic regression models

Study	n	diagnostic criterion	Predictive variables	S (%)	E (%)	ROC
Crocker (1990)	214	AHI>15	age, observed apneas, BMI, HTA	85	61	-
Viner (1991)	410	AHI >10	age, BMI, sex, snoring	94	28	0.77
Rauscher (1993)	300	AHÍ∃10	Sex, %ideal weight, sleep while reading, observed apneas	94*	45*	-
Kump (1994)	456	IAA>5(<15yrs) IAA>10(15-50y) IAA>15(>50 y)	snoring, observed apneas, sleep while driving +(BMI, sex, age)**	-	-	0.78 0.87
Dealberto (1994)	129	AHI∃10	sex, age, BMI, snoring, observed apneas	95	64	-
Flemons (1994)	180	AHÍ >10	circumference of thorax, change of weight, observed apneas, HTA	(†)	(†)	(†)
Maislin (1995)	427	AHÍ∃10	index 1(‡), BMI, IMC, age, sex	-	-	0.78
Deegan (1996)	250	AHÍ∃15	sex, age, snoring, observed apneas, BMI, alcohol consumption, sleep while driving	100	11	-

S: sensibility. E: specificity. ROC: area under the curve. AHI:apnea-hipopnea index. IAA: index of increase in apneic activity. BMI: body mass index. HTA: arterial hypertension. (*) data obtained after model verification. (**): model which includes the previous symptoms and those in parenthesis. (†): refer to data similar to that of Viner and Crocker. (‡): includes intense snoring, observed apneas and respiratory insufficiency. (-): data not available.

The reference method for OSAS diagnosis is the polysomnogram. It consists of the simultaneous recording of a number of sleep parameters, which allow us to identify its different phases and the correlation of these with cardiorespiratory events such as apneas, desaturation of oxyhaemoglobine and changes in cardiac rhythm. For sleep measurement, including body position changes, respiratory effort and efficiency in ventilation, there exist multiple methods and each clinic tends to use its own variables which are obtained with the resources available in each centre.

At present, it is not appropriate to define rigid diagnostic criteria in this rapidly developing area. Neither is it possible to identify the ideal equipment for sleep studies.

The Polysomnogram is a technique which is complex to realise and to interpret, and its economic cost is high. This provokes a saturation of the few installations available for its practise and the resulting diagnostic delay. The situation has obliged the search for simpler diagnostic alternatives, the majority of which are based in the registration and evaluation of the cardiorespiratory parameters, or in the use of simplified portable equipment for home diagnosis. The validation of many of these diagnostic kits is still being studied. A previous filtering of patients is recommended to select those which are most appropriate to be given a Polysomnogram. An effort has to be made to define which is the most useful method for this objective, given that a filtering method has to be sensitive, specific and economical. Due to the high prevalence of respiratory alterations during sleep, we should evaluate the cost/benefit of these methods in order to identify, diagnose and treat the majority of the sick.

1.3 Main contributions

The following section summarises the main contributions of the thesis to the chosen fields of study, namely: mixed data type processing and representation; a novel use of the Hartigan 'joining algorithm' using as input a 'fuzzy covariance matrix'; theoretic conception, development and testing of a fuzzy covariance calculation; analysis of ICU medical data using standard data mining algorithms with the motivation of comparing the results to analysis of the same data using Hartigan's 'joining algorithm' with fuzzy and crisp covariance matrices as input, and using fuzzy c-Means for analysis of the relationship between the variables and the clusters; the use of a genetic algorithm for weight vector selection for the WOWA aggregation operator; modification of WOWA for variable weight vector and missing values processing; data representation and membership function design for fuzzy question responses in Apnea screening questionnaire.

1.3.1 Analysis of existing algorithms

A selection of existing AI Data Mining and statistical techniques are executed against the medical test data sets in order to establish their capacity to produce coherent results from the data. The AI and statistical techniques include C4.5 and ID3 rule induction, feedforward neural network, statistical analysis (covariance, max, mix, mean, median, distribution plots), Kohonen SOM clustering. We identify areas which the standard techniques produce reasonable results, and those areas where the results have room for improvement. The ICU data from the Hospital of Sabadell is thoroughly analysed by a battery of AI and statistical analysis techniques, which identifies some of the strengths and weaknesses of each [Nettleton96][Nettleton99a]. Then this is contrasted by processing the data with Hartigan's 'joining algorithm' using fuzzy and crisp covariances as input, and fuzzy c-Means to cluster the data and show relationships between variables and the fuzzy cluster prototypes. The techniques are discussed in Chapter 2 of the thesis, and the results are given in Section 4.1.

1.3.2 Mixed data type processing and representation

Mixed data type processing is an area which is still unresolved or even not approached by many statistical techniques. The major data types (integer, categorical ordinal, nominal, binary, …) have been considered systematically to establish forms of representing , comparing and processing them together. For example, how would one calculate the covariance between a first variable defined as numerical and a second variable defined as non orderable categorical? Possible approaches are developed from basic notions, for establishing covariance between variables of different types, such as the point density diagrams of Figures 42 and 43. This section revisits different techniques for comparing variables of distinct types and presents some novel interpretations from statistical first principles. This area of the work is summarised in [Nettleton97][Nettleton98a] and in the thesis it is covered in Sections 3.1 and 4.1.

1.3.3 Novel use of Hartigan Joining Algorithm

The Hartigan clustering algorithms [Hartigan75] are a work of reference in the statistical field. In this thesis, the 'joining algorithm' has been used in a new context, that of factor reduction from covariances of mixed and fuzzy data types. It is applied to test data sets and real ICU and Apnea medical data sets for data reduction and factor analysis. One of the interesting characteristics of this algorithm is that it allows one to observe the successive 'joinings' of attributes in a tree, reducing the number of attributes by pairs in each iteration. This allows the study of the groupings of variables in a data set, such as that of the ICU data [Nettleton98b]. This area of work is covered in Sections 3.2 and 4.1 of the thesis.

1.3.4 Fuzzy covariance calculation

Bezdek, Gustafson and Kessel defined the basis for fuzzy covariances between the fuzzy prototype and a fuzzy data instance. In this thesis the formula has been extended to calculate the covariance between two fuzzy variables in a fuzzy set. Different versions of the algorithm were defined and tested against standard test data sets (Iris, Gustafson's Cross, …) and against a real medical dataset. The resulting covariances were compared against the standard SPSS covariances generated from the same data, and also with techniques such as principal components, neural network and rule induction to identify and rank the most significant variables in a data set [Nettleton98b]. This area of work is covered in Sections 3.1 and 4.2 of the thesis.

1.3.5 Genetic algorithm for learning of WOWA weights

One of the problems with algorithms which use weighting factors is how to assign the weights, and find the best values for the initial weight assignments. The WOWA aggregation operator uses two weighting vectors for relevance and reliability, and different tests were carried out to learn both vectors, just the relevance vector, or just the reliability vector, using a genetic algorithm to learn the weights. This is a new approach with respect to establishing weights for aggregation operators in general, and the WOWA operator in particular. The GA learning was benchmarked against the ASM Active Sets Method and by expert assignment of the weights, and compared favourably [Nettleton01b]. This area of work is covered in Sections 3.2 and 4.3 and 4.4 of the thesis.

1.3.6 WOWA modified for variable weight vector and missing values processing

The standard WOWA aggregation operator was modified to enable processing of data for which one or more data values of one or more variables was missing, or undefined. This enables processing of real data sets in which missing data is often a problem. The algorithm detects the missing values in a pre-processing phase and 'contracts' the weighting and data vector to cover only the known values. Also methods were tested for enabling the reliability weights to be dynamically interpreted for each case, which would allow the algorithm to adapt to different data distributions which may exist from one dataset to another, or from one application domain to another [Nettleton01b]. This area of work is covered in Section 3.2 of the thesis. The results of using dynamical interpretation of the reliability weights are detailed in Section 4.3.

1.3.7 Data representation for fuzzy processing – Apnea questionnaire

Screening of Apnea patients is a practise which can avoid costly and unnecessary admission and testing of patients who do not have the ailment. Unfortunately, questionnaire screening, the standard method, does not have a high precision rate. One of the reasons may be the lack of more subtle interpretation of the responses; we consider that the use of 'membership grades' may provide a solution. A scalar format was introduced for the questionnaire responses, and we designed an appropriate membership function curve to be overlaid on the corresponding scale. This permits a fuzzy response and the calculation of a grade of membership for the five defined linguistic labels. A reliability vector and a relevance weight were also assigned to each variable-attribute. These weights give a greater flexibility in processing data which is susceptible to variability in terms of its reliability and relevance, and controls the impact that these aspects have on the overall outcome (diagnosis). The work related to Apnea data aggregation and representation is summarised in [Nettleton99b] [Nettleton99c] [Nettleton99e] [Nettleton01a]. This area is covered in Sections 3.1, 4.3 and 4.4 of the thesis.

1.3.8 Application of AI techniques to Apnea Diagnosis

In contrast to ICU patient prognosis, which is an area which has received considerable investigation both with statistical and AI techniques, Apnea diagnosis is little explored with AI techniques. We apply WOWA aggregation to give a diagnostic output, employing reliability and relevance weights to refine the model, and contrasting expert weight assignment to weight learning with a genetic algorithm. Also we benchmark neural nets, induction algorithms, principal components and the OWA aggregator against the same data, the results of which are detailed in Sections 4.3 and 4.4 of the thesis.

Chapter 2. Some Preliminaries

This Chapter details the background of the most relevant authors and methods which have served as the basis for the areas associated with the thesis work. The following themes are covered: classical statistics, fuzzy set theory and fuzzy covariance, aggregation, clustering, fuzzy clustering and classification. As outlined in Chapter 1, and summarised in Figures 1 and 2, we understand the Data Mining process as consisting of three main steps: data definition and representation, data exploration, and data modelling. Thus for each of these steps, we have studied corresponding themes. In the case of data definition and representation we consider classical statistical methods, fuzzy set theory and issues related to representation of data in the fuzzy form, such as the definition of membership functions and quantifiers. In the case of data exploration we consider how to identify relations in the data and between variables, with techniques such as correlation, covariance, fuzzy covariance, clustering and fuzzy clustering. Finally, for the data modelling phase we consider rule induction methods such as ID3 and C4.5 to establish a meaningful classification for the data and also serve as predictive or diagnostic models. Aggregation operators are also considered a method of data modelling, given that their output is interpreted as a predictive value or diagnosis. Emphasis is given to the background of aggregation operators, such as OWA (ordered weighted average) and WOWA (weighted OWA), given that this is the method experimentally developed in Chapter 3 and applied in Chapter 4 to real data domains. Fuzzy data processing and fuzzy representation are also given extensive coverage, as in Chapter 3 and 4 fuzzy techniques play a central role in the development of methods for data inputs and outputs whose most adequate representation is in the fuzzy form.

2.1 Classical statistics

In this section we cover a selection of the standard statistical concepts and methods used in data analysis, and some of which are used in later sections to analyse the datasets, and compare with other methods such as those of AI and the techniques developed in this thesis. The methods covered are: variance, covariance, correlation, multivariate variance analysis, likelihood, variance analysis (ANOVA), covariance analysis and regression models.

Variance, Covariance and Correlation [Lebart85], pp24.

Let $(\Omega, P(\Omega), \mathbf{P})$ be a finite probability space in which the random variables X and Y have been defined; we note that $\mu x = E(X)$ and $\mu y = E(Y)$. The following quantities are defined:

(1) variance of X : $\quad Var(X) = E[(X - \mu x)^2]$ (2.1)

(2) covariance between X and Y : $\quad Cov(X,Y) = E[(X - \mu x)(Y - \mu y)]$ (2.2)

(2) correlation between X and Y : $\quad \rho(X, Y) = Cov(X,Y) / \sqrt{(Var(X)\, Var(Y))}$ (2.3)

Multivariate Variance Analysis [Cuadras80], pp503.

The observed variables are indicated by Y (in the univariate case) or $Y_1, ..., Y_n$ (in the multivariate case). In 'linear mode', the definition is as follows: let Y be an observable variable of which a sample of size N has been obtained, in different experimental conditions. We indicate the sample by the column vector

$$y = (y_1, ..., y_N)'$$

The linear mode of the variance analysis consists of the following elements:

1) m unknown parameters $\beta_1, ..., \beta_m$ known as regression parameters. In vectorial notation

$$\beta = (\beta_1, ..., \beta_m)'$$

2) A matrix of known elements

$$A = \begin{pmatrix} a_{11} \ldots\ldots\ldots a_{1m} \\ a_{21} \ldots\ldots\ldots a_{2m} \\ \ldots\ldots\ldots\ldots\ldots \\ a_{N1} \ldots\ldots\ldots a_{Nm} \end{pmatrix}$$

called a factorial design matrix. The range of **A** is known as the design range. If the range is **m** it is said that the design has maximum range.

3) The linear model that relates the observations to the parameters

$$y_i = a_{i1}\beta_1 + ... + a_{im}\beta_m + e_i \; i = 1, ... , N \qquad (2.4)$$

where e_i is the error or random deviation of the model.

Indicating $\mathbf{e} = (e_1, ... , e_N)'$, the matricial expression of the model is

$$y = A\beta + \mathbf{e} \qquad (2.5)$$

4) It is assumed that $e_1, ... , e_N$ are independent, with mean 0 and variance σ^2 . It is also assumed that the variance, which is another unknown parameter of the model, is the same for each e_i (condition of homoscedacity). In consequence $y_1, ... , y_N$ are also independent, with the same variance σ^2 , and with means

$$E(y_i) = a_{i1}\beta_1 + ... + a_{im}\beta_m \; i = 1, ... , N,$$

in matrix notation, this is written as

$$E(y) = A\beta \qquad (2.6)$$

5) If it is also assumed that each e_i follows the normal distribution, we can then refer to a normal linear model.

As part of the definition, one can also define the 'reduced design matrix' and the estimation of parameters $\beta = (\beta_1, ..., \beta_m)'$, by the criteria of squared minimums, and of σ^2 with an unslanted estimator.

Likelihood

Given a random variable **X** that can assume a series of values $x_1, x_2,...x_n$ with probabilities respectively equal to $p_1, p_2,...p_n$, we can define its likelihood $E(\mathbf{X})$ as the expression:

$$E(\mathbf{X}) = \sum_{i=1}^{n} x_i \, P_i \qquad (2.7)$$

If a series of **s** independent tests is realised and their average value calculated, the probability of this tends to the likelihood when **s** approaches infinity. The likelihood possesses interesting properties of linearity:

$$E(\mathbf{X} + \mathbf{Y}) = E(\mathbf{X}) + E(\mathbf{Y}) \qquad (2.8)$$

$$E(a\mathbf{X}) = aE(\mathbf{X}) \qquad (2.9)$$

Variance Analysis. [Peña84], Vol II, pp29-77.

Variance Analysis is a procedure, created by R.A.Fisher in 1925, to decompose the variability of an experiment in independent components that can be assigned to distinct causes.

Example:
Suppose that we wish to establish if the life of the elements produced by a group of **I** machines is the same in the long term (does not depend on the machine). We assume that the life of the elements produced by the same machine varies due to many non-controllable factors (pureness of the raw materials, random loss of precision of the machine, running temperature, operator skill, etc.), and that we have measured the life of n_1 elements of machine **1**, and n_i of machine **i**, with a total of **n** data for the set of **I** machines:

$$\Sigma n_i = n$$

Let y_{ij} be the random variable 'life of element **j** produced by machine **i**'. The objective of the study is:

(1) verify if all the machines are identical: that is, produce elements with the same average life;
(2) if the machines are not equal, estimate the average life of the elements produced by each one.

To achieve this, we have to formalise the situation with a mathematical model, which is now briefly detailed.

The Model

We permit that the average life oscillates randomly about an unknown value μ_i , that characterises a machine. The differences between the effectively observed values for a machine y_{ij} , and its mean, μ_i , are the result of multiple factors that we encompass in a term known as *'experimental error'* or *'perturbation'*. Thus:

$$y_{ij} = \mu_i + \mu_{ij} \qquad (2.10)$$

We assume that the perturbations $\mu_{ij} = y_{ij} + \mu_i$, verify the following hypotheses:

a) $E[\mu_{ij}] = 0$ $\forall i, j$
b) $Var[\mu_{ij}] = \sigma^2$ $\forall i, j$ (2.11)
c) $E[\mu_{ij}\mu_{rk}] = 0$ $i \neq r$, or $j \neq k$
d) Its distribution is normal

Condition (a) requires that the **n** random variables μ_{ij} have average zero, and is equivalent to requiring that the n_i observations proceeding from the machine **i** have the same mean u_i. In order for this to occur, the distinct measures of the life of the elements have to have been taken in homogeneous conditions.
Condition (b) requires that the perturbations have the same variability in all the machines and that, besides, this variability is stable – that is, it does not tend to increase and to decrease – in the experiment. This condition is equivalent to saying that the variance of the **n** random variables y_{ij} must be the same.
Condition (c) imposes that the experimental errors or perturbations are produced in an independent manner, from one observation to another, which implies the independence of the observations y_{ij} . This hypothesis is difficult to test in practice and, as we see later, one of the objectives if designing an experiment is to guarantee this independence.
Finally, the hypothesis of normality is justified by the central theorem of the limit: assume that the perturbations cannot be predicted or assigned to concrete causes, but that they result from the accumulated effect of many distinct factors, none of which is predominant.
To summarise, the model established by (2.10) and (2.11) above, specifies that the n_i observations of machine **i** are a simple random sample of a random variable with normal distribution, $N(\mu_i, \sigma)$.

Variance analysis can be understood in two forms:

(i) A procedure to compare groups that may or may not differ in their averages.
(ii) A type of statistical model in which a qualitative variable is used to explain the possible differences between quantitative variables: the group to which they belong.

In emphasising the construction of a model, we show the contrast depends on certain hypotheses, which, on verification, guarantee its validity.

Decomposition of the Variability.
The deviations between the observed data and the general mean can be expressed by the identity:

$$y_{ij} - \ddot{y}.. = (\ddot{y}_{i.} - \ddot{y}..) + (y_{ij} - \ddot{y}_{i.}) \qquad (2.12)$$

which decomposes the variability between the data and the mean in two terms: the variability between the means and the general mean, and the residual, or variability in side the group. We raise to the square, and sum for the *n terms* . Now, 'Total Variability' will be:

$$TV = \sum_i \sum_j (\ddot{y}_{ij.} - \ddot{y}..)^2$$

Explained Variability will be:

$$EV = \Sigma\, n_i(\ddot{y}_{i\cdot} - \ddot{y}_{\cdot\cdot})^2$$

UnExplained Variability (or residual) will be:

$$UEV = \Sigma\Sigma\, (\ddot{y}_{ij\cdot} - \ddot{y}_{\cdot\cdot})^2 = \Sigma\Sigma e^2_{ij}$$

we will have that:

$$VTV = EV + UEV \qquad (2.13)$$

This result is due to the fact that the sum of the double products resulting from raising to the square (2.12, above) is null:

$$\sum_i \sum_j (\ddot{y}_{i\cdot} - \ddot{y}_{\cdot\cdot})(y_{ij} - \ddot{y}_{i\cdot}) = \sum_i (\ddot{y}_{i\cdot} - \ddot{y}_{\cdot\cdot}) \sum_i e_{ij} = 0 \qquad (2.14)$$

being that, the sum of the residuals is zero inside each group.

Table 3. ANOVA - Analysis of Variance

Source of Variation	Sum of Squares	Grades of Freedom	Variances
Between Groups (EV)	$\Sigma n_i(\, y_{i\cdot} - \ddot{y}_{\cdot\cdot})^2$	**I - 1**	$s'^2_e = \dfrac{EV}{I - 1}$
Internal, not explained or residual (UEV)	$\Sigma\Sigma(\, y_{ij} - \ddot{y}_{i\cdot})^2$	**n - I**	$s'^2_R = \dfrac{UEV}{n - I}$
TOTAL	$\Sigma\Sigma(\, y_{ij} - \ddot{y}_{\cdot\cdot})^2$	**n - 1**	s'^2_y

Methodology

(a) Specify the model

1) $y_{ij} = \mu_i + u_{ij}$

2) $u_{ij} \sim N(0,\sigma^2)$

(b) Estimate the parameters

$(\mu_1, ..., \mu_I ; \sigma^2)$

$\mu'_i = \ddot{y}_i$

$$s'^2_R = \frac{\Sigma\Sigma(\, y_{ij} - \ddot{y}_{i\cdot})^2}{n - I}$$

(c) Contrast if a simplification is possible (ADEVA contrast)

$H_0 : \mu_1 = \mu_2 = ... = \mu_I = \mu$

$$F = \frac{\Sigma\, n_i(\, y'_{i\cdot} - \ddot{y}'_{\cdot\cdot})^2}{(I - 1)\, s'^2_R}$$

(d) Construct confidence intervals for the parameters and revise the basic hypotheses.

(1) if the averages are the same, intervals for μ will be:

$$s'^2_R = \frac{\Sigma\Sigma(y_{ij} - \ddot{y}_{i.})^2}{n - I}$$

(2) if the averages are not the same, intervals for $\mu_i - \mu_j$, multiple contrasts.

(e) If the model is not adequate it will have to be reformed (data transformations, introduction of new explicative variables, modify the hypothesis).

Diagnosis and validation.

- homogeneity $E[\mu_{ij}] = 0$?
- homocedasticity $Var[\mu_{ij}] = \sigma^2$?
- normality
- independence

Covariance [Cuadras80], Vol.I, pp.233.

Definition of Covariance: we assume that we have a sample of **n** pairs of observations of two variables X and Y

X: $x_1\ x_2 \ldots x_n$
Y: $y_1\ y_2 \ldots y_n$

Let $x' = \frac{1}{n} \Sigma x_i$, $y' = \frac{1}{n} \Sigma y_i$.

The following is called the covariance of the sample

$$S_{xy} = \frac{1}{n} \sum_{i=1}^{n} (x_i - x') (yi - y')$$

We verify that

$$S_{xy} = \frac{1}{n} \sum_{i=1}^{n} x_i\ yi - x'\ y'$$

The generalisation of the covariance to random variables consists in defining

$$cov(X, Y) = E[(X - E(X)) . (Y - E(Y))]$$

for two random variables X, Y, assuming that E(X), E(Y) and E(X . Y) exist.

Covariance of qualitative values [Cuadras80], Vol. II, pp.324.

Covariance analysis: is a synthesis of 'variance analysis' and 'regression' methods. It thus combines a number of qualitative variables with a number of quantitative variables. It proposes to relate an observable variable Y with a second variable X, called the concomitant, which influences in the design, and establishes hypothesis taking into account this relation.

The Chi-Squared test (χ^2) [Cuadras80], Vol. II, pp.211.

This is a statistical test which lets us decide if some observed frequencies adjust to some expected frequencies. It is based in two theorems, the first relates to the case in which the probabilities p_i are known, and the second relates to the case in which they depend on certain parameters which must be estimated.

Regression analysis

Linear regression is a statistical modelling technique which examines the relationship between a dependent variable and a set of independent variables. For example, we could try to predict a customers' total yearly purchases (the dependent variable) from independent variables such as age, socio-economic indicator, years as customer, and region of residence. Both the dependent and independent variables must be measured on an interval scale. Nominal variables such as gender or region of residence have to be recoded as binary variables. To establish how well the regression model fits the data, one can examine the residuals and identify any outlier values which may be present. **Linear regression** analyses the relationship between two variables, X and Y. For each case in the dataset being analysed, the values of X and Y are known and the objective is that of finding the best straight line through the data. For some data domains, the slope and/or intercept have a interpretable meaning. In other cases, the *linear regression* line is used as a standard to find new values of X from Y, or Y from X. The goal of *linear regression* is to adjust the values of slope and intercept to find the line that best predicts Y from X. More precisely, the goal of regression is to minimise the sum of the squares of the vertical distances of the points from the line.

Non-linear regression is a general technique to fit a curve through a given dataset. It fits data to any equation that defines Y as a function of X and one or more parameters. It finds the values of those parameters that generate the curve that comes closest to the data (minimises the sum of the squares of the vertical distances between data points and curve). Except for a few special cases, it is not possible to directly derive an equation to compute the best-fit values from the data. Instead *non-linear regression* requires a computationally intensive, iterative approach, with a basis in matrix algebra.

If the equation that we wish to fit is known, and its parameters are non-linear, a **non-linear technique** can be used. If the dependent variable is binary, such as whether a particular diagnosis is positive or negative, the *logistic regression* model is used. If the dependent variable is censored, such as survival time after surgery, some possible techniques would be *Life Tables*, *Kaplan-Meier*, or a *Cox Regression*.

Logistic regression estimates regression models in which the dependent variable is binary. For example, one could use a *logistic regression* to estimate the probability of a rise in the share value of a company in the stock market based on the expertise, past performance, type of business, and zone of operations of the company. Or one could estimate the probability that a patient will survive based on characteristics of the patient and the severity of the disease. Typically, a variable selection technique is used to identify a subset of independent variables that are best related to the outcome of interest. This is accompanied by diagnostic procedures to assess how well the model fits and to identify outliers. *Logistic regression* produces a formula that predicts the probability of the occurrence as a function of the independent variables. A special s-shaped curve is fitted by taking the linear regression, which could produce any *y*-value between minus infinity and plus infinity, and transforming it with the function: $p = \text{Exp}(y) / (1 + \text{Exp}(y))$ which produces *p*-values between 0 (as *y* approaches minus infinity) and 1 (as *y* approaches plus infinity).

If the dependent variable has more than two categories, a *discriminant analysis* can be used to identify variables which permit assignment of the cases to the various groups. If the dependent variable is continuous, one can also use a *linear regression* to predict the values of the dependent variable from a set of independent variables.

2.2 Fuzzy sets and fuzzy data processing

In this section we revise some of the key concepts associated with fuzzy set theory, namely: fuzzy set, fuzzy relation, membership function, fuzzy variable, fuzzy number, the concept of fuzzy membership and the elicitation of membership functions. This summary is drawn from the definitions of key investigators in the field, such as Zadeh and Bezdek.

2.2.1 Basic concepts

Uncertainty: the presence of 'uncertainty' in a mathematical model may be due to: imprecise measurements; random occurrences; vague descriptions, that are manifested in deterministic probabilistic and fuzzy models, respectively.

Deterministic: the result may be predicted with total certainty, by replication of the circumstances that define it.

Probabilistic: the result of a physical process is random, with an element of 'chance', that belongs to the evolution of a process which is not affected by imprecision in the environment (e.g. throw a coin). Allows 'stochastic' laws to be derived which allow the evaluation of the probability of observing a given result.

Fuzzy: exists in a physical situation which manifests a non-stochastic source of uncertainty (e.g. class of people who are almost two metres tall). Introduces membership grades. It is neither deterministic or probabilistic.

In the case of the 'hospital admissions' (ICU) data set considered in Sections 3.1 and 4.1, we have to evaluate each attribute to decide whether it is deterministic, probabilistic, or fuzzy.

Concept of Fuzzy Membership

A fuzzy subset, F, has a membership function μ_F, defined as a function from a well defined universe (the referential set), X, into the unit interval as: $\mu_F : X \rightarrow [0, 1]$. Hence, the vague predicate "Patient (x) is Long Stay (S)" is represented by a number in the unit interval $\mu_S(x)$. There are several possible answers to the question "What does it mean to say $\mu_S(x) = 0.7$?"

likelihood view	70% of a given population declared that Patient is Long Stay.
random set view	70% of a given population described "Long Stay" as an interval containing Patient's duration.
similarity view	Patient's Length of Stay is away from the prototypical object which is truly "Long Stay" to the degree 0.3 (a normalised distance).
utility view	0.7 is the *utility* of *asserting* that Patient is Long Stay.
measurement view	when *compared* to others, Patient is Longer Stay than some and this fact can be encoded as 0.7 on some scale.

These interpretations can be further summarised as: subjective Vs objective on one dimension and individual Vs group on the other.

Zadeh is possibly *the* key author in Fuzzy Set Theory. His landmark paper [Zadeh65] coined the term 'fuzzy set' and defined its properties. He cited examples of fuzzy classes such as "the class of tall men" or "the class of all real numbers much greater than 1". He states that a "fuzzy set" is a "class" with a continuum of grades of membership, and proposed that "fuzzy sets" provide a framework similar to ordinary sets, but more general and with a potentially wider scope and applicability in fields such as pattern classification and information processing. Zadeh's electrical engineering background suggest the possibility that analogies in electrical theory contributed in giving rise to these new ideas.

The key definition of a fuzzy set is the following [Zadeh65]: Let X be a space of points (objects), with a generic element of X denoted by x. Thus, $X = \{x\}$.A fuzzy set (class) A in X is characterised by a membership (characteristic) function $f_a(x)$ which associates with each point in X a real number in the interval [0, 1], with the value of $f_a(x)$ at x representing the "grade of membership" of x in A. Thus the nearer the value of $f_a(x)$ to unity, the higher the grade of membership of x in A.

Examples and illustrations are given of unions and intersections on fuzzy sets, algebraic operations, and convexity and non-convexity. Finally a proof of the separation theorem for fuzzy sets is given in terms of a hyperplane H and the consideration of specific points on it.

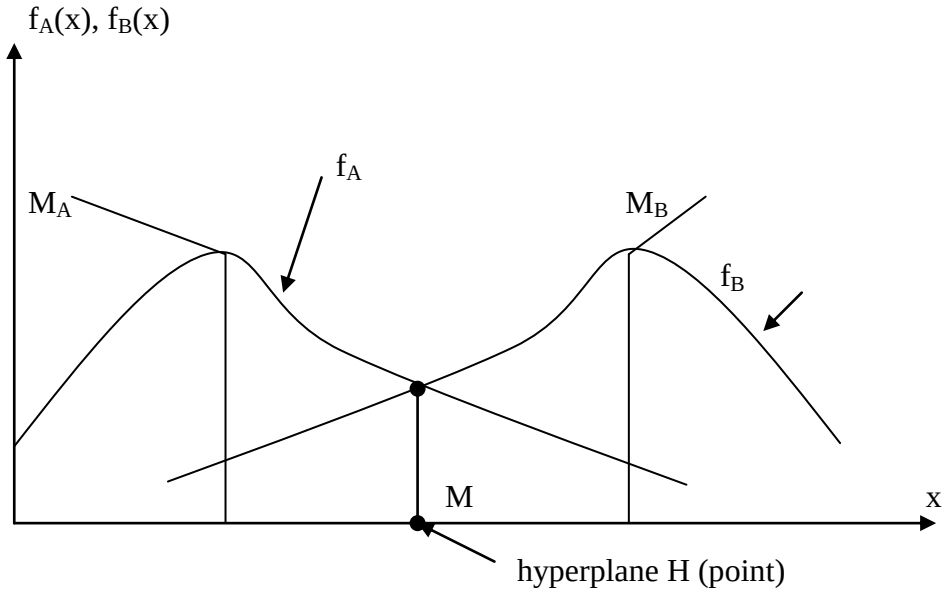


Figure 9. Illustration of the separation theorem for fuzzy sets in a real Euclidean space of E^1

In Figure 9, a hyperplane H exists which realises 1 – M as the degree of separation of A and B, and M_A and M_B are the maximal membership grades achieved by fuzzy sets A and B, respectively.

'Fuzzy Relations'

Zadeh's paper [Zadeh71], introduces three basic concepts for fuzzy set theory: "similarity", "similarity relation", and "fuzzy ordering". "Similarity" is defined as a generalisation of the notion of equivalence.

A fuzzy (binary) relation R is defined as a fuzzy collection of ordered pairs. Thus, if $X = \{x\}$ and $Y = \{y\}$ are collections of objects denoted generically by x and y, then a fuzzy relation from X to Y or, equivalently, a fuzzy relation in $X \cup Y$, is a fuzzy subset of $X \times Y$ characterised by a membership (characteristic) function μ_R which associates with each pair (x, y) its "grade of membership", $\mu_R(x, y)$, in R. The range of μ_R is assumed for simplicity to be the interval [0, 1] and the number $\mu_R(x, y)$ is referred to as the strength of the relationship between x and y.

A "fuzzy partial ordering" of a fuzzy relation P in X exists iff it is reflexive, transitive and antisymmetric, where antisymmetry of P is defined as:

$$\mu_p(x, y) > 0 \quad \text{and} \quad \mu_p(y, x) > 0 \Rightarrow x = y, \quad x, y \in X. \tag{2.15}$$

A relation matrix and a 'fuzzy Hasse diagram' are used to illustrate an example of a fuzzy partial ordering.

Zadeh, in [Zadeh71], adds the following definitions to his paper of 1965:

Similarity Relation: S, is a fuzzy relation which is reflexive, symmetric and transitive. Thus, let x, y be elements of a set X and $\mu_s(x,y)$ denote the grade of membership of the ordered pair (x, y) in S. Then S is a similarity relation in X if and only if, for all x, y, z in X, $\mu_s(x, x)=1$ (reflexivity), $\mu_s(x, y)= \mu_s(y,x)$ (symmetry), and $\mu_s(x, z) > \vee (\mu_s(x,y) \wedge \mu_s(y,z))$ (transitivity), where $\vee$ and $\wedge$ denote max and min, respectively.

Fuzzy Ordering: is a fuzzy relation which is transitive. In particular, *a fuzzy partial ordering*, P, is a fuzzy ordering which is reflexive and antisymmetric, that is ($\mu_p(x, y) > 0$ and $x \neq y) \Rightarrow \mu_p(y, x) = 0$. A *fuzzy linear ordering* is a fuzzy partial ordering in which $x \neq y \Rightarrow \mu_s(x, y) > 0$ or $\mu_s(y, x) > 0$. A *fuzzy preordering* is a fuzzy ordering which is reflexive. A *fuzzy weak ordering* is a fuzzy preordering in which $x \neq y \Rightarrow \mu_s(x, y) > 0$ or $\mu_s(y, x) > 0$.

The 'Fuzzy' Relation: in the case of the 'hard' relation, a data item is a member of a cluster (membership=1), or it is not (membership=0). If all the data in the data space is classified in clusters, the fact that one data item has membership=0 for one cluster, implies that in some other cluster, it must have membership=1. It may only have membership=1 for one cluster in the data space.

In the fuzzy case, 'membership grades' are introduced, which assume values in a continuous range between 0 and 1, for example: 0.55, 0.1, 0.965, 0.73, etc. Thus it may be said that data item **a** is a member of cluster A with membership grade 0.2 (which is little), and also that data item **a** is a member of cluster B with membership grade 0.77 (which is quite high). This implies that data item **a** is a member of clusters A and B, although it has a greater membership grade to cluster B.

The standard definition of a *relation* is that of a set of ordered pairs. For example, the set of all ordered pairs of real numbers x and y such that $x \geq y$. In the context of fuzzy sets, a fuzzy relation in X is a fuzzy set in the product space $X \times X$. For example, the relation denoted by $x >> y$, $x,y \in R^1$, may be considered a fuzzy set A in R^2, with the membership function of A, $f_A(x,y)$, having the following representative values: $f_A(10,5) = 0$; $f_A(100,10) = 0.7$; $f_A(100,1) = 0.1$; and so on.

More generally, an n-ary **fuzzy relation** in X is defined as a fuzzy set A in the product space $X \times X \times ... \times X$. For such relations, the membership function is of the form $f_A(x_1,...,x_n)$, where $x_1 \in X$, i = 1, ... , n.

In the case of binary fuzzy relations, the composition of two fuzzy relations A and B is denoted by $B \circ A$ and is defined as a fuzzy relation in X whose membership function is related to those of A and B by

$$f_{B \circ A}(x,y) = \mathrm{Sup}_v \, \mathrm{Min}[f_A(x, v) , f_B(v, x)]$$

Definitions for Fuzzy Relations

(i) *Convexity:* A fuzzy set A is *convex* if and only if the sets Γ_α defined by

$$\Gamma_\alpha = \{x \mid f_A(x) \geq \alpha\} \qquad (2.16)$$

are convex for all α in the interval (0,1].

(ii) *Boundedness:* A fuzzy set A is *bounded* if and only if the sets $\Gamma_\alpha = \{x \mid f_A(x) \geq \alpha\}$ are bounded for all $\alpha > 0$; that is, for every $\alpha > 0$ there exists a finite $R(\alpha)$ such that $\|x\| \leq R(\alpha)$ for all x in Γ_α .

If A is a bounded set, then for each $\varepsilon > 0$ there exists a hyperplane *H* such that $f_A(x) \leq \varepsilon$ for all x on the side of *H* which does not contain the origin. For example, consider the set $\Gamma_\varepsilon = \{x \mid f_A(x) \geq \varepsilon\}$. By hypothesis, this set is contained in a sphere S of radius $R(\varepsilon)$. Let *H* be any hyperplane supporting S. Then, all points on the side of *H* which does not contain the origin lie outside or on S, and hence for all such points $f_A(x) \leq \varepsilon$.

(iii). *Strict Convexity:* A fuzzy set A is *strictly convex* if the sets Γ_α , $0 < \alpha \leq 1$ are strictly convex (that is, if the midpoint of any two distinct points in Γ_α lies in the interior of Γ_α). Note that this definition reduces to that of strict convexity for ordinary sets when A is such a set.

(iv) *Strong Convexity*: A fuzzy set A is *strongly convex* if, for any two distinct points x_1 and x_2, and any λ in the open interval (0,1)

$$f_A[\lambda x_1 + (1 - \lambda) x_2] > \mathrm{Min}[f_A(x_1), f_A(x_2)].$$

Note that strong convexity does not imply strict convexity or vice-versa. Note also that if A and B are bounded, so is their union and intersection. Similarly, if A and B are strictly (strongly) convex, their intersection is strictly (strongly) convex.

(v) *Separation of Convex Fuzzy Sets*: the classical separation theorem for ordinary convex sets states , in essence, that if A and B are disjoint convex sets, then there exists a separating hyperplane *H* such that A is on one side of *H* and B is on the other side.

Can this theorem can be extended to convex fuzzy sets, without requiring that A and B be disjoint ? We wish to avoid the condition of disjointedness given that it is too restrictive in the case of fuzzy sets. The following shows that the classical separation theorem for ordinary convex sets can be extended to convex fuzzy sets.

First we define A and B as two bounded fuzzy sets and *H* as be a hypersurface in E^n defined by an equation h(x) = 0, with all points for which $h(x) \geq 0$ being on one side of *H* and all points for which $h(x) \leq 0$ being on the other side. K_H is defined as a number dependent on *H* such that $f_A(x) \leq K_H$ on one side of H and $f_B(x) \leq K_H$ on the other side. M_H is defined as Inf K_H . The number $D_H = 1 - M_H$ is defined as the *degree of separation* of A and B by H. The case is generalised from that of a given hypersurface *H* to that of a family of hypersurfaces $\{H_\lambda\}$, with λ ranging over E^m. The problem is stated as that of finding a member of this family which achieves the highest possible degree of separation.

Taking a special case of this problem, where the H_λ are hyperplanes in E^n , with λ ranging over E^n . In this case, we define the degree of separability of A and B by the relation

$$D = 1 - \bar{M} \tag{2.17}$$

where

$$\bar{M} = \mathrm{Inf}_H M_H \tag{2.18}$$

with the subscript λ omitted for simplicity.

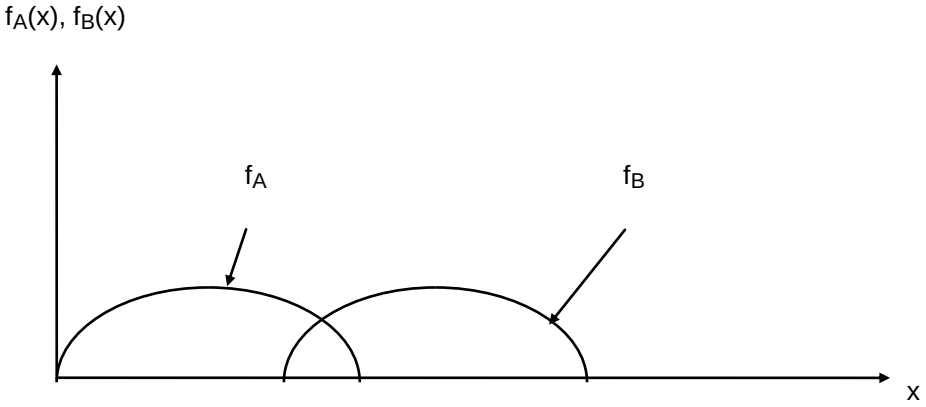


Figure 10. Illustration of the union and intersection of fuzzy sets in R^1

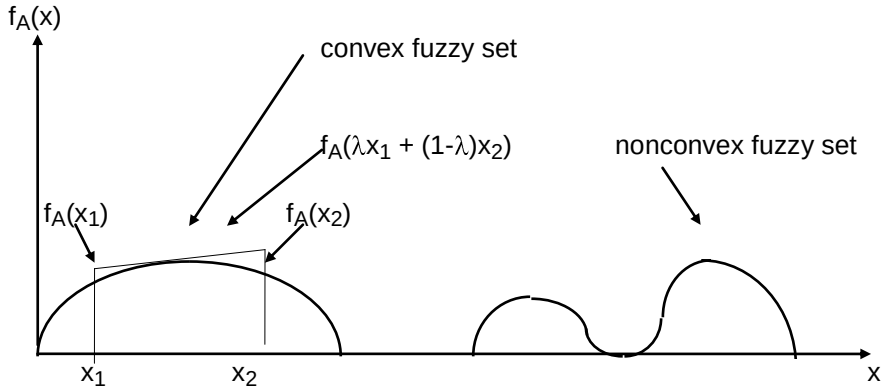


Figure 11. Convex and nonconvex fuzzy sets in E^1

Definition of a Fuzzy Set

Bezdek, in [Bezdek81], gave the following illustration of a fuzzy set: X is defined to represent a sample of n people, and A_1 is defined as the subset of X for which h(x) is exactly two metres:

$$A_1 = \{x \in X \mid h(x) = 2\} \tag{2.19}$$

If we agree to say that x is nearly two metres if and only if x belongs to A_1 ($x \in A_1$), then A_1 will be a very sparse set, its obvious shortcoming being that we cannot measure h(x) exactly. To overcome this deficiency, consider the set

$$A_2 = \{x \in X \mid h(x) = 2 \pm 0.005\} \tag{2.20}$$

If membership in A_2 is equivalent to nearly two metres, the resulting decision rule will identify many people that are nearly two metres tall. But, the threshold $\pm$ 0.005 would exclude, for example, person y, whose observed height h(y) is 2.0051 metres. Another set of problems occur which stochastic or possibilistic models.

A more appropriate model was suggested by [Zadeh65]: since set membership is the key to our decisions, let us alter our notion of sets when the process suggests it, and proceed accordingly. Mathematical realisation of this idea is the following. We let

$$A_3 = \{x \mid x \text{ is nearly two metres tall}\} \tag{2.21}$$

Since A_3 is not a conventional (hard) set, there is no direct interpretation for it in traditional set theory. We can, however, imagine a function-theoretic representation, by a function, say u_3: $X \rightarrow [0,1]$, whose values u_3:(x) give the grade of membership of x in the fuzzy set u_3. This is a natural generalisation of the function-theoretic relation of sets A_1 and A_2 by their characteristic (or indicator) functions, say u_1 and u_2, respectively, where

$$u_1(x) = \begin{pmatrix} 1; & x \in A_1 \\ 0; & \text{otherwise} \end{pmatrix} \tag{2.22}$$

$$u_2(x) = \begin{pmatrix} 1; & x \in A_2 \\ 0; & \text{otherwise} \end{pmatrix} \tag{2.23}$$

u_3 embeds the two-valued logic of {0,1} in the continuously valued logic [0,1] .

We might define a discrete fuzzy model such as

$$u_3(x) = \begin{pmatrix} 1, & 1.995 \le h(x) \le 2.005 \\ 0.95, & 1.990 \le h(x) < 1.995 \text{ or } 2.005 < h(x) \le 2.010 \\ & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ 0.05 & \ldots \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \end{pmatrix} \tag{2.24}$$

Zadeh in [Zadeh65], defines the following: let X be a space of points (objects), with a generic element of X denoted by x. Thus X = {x}. A *fuzzy set* (class) A in X is characterised by a *membership (characteristic) function* $f_A(x)$ which associates with each point in X a real number in the interval [0,1], with the value of $f_A(x)$ at x representing the "grade of membership" of x in A.

Thus, the nearer the value of $f_A(x)$ to unity, the higher the grade of membership of x in A. When A is a set in the standard sense of the term, its membership function can take on only two values 0 and 1, with $f_A(x) = 1$ or 0 according as x does or does not belong to A. Thus, in this case $f_A(x)$ reduces to the familiar characteristic function of a set A.

Definitions for Fuzzy Sets:

(i) Empty Set: a fuzzy set is empty if and only if its membership function is identically zero on X.

(ii) Equality: two fuzzy sets A and B are equal, written as A = B, if and only if $f_A(x) = f_B(x)$ for all x in X.

(iii) Complement: the complement of a fuzzy set A is denoted by A' and is defined by

$$f_{A'} = 1 - f_A . \tag{2.25}$$

(iv) Containment: A is contained in B (or, equivalently, A is a subset of B, or A is smaller than or equal to B) if and only if $f_A(x) \leq f_B(x)$. In symbols

$$A \subset B \Leftrightarrow f_A(x) \leq f_B(x). \tag{2.26}$$

(v) Union: The union of two fuzzy sets A and B with respective membership functions $f_A(x)$ and $f_B(x)$ is a fuzzy set C, written as $C = A \cup B$, whose membership function is related to those of A and B by

$$f_C(x) = \text{Max}[f_A(x) , f_B(x)], \qquad x \in X. \tag{2.27}$$

or, in abbreviated form,

$$f_C(x) = f_A(x) \vee f_B(x) \tag{2.28}$$

(vi) Intersection: the intersection of two fuzzy sets A and B with respective membership functions $f_A(x)$ and $f_B(x)$ is a fuzzy set C, written as $C = A \cap B$, whose membership function is related to those of A and B by

$$f_C(x) = \text{Min}[f_A(x) , f_B(x)], x \in X. \tag{2.29}$$

or, in abbreviated form

$$f_C(x) = f_A(x) \wedge f_B(x) \tag{2.30}$$

Membership Function: Zadeh, in [Zadeh71], defines the following: A *membership (characteristic) function* $f_A(x)$ associates with each point in X a real number in the interval [0,1], with the value of $f_A(x)$ at x representing the "grade of membership" of x in A. We could define a membership function as any function from a well defined referential set X into the unit interval:

$$\mu : X \rightarrow [0, 1]$$

Therefore, we could say that $\mu(x) = 0.4$ where x is a certain object. This could be the grade of membership of the sponge (x) to the fuzzy set ' large' is 0.4. A membership function may be triangular, trapezoidal, convex, concave, etc.

Eight methods for eliciting membership functions:

polling: do you agree that the Patient is Long Stay? (Yes/No).

direct rating (point estimation): classify colour A according to its darkness, classify Patient according to his Length of Stay. In general, the question is: "How F is a ?".

reverse rating: identify the Patient who is Long Stay to the degree 0.6? In general, identify a who is F to the degree $\mu_F(a)$.

interval estimation (set valued statistics): give an interval in which you think colour A lies, give an interval in which you think the Length of Stay of Patient lies.

membership function exemplification: what is the degree of belonging of colour A to the (fuzzy) set of dark colours? What is the degree of belonging of Patient to the set of Long Stay patients? In general, "To what degree a is F?".

pairwise comparison: which colour, A or B, is darker (and by how much?)

clustering methods: given a set of input data extract the fuzzy subset of Long Stay patients.

neural-fuzzy methods: given a set of input data and a neural structure, extract the fuzzy subset of Long Stay patients.

Fuzzy Algorithm

Another key definition of [Zadeh73] is that of the fuzzy algorithm. He states that in broad terms, a fuzzy algorithm is an ordered set of fuzzy instructions which upon execution yield an approximate solution to a specified problem. Humans use fuzzy algorithms all the time to: park the car, cook a meal, find a number in a telephone directory, and so on. A simple example of a relational algorithm R defines a fuzzy ternary relation R in the data space U = 1 + 2 + 3 + 4 + 5 with small and large defined as follows:

small = 1/1 + 0.8/2 + 0.6/3 + 0.4/4 + 0.2/5 (2.31)

large =0.2/1 + 0.4/2 + 0.6/3 + 0.8/4 + 1/5 (2.32)

algorithm R(*x*, *y*, *z*)
{
IF *x* is *small* AND *y* is *large* THEN *z* is *very small* ELSE *z* is not *small*;
IF *x* is *large* THEN (IF *y* is *small* THEN *z* is *very large* ELSE *z* is *small*) ELSE *z* and *y* are *very very small*;
}

The relation R is the result of the intersection of the relations defined by each of the two instructions. In the definitions (2.31) and (2.32) above we can see some similarities with the definitions of weighted aggregation operators and the work of Yager. Another example consists of rules such as "IF *x* is *small* and *x* is increased *substantially* THEN *y* will increase *substantially*".

Fuzzy Variable

Is a variable whose values are definable as members of a fuzzy set, with a given membership function. The concepts of 'Fuzzy' as conceived by Zadeh or Bezdek do not refer to fuzzy variables as such. Rather they begin with a definition of a 'fuzzy set theory' as an extension of classical set theory. Zadeh introduced notions of 'fuzzy set', 'membership function', 'similarity relation', 'fuzzy ordering' using traditional set theory as a starting point.

Therefore the concept of a fuzzy variable is not directly defined. Rather we have a crisp variable with crisp values, which are passed through a membership function to give grades of membership to the fuzzy sets identified. In contrast, consider a variable which 'begins life' as fuzzy (the initial reading is in terms of grades of membership to fuzzy sets). This implies a previous interpretation by some membership function. Often, 'linguistic variables' are good candidates for fuzzy representation.

Consider an object, such as a sea sponge, which may have many variables of different types associated with it. Possibly, for some of those variables, the most adequate representation is the fuzzy form. For example, the variable 'diameter' may be representable by three fuzzy sets: 'small', medium' and 'large'. For each object (sea sponge), the variable 'diameter' would then be expressed as three values, each one being a membership grade for the respective fuzzy sets defined previously. The fuzzy sets will be defined for all objects for each fuzzy variables, by a unique membership function. Rather than fuzzy variable, reference is usually made in the literature to 'fuzzy number' or 'fuzzy value', such as in [Delgado95].

Fuzzy Number

[Kahraman97] defines a fuzzy number as a normal and convex fuzzy set with membership function μ (x) which both satisfies

normality : $\mu(x) = 1$, for at least one $x \in R$

and

convexity: $\mu(x') \geq \mu(x_1) \wedge \mu(x_2)$

where $\mu(x) \in [0, 1]$ and $\forall x' \in [x_1, x_2]$.

Fuzzy numbers are very useful in promoting the representation and processing of information under a fuzzy environment. A trapezoidal fuzzy number (TzFN) can be denoted by (a,b,c,d) while a triangular fuzzy number (TFN) can be denoted by (a',b',c'). The membership grades are simple readings off the graphical representations of the membership functions.

2.2.2 Quantifiers

In [Zadeh73] Zadeh focuses on the problem of processing linguistic variables, such as "tall", "not tall", "very tall", "very very tall", and so on. A linguistic variable is defined as one whose values are sentences in a natural or artificial language.

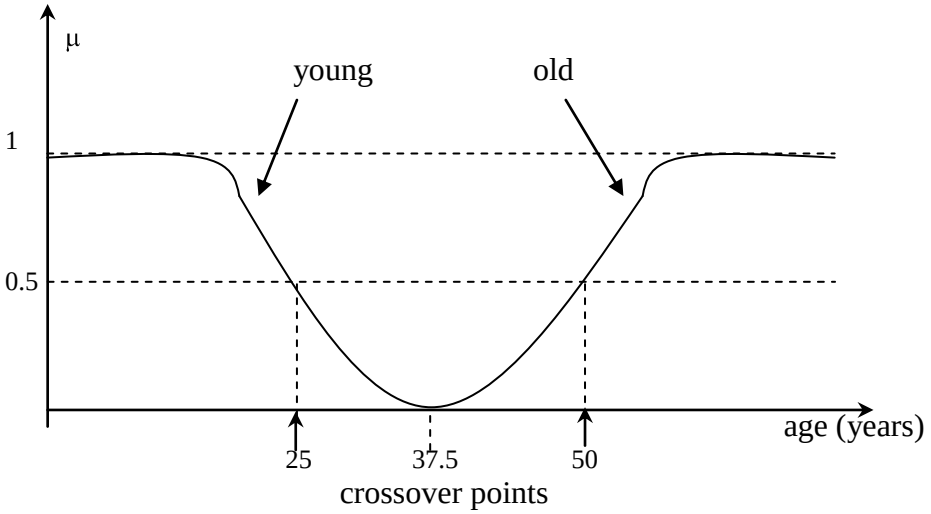


Figure 12. Graphical representation of 'young' and 'old'

Zadeh considers defining a fuzzy linguistic label by the union of fuzzy singletons, that is fuzzy sets whose support is a single point in U, over a given range. In Figure 12, for example, if the universe U is the interval [0, 100] then the label 'young' could be defined by the union of the fuzzy singletons over 0 to 37.5 (years), while 'old' could be defined by the union of the fuzzy singletons over 37.5 to 100.

This leads on to the definition of 'fuzzy conditional statements', which are expressions of the form: IF x is *very small* THEN y is *quite large*. One key aspect in Zadeh's opinion is that the meaning of such statements when used in communication between humans is poorly defined. Zadeh demonstrates that the condition statement IF A THEN B can be given a precise meaning even when A and B are fuzzy rather than nonfuzzy sets, as long as the meanings of A and B are precisely defined as specific subsets of the universe of discourse.

In [Zadeh73] the idea of 'hedges' is also introduced. The idea 'hedges' consists of two basic concepts: (i) primary terms such as 'young' and 'old'; (ii) 'modifiers' of the primary terms such as 'very', 'much', 'slightly', 'more or less', and so on. In terms of the membership function curves, if x = 'old' then, for example, x^2 = 'very old'. As a consequence, the derivative of the curve which represent 'old' becomes steeper with respect to the y-axis and shifts proportionately to the right on the x-axis, as can be seen in Figure 13. This is calculated by applying the increase of the order of the root to the right hand side of the equation of the derivative.

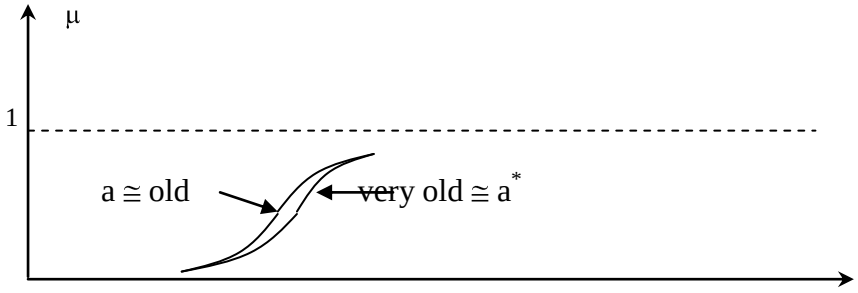


Figure 13. Effect of hedge 'very'

With respect to 'quantifiers', Yager [Yager88] makes reference to Zadeh's [Zadeh83] concept of linguistic quantifier. Yager applies this to multicriteria decision making to give a more profound interpretation of the weighting function 'W' associated with an aggregation operator 'F'. Zadeh said [Zadeh83] that quantifiers are of at least two kinds – those which say something about the number of elements and those which say something about the proportion of elements. Quantifiers can be represented a fuzzy subsets of either the unit interval or the real line. The distinction is based on whether the quantifier is related to an absolute or is a proportion type statement. It follows that if **Q** is relative to a quantity such as "most" then **Q** may be represented as a fuzzy subset of I such that for each $r \in I$, $\mathbf{Q}(r)$ indicates the degree to which r portion of the objects satisfies the concept denoted by **Q**, as can be seen in Figure 14.

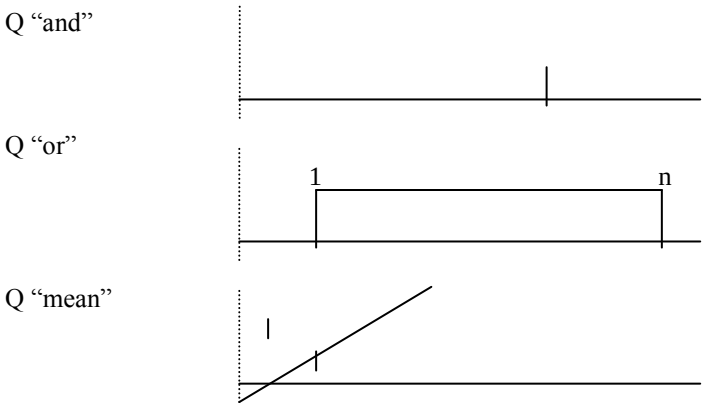


Figure 14. Three different quantifiers and their linguistic interpretation

The weights associated with the OWA function, described in Section 2.3, determine the kind of 'quantifier' it is effecting. By varying the assignment of the weights in W, we can move from a Min type "for all" quantifier, to a Max type "there exists" quantifier. Also, aggregations which emulate concepts like "most" can be represented. Yager explains how the degree of "andness" and "orness" may be measured.

2.2.3 T-norms, t-conorms and indistinguishability

The work of Jacas [Jacas88,90,93,95] includes the study of relations of indistinguishability and its application to classification processes, similarities, fuzzy equalities, and the study of properties and uses of t-norms and t-conorms. The basic elements are things such as fuzzy sets; membership functions, types of membership functions such as trapezoidal, triangular, and so on; union, intersection, complement; fuzzy relations; max-min and min-max products and transitivity.

With respect to t-norms and t-conorms, key aspects are Lings theorem, De Morgans terms, continuous t-norms and t-conorms, additive generators. In the case of classifications, relations and equivalence classes are considered, along with partitions, discrete pseudo distances, and fuzzy classifications, especially that of fuzzy c-Means, supervised and non supervised classifications. Fuzzy c-Means is described in Section 2.5 of the thesis.

In the case of relations of indistinguishability, the following are considered: t-indistinguishability, s-metrics, generation via Max-T and Min-S, generators, bases and dimension. In the case of t-indistinguishability: theorems of characterisation (e.g. Poincaré), similarities, algorithms which find the base. In the case of m-partitions,: classification with respect to a given metric, relation between m-partitions, t-indistinguishabilities and s-metrics.

Similarity is considered a generalisation of the notion of equivalence. A similarity relation has the following properties: it is reflexive, symmetric and transitive, where:

reflexive: $\mu s(x,x) = 1$

symmetric: $\mu s(x,y) = \mu s(y,x)$

transitive: $\mu s(x,z) > \vee(\mu s(x,y) \wedge \mu s(y,z))$

and

$\vee$ = max, and $\wedge$=min

The grade of membership

$\mu r(x,y) = 1$

is considered the strength of relation between x and y. A possibility is considered equivalent to a similarity which is equivalent to an indistinguishability.

T-norms

T-norms are a class of ordered topological semigroups in the unit interval. In 1942, Karl Menger introduced a probabilistic generalisation of metric spaces by replacing the real values d(p,q) by a probability distribution function F_{pq} . The main problem with the theory was how to generalise the classical triangle inequality. Menger analysed a relation of the form $F_{pr}(x+y)$ $T(F_{pq}(x), F_{qr}(y))$, where the function T from [0,1] × [0,1] into [0,1] was supposed to satisfy some special requirements:

(i) $T(a, b) = T(b, a)$
(ii) $T(a, b) \leq T(c, d)$ whenever $a \leq c$ and $b \leq d$
(iii) $T(a, 1) > 0$ whenever $a > 0$, and $T(1, 1) = 1$

In 1956 Berthold Schweizer and Abe Sklar rediscovered Menger's inequality and in 1960 published a paper where condition (iii) was replaced by the boundary condition:

(iii') $T(a, 1) = a$ for all a in [0,1] ,

and the associativity of T was also assumed:

(iv) $T(a, T(b, c)) = T(T(a, b), c)$.

Since the operations T satisfying (i), (ii), (iii) and (iv) were related to a class of triangle inequalities, they were named 'triangular norms', abbreviated to 't-norms'. Thus we have:

Definition 1. A two place function T from [0,1] × [0,1] into [0,1] is a t-norm if T satisfies the following conditions for all a,b,c,d in [0,1]:

(a) $T(a, 1)=a$;
(b) $T(a, b) \leq T(c, d)$ whenever $a \leq c$, $c \leq d$;
(c) $T(a, b) = T(b, a)$
(d) $T(a, T(b, c)) = T(T(a, b), c)$.

A **t-conorm** is analogous to a t-norm: an operation S of [0,1] × [0,1] in [0,1] is called a continuous t-conorm if it satisfies the following properties:

(i) associative: S(x,S(y,z) = S(S(x,y),z);
(ii) monotonous: if x ≥ x' then S(x,y) ≤ S(x', y);
if y ≥ y' the S(x,y) ≤ S(x,y');
(iii) conditions of contorn: S(x,0) = S(0,x) = x;
(iv) continuity: S is continuous as a function of two variables

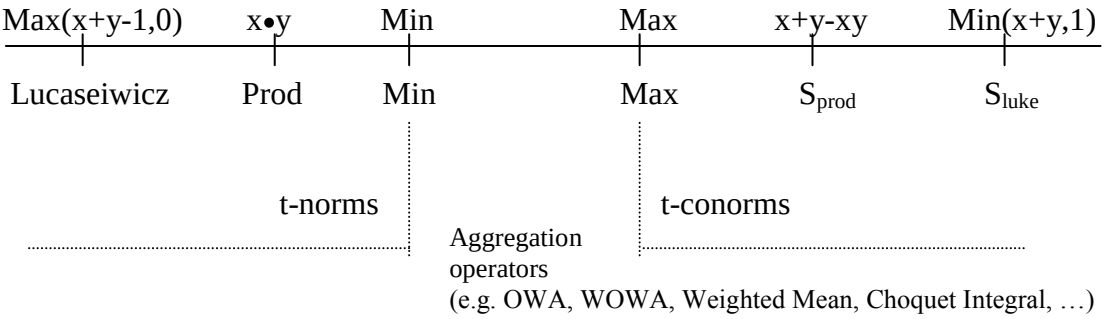


Figure 15. T-norms and t-conorms are a special type of topological semigroup ordered in the real index.

With reference to Figure 15, aggregation operators, such as OWA and WOWA, could be considered to exist in the range between 'Min' and 'Max', with the following conditions: (i) Min $a_i \leq C(a_1, \ldots a_n) \leq$ Max a_i; (ii) C(a, …., a) = a. Aggregation operators are covered in detail later in Section 2.3.

In [Jacas88] the structure of the generator's set of a t-indistinguishability operator is analysed. A suitable characterisation of such generators is given. T-indistinguishability operators generated by a single fuzzy set, in the sense of the representation theorem, are studied.

The following is the representation theory of a t-indistinguishability: a map *E* from $X \times X$ into [0,1] is a T-indistinguishability operator if, and only if there exists a family $\{h_j\}_{j \in J}$ of fuzzy subsets of *X* such that

$$E(x, y) = \inf_{j \in J} \hat{T} (\max(hj(x),hj(y))|\min(hj(x),hj(y)))$$

In [Jacas93] a general vision is presented of the concept of equality. The key aspect resides in achieving a good generalisation of the concept of transitivity which allows us to include different approximations of the idea of equality present in different branches of knowledge (physics, psychology, social sciences, fuzzy set theory and information theory) in a widened model. The definition of fuzzy equality presented allows a formulation 'more realistic' of the equality between objects which is shown to have a duality with the concept of distance.

One of the key ideas consists of defining a distance in a set X as an application m which assigns a positive number m(x, y) to each pair of elements (x, y) of X, which satisfies:

(i) m(x, x) = 0
(ii) m(x, y) = m(y, x)
(iii) m(x, y) + m(y, z) ≥ m(x, z) {triangular property}

An *indistinguishability operator* in a set X is an application $E: X \times X \rightarrow L$ such that

(i) $E(x, x) \geq \lambda$
(ii) E(x, y) = E(y, x)
(iii) E(x, y) * E(y, z) ≤ E(x, z)

Irrespective of what x, y, z of X are, and assuming that (L, ≤) is a partially ordered set (L, *) is a semigroup and λ is a distinguished element of L.

In [Jacas95], the set of generators of a generalised equality relation (T-indistinguishability operator) is studied. This set is identified with the set of the eigenvectors of the relation. The relation between the fuzzy and 'metric' topologies derived from these equalities is established. The concept of basis is introduced and the construction of a procedure is proposed in order to explicitly calculate the basis of a T-indistinguishability operator, for T archimedean.

The following is the algorithm to calculate a basis of T-indistinguishability E on a finite set X (# X = n) for T = Π or T = L.

(i) calculate the edges of the set H_E .
(ii) *count* = 1
(iii) build a set A obtained by taking a generator from each edge of H_E
(iv) define B(*count*) = the set of subsets of A of *count* elements.
(v) select a set H of B(*count*) and generate the relation E_H .
(vi) if E_H = E then stop
(vii) do step (v) and step (vi) for all different elements of B(*count*) .
(viii) *count* = *count* + 1. Go to step (iv) .

We observe that the elements of a preceding basis belong to different edges and since the number of edges is finite, a method can be derived to calculate a basis of E.

2.2.4 Fuzzy data representation

We now consider different aspects of fuzzy data representation, including the representation of fuzzy linguistic labels, binary variables, and the homogeneous fuzzy representation of variables of different types

Heterogeneous representation for fuzzy data

Hathaway and Bezdek proposed, in [Hathaway96], the following scheme for representing any type of data (including fuzzy) in the same scheme. Figure 16 shows the four kinds of symmetrical trapezoidal fuzzy numbers (STFN) which are considered: real numbers, intervals, and symmetrical triangular and trapezoidal fuzzy numbers, represented in the figure as $m_a(x:\mathbf{e})$, $m_a(x:\mathbf{f})$, $m_a(x:\mathbf{g})$, and $m_a(x:\mathbf{h})$, respectively. The notation, $m_a(x:a_1, a_2, a_3) = m_a(x:\mathbf{a})$ is used, where $\mathbf{a} = (a_1, a_2, a_3)$ is the vector of parameters that specifies **m** in the chosen representation. $\mathbf{m_a}$ defines the standard representation of an STFN, where a_1 is the *centre*, a_2 is the *inner radius* and a_3 is the *outer radius* of the structure specified by $m_a(x:\mathbf{a})$.

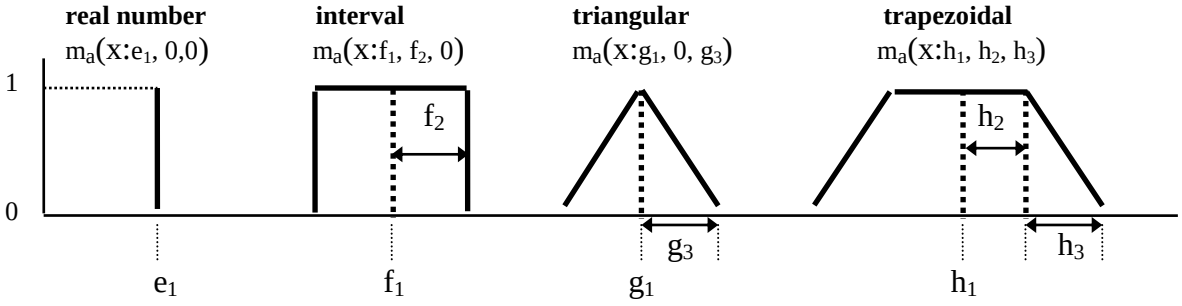


Figure 16. Representation of real, interval, triangular and trapezoidal fuzzy variables with symmetrical forms

With reference to Figure 16, the definition of a real number, for example, 1.1, using this scheme would be $m_a(x:1.1, 0.0, 0.0)$. In a similar manner, the definition of an interval would be $m_a(x:1.5, 1.0, 0.0)$. Note that because the scheme applicable to all four forms, one or more elements in the vector of parameters may be redundant and is assigned 0 in this case. This representation method is very flexible, and we will see in Section 3.1.2 of the thesis, how this idea is generalised to include parmenidean pairs, and the data format is extended.

Representation of Fuzzy linguistic labels

Linguistic labels, such as 'high', 'low', 'medium', 'strong', 'weak', are one of the prime targets for fuzzy representation, given their inherent imprecision and context dependency. The areas of investigation range from those based on linguistic theory, to geometrical descriptions, and from those which use neural networks for processing, to those which use rule based systems.

Baldwin's work [Baldwin95] deals with the modelling of words using Cartesian granule features (CGF). A 'Cartesian granule' is a collection of 'words', each 'word' being represented by a fuzzy set, and a T-Norm 'min' or 'product' being applied. One of the objectives of this approach was to demonstrate an improvement over other approaches.

The process is as follows: first, the data is pre-processed with a Kohonen net or with Fuzzy c-Means. Then the following process is applied: (i) extract CGF from data by induction; (ii) choose which features to use; (iii) create linguistic partition; (iv) then add features one by one. Figure 17 shows the entity relationship between Cartesian Granules, Words and Fuzzy Sets.

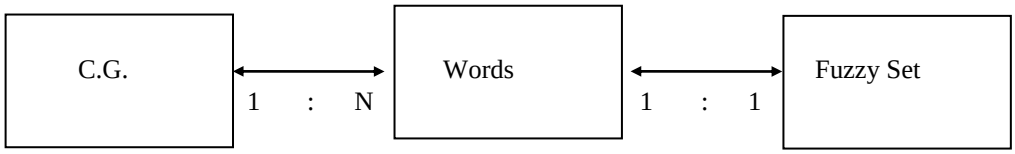


Figure 17. Entity relationship between 'Cartesian granules', 'words' and 'fuzzy sets'.

The algorithm uses a weight measure to represent the discriminant power of the features. The real application to which the method is applied is that of image recognition of a street with a couple of parked cars - 83% of the image area was correctly classified. There were 28 input features (unseen data), which were less features than those needed by a Neural Network.

Table 4. Properties of a feature

intensity.....
centroid.....
X,Y dimensions.....
Texture.....
Green-Blue measure (G-B).....
Yellow-Green measure (Y-G).....
Red-Blue measure (R-B).....

In summary, the work of Baldwin, 'feature and granularity selection', is based on the selection of features by the use of a semantic discriminant analysis. One of the key criteria applied is to look for features with a good class separation. In Figure 18a, feature A is comprised of three fuzzy classes, which could be 'low', 'medium' and 'high', for example. Each hemisphere in the figure represents a fuzzy class, and we observe that there is little overlap. This means that for feature A, each of the three fuzzy classes which comprise it are mainly distinct from one another. On the other hand, in Figure 18b, we see that for a given feature B, the three hemispheres overlap to a greater extent, and therefore there is a proportionately reduced distinction between the corresponding fuzzy classes. We would conclude that feature A has a better class separation, and is therefore has greater discriminative power than feature B. The system was developed in FRIL, C++, Java, and SNNS (neural network software).

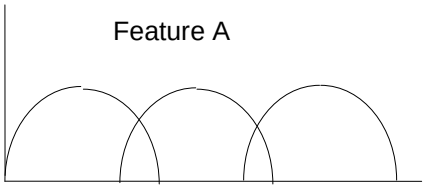


Figure 18a. Example of good separation of fuzzy sets

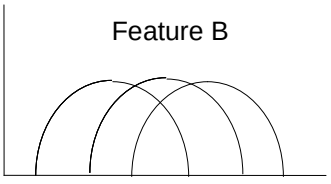


Figure 18b. Example of bad separation of fuzzy sets

On the fuzzy representation of binary type variables

Nominal (binary) variable. (respiratory failure{yes,no}, Duration of stay in ICU for 24 hours or more{yes,no})

This could be considered as a special case of nominal variables: In clinical records, such as the ICU dataset, many values are often the binary type, with a 1 or 0 response to clinical states and states of concentrations, presences, durations, and so on. Bezdek [Bezdek81], pp86, elaborated a method specially for attributes which assume binary values in medical data. The details of the method are extensively detailed in [Bezdek81]. The following depicts the table which gives the fuzzy cluster centres for each of the 11 variables considered. Fuzzy c-Means calculates these centres, together with the membership grades of each patient for each value of the binary variable (Hernia).

Table 5. Prototypes for Membership functions

Symptom	(Hernia) ν_{1j}	(Gallstone) ν_{2j}	Absolute differences, $\mathbf{f}_{12,\,j}$
1	0.57	0.27	0.30
2	0.98	0.67	0.31
3	0.06	0.93	0.87
4	0.22	0.55	0.33
5	0.17	0.10	0.07
6	0.77	0.84	0.07
7	0.42	0.05	0.37
8	0.39	0.84	0.45
9	0.48	0.04	0.44
10	0.02	0.16	0.14
11	0.12	0.25	0.13

In Table 5 (above) we observe that attribute (symptom) 2 is the most significant (as identified by fuzzy c-Means) for *Hernia*, while attribute 3 is the most significant for *Gallstone*.

All the patients have Hernia or Gallstone. Thus we have a data set which should fall into two clusters, in a binary fashion. Nevertheless, it is not completely clear how to assure that with the number of clusters assigned to two, fuzzy c-Means is able to identify which is the flag attribute (hernia={yes,no}) which distinguishes the groups, and not any of the other 11 attributes (or permutation groupings of the other 11 attributes). It would seem reasonable that the values represent the weighted mean of all the cases, for each variable.

The method detailed assumes that all the attributes are in a binary form (not only the possible classes, c being equal to 2). Thus, for variables of different types we cannot use this method, being restricted to subsets of the attributes which are binary. Nevertheless, the interpretation of the membership grades is by the standard fuzzy c-Means algorithm, and we may use this for all types of variables.

In [Bezdek81], pp86, the selection of variables from binary data is considered, for principal medical symptoms.

A method is developed for selecting attributes for *binary data*, based on *diffuse prototypes* (fuzzy) $\{\nu_i\}$ derived from the fuzzy c-Means algorithms. For the demonstration, data set $\mathbf{X}$ must possess binary values in each attribute; each $\mathbf{K}_j$ is the set $\{0,1\}$, and for ρ attributes, we have:

$$\mathbf{X} \subset (\{0,1\} \times \{0,1\} \times ..... \times \{0,1\}) = [\{0,1\}]^{\rho} \subset \Re^{\rho} \qquad (2.33)$$

It is assumed that $\chi_{kj} = 0$ or 1, respectively, that represents if "patient" χ_k has or does not have symptom j . In general, 0 (= absent) and 1 (= present) are attributes observed in many applications, and the method which is detailed as follows is pertinent to them, given that the objective is quite intuitive:

> A doctor collects ρ responses to clinical questions. Which of the questions (which of the attributes measured) of patient **k** allows the doctor to make a correct diagnosis ? Are there redundant attributes? Confusing

symptoms? Too much data? Insufficient data? To summarise, we are looking for a means of identifying the "best" attributes (features) for medical diagnosis [Bezdek81].

With reference to Section 4.1, we can use this representation for attributes such as *'Increase in Creatinine'* and *'Vital state on leaving the ICU'.*

Non-linear membership functions

The generation of membership functions is also related to quantifiers which interpolate a set of points to form a 'continuous' curve. There are many functions we can choose to generate a curve. Notwithstanding, we are interesting in functions whose parameters we can control in order to model linguistic labels and phrases. To this end, we consider 'hedges' as auxiliary qualifiers to linguistic labels which strengthen or weaken the initial concept. An example of a hedge would be the use of 'very' to strengthen the concept 'cold'.

Zadeh's S-Function is an example of an adequate function with which 'hedges' can be applied to non-linear membership functions. It is defined as follows:

$$S(x;\alpha,\beta,\gamma) = \begin{cases} 0 & x \le \alpha \\ \left(\frac{x-\alpha}{\beta-\alpha}\right)^2 & \alpha < x \le \beta \\ \left(\frac{x-\beta}{\gamma-\beta}\right)^2 & \beta < x \le \gamma \\ 1 & \gamma < x \end{cases} \qquad (2.34)$$

Now

$$f(x) = \begin{cases} \frac{1 + \sqrt[3]{(x-1/2)}}{2 \quad 2\times\sqrt[2]{(1/2)}} & x > ½ \\ \frac{1 - \sqrt[3]{(1/2-x)}}{2 \quad 2\times\sqrt[2]{(1/2)}} & x \le 1/2 \end{cases} \qquad (2.35)$$

The use of $f(S(x;\alpha,\beta,\gamma))$ increases all the membership values above 0.5, and decreases all the others. This is the definition for "very"; for "extremely" we can replace in formula 2 the 3rd root by the nth root (for a suitable $n > 3$, n odd).

Graphical display of fuzzy memberships

Kaufman and Rousseeuw in [Kaufman90], pp195,discuss ways of representing fuzzy memberships graphically. Lists of membership coefficients are often produced as output by programs and do not lend to an easy interpretation. In [Rousseeuw89] a method was proposed for computing the principal components of the membership coefficients. This involved simply applying a standard principal components program such as that found in SPSS or SAS, to the memberships, in the same way that it is usually applied to measurements. The number of nondegenerate principal components is the number of fuzzy clusters minus 1, given that the sum of the memberships is constant for each object.

As an example, Kaufman & Rousseeuw [Kaufman90] applied their fuzzy analysis program FANNY to a 'countries' dataset with *c*=3 and 12 countries. Given there were 3 fuzzy clusters, 2 principal components were obtained, as can be seen in Figure 19. The vertical component was interpreted as the countries political orientation while the horizontal component seemed to correspond to the degree of industrialisation. Egypt seems to hold a more intermediate and ambiguous position. Note that the criteria for plotting the positions is in the historical context taken at the end of the 1980's !

For two fuzzy clusters, the membership u_{i1} may be plotted for each object in the first cluster, then the membership in the second cluster can be read from right to left as $u_{i2} = 1 - u_{i1}$.

For three fuzzy clusters, each object has memberships (u_{i1}, u_{i2}, u_{i3}) and the possible combinations fill an equilateral triangle in 3-D space. Or we can use principal components to recover the triangle, or plot the memberships using barycentric (or trilinear) co-ordinates.

For more than three clusters, the two components may be shown which have the largest eigenvectors, thus explaining the largest portion of the variability. Alternatively a 3-D plot can be made or 2-D plot of different pairs of principal components.

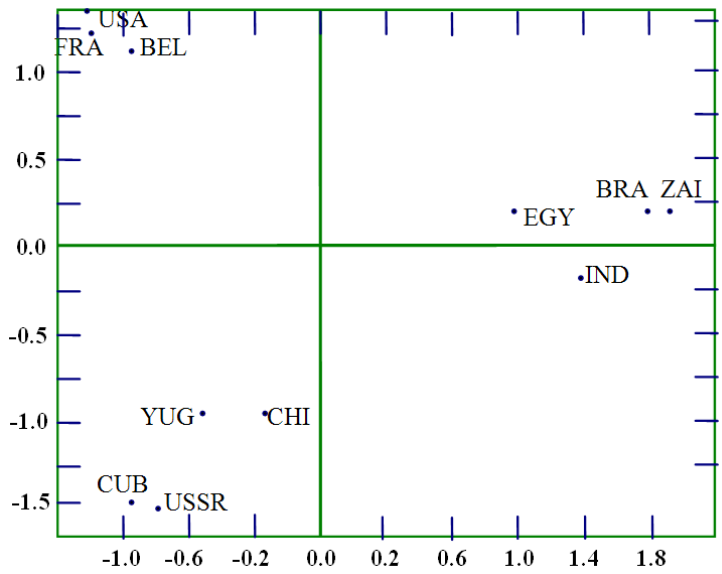


Figure 19. Principal components of memberships of 12 countries in three fuzzy clusters

2.2.5 Fuzzy data analysis

We now consider work relating to different aspects of fuzzy data analysis, including fuzzy clustering, fuzzy data modelling, fuzzy neural modelling, fuzzy rule induction and fuzzy factorial analysis.

Bezdek is a second key author (after Zadeh) in the fuzzy clustering field and is best known for his work related to the fuzzy c-Means family of clustering algorithms. In [Bezdek81], a feature selection scheme is detailed for medical binary data based on fuzzy prototypes $\{v_i\}$ derived from the fuzzy c-Means algorithms. This is interesting because medical diagnosis often involves many input variables which are of a nominal binary type, defined by a state or condition. One example would be the attribute *'respiratory failure'* which has two possible values, {yes,no}; a second example would be *'duration of stay in ICU for 24 hours or more'*, also with two possible response values, {yes,no}.
Bezdek defined a measure to establish the features which possess optimal discriminatory power for interclass separation. For $i \neq j$, where j are the different symptoms and i are the possible diagnoses, he defines a 'separation vector':

$$\mathbf{f}_{ij} = (|v_{i1} - v_{j1}| , |v_{i2} - v_{j2}| , \ldots , |v_{ip} - v_{jp}|)$$

Bezdek follows with the definition of a plausible heuristic for ranking features k as discriminators of classes i and j by ordering them by the components of $\mathbf{f}_{ij}$. If c, the predefined number of clusters given to fuzzy c-Means is equal to 2, the procedure is direct; for $c > 2$ the average of $\mathbf{f}_{ij,k}$ is calculated over the $c(c\text{-}1)/2$ pairs (i, j) with $i \neq j$. This gives an overall average efficiency for feature k.

$$\mathbf{f}_k = (2/(c)(c\text{-}1)) \sum_{i=1}^{c-1} \sum_{j=i+1}^{c} \mathbf{f}_{ij,k}$$

$\mathbf{f}_k$ then measures the relative ability of feature k for the interclass separation over all the distinct pairwise fuzzy clusters in the binary valued data, X. Optimal features are then selected by ordering the $\{f_k | 1 \leq k \leq p\}$, where p is each individual response.

Also in [Bezdek81], a method for obtaining 'shape descriptions' is given, with fuzzy covariance matrices. The shape of the cluster (in a two dimensional feature space) is defined by the norm. Thus if the norm can be varied locally, this makes it possible to alter the shape of individual clusters in the same feature space, thus more truly representing the underlying characteristics of the dataset. The mechanism to achieve this includes the use of the fuzzy covariance matrix defined by Gustafson and Kessel in [Gustafson79], which each fuzzy covariance matrix induces a different norm.

The sample covariance matrix measures the covariance of each sample with respect to each cluster prototype:

$$C_i = \sum_{x_k \in u_{ig}} (x_k - v_i)(x_k - v_i)^T / n_i$$

The memberships are effectively distributed to minimise the overall "fuzzy scatter volume" of the *c* fuzzy clusters

The number $(x_k - v_i)^T C_i^{-1} (x_k - v_i)$ is the squared Mahalonobis distance between $x_k \in u_i$ and its sub sample mean v_i, C_i^{-1} being the inverse of the sample covariance matrix of the points in u_i.

The fuzzy c-Means algorithm modified to incorporate the fuzzy covariance calculation, and defined by Gustafson and Kessel in [Gustafson79], is as follows:

Step 1: assign c, $2 \le c \le n$, assign m $\in(1,\infty)$, assign c volume constraints $\rho_j \in (0,\infty)$, $1 \le j \le c$. Initialise $U^{(0)} \in M_{fc}$. And so on at step l, l=0,1,2, …
Step 2: calculate the c fuzzy cluster centres $\{v_i^{(l)}\}$.
Step 3: calculate the c fuzzy scatter matrices $\{S_{fi}^{(l)}\}$. Calculate their determinants and their inverses.
Step 4: calculate the norm-inducing matrices $\{Aj^{(l)}\}$.
Step 5: update $U^{(l)}$ to $U^{(l+1)}$. Distance $d_{ik}^{(l)} = \|x_k - v_i^{(l)}\|_{Ai}$; $1 \le i \le c$, $1 \le k \le n$.
Step 6: compare $U^{(l)}$ to $U^{(l+1)}$ in a convenient matrix norm: if $\| U^{(l+1)} - U^{(l)} \| \le \varepsilon_L$, stop. Otherwise return to Step 2 with l = l + 1.

In [Bezdek77] the fuzzy ISODATA algorithms are used to address: (i) feature selection for binary valued data sets; (ii) the design of a fuzzy one-nearest prototype classifier. Feature selection has already been covered in the notes on [Bezdek81]. In the case of prototype classification, an average classifier performance of 62% was reported using fuzzy 1-NP classification for patients known to have one of the six stomach disorders under consideration. The objective is that the cluster centres are good classifiers, and the method is compared with k-nearest neighbour classifiers.

Fuzzy clustering with weighting of data variables

[Keller00] considers fuzzy clustering with weighting of data variables. An objective function-based fuzzy clustering technique assigns one influence parameter to each single data variable for each cluster. The concept consists of weighting single attributes for each cluster using a distance measure in which the distance between a datum x_k and a cluster (vector) v_i is defined by

$$d^2(v_i, x_k) = \sum_{s=1}^{p} \alpha_{is}^t \cdot (x_k^{(s)} - v_i^{(s)})^2$$

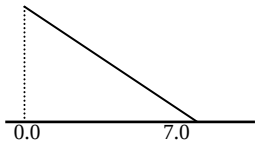
$x_k^{(s)}$ and $v_i^{(s)}$ indicate the sth coordinates of the vectors x_k and v_i, respectively. The number of variables or attributes is denoted by *p*. α_{is}^t is a parameter determining the influence of attribute (coordinate) *s* for cluster *i*. $t \in \Re_{>1}$ is a real-valued parameter which allows for the definition of the strongness of the emphasis that is put on the attribute weighting task.

As an example of how the influence parameter works, consider a partition of four clusters: for cluster 2, the attribute influence parameters α_{is} have nearly the same value, while the data co-ordinates are approximately uniformly distributed for the two domains of the cluster. For clusters 3 and 4, the data values for attribute *x* are scattered widely whereas the values for attribute *y* have a small range – thus the influence parameters α_{ix} are small in comparison to α_{iy} for clusters 3 and 4. In the case of cluster 1 the data values for attribute *y* are scattered widely, resulting in a high value for influence parameter α_{ix}.

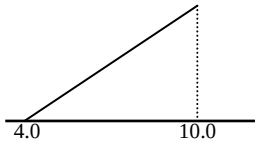
Fuzzy data modelling

The Sugeno-Takagi fuzzy model [Takagi85] determines an optimal structure by using a criterion chosen by cross validation. With respect to initialisation, a supervised Gustafson & Kessel algorithm is applied to the input-output space, in which compatible clusters are merged, and fuzzy rule antecedents are generated by the projection of the clusters into the input space. The algorithm splits a rule to determine the parameters of the children and the optimal antecedents (called priors). The results were benchmarked using [Platt91], and with a training set of 500 points it is easy to identify the point in which overfitting starts. One drawback of the method is the high consumption of processing time.

Rule models in [Takagi85] are constructed from rules defined in the following manner:



If *x* is **then *y* = 0.6*x* + 2**



If *x* is **then *y* = 0.2*x* + 9.0**

Figure 20. Example of Takagi-Sugeno fuzzy rule definitions

The rules in Figure 20 (above) represent a perfect (linear) distribution of the points. If noise is introduced into the dataset as can be seen in Figure 21 (below) where the points no longer lie perfectly along the line, but are dispersed as in a real dataset, the method is able to adapt to it by altering the parametric values in the rules.

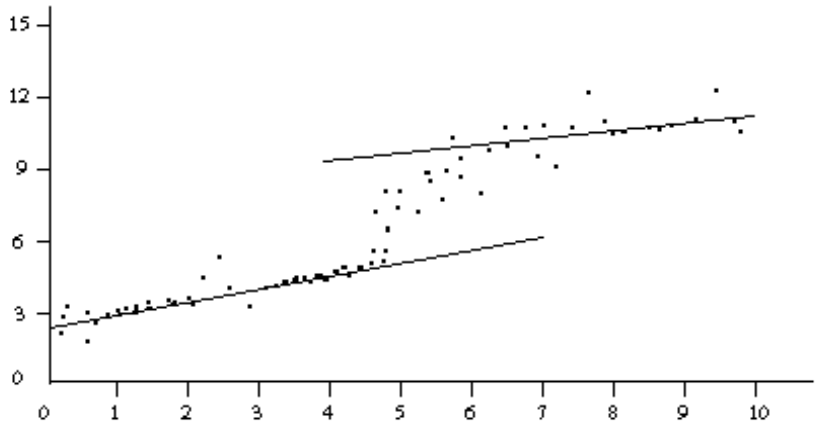


Figure 21. The effect of noise data on fuzzy rules

When the premise parameters are derived again from the new data, we see some small but significant changes to the coefficients, as illustrated in Figure 22, which model the dataset of Figure 21.

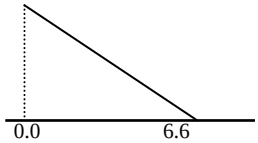


Figure 22. Example of Takagi-Sugeno fuzzy rule definitions after the addition of noise to the data

A Fuzzy Q-Learning method [Glorennec94] uses agents to learn about the input space. There are three variants: discrete action space, knowledge extraction and knowledge fusion. A predefined number of agents compete against each other to control the system. One of the objectives is that of knowledge fusion from several experts, a similar objective to aggregation algorithms. Learning involves a knowledge extraction phase followed by a knowledge integration phase. In order to reach a certain type of behaviour a reward-punishment tactic is used, to reinforce the desired behaviour of the system. Rule based optimisation is shown to be an efficient process, and may be changed by expert. Actions are considered to be continuous, as opposed to discrete.

[Baldwin95]. The problem of representing attributes such as 'age' or 'hobbies' in a fuzzy form is considered from a semantic point of view. Binary relations are considered in which vagueness is defined as the crispness of an object, or that of not being sure in which set it is, that object having a membership to more than one set. 'Vagueness' is contrasted with 'uncertainty', which is considered as applying only to binary variables. A fuzzy object can be an 'uncertain object', or an 'incomplete object', which may have a subjective or objective interpretation. It is stated that an object may be at the same time 'uncertain' and 'incomplete'. The purpose of fuzzy types is considered: they can be used in the case of an incomplete specification, can permit vagueness in the case of ranges and default values, and can interpret subjective as opposed to objective definitions. It was stated that one of the objectives was to define the properties of 'type', as well as the meta-properties of 'type'.

[Bouaziz96] defines a set of trapezoidal 'linguistic types', which employ a squeezing mechanism to 'squeeze' the membership graph towards the origin, which helps to cope with 'data explosions'. The squeeze factor depends on events – a smaller squeeze implies a weaker reaction. The form of codification of the trapezoidal functions allows for the creation of new linguistic types'.

[Cordon97] in his work on 'new reasoning methods in a classification system based on fuzzy rules' considers fuzzy rules with grade of truth, which can be applied to fuzzy classes. [Delgado95] in his work on 'hybrid techniques for generating and tuning rules in fuzzy modelling' treats of the evolution of fuzzy rules from an initial state. The end result is susceptible to local minimum's and a lucky (or unlucky) first guess at the first rule to evolve. He mentions fuzzy fusion as being in two steps: (a) correlation of homogeneous values and (b) distances between membership grades. [Castro98], in his work on 'a methodology for SBC development' is partially inspired by MILORD [Sierra89]. He has an interesting methodology, which consists of the following: (a) extraction phase of relevant variables ; (b) selection of a representative set of examples; (c) knowledge acquisition phase; (d) validation phase; (e) verification phase. His test data was the Iris set, and the rule generation step produced three rules. A hierarchic control is used in the movement of the rules. If it doesn't arrive at a satisfactory result, it looks for a more general rule.

[Flores-Sintas97] tested the fuzzy-minimals algorithm using real data. We recall that fuzzy-minimals is one of the fuzzy c-Means developed by Bezdek. They have realised studies based on covariance matrices, employing a 'defuzzification' process to tackle a real problem posed by the 'Caja de Murcia', that of classifying the offices of the bank. A partition tree was generated , containing 9 groups of which 3 were 'spurious'. [López-García97] defined and studied some of the conditions for existence of 'Gastwirth type fuzzy inequality measures'. They have employed an interesting type of membership function based on hyperbolic indexes to represent 'relevance grade' responses of scientific-professional training. (very low, low, medium, high, very high, don't know/don't answer).

Fuzzy neural modelling

[Kim96] defines the following simple algorithm for a 'fuzzy neural classifier': (i) find nearest k neighbours of x; (ii) determine class of x by voting of all k neighbours and the majority wins. A 'condensing' phase is used to remove redundant (deeply embedded data) and to remove boundary data (that which is borderline between 2 clusters). An example of how the 'rule base' is defined is as follows:

IF **x** is close to (A_1, A_2, A_3) THEN class = **O**
IF **x** is close to (B_1, B_2, B_3) THEN class = χ

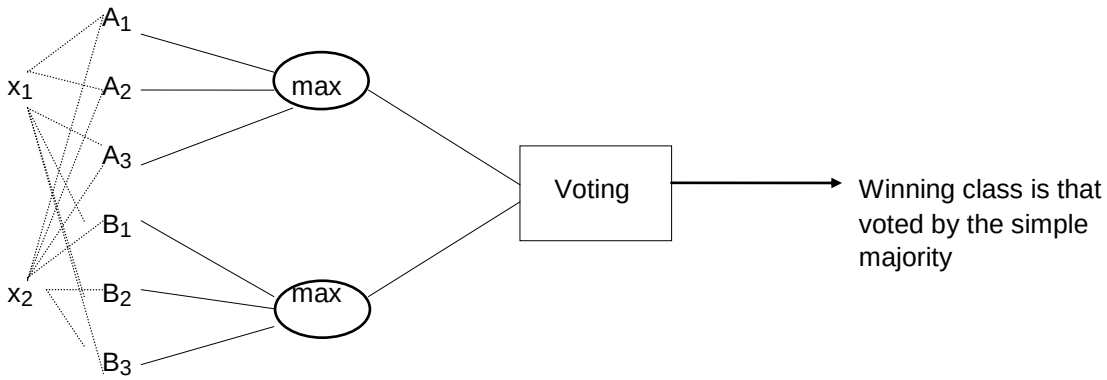


Figure 23. Example architecture of a fuzzy neural model

For testing, 578 test cases and 1063 training cases were used, with three clusters. A sample size of 497 was reduced to 64 by the 'condensing' phase. The sample size was further reduced to 60 after processing with Fuzzy c-Means to find a reduced sample set.

[Juang97] defines a recurrent self-organising neural fuzzy inference network which includes a time dimension. An example of one of the rules is as follows:

IF at '1' THEN have to predict '2'. IF at '2' THEN have to predict '3'.

This is depicted graphically in Figure 24 (below).

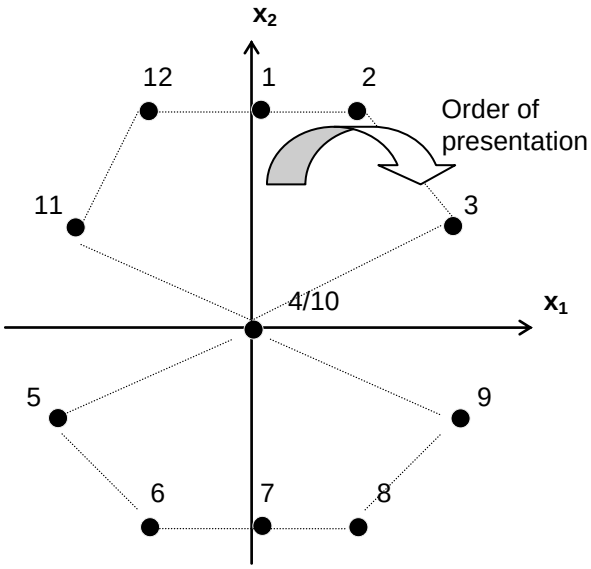


Figure 24. Graphical representation of the temporal dependencies of the tests a, b and c

A 'best trajectory' is achieved by their algorithm 'RSONFIN', which they claim is better than the trajectory found by a feedforward fuzzy neural network. The benefits of the model are that it is constructed from a recurrent net, has no preassignment, and has a small number of parameters.

Fuzzy rule induction

The Fuzzy Projection Pursuit method, called ID3* and developed by [Miyoshi97], unifies the Fuzzy ID3 approach of [Umano94] and the Projection Persuit approach of [Friedman74]. A brief description of the method is as follows:

Equation (1) - In a leaf node **b**, a sample data **x** belongs to a fuzzy set with membership

$$\mu_b = \prod_{(i,l) \in Q_b} \mu_{il} \qquad (2.36)$$

where μ_{il} is a membership function of the attribute value, (i, l) means the lth attribute value (fuzzy set) of the ith attribute and Q_b is a set of the pair (i, l) along the branches from the root to a node **b**.

Miyoshi [Myoshi97] proposed a method which generates a decision tree whose nodes consist of the projections of attribute values. For example, the data whose attributes are x_1 and x_2 is classified into two classes, B and S, as shown in Figure 25 (a). The standard ID3 algorithm has difficulty in obtaining a simple decision tree from the given data, but the projection persuit ID3 can produce a simple decision tree, as shown in Figure 25 (b), if an appropriate projection vector is given as defined in [Myoshi97].

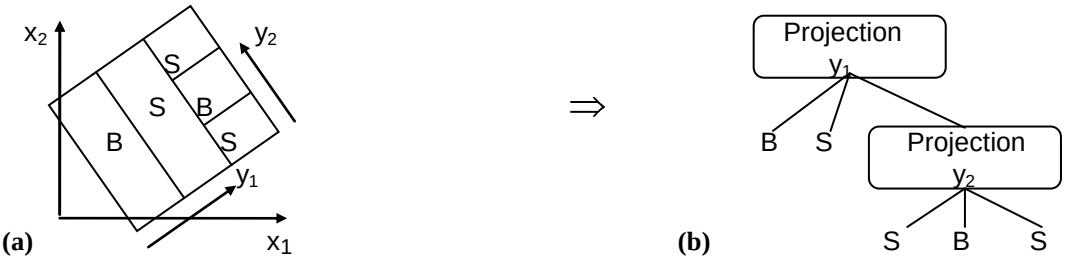


Figure 25. Example of projection persuit ID3: (a) fuzzy partitions; (b) decision tree corresponding to the fuzzy partition in (a)

[Branco94] considers methods for decomposing 'Sugeno rules', with the values of the variables taken as features. The resulting table is converted into a decision tree. For example:

Table 6. Classification of different geometrical shapes in terms of the number of sides and angles

	sides		angles=		
	3	4	2	3	4
Equilateral triangle	1	0	0	1	0
Isosceles triangle	1	0	1	0	0
Square	0	1	0	0	1
Rhombus	0	1	1	0	0

is converted to (next page):

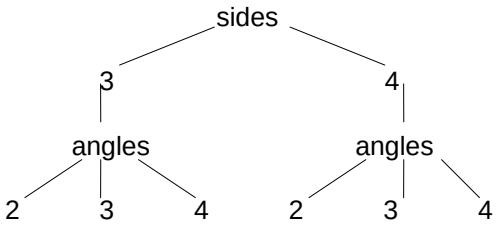


Figure 26. Tree representation of geometric features generated from Table 6.

A table such as that shown in Table 6, is constructed for the objects and features, where the objects in this example are geometric shapes (square, rhombus, …) and the features are 'sides' and 'angles'. A supervised learning method is then used to generate a tree representation, such as that seen in Figure 26. A learning set of 250 points was used, derived from PROBEN1, a collection of data sets for neural network learning.

The results showed that the classification error depends on the number of fuzzy sets, one drawback being that the method depends on the fuzzy partition. Trapezoidal and triangular fuzzy sets were used, and it combines rules to incorporate those elicited from the data and from the expert.

[Loutchmia97] defines a method for inductive learning using similarity measures on a lattice-fuzzy set. This employs a 'case based' approach for learning, and the 'lattice' approach is used to handle uncertainty and imprecision. The method has been tested on the 'sponge' and 'iris' datasets, and the distance measure used is as follows:

$$\delta(x, y) = \frac{\sum_{k=1}^{\rho} \mu_k \, d_b(x_k, y_k)}{\sum_{k=1}^{\rho} \mu_k} \qquad (2.37)$$

where the observations are described by a set of ρ attributes which are considered elements of a lattice T, μ_k is a weight associated to the attribute A_k, d_b is the bipartite distance, δ is the average bipartite distance and $x = (x_1, \ldots, x_\rho)$, $y = (y_1, \ldots, y_\rho)$ are two elements of the lattice T.

Fuzzy factorial analysis

[Inuiguchi97] introduces a 'mean-absolute deviation based fuzzy linear regression analysis', whose objective is to achieve automatic deduction from data. The motivations are that it is not based on minimising the deviations, and performs a minimum range estimation, without any initial parameters. Its strong points are that it avoids non-intuitive results, is non-parametric, and the regression is obtained by a sequential simplex method.

[Dubois97] describes a user-driven summarisation of data based on gradual rules. This consists of three main steps: (i) identify typical points along the data; (ii) compute cores of each output fuzzy set using 'convex hull' method; (iii) rule refinement. One problem which may occur in step three is that the core may be found to be too large, as can be seen in Figure 27 below. Also, the support of a fuzzy output may be too large. An example of a 'gradual rule', would be: the more X is A, the more Y is B .

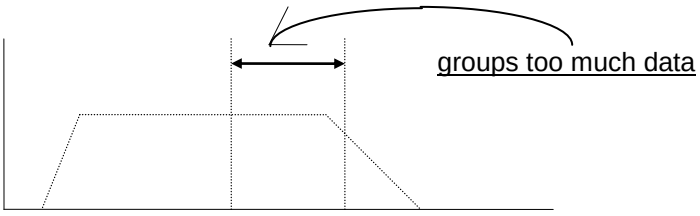


Figure 27. Illustration of generation of a core with excessive grouping of data

[Watada94] considers the processing of crisp data into fuzzy sets. As can be seen in Table 7, a fuzzy factor loading is considered as the correlation between a given principal component and a given attribute-variable.

Table 7. Definition of a fuzzy factor loading $r(Z_k, x_i)$ which is considered as the correlation between the Principal Component Z_k and the attribute x_i

	factor 1			factor 2			…….	factor N		
Item	L	C	U	L	C	U	.....	L	C	U
1 2 3 4 ..... N										

L=Lowerbound, C=Centre, U=Upperbound.

One of the more interesting references in the field of factor analysis for fuzzy data is that of [Nakamori97]. He states that one reason that factor analysis for fuzzy data has not been developed is the difficulty to calculate the second moment of fuzzy data given by interval fuzzy. Nakamori uses intervals (min, max) for the distances between attributes, in which the first step is to define a Fuzzy Correlation matrix. Nakamori´s proposal for fuzzy factor analysis uses a correlation matrix between measures $R = (r_{ij})$, and for subject *k* it is $R^k = \{r_{ij}^k\}$.

First the correlation matrices are computed based on the data of individual subjects. Then, the standard deviations σ_{ij} of (i, j)-elements of all correlation matrices are calculated. The variance of the correlation $\{r_{ij}^k\} \rightarrow \sigma_{ij}^2$

Now the following range is defined:

$$r_{ij}^L = \max\{-1, r_{ij} - \gamma\sigma_{ij}\},$$
$$r_{ij}^R = \min\{r_{ij} + \gamma\sigma_{ij}, +1\}, \quad (2.38)$$

where γ is the parameter indicating the degree of fuzziness. The fuzzy correlation matrix

$$\Re = (r_{ij}) = ([r_{ij}^L, r_{ij}^R]), \quad i, j = 1,2, \ldots, N, \quad (2.39)$$

Where $[r_{ij}^L, r_{ij}^R]$ denotes an interval fuzzy number. The fuzzy correlation matrix holds the relative fuzziness of correlation coefficients.

The fuzzy distance between two fuzzy objects O_i and O_j can be defined as a fuzzy number:

$$d_{ij} = [d_{ij}^L, d_{ij}^R] \quad (2.40)$$

First the minimum and maximum distances along each factor axis is computed, then the lower (L) and upper (R) distances are defined. The same mechanism for traditional factor analysis (loading matrix holding the loading factors,

rotation matrix, and correlation matrix) exist and implies that this method follows similar lines to the Hartigan joining algorithm.

In the case of fuzzy distance measures, [Loutchmia97] considers inductive learning using similarity measures, and refers to the ID3 [Quinlan86] algorithm and to fuzzy decision trees [Dubois91] [Okamoto94] [Zeidler96].

Selection of Characteristics

For the constitution of a 'data space' S, it is necessary to evaluate if the characteristics of data item $x_k \in \mathbf{X}$, are sufficiently representative of the physical process, to permit the construction of clusters which classify and which are realistic. It is necessary to evaluate if we possess the correct data space. ¿Is it necessary to eliminate some attributes from x_k, modify them, enrich them, transform them?

It is necessary to look for the 'internal' structure in the data, the objective being to improve its utility for the clustering process and / or classification in the data set. In the case of the *'hospital admissions'* data set (see Section 4.1), there are two considerations: **(i)** reduction of the number of attributes in order to only keep the most significant ones, with respect to a key 'objective' attribute, for example, *'duration of stay in hospital in days'*; **(ii)** fuse the most significant attributes in two or three 'super-attributes', thus allowing data analysis in these dimensions. We assume that these two aspects will be carried out separately and with different techniques.

2.2.6 Fuzzy covariances

In this section we introduce some concepts and definitions of fuzzy covariances, which are considered in more detail in Section 3.1 of the thesis. In the literature, the term fuzzy covariance is often used to describe different things, and it is difficult to establish a strict definition. In conceptual terms, [Gustafson79] understands fuzzy covariance as the covariance between a fuzzy instance in a dataset and the centroid of a corresponding fuzzy cluster. The formal definition of this is given below. Other authors, such as [Nakamori97] or [Watada94] define application specific fuzzy covariance calculations. The definitions we later make in Section 3.1 of the thesis in order to define a fuzzy covariance which defines the covariance between two variables defined in the fuzzy form, are derived from the definition given by [Gustafson79].

Derivation of the Gustafson & Kessel fuzzy covariance matrix [Gustafson79]

Covariance matrix: the Gustafson and Kessel algorithm, as detailed in [Bezdek81,pp168], takes as starting point a 'simple' data set, and is based on a modified version of fuzzy c-Means.

The family of functions $\{J_m \mid 1 \le m < \infty\}$is considered, where J_m is an infinite family of fuzzy clustering algorithms, based on a least-squared error criterion. M_{fc} is a fuzzy c-partition [Bezdek81, pp26], and $\Re^{cp}$ is a real p-dimensional vector space, for c fuzzy partitions[Bezdek81, pp48],. Let $J_m : M_{fc} \times \Re^{cp} \rightarrow \Re^{+}$ be

$$J_m(U, \boldsymbol{\nu}) = \sum_{k=1}^{n} \sum_{i=1}^{c} (u_{ik})^m (\mathbf{d}_{ik})^2 \qquad (2.41)$$

where

$$U \in M_{fc}$$

is a fuzzy c-partition of X;

$$\boldsymbol{\nu} = (\boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \ldots, \boldsymbol{\nu}_c) \in \Re^{cp}, \text{ with } \boldsymbol{\nu}_i \in \Re^{p}$$

is the cluster centre or prototype of u_i, $1 \le i \le c$;

$$(d_{ik})^2 = \| \mathbf{x}_k - \boldsymbol{\nu}_i \|^2 \quad \text{and} \quad \| \cdot \|$$

is any inner product induced norm on $\Re^p$; and weighting exponent $m \in [1, \infty)$

Examination of J_m reveals that the dissimilarity measure is $d_{ik} = \| \mathbf{x}_k - \boldsymbol{\nu}_i \|$, which represents the distance between each data point $\mathbf{x}_k$ and the fuzzy prototype v_i ; then the squared distance is weighted by $(u_{ik})^m = (u_i (\mathbf{x}_k))^m$, the mth square of the membership of $\mathbf{x}_k$ in fuzzy cluster u_i. Given that each term of J_m is proportional to $(d_{ik})^2$, J_m is a squared error clustering criteria, and solutions of

$$\begin{array}{l} \text{minimise } \{ J_m (U, \boldsymbol{\nu})\} \\ M_{fc} \times R^{CP} \end{array} \qquad (2.42)$$

are minimum squared error stationary points of J_m [Bezdek81,pp66].

Bezdek defines the following theorem 2.43, that allows the calculation of the fuzzy cluster centres (prototypes).

The conditions are valid for any norm-metric induced on the interior product. It follows that any positive defined matrix $A \varepsilon V_{pp}$ induces this norm via the weighted interior product :

$$\langle \mathbf{x}, \mathbf{y} \rangle_A = \mathbf{x}^T \mathbf{A} \mathbf{y} = \sum_{i=1}^{\rho} \sum_{j=1}^{\rho} \mathbf{x}_i \, a_{ij} \, y_j \qquad (2.43)$$

$\forall_{\mathbf{x}}, \mathbf{y} \varepsilon \Re^p$. With respect to this special class of norms, we may write J_m as:

$$J_m (U, \boldsymbol{\nu}, A) = \sum_{k=1}^{n} \sum_{i=1}^{c} (u_{ik})^m \| \mathbf{x}_k - \boldsymbol{\nu}_i \|^2_A \qquad (2.44)$$

where

$$(d_{ik})^2 = \| \mathbf{x}_k - \boldsymbol{\nu}_i \|^2_A = \langle \mathbf{x}_k - \boldsymbol{\nu}_i , \mathbf{x}_k - \boldsymbol{\nu}_i \rangle_A = (\mathbf{x}_k - \boldsymbol{\nu}_i)^T A (\mathbf{x}_k - \boldsymbol{\nu}_i)$$

This form makes an emphasis on the dependence of J_m on the matrix **A**, defining the norm for R^P through (2.43). There are two reasons for doing this:

(i) Under certain special conditions, **A** may be included as *theoretical* variable for optimisation, as in the modification made by Gustafson and Kessel.

(ii) In any case, **A** is an *algorithmic* variable, for the fuzzy c-Means methods.

A popular idea often associated with fuzzy sets is their "core". The usual definition involves a threshold, for example α , to identify the core.

Definition 1. (α-Level-Set). Let u_i be a fuzzy subset of **X**. The α-core or α-level set of **X** derived from u_i at each $\alpha \in [0, 1]$ is the hard set

$$C(u_i ; \alpha) = \{x \in \mathbf{X} \mid u_i (x) > \alpha\} \qquad (2.45)$$

Note that u_i is its own core for every α in case u_i is hard.

The core of a fuzzy set provides a different way to compare c-partitions of data.

Definition 2 (Fuzzy Scatter Matrix). Assume $m \in (1, \infty)$; $\mathbf{X} = \{ x_1 , x_2 ,..., x_n\} \subset \Re^p$; and $(U, \mathbf{v}) \subset M_{fc} \times \Re^{cp}$. The following is defined:

Fuzzy centroid or prototype of cluster u_i :

$$\mathbf{v}_i = \frac{\sum_{k=1}^{n} (u_{ik})^m \mathbf{x}_k}{\sum_{k=1}^{n} (u_{ik})^m} \tag{2.46}$$

Fuzzy scatter matrix of cluster u_i :

$$S_{fi} = \sum_{k=1}^{n} (u_{ik})^m (\mathbf{x}_k - \mathbf{v}_i)(\mathbf{x}_k - \mathbf{v}_i)^T \tag{2.47}$$

Fuzzy within-cluster scatter matrix:

$$S_{fW} = \sum_{i=1}^{c} S_{fi} \tag{2.48}$$

The fuzzy scatter matrices $\{ S_{fi} \}$ at (2.47) arise naturally in the generalisation of $\mathbf{J}_m$ detailed in the fuzzy "covariance" algorithm of Gustafson and Kessel.

Evidently, (2.48) is a *generalised* minimum variance partitioning problem. There is a statistical interpretation to the criterion $\mathbf{J}_m$ which is entirely analogous to that enjoyed by $\mathbf{J}_1$ as long as the measure of dissimilarity is Euclidean.

Summary

Now we reach the 'objective', which is the fuzzy covariance matrix itself, having seen the definition of $\mathbf{J}_m$ as the squared error clustering criteria, and **A** as the positive defined matrix. **A** may be included as a *theoretical* variable to be optimised, as in the modification given by Gustafson and Kessel.

Solutions of (2.48) can be regarded as solutions of generalised minimum variance partitioning problems by introducing a fuzzy extension of the hard scatter matrices of Wilks.

Fuzzy scatter matrices $\{ S_{fi} \}$ at (2.47) arise naturally in the generalisation of $\mathbf{J}_m$ detailed in the fuzzy "covariance" algorithm of Gustafson and Kessel.

The line of reasoning follows from the definition of fuzzy c-Means by Bezdek, followed by the modified version of fuzzy c-Means of Gustafson and Kessel, which uses the **A** matrix. It is followed by Bezdek's 'fuzzy scatter matrix' algorithm from which we reach the fuzzy covariance algorithm of Gustafson and Kessel.

Fuzzy covariance matrix

We assume that $m \in (1, \infty)$; $\mathbf{X} = \{ \mathbf{x}_1 , \mathbf{x}_2 ,...,\mathbf{x}_n\} \subset \Re^p$; y $(U, \mathbf{v}) \in M_{fc} \times \Re^{cp}$. The fuzzy covariance matrix of cluster u_i is:

$$C_{fi} = \frac{\sum_{k=1}^{n} (u_{ik})^m (\mathbf{x}_k - \mathbf{v}_i)(\mathbf{x}_k - \mathbf{v}_i)^T}{\sum_{k=1}^{n} (u_{ik})^m} = \frac{S_{fi}}{\sum_{k=1}^{n} (u_{ik})^m} \tag{2.49}$$

where C_{fi} is each element of the matrix, and we consider the class of interior product norms induced on $\Re^p$ (space of-tuples of real numbers) by symmetrical and positive matrices in $\mathbf{V}_{pp}$ (**V** is a vectorial space of p × p matrices). **X** is a data set which contains **n** clusters.

2.3 Aggregation

In this section, we consider data aggregation methods based on the use of weighting vectors to bias the data, with respect to relevance and reliability. Three methods are considered: principal components (PC), ordered weighted average (OWA), and weighted ordered weighted average(WOWA). These methods contrast different weighting factors; PC correlates the input variables in order to reduce the dimensionality in one or more factors, OWA weights the data values, and WOWA weights both the variables and the data values. Weighted mean (WM) is an aggregation technique which has as input a data vector and a weight vector. The weight vector contains one degree of reliability value between 0 and 1, for each corresponding variable.

A special focus is given to the WOWA data aggregation operator [Torra97a]. This operator is a hybrid of the weighted mean and the OWA Ordered Weighted Average operators, as described in [Torra97a]. The OWA operator was first introduced in [Yager88] and is one of the key references of a technique for aggregating data values, using a weighting vector to allow the introduction of a reliability factor for each value. The WM operator introduces a weighting vector for the variables, thus WOWA combines both approaches to allow two data vectors which weight the data values and the variables in the same pass. All the methods WM, OWA and WOWA are considered as Choquet Integrals; as is illustrated in Figure 28 (page 69), which illustrates that the Choquet Integral is a generalization of WM, OWA and WOWA.

2.3.1 Basic definitions

PC – Principal Components

PC is a standard statistical technique which correlates the input variables in order to reduce the dimensionality in one or more factors. In order to combine two variables into a single factor, we could summarise the correlation between the two variables in a scatterplot. A regression line could then be fitted which represents the *'best'* summary of the linear relationship between the variables. If we could define a variable that would approximate the regression line in such a plot, then that variable would capture most of the *'synthesis'* of the two items. The individual scores of cases on the new factor, represented by the regression line, could then be used in future data analyses to represent the *'synthesis'* of the two variables. In a sense we have reduced the two variables to one factor.

This example illustrates the basic idea of *factor analysis*, or of *principal components analysis*. If we extend the two-variable example to multiple variables, then the computational effort increases, but the basic principle of expressing two or more variables by a single factor remains the same.

The extraction of principal components can be summarised as a variance maximising (varimax) rotation of the original variable space. For example, in a scatterplot we can think of the regression line as the original X axis, rotated so that it approximates the regression line. This type of rotation is called variance maximising because the criterion for (goal of) the rotation is to maximise the variance (variability) of the "new" variable (factor), while minimising the variance around the new variable. When there are more than two variables, we can think of them as defining a "space," just as two variables defined a plane. Thus, when we have three variables, we could plot a three dimensional scatterplot and then a plane could again be fitted through the data.

OWA – Ordered Weighted Average

OWA is a data aggregation method which was originally defined in [Yager88]. Ordered Weighted Average has two vectors as input: a data vector and a weight vector. The weight vector contains two or more degree of relevance values between 0 and 1, which are used to interpret the data values. OWA permits an AND/OR effect on the data inputs, controlled by the relevance weights.

Definition: A mapping *F* from

$$I^n \rightarrow I \text{ (where } I = [0, 1])$$

is called an OWA operator of dimension n if associated with *F*, is a weighting vector ω,

$$\boldsymbol{\omega} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ ..... \\ \omega_n \end{bmatrix}$$

such that

1) $\omega_i \in (0,1)$
2) $\Sigma_i \omega_i = 1$

and where

$$F(a_1, a_2, ..., a_n) = \omega_1 b_1 + \omega_2 b_2 + ... + \omega_n b_n, \qquad (2.50)$$

where b_i is the ith largest element in the collection $a_1, a_2, ..., a_n$. **B** is called an ordered argument vector if each element $b_i \in [0,1]$ and $b_i \geq b_j$ if $j > i$. Given an OWA operator *F* with weight vector $\boldsymbol{\omega}$ and an argument tuple $(a_1, a_2, ..., a_n)$ we can associate with this tuple an ordered input vector **B** such that **B** is the vector consisting of the arguments of *F* put in descending order. It is important to note that weights are associated with a particular ordered position rather than a particular element.

Yager's paper [Yager93], published five years later than [Yager88], shows the evolution of his work towards more specialised OWA operators, enhancements, and ways of establishing (learning) the weights.

An example of applying the OWA operator is as follows. Assume:

$W = [0.4, 0.3, 0.2, 0.1]^T$

Then f(0.7, 1.0, 0.3, 0.6) = (0.4)(1.0) + (0.3)(0.7) + (0.2)(0.6) + (0.1)(0.3)=0.76.

Different OWA operators are distinguished by their weighting function. Yager distinguishes three special cases of OWA aggregation:

(i) F^* : in this case, $W = W^* = [1 \ 0 \ \ldots \ 0]^T$,
(ii) F_* : in this case, $W = W_* = [0 \ 0 \ \ldots \ 1]^T$,
(iii) F_A : in this case, $W = W_{ave} = [1/n \ \ldots. \ 1/n]^T$.

It can be seen that

$F^*(a_1, \ldots a_n) = \underset{i}{\text{Max}}(a_i)$,

$F_*(a_1, \ldots a_n) = \underset{i}{\text{Min}}(a_i)$,

$F_{Ave}(a_1, \ldots a_n) = 1/n \ \Sigma \ (a_i)$,

WOWA – Weighted Ordered Weighted Average

The work of Torra is characterised by the study of aggregation operators such as WOWA [Torra96], and the construction of membership functions using interpolation methods [Torra99a]. One of the original contributions of Torra to the field of aggregation is the WOWA operator , which combines the characteristics of the OWA and WM operators. It uses two weighting vectors, one relating to 'relevance' and another to 'reliability' of the sources, which are used to aggregate the values.

The WOWA (Weighted OWA) aggregation operator is first presented in [Torra96]. It is stated that the OWA operator satisfies the commutative property, while the weighted mean does not. The commutative property implies equal reliability of all the information sources which supply the data. First the properties of the OWA operator are formally defined, followed by those of the weighted mean and of the WOWA operator itself. The determination of the WOWA

operator is explained, followed by some examples of interpolation and aggregation outputs with simple datasets. Finally, possible variants such as 'Quasi-WOWA', which is derived from the quasi-arithmetic mean and quasi-OWA, is a generalisation; while the 'Linguistic WOWA' serves to combine information in a qualitative form (instead of quantitative), based the convex combination of linguistic labels.

Definition 1. A vector $\mathbf{v}= [v_1\ v_2\ ....\ v_n]$ is a *weighting vector* of dimension n if and only if

$$v_i \in [0,1] \qquad \Sigma_i\, v_i = 1$$

Definition 2 [Torra97a]. Let **p** be a weighting vector of dimension n, then a mapping WM: K^n -> K is a *weighted mean* of dimension n if $WM_{\mathbf{p}}(a_1,...,a_n) = \Sigma_i\, p_i\, a_i$.

Definition 3 [Yager88]. Let **w** be a weighting vector of dimension n, then a mapping $OWA_{\mathbf{w}}$: K^n -> K is an *Ordered Weighted Averaging (OWA) operator* of dimension n if

$$OWA_w(a_1,...,a_n) = \Sigma_i\, w_i\, a_{\sigma(i)}$$

where $\{\sigma(1),...,\sigma(n)\}$ is a permutation of {1,..,n} such that $a_{\sigma(i-1)}\ \Sigma a_{\sigma(i)}$ for all i=2, ..., n. (i.e., $a_{\sigma(i)}$ is the i-th largest element in the collection $a_1,..., a_n$).

Definition 4. Let **p** and **w** be two weighting vectors of dimension n, then a mapping WOWA: K^n -> K is a *Weighted Ordered Weighted Averaging (WOWA) operator* of dimension n if

$$WOWA_{p,\,w}(a_1,...,a_n) = \Sigma_i\, \omega_i\, a_{\sigma(i)}$$

where $\{\sigma(1),...,\sigma(n)\}$ is a permutation of {1,..,n} such that $a_{\sigma(i-1)}\ \Sigma a_{\sigma(i)}$ for all i=2, ..., n. (i.e., $a_{\sigma(i)}$ is the i-th largest element in the collection $a_1,..., a_n$), and the weight ω_i is defined as

$$\omega_i = w^*(\Sigma_{j\le i}\, p_{\sigma(j)}) - w^*(\Sigma_{j<i}\, p_{\sigma(j)})$$

with w^* a monotone increasing function that interpolates the points $(i/n, \Sigma_{j\le i}\, w_j)$ together with the point (0,0). The function w^* is required to be a straight line when the points can be interpolated in this way.

Proposition 1 [Torra97a]. The WOWA operator satisfies the following properties:

(1) It is an aggregation operator which remains between the minimum and the maximum.
(2) It satisfies idempotency (unanimity).
(3) It is commutative if and only if $p_i=1/n$ for all i=1,…,n such that $w_i \neq 0$.
(4) It is monotone in relation to the input values a_i.
(5) It leads to dictatorship of the i-th value when $p_i = 1$ and $p_j = 0$ for all j = 1, …, n but $j \neq i$.
(6) It leads to the arithmetic mean when $p_i = 1/n$ and $w_i = 1/n$ for all i=1, …, n
(7) It leads to the weighted mean when $w_i=1/n$
(8) It leads to the OWA operator when $p_i=1/n$

Definition of the Choquet integral

The Choquet integral generalises the OWA and WM operators, and is defined as follows. Let μ be a fuzzy measure on X. A fuzzy measure on X is defined as a monotonic set function $\mu : 2^N \rightarrow [0,1]$ with $\mu(\Re) = 0$ and $\mu(N)=1$. Monotonicity implies that $\mu(S) \le \mu(T)$ whenever $S \subseteq T$. The Choquet integral of a function $f: X \rightarrow \Re$ with respect to μ is defined by:

$$(C)\ \int f\, d\mu = \sum_{i=1}^{n} \left(f(x_{s(i)}) - f(x_{s(i-1)}) \right)\ \mu(A_{s(i)}) \qquad (2.51)$$

where $f(x_{s(i)})$ indicates that the indices have been permuted so that $0 \le f(x_{s(1)}) \le .... \le f(x_{s(N)}) \le 1$, $A_{s(i)} = \{ x_{s(i)}, \dots , x_{s(N)} \}$ and $f(x_{s(0)})=0$. This definition shows that in a Choquet integral each segment $f(x_{s(i)}) - f(x_{s(i-1)})$ is considered (weighted) according to all the elements x_j such that $f(x_{s(j)}) \ge f(x_{s(i)})$. That is, the importance of each segment $f(x_{s(i)}) -$

$f(x_{s(i-1)})$ corresponds to the measure of all the elements whose evaluation embeds that segment (i.e. x_j such that $f(x_{s(j)}) \geq f(x_{s(i)})$).

Definition of the Sugeno integral [Sugeno74]

The Sugeno integral generalises the 'weighted min' and 'weighted max' operators, and is defined as follows. Let μ be a fuzzy measure on X. The Sugeno integral of a function f: $X \rightarrow \Re$ with respect to μ is defined by:

$$S_{\mu, f}(X=(x_1, \ldots, x_n)) = \mathrm{Max}_i \mathrm{Min}(f(x_{s(i)}), \mu(A_{s(i)})) \quad (2.52)$$

where $f(x_{s(i)})$ indicates that the indices have been permuted so that $0 \leq f(x_{s(1)}) \leq \ldots. \leq f(x_{s(N)}) \leq 1$, $A_{s(i)} = \{ x_{s(i)}, \ldots, x_{s(N)} \}$ and $f(x_{s(0)})=0$.

Definition of the Fuzzy t-integral [Murofushi91]

The Fuzzy t-integral generalises both the Choquet integral and the Sugeno Integral. It is defined over a tuple called a t-conorm system for integration, and an operator $-_{\Delta}$ based on one of the elements of this tuple. Let (X, X, μ) be a fuzzy measure space and (<, ζ, $\curlywedge$, 1) a t-system. For a measurable function *f*: $X \rightarrow [0,1]$, the fuzzy t-conorm integral (or fuzzy t-integral) of *f* based on (<, ζ, $\curlywedge$, 1) with respect to μ is defined as follows:

$$(\curlywedge) \int f \, 1 \, d\mu = \lim_{n \rightarrow \infty} (\curlywedge) \int f_n \, 1 \, d\mu \quad (2.53)$$

Where $\{f_n\}$ is a non-decreasing sequence of simple functions which pointwise converges to *f*.

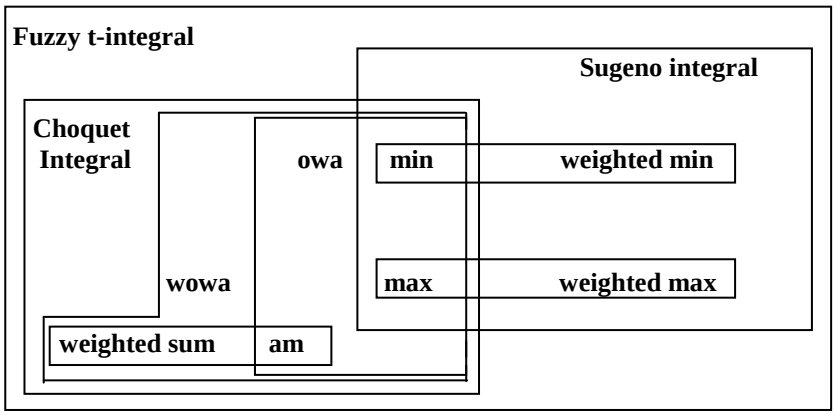


Figure 28. Relation between several numeric aggregation operators

Figure 28 depicts each aggregation operator as a polygon form or a rectangular form. A $form_1$ is inside a $form_2$ if the operator corresponding to $form_2$ is a generalisation of the operator corresponding to $form_1$.

'Orness' and 'Andness'

Yager , in [Yager93], formally defined two concepts, 'orness' and 'andness', in which the former is a type of bias which is related to 'optimism', which can be quantified and applied to a set of attribute-values. The opposite of 'orness' in this context is 'andness', which equates with 'pessimism'. This is an attempt to capture the subjectivity which exists whenever an expert makes a judgement with respect to a given attribute or attribute-value.

The following is a definition of 'orness', or the grade to which an OWA operator tends towards a pure 'or' condition.

$$\mathrm{orness}(F) = \tfrac{1}{2} + \sum_i ((n-i)/(n-1))w_i - \tfrac{1}{2} \quad (2.54)$$

A second interesting definition is that of 'biasness', which is as follows:

$$\text{biasness}(W) = ½\,(\text{orness}(W) - ½) \tag{2.55}$$

This is said by Yager [Yager93] to be the biasness of the average. If it takes a positive value it accentuates the higher values, and if it takes a negative value it accentuates the lower values.

Another concept relevant to decision making is that of 'optimism'. It observed that in a decision making environment the measure of dispersion may be interpreted as the entropy of the probability distribution. Further, the measure of 'orness' of W may be interpreted as a measure of the optimism of the decision made, while the measure of 'andness' could be a measure of pessimism. For the Hurwicz model the following is obtained:

$$\text{optimism}(W) = 1/(n-1) \sum_{i=1}^{n} (n-i) \bullet w_i \tag{2.56}$$

$$= 1/(n-1)\,((n-1)\alpha)+(n-n)/(n-1)\,(1-\alpha)$$
$$= \alpha$$

Yager in [Yager93] comments that in some applications the weights associated with an OWA operator must be learned from observations. Let F be an OWA operator of dimension n with weighting vector W. Assume there is a collection of m pieces of data each of which is an n+1 tuple of the form

$(a_{i1}, a_{i2}, \ldots a_{in}, y_i)$.

where the a_{ij}, j=1, …. , n are the input (aggregate) values for the ith sample and y_i is the aggregated value for this ith sample. The objective is to find the weights of an OWA operator to model this process. A simple neural net is proposed to learn the approximate weights, which has a sorter pre-process to order the data values. The net is run until the successive change in weights is less than a predetermined value.

Degree of disjunction: Marichal, in [Marichal98] also considers the concept of 'orness' and derives a definition in terms of the Choquet Integral. The Choquet Integral permits, by appropriate choice of the fuzzy measure, to move continuously from the *min* operator to the *max* operator. In order to classify these Choquet integrals in terms of their location on this continuum a measure of disjunction can be defined as follows. Let the average value of the Choquet integral be:

$$m(C_\mu) := \int_{[0,1]^n} C_\mu(x)\,dx. \tag{2.57}$$

Where μ is a fuzzy measure and C_μ is the Choquet integral.

Now, a degree of orness of C_μ will be:

$$\text{orness}(C_\mu) := \frac{m(C_\mu) - m(\min)}{m(\max) - m(\min)} \tag{2.58}$$

It is observed that orness(C_μ) is always in the unit interval, where orness(min)=0 and orness(max)=1. One characteristic is that the closer orness(C_μ) is to 0, the nearer C_μ is to 'min' and has a conjunctive behavior, while the closer orness (C_μ) is to 1, the nearer C_μ is to 'max' and has a disjunctive behavior. Thus, the degree of orness can be interpreted as a measure of the 'strictness' of the decision maker. Less strict decision makers allow that only *some* criteria have to be satisfied, which corresponds to a disjunctive behavior (orness (C_μ) > 0.5), with extreme example of 'max'. In contrast, more strict decision makers decision makers requiere that *most* criteria be satisfied, which corresponds to a conjunctive behavior (orness(C_μ) < 0.5), and whose extreme example will be 'min'. It follows that orness(C_μ) = 0.5 would represent an equitable decision maker.

It is noted that, in the case of Marichal's definition of orness, the degree of orness corresponds to decision making problems which are modelled by the Choquet integral, although it may be defined for any compensative aggregation operator.

2.3.2 Mechanisms for learning weights

In this section we consider the determination of the parameters (weights) for aggregation functions. We can divide the determination methods into two camps: those which rely on expert judgement, and those which rely on automated algorithmic means. In relation to the first approach, we can cite Saaty's Analytic Hierarchy Process [Saaty80] for the weighted mean or the method defined in [O'Hagan88] for the OWA operator. These methods are based on a user, or a set of users/experts, who supply critical information which is subsequently used in a certain way to define or extract the weights. This approach can be applied when there is some background knowledge about the system we want to model or about the decision process. To deal with the case when this background knowledge does not exits, there exist 'automated' approaches such as those based on machine learning techniques. These latter techniques attempt to learn the parameters from a set of examples. This is the case in [Filev98] which learns the weighting vector for OWA operators, in [Torra99b] for the weighted mean and OWA operators, or in [Marichal99] for Choquet integrals. In all these works, a set of examples (defined as a set of input parameters and their corresponding output) are given and from them the weighting vectors or the fuzzy measures are inferred.

The advantage of the automated approach is that it does not require *a priori* background knowledge. By automating the whole process through the use of examples, learning the weights allows the extension of these operators to larger problems where the number of parameters is great. This is not possible when all the information has to be supplied by experts as a large amount of knowledge is then required.

On the other hand, expert assignment of the weights implicitly includes the experience, intuition and common sense of those domain experts. This can save time in avoiding nonsensical or incongruent results, and is independent of a given dataset which could incorporate noise, missing values or skew distributions for certain attributes. An intermediate approach is to combine automated learning with expert assignment, or expert review of the automated assignments for validation. This joint approach is later compared with the exclusively automated approach and the exclusively expert assignment approach, in Sections 4.3 and 4.4 of the thesis.

Expert weight assignment

O'Hagan, in [O'Hagan88], employs the methodology described by Saaty in [Saaty77] for use in hierarchical structures with data values in the [0,1] interval, together with the methodology described by Yager in [Yager 88] for normalised weighting. O'Hagan considers the situation in which a panel of experts agree to a set of rankings for the relative importance of the following events:

IF (Event *1* observed with confidence a_1, AND
Event *2* observed with confidence a_2, AND ….
Event *n* observed with confidence a_n)
THEN conclusion with Max Confidence Factor (CF)

The set of rankings being:

- Event 2 is weakly more important in defining our conclusion than Event 1.
- Event 3 is somewhere between equal and weakly more important than Event 1.
- Event 2 is weakly more important than Event 3.

The above relative importance ranking results in a paired comparison matrix using Saaty's analytic hierarchical process [Saaty77] of:

$$A = \begin{pmatrix} 1 & 1/3 & 1/2 \\ 3 & 1 & 3 \\ 2 & 1/3 & 1 \end{pmatrix}$$

The Eigen solution of A yields a maximum real Eigen value of 3.05 and a unit normalised Eigen vector to be used as relative importance weights of:

Event	Eigen Vector (α_j)
1	0.157
2	0.594
3	0.249

The relative importances, denoted by a_j , are constrained to the unit interval by the method described by Yager in [Yager88]. A normalisation process is then performed using the general form given in [Yager88], which uses the *CF* of

the conclusion as the degree of *orness* or optimism. The general form with the α_j relative importances constrained to the unit interval is:

$$a_j = H(A_j(x), \alpha_j) = (\alpha_j - p) * A_j(x)^{(\alpha j - q)}$$

Given the fuzzy scores or confidences in the individual events, the general expression is used to normalise the values while still reflecting the general orness for the overall conclusion associated with the rules. Using a CF of 0.90, and the calculated relative importances, we have:

$$a_1 = (0.157 - 0.10) * (0.7)^{(0.157 - 0.9)}$$
$$= 0.157 * (0.7)^{0.90} = 0.11 ,$$

$$a_2 = (0.594 - 0.10) * (0.5)^{(0.594 - 0.9)}$$
$$= 0.594 * (0.5)^{0.90} = 0.32 ,$$

$$a_3 = (0.249 - 0.10) * (0.5)^{(0.249 - 0.9)}$$
$$= 0.249 * (0.5)^{0.90} = 0.13 .$$

The resulting normalised fuzzy values are then sorted and a set of OWA coefficients are applied to perform the aggregation. The resulting sorted values are called the *B* vector with elements b_j corresponding to the sorted a_js. Thus we have:

$$[a_2, a_3, a_1] = [0.32, 0.13, 0.11] = [b_1, b_2, b_3] = b'$$

Before the aggregation operator is executed, the OWA weights used in the aggregation process are computed, following the method which we have just described. Then the actual aggregation process can be considered a dot product operation $F = B'W$ with the *B* vector computed by sorting the a values as described previously:

$$F = B'W = [0.32, 0.13, 0.11] \begin{pmatrix} 0.825 \\ 0.150 \\ 0.025 \end{pmatrix} = 0.286$$

Automated weight assignment

Filev in [Filev98] considers obtaining the OWA weights based on historical data, that is, input data values and their corresponding aggregated outputs. Yager, in [Yager93] defines two concepts, 'orness' and 'optimism', which have already been discussed in Section 2.3.1 of the thesis. Filev makes use of these concepts in [Filev98]. Yager proposed that 'orness' is a type of bias which is related to 'optimism', both of which can be quantified and applied to a set of attribute-values. The opposite of 'orness' in this context is 'andness', which equates with 'pessimism'.

The method described by Filev in [Filev98] uses a gradient descent method to find the minimum error, which is illustrated by an example of learning from a collection of samples, as can be seen in Table 8, which constitutes the historical data. Each sample consists of 4 attribute values and the corresponding aggregate value.

Table 8. Historical data used for OWA weight learning

Sample	Attribute values				Aggregated value
1	0.4	0.1	0.3	0.8	0.24
2	0.1	0.7	0.4	0.1	0.16
3	1.0	0.0	0.3	0.5	0.15
4	0.2	0.2	0.1	0.4	0.17
5	0.6	0.3	0.2	0.1	0.18

The aggregated values for each sample are calculated by the Hurwicz [Engemann96] method for comprimised aggregation. This method defines that the aggregated value *d* obtained from a tuple of *n* arguments, $a_1, a_2, \ldots, a_n$, is defined as a weighted average of the Max and Min values of that tuple

$$\rho \max_i a_i + (1 - \rho) \min_i a_i = d ,$$

where parameter ρ represents the optimism of the decision maker, $0 \le \rho \le 1$. For example, the aggregated value for the samples in Table 8 were calculated using $\rho = 0.2$. For sample 1, the Max and Min values for the first sample produce:

$$0.2\ (0.8) + (1 - 0.2)\ (0.1) = 0.24$$

Small variations in the value of parameter ρ for each argument tuple reflect variations due to the individuality of different experts, which would result in slight differences of aggregation for different samples. The ρ parameter for samples 1 to 5 was, respectively, {0.2, 0.1, 0.15, 0.25, 0.18}.

The OWA weights have to satisfy the properties that they must sum to 1, be non negative and be in the range [0,1]. The following definition of the OWA weights guarantees these properties:

$$w_i = \frac{e^{\lambda i}}{\sum^{n}_{j=1} e^{\lambda j}}, \quad i = (1, \ldots, n) \qquad (2.59)$$

From definition 2.59, for any values of the parameters λ_i, the weights w_i will be positive and will sum to 1.

The learning algorithm was applied on the reordered argument tuples, started with initial values $\lambda_i(0)=0$, i=(1,4), and the OWA weights w_i initialised to 0.25. Using a learning rate $\beta = 0.35$, the estimated values of w_i after 150 iterations are:

w_1=0.08, w_2=0.11, w_3=0.14, w_4=0.67

Estimated aggregated values d_k at the end of the learning process are:

d_1=0.22, d_2=0.18, d_3=0.15, d_4=0.15, d_5=0.18

Using the learned OWA weights as above, and applying the formula for *orness* as defined in Section 2.3.1, a degree of orness of 0.199 is calculated. Filev concludes that this is a reasonable reflection of the total level of orness associated with the complete sample set, when contrasted with the different levels of optimism for each individual sample.

The learning of weights for the Choquet integral is considered in [Marichal99]. The Choquet integral has been defined and discussed previously in Sections 2.3.1. The problem is limited to 2-order fuzzy measures; which means a fuzzy measure which can be represented by a polynomial expression of degree 2. Also the problem involves the identification of weights of interacting criteria, that is attributes which have mutual influence. Semantic considerations about criteria are made and are as follows:

- *Importance of criteria.* Realised by a partial preorder on N, representing a ranking of the weights $\mu(i)$, $i \in N$. The ranking may or may not be defined by exact values.

- *Interaction between criteria.* This enables the appraisal of the degree of interaction among any subset of criteria. In order to formalise this concept, consider a pair $\{i, j\} \subseteq N$ of criteria, where N is a set of criteria $\{1, \ldots, n\}$. The difference

  $$a(ij) = \mu(ij) - \mu(i) - \mu(j)$$

  reflects the degree of interaction between i and j. The difference is zero when there is no interaction between i and j, and the difference is positive if an interaction with 'positive interference' exists between i and j. If the difference is negative then an interaction with 'negative interference' is said to exist. Thus, one possible 'interaction index' is $I(ij) = a(ij)$. This allows a partial preorder on the set of pairs of criteria. The sign of each interaction $a(ij)$ can be given, including exact values.

- *Symmetric criteria.* Two criteria i and j are symmetric if they can be exchanged without changing the aggregation mode. Then $\mu(T\ 4\ i) = \mu(T\ 4\ j)$ for all $T \subseteq \bar{N} \setminus ij$. This has the effect of reducing the number of coefficients.

- *Veto and favour effects.* A criterion $i \in N$ is said to be a veto for a decision problem modelled by the Choquet integral if $C\mu(x) \leq x_i$ for all $x \in \Re^n$. This implies that a low score on criterion i will lead to a bad global score, irrespective of the values of the remaining scores. This effect can be modelled by taking a fuzzy measure μ such that $\mu(S)=0$ whenever $i \notin S$. In the same manner, criterion i is said to be a favour if $C\mu(x) \geq x_i$ for all $x \in \Re^n$. In this

case, a good score on *i* will result in a good global score, irrespective of the remaining score values. This effect can be modelled by taking μ such that μ(*S*)=1 whenever $i \in S$.

The above four criteria assume that an expert is available who can make judgements on the relative importance of criteria, and interactions between criteria. It is said that experts find it easier to provide information in terms of weights u(i) and on interaction indices, than directly on the values of fuzzy measures. The success of elicitation from the expert also depends on asking the right questions and expressing them in an adequate manner.

The problem, defined in terms of the input data, is as follows:

- The set *A* of alternatives and the set *N* of criteria,
- A table of individual scores (utilities) x^a_i given on the same interval scale $X \subseteq \Re$,
- A partial preorder $\succeq_A$ on *A* (ranking of alternatives),
- A partial preorder $\succeq_N$ on *N* (ranking of criteria),
- A partial preorder $\succeq_P$ on the set of pairs of criteria (ranking of interaction indices),
- The sign of interaction between some pairs of criteria $a(ij) : >0, =0, <0$.

The above defined data can be formulated in terms of linear equalities or inequalities, which link the unknown weights 'μ'. This reduces to a linear constraints satisfaction problem, which, when written in terms of the Möbius representation [Roubens96] allows the following definition of a model for eliciting the weights:

Maximise $x = \varepsilon$

Subject to

$C(a) - C(b) \geq \delta + \varepsilon$ if $a \succ_A b$
$-\delta \leq C(a) - C(b) \leq \delta$ if $a \sim_A b$ } partial semiorder with threshold δ

$a(i) - a(j) \geq \varepsilon$ if $i \succ_N j$
$a(i) = a(j)$ if $i \sim_N j$ } ranking of criteria (weights on singletons)

$a(ij) - a(kl) \geq \varepsilon$ if $ij \succ_P kl$
$a(ij) = a(kl)$ if $ij \sim_P kl$ } ranking of pairs of criteria (interactions)

$a(ij) \geq \varepsilon$ (resp. $\leq -\varepsilon$) if $a(ij) > 0$ (resp. < 0)
$a(ij) = 0$ if $a(ij) = 0$ } sign of some interactions

$\sum_{i \in N} a(i) + \sum_{\{i,j\} \subseteq N} a(ij) = 1$ } boundary and
$a(i) \geq 0 \quad \forall i \in N$ } monoticity
$a(i) + \sum_{j \in T} a(ij) \geq 0 \quad \forall i \in N, \forall T \subseteq N \setminus i$ } conditions

$C(a) = \sum_{i \in N} a(i)\, x^a_i + \sum_{\{i,j\} \subseteq N} a(ij)[x^a_i \wedge x^a_i] \quad \forall a \in A$ } definition of Cμ

In [Torra99b] the learning of weights is considered for the weighted mean, OWA and WOWA operators. The results for the different operators are compared. In the case of WM, the ideal weights are considered to be those which approximate the solutions of the example with a minimum error. The distance to be measured is the difference between the ideal value (m) and the calculated one ($\Sigma a_i p_i$) for each example j, and the expression to be minimised is:

$$D(\mathbf{p}) = \sum_{j=1}^{M} (\sum_{i=1}^{N} a_i^j p_i - m^j)^2 \qquad (2.60)$$

where the dimension of the weighted mean is settled to N and the number of examples is M.

For the OWA, the expression used is the same, except that a permutation is introduced $a\sigma_i^j$ in substitution of a_i^j .

The least squares method is also defined for the WOWA operator, with the modification that the distance has to be defined in terms of the operator with the two sets of weights, thus:

$$D(\mathbf{p}) = \sum_{j=1}^{M} \left(\sum_{i=1}^{N} WOWA_{w,p}(a^j) - m^j \right)^2 \qquad (2.61)$$

Such that $\sum_{i=1}^{N} p_i = 1$, $\sum_{i=1}^{N} w_i = 1$, $p_i \geq 0$, $w_i \geq 0$

2.3.3 Construction of membership functions

Generation of membership functions from observations: in [Zhang93] a formal definition of fuzzy sets to describe fuzzy categories is introduced. The approach assumes that for any fuzzy category â to be described, there is a partition of the reference set into three subsets X_0, X_f and X_l: X_0 corresponds to all elements that do not belong to the fuzzy category â; X_f to the elements such that their membership to â is 'doubtful'; and X_1, to the elements that belong to â with total certainty. The three sets, X_0, X_f and X_l , are referred to, respectively, as the 0-subset, the fringe and the 1-subset for â and the partition $X=[X_0, X_f , X_l]$ is called the fringe partition.

In order to build a membership function for a fuzzy category â, [Zhang93] assumes a partial ordering $\geq_{\hat{a}}$ in the fringe of â, such that $x \geq_{\hat{a}} y$ means x is at least as qualified to be a member of â as y, and with respect to a reference measure η.

Torra, instead of building a membership function from the set X, considers the construction of the function from a set of observations Ξ obtained from a given experiment to elicit the fuzzy concept â.

Taking this into account, and letting Ξ_f be a set of observations which correspond to the fringe of a fuzzy category â, owa^Q be a combination function, and I^α is the I-family of fuzzy quantifiers. Then, a membership function for the fuzzy category â may be expressed as the set of α-cuts $\{A_\alpha\}_{\alpha \in [0,1]}$:

$$A_\alpha = [\, owa^{I^\alpha}(\Xi_f) , + \infty]$$

This may be understood in terms of fuzzy quantifiers as follows: if each observation x_i corresponds to an expert i who asserts that the concept is fully satisfied when $x \geq x_i$ and is totally unsatisfied when $x < x_i$, then $y = owa^{I\alpha}(\Xi_f)$ means approximately that a proportion α of the experts agree that the interval beginning at the right of y fully satisfied the concept. In the extreme case, all agree with the interval when $\alpha=1$, which corresponds to the intersection of the intervals, and $y=\Xi_f$, and only one agrees when $\alpha=0$ and $y = \min \Xi_f$.

Interpolation

One of the problems we face in defining a membership function is that of the construction of a smooth curve that passes through a given set of data points. The most common technique is that of cubic spline interpolation, although this technique may produce undesirable oscillations that make the curve non-compliant with some characteristics which we may require of the membership function. Such characteristics may be those of monotonicity and/or convexity, and we thus require an interpolation scheme which preserves the desired shape and characteristics. Algorithms which preserve a desired shape typically introduce additional 'knots' or modify the prescribed slopes in order to give the required shape as the final 'product' [Iqbal92]. We now look at three contrasting algorithms which preserve desired characteristics, that of Mc Allister and Roulier [McAllister81], that of Schumaker [Schumaker83] and that of Chen and Otto [Chen95].

The algorithm presented in [McAllister81] has the slope and knot assignments based on a geometrical argument which preserves monotonicity and/or convexity. The resulting quadratic piecewise polynomial is constructed from Bernstein polynomials. [Schumaker83] presents a similar algorithm using quadratic piecewise polynomials, which preserves monotonicity and/or convexity by the addition of one knot, if necessary, in each data subinterval. One notable aspect of this algorithm is that the user is able to modify the interpolant either by changing the slopes or by altering the knot locations.

The interpolation problem is defined formally as follows: given points $t_1 < \ldots < t_n$ and values $\{z_i\}^n_1$, find *s* such that

$$S(t_i) = z_i , \qquad i = 1,2, \ldots , n .$$

A method must be found which preserved the shape of the data. This means that in those intervals where the data is monotone increasing or decreasing, *s* should have the same property. Similarly, in those intervals where the data is convex or concave, the same should be true of *s*.

[Schumaker83] describes a relatively simple method which preserves a desired shape, by constructing the interpolant as a C^1 quadratic spline with knots at the data points $t_1, \ldots, t_n$ and with one additional knot in each subinterval (t_i , t_{i+1}), i=1, …, n–1. If $t_1 < t_2$, and it is assumed that z_1, z_2, s_1 and s_2 are real numbers, then a function $s \in C^1[t_1, t_2]$ is to be found such that

$$s(t_i) = z_i , s'(t_i) = s_i , i=1,2 \tag{2.62}$$

The following lemma shows that in certain cases, problem (2.62) can be solved by a quadratic polynomial.

Lemma l.a. There is a quadratic polynomial solving problem (2.62) if and only if

$$\frac{s_1 + s_2}{2} = \frac{z_1 - z_2}{t_2 - t_1} \tag{2.63}$$

In particular, if (2.63) holds, then

$$S(t) = z_1 + s_1(t - t_1) + \frac{(s_2 - s_1)(t - t_1)^2}{2 (t_2 - t_1)} \tag{2.64}$$

is a solution.

The following considers the conditions of monotonicity and convexity. This is preceded by some notational definitions:

$\mathbf{I}_i = [t_i, t_{i+1}]$ and

$$\delta i = \frac{z_i + 1 - z_i}{t_i + 1 - t_i}, \qquad i = 1, 2, \ldots, n -1.$$

If a knot has been inserted in the interval $\mathbf{I}_i$, it is denoted by ξ_i .

Monotonicity . To guarantee that s is monotone on the interval $\mathbf{I}_i$, it first has to be assured that $s_i \exists s_{i+1} \geq 0$. In addition, if a knot ξ_i is inserted in the interval $\mathbf{I}_i$, then another requirement is that the following expression has the same sign as s_i and s_{i+1}.

$$\underline{s}_i = \frac{2(z_{i+1} - z_i) - (\xi_i - t_i)s_i - (t_{i-1} - \xi_i)s_{i+1}}{(t_{i+1} - t_i)} \tag{2.65}$$

If $(s_i - \delta_i)(s_{i+1} - \delta_i) \geq 0$, in order to ensure (2.65) the size of s_i and s_{i+1} must be restricted , depending on the location of ξ_i. In particular, the following must hold:

$$2|z_{i+1} - z_i | \geq |(\xi_i - t_i)s_i + (t_{i+1} - \xi_i)s_{i+1}|. \tag{2.66}$$

These conditions show how to make *s* monotone in the interval $\mathbf{I}_i$. If the data is globally monotone, that is, $z_1 < z_2 \ldots < z_n$, then by selecting the slopes $\{ s_i \}^n_1$ correctly, *s* can also be made globally monotone.

Convexity. To guarantee that s is convex on the interval $\mathbf{I}_i$, we need to assure that the condition $s_1 \leq s' \leq s_2$ holds, while for concavity, the condition $s_2 \leq s' \leq s_1$ is required. These conditions can be guaranteed by choosing the knot ξ_i in the interval $\mathbf{I}_i$, satisfying the following:

$$t_i < \xi_i \leq t_i + \frac{2(t_{i+1} - t_i) (s_{i+1} - \delta_i)}{(s_{i+1} - s_i)} \qquad \text{if } |s_{i+1} - \delta_i | < |s_i - \delta_i | , \tag{2.67}$$

$$t_{i+1} + \frac{2(t_{i+1} - t_i) (s_i - \delta_i)}{(s_{i+1} - s_i)} \leq \xi_i < t_{i+1} \qquad \text{if } |s_{i+1} - \delta_i | < |s_i - \delta_i | , \tag{2.68}$$

respectively. If the data is globally convex, that is $\delta_1 < \delta_2 < \ldots < \delta_n$, *s* will be globally convex by choosing $s_1 < s_2 < \ldots < s_n$ such that *s* is locally convex in each subinterval. A similar assertion holds for global concavity.

The algorithm detailed in [McAllister81] produces a local C1 quadratic spline interpolant which preserves monotonicity and/or convexity of the data by inserting at most two additional knots per data interval. The selection of slopes and knots is based on the geometric arguments given below, while the polynomial pieces are constructed using Bernstein polynomials. S=(x_i, y_i) and T=(x_{i+1} ,y_{i+1}) are defined as two nondecreasing data points with $x_i<x_{i+1}$ having slopes di and di+1, respectively. L_1 and L_2 are defined as the two straight lines through points S and T with slopes d_i and d_{i+1} , respectively. R is defined as the set of points,

$$R = \{(x, y): x_i \leq= x \leq= x_{i+1}, \text{ and } y_i \leq= y \leq= y_{i+1}\} - \{S, T\}$$

where R is the boundary and interior of the rectangle defined by the points (x_i, y_i), (x_i, y_{i+1}), (x_{i+1}, y_{i+1}), and (x_{i+1}, y_i) minus the points S and T. M is then defined as the midpoint line segment through the points

$$F = \left(\frac{x_i + x_{i+1}}{2}, y_i\right) \quad \text{and} \quad G = \left(\frac{x_i + x_{i+1}}{2}, y_{i+1}\right)$$

Let Z = (z_1, z_2) be a point of intersection of line segments L_1 and L_2. The following shows how to construct the desired quadratic spline interpolant *p*.

In **Case 1**, L_1 and L_2 intersect each other at point Z = (z_1, z_2) in R as shown in Figure 29, where

$$z_1 = \xi_i = (y_i - y_{i+1} + d_{i+1} x_{i+1} - d_i x_i)/(d_{i+1} - d_i) \tag{2.69}$$

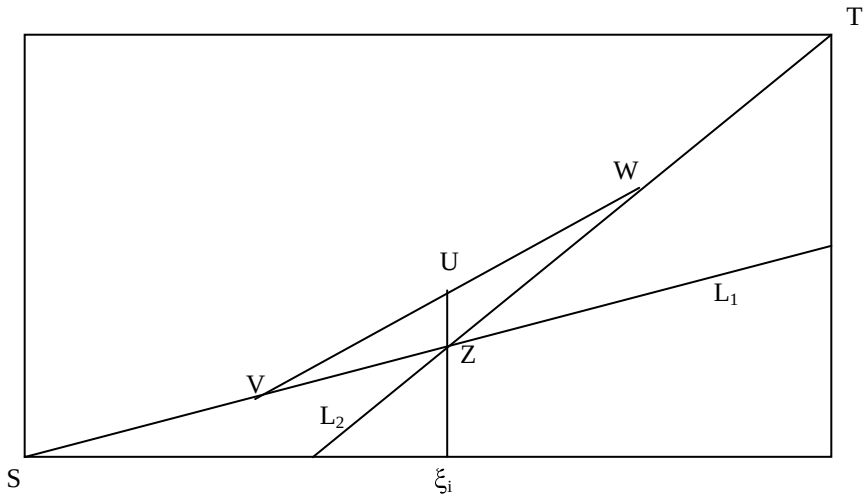


Figure 29. The case in which straight line segments L_1 and L_2 intersect.

The algorithm inserts an additional knot at x = ξ_i . Now assume that

$$V = (v_1, v_2) = \left(\frac{x_i + \xi_i}{2}, L_1\left(\frac{x_i + \xi_i}{2}\right)\right) \tag{2.70}$$

$$W = (w_1, w_2) = \left(\frac{x_{i+1} + \xi_i}{2}, L_2\left(\frac{x_{i+1} + \xi_i}{2}\right)\right) \tag{2.71}$$

Now $\eta_i=L(\xi_i)$ is defined, *L* is the line passing through the points *V* and *W*. ρ on [x_i , x_{i+1}] is then defined with a join point *U* = (ξ_i , η_i) as follows:

$$P(x) = \begin{cases} \frac{1}{(\xi_i + x_i)^2}\ [yi(\xi_i - x)^2 + 2v_2(x\text{-}x_i) \\ \quad \times (\xi_i - x) + \eta_i(x - x_i)^2\ ], \quad x \in [x_i, \xi_i] \\ \frac{1}{(x_{i+1} - \xi_i)^2}\ [\eta_i\ (x_{i+1} - x)^2 + 2w_2(x - \xi_i) \\ \quad \times (x_{i+1} - x) + y_{i+1}(x - \xi_i)^2\ ], \quad x \in [\xi_i, x_{i+1}] \end{cases} \tag{2.72}$$

If the first degree spline defined by join points S, V, U, W and T is convex (concave) and/or monotone, then ρ is also convex (concave) and/or monotone.

In **Case 2**, L_1 and L_2 do not intersect in R; instead both intersect the line segment M, as can be seen in Figure 30, and the method introduces one additional knot in (x_i, x_{i+1}), which is:

$$\xi_i = (x_i + x_{i+1}) / 2 \tag{2.73}$$

V, W U, and the spline ρ are then defined with a common point $U = (\xi_i, \eta_i)$ on $[x_i, x_{i+1}]$ as in Case 1. Then ρ will have a continuous first derivative and preserve the shape of the data on $[x_i, x_{i+1}]$.

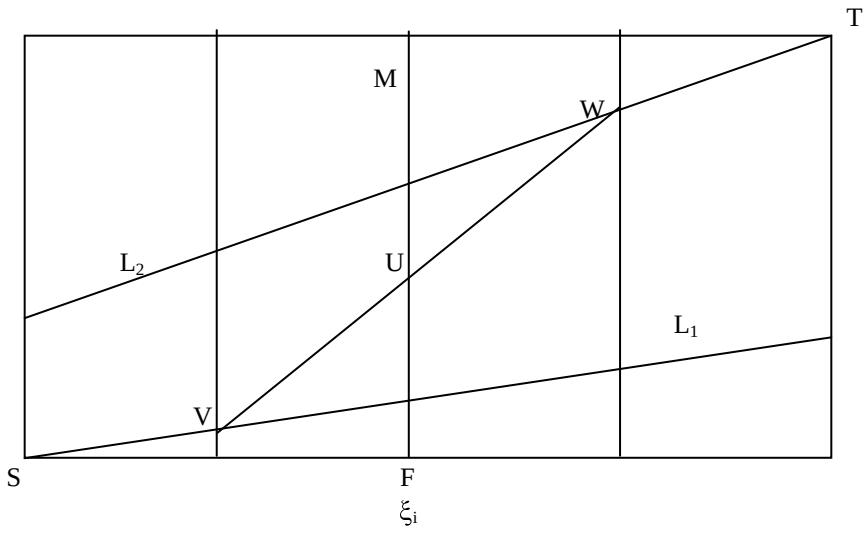


Figure 30. The case in which straight line segments L_1 and L_2 do not intersect.

In [Torra99a] some corrections are made to Chen and Otto's interpolation method for defining membership functions from a set of points. Some new conditions are also introduced on the boundaries which are more general than those of the original work. The relation of this method to the WOWA aggregation operator is that it is then used by WOWA as the interpolation method to calculate a set of point values from a combination of the two input weighting vectors. The desired 'smooth' curve can then be plotted from the point values. The point values are stored in an appropriate vector.

The interpolation procedure of Chen and Otto follows the following steps:

(i) for each point d_i , its slope m_i is calculated.
(ii) A knot point o_i is inserted between each d_i and d_{i+1} and a second degree Bernstein polynomial is defined between contiguous pairs of points.

This procedure follows the guidelines of Chen and Otto's original paper [Chen95]:

1. With respect to m_i for all i in {2, …, n};
   (a) m_i must be consistent with the monotonicity and convexity of the piecewise linear function determined by the data points d_{i-1} , d_i and d_{i+1}.
   (b) m_i must vary continuously with respect to changes in s_i and s_{i+1} (s_i is defined below as $s_i = (u_i - u_{i-1}) / (x_i - x_{i-1})$).
   (c) Points that are maximum or minimum points are fixed to have a slope equal to zero (i.e. , m_i=0 for al points di such that $u_i > \max(u_{i-1}, u_{i+1})$):

2. With respect to m_1 and m_n :
   (d) Extreme points d_1 and d_n are also required to have a slope equal to zero (i.e. , $m_1=m_n=0$).

3. With respect to knot points:
   (e) Only one knot point should be required between two data points. This is to minimise the complexity of the algorithm.

Membership functions for medical data

The construction of membership functions for medical diagnostics is considered by Amaya and Beliakov in [Amaya95]. The laboratory tests in medical diagnostics are precise in nature. A result of such a test is typically represented as a number, for example, temperature = 37.2ºC, or haemoglobin=14g/dL. Nevertheless, for this precision to be useful we need to specify a normal range, and the specific context of the test. It is common in medicine to use symbols such as ↑↑↑,↑↑,↑,N,↓,↓↓,↓↓↓ to denote variations, where N indicates normal. A physician who performs a given test usually relies on a table of normal ranges and on previous experience, especially when the result is 'borderline', or on a boundary. In practise, a coding with ↓'s and ↑'s is sometimes too restrictive and a scale between 0 and 1 is more convenient. In the medical context, there are two main causes of fuzziness: (i) classification in an under or overdimensioned universe, and (ii) intersubject differences in respect to the membership functions. Fuzziness due to inexact conditions of data recording for precise laboratory tests is not considered as significant.

The usual practise in medicine is to specify the range of 'normal' values for each test. Thus the statement 'haemoglobin=14g/dL' can be interpreted as 'haemoglobin level is normal if 14g/dL is inside this range and otherwise is abnormal'. The range of normal values specifies the necessary context inside which the precise value of haemoglobin may be interpreted. Notwithstanding, problems arise when values are on boundaries. For example, the normal range for haemoglobin is between 14 and 18 g/dL for males at sea level. But, using a crisp boundary has the consequence that 14g/dL is a normal value but 13.99g/dL is abnormal. One solution to this problem is to fuzzify the range of normal values. The remaining problem is that of establishing how to best fuzzify the range.

The boundary problem, in practise, may result in the physician who performs the test considering that a value outside the normal range is normal. This may occur as a consequence of the cause of fuzziness (i) stated previously: classification in an under or overdimensioned universe. If the universe is underdimensioned, the physician who performs the classification can only measure some of the necessary parameters and estimates the remaining ones. In the case that the universe is overdimensioned, the physician possesses additional information but is required to classify using just one criterion, the result of the test in the given context. It is not possible to directly establish the dimension of the universe in each case. Thus, the membership functions are defined in terms of the additional information, that is, the context together with the range of normal values.

It is supposed that sufficient data has been collected to approximate the probability density function p(x) which defines the behaviour of the random variable *x* (the result of the test).

A probability density function (or probability distribution function) is a function p defined on an interval (a, b) and having the following properties.

(a) $p(x) \geq 0$ for every x

(b) $\int_a^b p(x)\, dx = 1$

As a practical example, consider a survey which finds the following probability distribution for the haemoglobin level level of adult males:

Haemoglobin level (g/dL)	10-12	12-14	14-16	16-18	18-20
probability	0.05	0.20	0.35	0.30	0.10

We could represent this data as a histogram and plot a curve of the corresponding distribution. This curve would be the graph of the probability density function p. We note that the sum of the probabilities sum to 1 and the interval of values for haemoglobin level is defined from 10 to 20, where, as previously stated, normal values are considered to be between 14 and 18 for males.

This allows the distance to measured between numbers with respect to a pseudo-Euclidean metric induced by the probability density [Beliakov94]

$$d(x,y) = \left| \int_x^y \rho(t)\, dt \right|$$

A method for smoothing the margin of the crisp set, for example, 'bigger than *a*' is to associate its membership function with the normalised distance from the boundary, thus:

$$\mu_{\geq a}(x) := 1 - \frac{d(x,a)}{d(-\infty,a)} = 1 - \frac{\int_x^a \rho(t)\, dt}{\int_{-\infty}^a \rho(t)\, dt}$$

$$= \frac{\int_{-\infty}^x \rho(t)\, dt}{\int_{-\infty}^a \rho(t)\, dt} \quad \text{, for } x < a \qquad (2.74)$$

and

$$\mu_{\geq a}(x) := 1 \quad \text{, for } x \geq a$$

where $\mu_{\geq a}(x)$ denotes the membership function of the set 'bigger than a', where the quantifier 'bigger' is defined in the fuzzy form. It can be noted that the membership function of the crisp set 'bigger than a' is not modified in the interval $x \geq a$, where $\mu(x) = 1$.Furthermore, a physician would consider the result of the test normal if inside the normal range, thus keeping it possible for the physician to explain his or her decision. In a similar manner, the membership function of the fuzzy set 'smaller than b' can be defined as 1 for $x \leq b$ and to decrease from 1 to 0 in proportion to the distance from b to x for $x > b$. This is formalised in the following two definitions.

Definition 1: the membership function of the fuzzy set "bigger than a" is defined by the formula

$$\mu_{\geq a}(x) := \frac{\int_{-\infty}^x \rho(t)\, dt}{\int_{-\infty}^a \rho(t)\, dt} = \frac{P(t \leq x)}{P(t \leq a)} \quad \text{, for } x < a \qquad (2.75)$$

and

$$\mu_{\leq a}(x) := 1 \quad \text{, for } x \geq a$$

Definition 2: the membership function of the fuzzy set "smaller than b" is defined by the formula

$$\mu_{\leq b}(x) := \frac{\int_x^{\infty} \rho(t)\, dt}{\int_b^{\infty} \rho(t)\, dt} = \frac{P(t \geq x)}{P(t \geq b)} \quad \text{, for } x > b \qquad (2.76)$$

and

$\mu_{\leq b}(x) := 1$, for $x \leq b$

Returning to the consideration by the physician of what are normal values and what are abnormal values, a fuzzy definition is given for each. If the range of normal values is [a,b], for a ≤ b, then the membership function of the fuzzy set 'normal' will be the intersection of the sets 'bigger than a' and 'smaller than b', and is obtained by applying the min operation:

$\mu_{[a,b]}(x) := \min(\mu_{\geq a}(x), \mu_{\leq b}(x))$

The fuzzy set of abnormal values will be the union of the fuzzy sets 'bigger than b' and 'smaller than a'. The membership functions of these sets, $\mu_{\geq b}(x)$, and $\mu_{\leq a}(x)$, are constructed using definitions (1) and (2), and the membership function of the union is:

$\mu_{\neg[a,b]}(x) := \max(\mu_{\geq b}(x), \mu_{\leq a}(x))$

The resulting membership function is shown in Figure 31: on the x-axis of Figure 31, we see the two points 'a' and 'b' which represent the minimum and maximum 'crisp' limits of what are considered 'normal' values. Two fuzzy sets are defined by the membership curves: 'bigger than a' and 'smaller than b'. The intersection of these two fuzzy sets will be the fuzzy set of 'normal', with their corresponding membership grades. The descending curve which starts at point b (μ=1), shows a gradual change from cases with a high membership to the 'smaller than b' fuzzy set, to cases with a low membership to the 'smaller than b' value fuzzy set, and therefore a high membership to the 'greater than b' fuzzy set. The same is evident for point 'a', where the ascending curve on the left indicates cases with an increasingly higher membership to the 'bigger than a' fuzzy set.

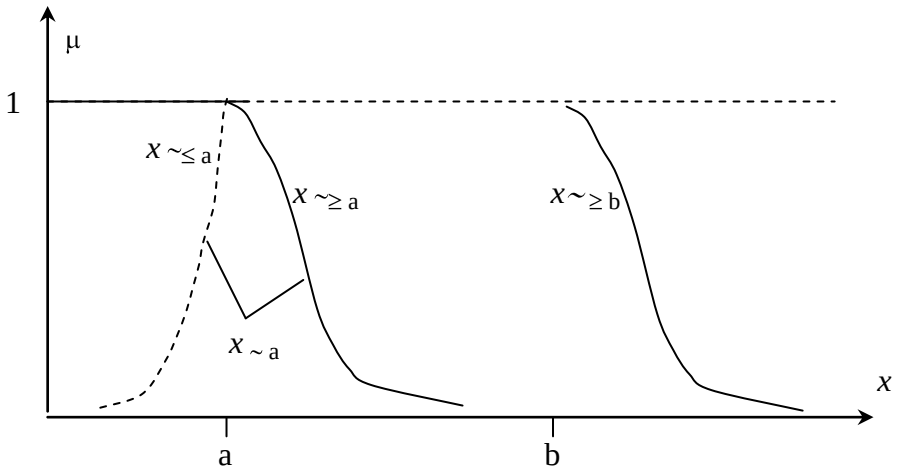


Figure 31. Membership curves of the fuzzy sets corresponding to 'bigger than a' and 'smaller than b' values

In medical tests, two characteristics which are typically used are sensitivity and specificity. These characteristics can be explained in terms of two groups of patients, one of which consists of those who have a given disease, and the other which consists of those who do not. The probabilities that the test result belongs to the interval $[x, x + dx]$ are denoted by $p_+(x)dx$ and $p_-(x)dx$, respectively. These are shown in Figure 32. Negative results are typically associated with small values of x while positive results have values bigger than a given threshold a. The sensitivity of the test is thus defined as the probability of the positive test result for patients having the disease. The specificity is thus defined as the probability of the negative test result for the patients without the disease. The positive or negative predictive value is the probability that a patient has or does not have the disease given a positive or negative result. When the threshold is fuzzified the sensitivity or specificity becomes the probability of a fuzzy event that the test result is positive or negative, respectively.

$$\text{Sensitivity} = \int_{-\infty}^{\infty} \rho_+(x)\, \mu_+(x)\, dx$$

$$\text{Specificity} = \int_{-\infty}^{\infty} \rho_-(x)\, \mu_-(x)\, dx$$

where $\mu_+(x)$ and $\mu_-(x)$ denote the membership functions of the fuzzy sets 'bigger than a' and 'smaller than a', respectively.

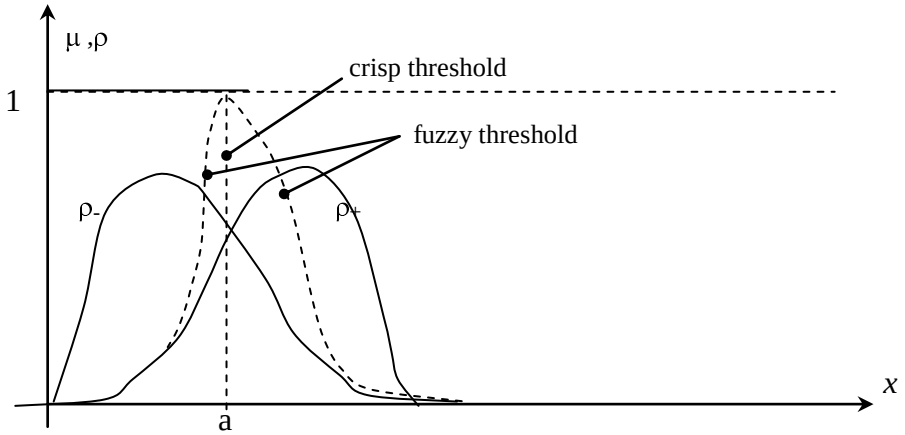


Figure 32. Plot of the probabilities that the test result belongs to the interval [*x*, *x* + *dx*], denoted by $p_+(x)dx$ and $p_-(x)dx$, 'have disease' and 'do not have disease', respectively.

2.4 Factor analysis and attribute fusion

In this section we consider the use of Hartigan's 'Joining Algorithm' [Hartigan75] for factor analysis. This algorithm serves two objectives: the first is that of attribute reduction via the progressive unification of the attributes; the second objective is the identification of the most significant factors and those factors which have most inter-relation. In fact, the second objective is a prerequisite for the first. The details of implementation of this algorithm by Nettleton are given in Section 3.2.2 of the thesis. There are other 'joining algorithms', as detailed in Hartigan's book, and it is interesting to compare their functional differences, as we do later in Section 2.4.

Preliminaries: A matrix of covariances $\{C(I,J),\ 1 \le I \le N,\ 1 \le J \le N\}$, where N is the initial number of variables to be processed, will be approximated by the product of loading matrices B, in which B has a *simple tree structure*. This implies that each column of B has constant non-zero elements (possibly one different constant in different columns) and that the clusters of variables defined by the non-zero elements in each column form a tree. This assumes that the data representation has been adequated to the form of representation which allows the C matrix to be defined. The prerequisite is the ability to calculate the C matrix: for non- numerical data an appropriate distance metric is required to calculate the covariances, and the B matrix is assigned initial default values (Step 1, below).

A covariance matrix **C** is exactly equal to a product of loading matrices of this type, if and only if **–C** is an ultrametric, that is, if and only if for each three variables I,J,K, $C(I,J) \ge \min [C(I,K),C(J,K)]$. We observe that, if **–C** is an ultrametric for a scaling of the variables, it is not necessarily so for another scaling, and as a consequence a careful scaling of the variables may improve the 'fit' of the model.

The algorithm proceeds to find the pair of variables with the biggest covariance and it joins them to construct a new factor, whose covariance with respect to each other variable will be the weighted mean of the covariances of the fused variables for that variable. Then, the next highest covariance will indicate the next pair to be fused. This gives the same result as the 'standard distance and amalgamation' procedure.

During the execution of the algorithm, the clusters (or factors) are constructed: {1,2,.....,2N - 1}.
The first N clusters are the original variables. The structure of the cluster is written in the vector JT, where JT(I) is the cluster constructed by the fusion of I to some other cluster. If clusters I and J are fused to form cluster K, then the loading in the Ith column of the loading matrix will be:

$$\{ C(I,I) - \min[C(I,I), C(J,J), C(I,J)]\}^{1/2} .$$

Step 1: Assign K, the number of clusters, to N. For each I ($1 \le I \le N$) we define WT(I) = 1, JT(I)=0. We define B(I,I) =1 ($1 \le I \le N$) and B(I,J) = 0 for all others I,J ($1 \le I \le N,\ 1 \le J \le 2N-1$).

Step 2: Find the pair $I \ne J$ with JT(I) = JT(J) = 0, such that C(I,J) is a maximum.

Step 3: Increment K by 1. Define JT(I) = K, JT(J) = K, C(K,K) = min[C(I,I), C(J,J), C(I,J)] , WT(K) = WT(I) + WT(J). Define $B[L,I) = [C(I, I) - C(K, K)]^{1/2}$ always such that B(L, I) = 1($1 \le L \le N$). Define $B(L,J) = [C(J,J) - C(K, K)]^{1/2}$ always such that B(L, J) = 1($1 \le L \le N$). Define B(L,K) = 1 always such that B(L, I) or B(L,J) are non-zero ($1 \le L \le N$). Assign JT(K) = 0.

Step 4: For each L, JT(L) = 0, define C(L,K) = C(K,L) = [WT(I) C(I,L) + WT(J) C(J,L)]/WT(K). If $K < 2N - 1$, return to Step 2.

Note 1. There will be 2N-1 clusters, or factors, following the calculations (some of them could have zero loadings and can be discounted). The loading matrix can be reduced to an $N \times N$ matrix in the following manner. We begin with the smallest clusters and go on to the biggest ones. If I and J are fused to form cluster K, we assume that DI, DJ, DK are the corresponding non-zero loadings. If and only if $B(I,L) \ne 0$, we assign $B(I,L) = DI^2/(DI^2+DJ^2)^{1/2}$. Always in the case that $B(J,L) \ne 0$, we assign $B(I,L) = -DJ^2/(DI^2+DJ^2)^{1/2}$. We completely eliminate the column $\{B(J, L) , 1 \le L \le N\}$. We substitute $B(K,L) \ne 0$ by $B(K,L) = [DK^2 + DI^2DJ^2/(DI^2+DJ^2)]^{1/2}$. During this procedure, we eliminate N−1 columns. The basis for eliminating is the colinearity of the columns I,J,K when I and J are fused to form K.

Note 2. The algorithm may be initialised as a mean fusion algorithm, by using Euclidean distances when the variances are all equal to 1 (unity). If $\rho(I,J)$ is the correlation between variables I and J, $D(I,J) = [1-\rho(I,J)] / 2M$ is the squared

Euclidean distance between the standardised variables. The distance between clusters of variables is defined as the mean distance on pairs of variables, one from each cluster. Then we obtain exactly the same fusion sequence over the distances, as the previous algorithm.

The covariance between pairs of clusters is the mean covariance between the variables in the two clusters. It is natural to associate a factor to each cluster that is equal to the mean of all the variables in the cluster, given that the covariance between clusters is equal to the covariance of these two factors. Of course, those factors will be oblique. We obtain another convenient set of factors for a binary tree by the association of a factor with each division in two clusters – the difference of the averages of the variables in the two clusters. These factors are also oblique, while the columns of the loading matrix are orthogonal.

Practical Example of execution of the Hartigan joining algorithm

With reference to the four steps defined previously in the preliminary description of the Hartigan joining algorithm, we now run through the first iteration of execution of these steps using the data defined below in Table 9. The initial state of the data consists of a covariance matrix for the seven variables to be 'joined'.

Step 1: Assign K, the number of clusters, to 7. Define WT(I) = 1, JT(I)=0. Define B(I,I) =1 for $1 \leq I \leq 7$ and B(I,J) = 0 for all others I,J ($1 \leq I \leq 7$, $1 \leq J \leq 13$).

Step 2: The pair FM and FR have the highest covariance, thus I=4, J=5.

Step 3: Increment K to 8. Define JT(4) = JT(5) = 8, C(8,8) = 0.846 [given that C(4,5) is less than C(4,4) or C(5,5)], WT(8) = 2. Define B(4,4) = $[1 - 0.846]^{1/2}$ = 0.392, B(5,5) = $[1 - 0.846]^{1/2}$ = 0.392, B(4, 8) = B(5,8) = 1. JT(8) = 0.

Step 4: The following is defined:

C(1,8) = 1/2[C(1,4) + C(2,4)] = 1/2(0.305 + 0.301) = 0.303.

In the same manner, the other covariances are defined with the new cluster or factor, by calculating the mean of the previous ones. Because K < 13, we return to Step 2, and so on ….

Table 9. Example of applying the fusion algorithm to a simple dataset of measurements

Initial state of data.

1.HL	1000						
2.HB	402	1000					
3.FB	395	618	1000				
4.FM	305	135	289	1000			
5.FR	301	150	321	846	1000		
6.FT	339	206	363	797	759	1000	
7.HT	340	183	345	800	661	736	1000
	HL	HB	FB	FM	FR	FT	HT

Step 1. Fuse FR with FM

HL	1000					
HB	402	1000				
FB	395	618	1000			
FMFR	303	142	305	846		
FT	339	206	363	778	1000	
HT	340	183	345	730	736	1000

Step 2. Fuse FMFR with FT

HL	1000	
HB	402	1000

FB	395	618	1000		
FMFRFT	315	163	778		
HT	340	183	345	732	1000

Step 3. Fuse FMFRFT with HT

HL	1000			
HB	402	1000		
FB	395	618	1000	
FMFRFTHT	321	168	328	732

Step 4. Fuse HB with FB

HL	1000		
FBHB	398	618	
FMFRFTHT	321	248	732

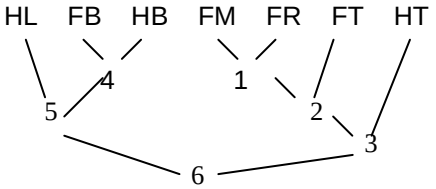
Step 5. Fuse HBFB with HL

HLFBHB	398	
FMFRFTHT	270	732

Step 6. Fuse HLFBHB with FMFRFTHT

HLFBHBFMFRFTHT	270

The resulting tree of sequence of fusions will be:



where the first fusion, FM with FR is denoted by '1', the second fusion of FMFR with FT is denoted by '2', and so on. The pairs with the highest covariances are fused. New covariances are considered to be the weighted means of the previous ones.

Non-fuzzy distance metrics between cases. [Hartigan75]

In [Hartigan75], pp66, the interpretation of the 'distance' between variables of different scales and types is considered (we call this a non-fuzzy interpretation). The process becomes more complex when we introduce variables with distinct scales and types.

(1) For two real variables measured using scales, an appropriate recalculation of scale results in a mean of 0 and variance of 1. It follows that the Euclidean distance is proportional to **1- the correlation**.

(2) For two ordered variables, a feasible distance would be:

$$P(X < X^* \mid Y < Y^* , X \neq X^*),$$

where X, Y and X*, Y* are all taken from a random sample of both variables.

(2) For two categorical variables, which may be ordinal or non-ordinal, P(I,J) is assigned the proportion of cases which have value I for the first variable, and value J for the second variable (supposing that I assumes values 1,2,….,M and J assumes values 1,2,….,N). The following is assigned:

$$P(I, 0) = \Sigma\{1 \le J \le N\}\ P(I, J)$$
$$P(0, J) = \Sigma\{1 \le I \le M\}\ P(I, J).$$

A similarity measure is:

$$\Sigma\{1 \le I \le M,\ 1 \le J \le N\}\ P(I, J)\ \log P(I, J)$$
$$-\Sigma\{1 \le I \le M\}\ P(I, 0)\ \log P(I, 0)$$
$$-\Sigma\{1 \le J \le N\}\ P(0, J)\ \log P(0, J)$$

(3) For a categorical variable and a real variable, a natural similarity measure (invariant when subject to permutation of categories and linear transformation of the real variable) is the ratio of the "between category squared mean" to the "inside-category squared mean".

In [Hartigan75], pp64-65 , Hartigan details calculations on Euclidean and non-Euclidean distances between variables, and also details how to graphically represent (plot) distances to 'detect' clusters.

2.5 Clustering

This section presents the main clustering methods applied in Sections 4.1 and 4.2 of the thesis, namely: Hartigan's 'joining algorithm' [Hartigan75], Kmeans [Dubes88], Fuzzy c-Means [Bezdek81], Kohonen SOM [Kohonen82]. We also summarise Ghosh's [Ghosh95] fuzzy version of the Kohonen SOM. Another system which we review as one of the classification systems in the literature is Linneo+.

LINNEO+

Linneo+ [Béjar94] uses as its starting point the Kmeans algorithm, thus giving it a basis in the traditional statistical field. Notwithstanding, the concept of distance is considered as a fuzzy similarity value, as in [Bezdek81]. The system is considered to be an agglomerate classification method based on the optimisation of a quality function of the obtained groups. A simplification of the algorithm is as follows:

1. Establish the number of desired classes K.

2. Choose K initial objects which we wish to optimise, or produce movements of observations between classes.

   a) Assign each one of the observations to one of the K groups depending on a similarity function.

   b) Calculate the prototype of each one of the K groups as the mean of the values of all the assigned objects.

   c) Calculate the quality function.

Search by 'hill-climbing' method:

This involves a function which establishes the similarity between the observations, the representatives of each group and the form of calculating the prototype of the groups. The function F chosen as the optimisation criteria, is the minimisation of the sum of the distances of the objects O_1 to O_{Ni} of each group to their respective prototypes C_1 to C_k.

$$F = \sum_{i=1}^{k} \sum_{j=1}^{N_i} d\,(O_j , C_i) \qquad (2.77)$$

The following datasets were used in [Béjar94] to test the algorithm: 'marine sponges', 'mental illness' and 'water treatment'.

The 'classification step' used by Linneo+ :

In [Bejar94], the distance used for determining the similarity between two objects O_i and O_j, is the generalised hamming distance:

$$d\,(O_i , O_j) = \sum_{k=1}^{n} diff\,(O_{ik} , C_{jk}) \qquad (2.78)$$

where *diff* (O_{ik} , C_{jk}) depends on the type of the attribute *k* as follows: if k is a qualitative attribute (ordinal or nominal categorical) the expression *diff* (O_{ik}, C_{jk}) will be 0 if the values are equal and 1 if they are different. If *k* is a quantitative attribute (a number) the expression evaluates to the absolute value of the difference between the two values. If one of the values of *k* is missing or null, its value is assigned ½ , and if both values of k are missing or null, they are both assigned 0.

The following types of distances are considered: Minkowski's metric, Mahalonobis distance χ^2. (chi-squared) distance and the Cosine distance. The Cobweb system [Fisher87] is also contrasted. Cobweb is a concept formation system which incorporates metrics and heuristics to calculate terms of 'predictability' and 'previsibility'.

Linneo+ uses a strategy of non supervised learning. A number of non-preestablished concepts are induced, in the case in which there is no 'expert' available who knows the concepts to be learned in advance. Linneo+ incorporates a methodology for non supervised learning for concept creation and the automatic construction of knowledge bases derived from sets of unclassified observations in 'ill-structured' domains.

Some of the key areas of Linneo+ are: 'incremental learning', the introduction of information which describes the observations and which allows the qualification of the values assumed by the attributes. It also contains heuristics which mitigate the dependency of the results of the incremental algorithm on the order of input of the observations. Semantic information is expressed in a declarative form to achieve a semantic bias of the results. Some of the quality measures for a classification, used for benchmarking with other methods, are: mean distance between the class centres; Mean distance between the objects and the centres of their classes; attribute dispersion; mean attribute dispersion, and the number of attributes with zero dispersion.

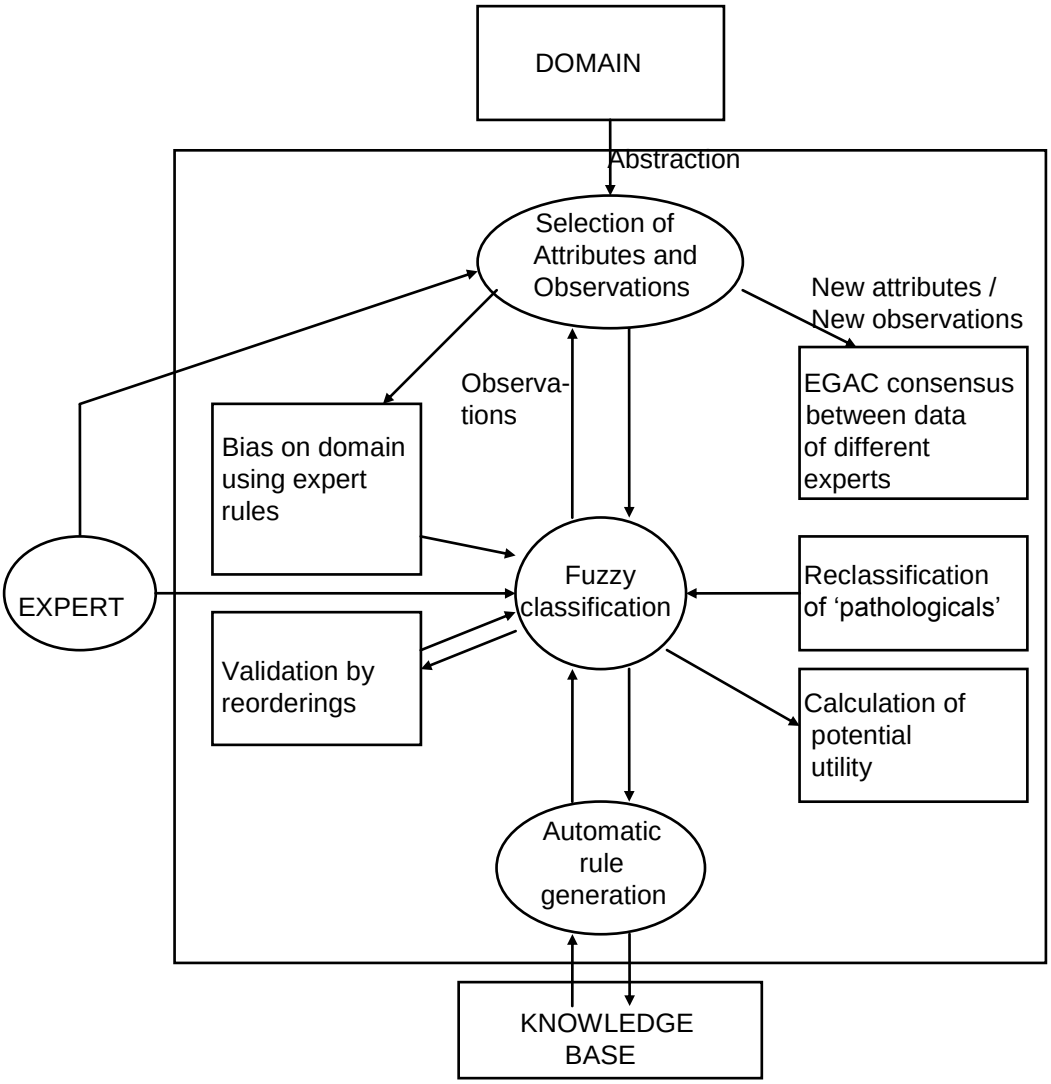


Figure 33. Functional representation of Linneo

Hartigan's clustering algorithms

In [Hartigan75], Hartigan defines CLUSTER, which is a sublibrary of Fortran subroutines for cluster analysis and related line printer graphics. It includes routines for clustering variables and/or observations using algorithms such as direct joining and splitting, Fisher's exact optimisation, single-link, K-means, minimum mutations, and routines for estimating missing values.

The types of clustering algorithms used are:

Sorting – an important variable is chosen somehow, and the observations are partitioned according to the values taken by this variable. Within each of the clusters of the partition, further partitioning takes place according to further important variables

Switching – an initial partition of the observations is given, and new partitions are obtained by switching an observation from one cluster to another, with the algorithm terminating when no further switch improves some criterion.

Joining – initially each observation is a cluster and the closest pair of clusters are joined to form a new cluster, continuing this step until a single cluster containing all the original observations is obtained.

Splitting – initially all observations are one cluster and then a cluster is chosen according to some criterion and split into smaller clusters.

Adding – a clustering structure (partition or tree) already exists, and each object is added to the closest cluster by some criterion in turn.

Searching – search over a subset of all possible clusters for the optimal one.

Fuzzy c-Means

As a precursor to Fuzzy c-Means, we will first comment Kmeans [Dubes88], given that Fuzzy c-Means is based on this essentially simple algorithm. Kmeans is is a crisp algorithm which establishes a distance between the observations, in which a predefined number of observations are selected to be used as seeds for the cluster construction process. Each observation is assigned a cluster by an iterative algorithm in the closest cluster as defined by the distance between the object to be clustered and the mean value of all the clusters. The clustering stops after a predefined number of iterations, in which point the current clusters are settled as the result of the clustering. We note that Kmeans is an algorithm which requires us to know *a priori* how many clusters we wish to cluster the data into. Hartigan and Wong later developed a supervised version of Kmeans [Hartigan79].

Fuzzy c-Means [Bezdek81] is a clustering algorithm which acts on cases described by numerical attributes. It is able to establish the best number of partitions for a given data set, by testing different numbers of partitions and using the cluster quality indicators to identify the best. It calculates the centres of the fuzzy clusters for the chosen number of clusters. It then calculates a membership grade for each case to each cluster, for each variable. From this information, by inspecting the calculated values, we can establish which variables are most significant for each cluster, and which are ambiguous.

Fuzzy c-Means is later applied, towards the end of Section 4.1 of the thesis, to the Hospital Clinic ICU data in order to generate a fuzzy clustering of the cases and of the variables and their inter-relationships, as indicated by the fuzzy cluster centres. This is contrasted with the analysis of the same data with statistical techniques such as principal components and the Hartigan 'joining algorithm', and AI techniques such as ID3, C4.5, feedforward neural nets and the Kohonen SOM.
Fuzzy c-Means [Bezdek77] generalises the function J_W , sum of squared error within groups. It suggests many infinite families of clustering algorithms, which have been developed upon by different investigators.

Algorithm

Clustering algorithms such as fuzzy c-Means are essentially 'Picard Iteration' for a set of given conditions.

Step 1: Fix **c** , $2 \leq \mathbf{c} < n$; select any interior product metric norm for $\Re^p$; and fix m, $1 \leq m < \infty$. Initialise $U^{(0)} \in M_{fc}$. Then to iteration l, l=0,1,2,.....,:

Step 2: Calculate the centres for the c-fuzzy clusters $\{ v_i^{(l)} \}$ with $U^{(l)}$.

Step 3: Actualise $U^{(l)}$ using $\{v_i^{(l)}\}$.

Step 4: Compare $U^{(l)}$ with $U^{(l+1)}$ a convenient matrix type norm: if $\|U^{(l+1)} - U^{(l)}\| \leq \varepsilon_L$ the algorithm terminates: otherwise, return to Step 2.

Calculation of cluster quality

For the number of clusters defined as input parameter ('kbegin' and 'kcease') the algorithm evaluates each cluster, calculating the following: Fstop, 1-Fstop, Entropy and Payoff. Two of the objectives are: maximise the grade of partition, and minimise the entropy. If kbegin=2 and kcease=3, the algorithm executes for 2 partitions and for 3 partitions, calculating the clusters, the centroids, and the quality indicators. The values for 'Payoff' can then be interpreted in order to identify the most favourable value of **c**.

Description of the 'Fuzzy c-Means' parameters

Fuzzy c-Means has the following algorithmic parameters: **c** , **m** , $\mathbf{U^{(0)}}$, $\|.\|_A$, ε_L .

c : is the number of expected clusters. This can be fixed (for example to 2) and the algorithm will try to create **c**=2 clusters/partitions in the data set. Or one can increment **c** , in an iterative manner (**c**=1,2,....., n) and compare the quality of the results for each **c** .

m : the bigger **m** is, the more 'fuzzy' the membership assignments will be. It is a grade of 'fuzziness', or a *weight exponent* which controls the grade of sharing of memberships between fuzzy type clusters in **X**. Typical values are 2, 1.25,1 , etc. ..

$\mathbf{U^{(0)}}$: matrix which contains the membership functions, with their initial value assignments.

$\|.\|_A$: is a norm induced on $\Re^p$ of internal product. $\Re^p$ is a space of p-tuples of real numbers. For example, the following three norms can be defined: N_E , the Euclidean norm; N_D , the Diagonal norm, and N_M , the Mahalonobis norm. N_D is typically used to compensate for distortions due to great differences between the variance of characteristics in samples in the directions of the co-ordinate axes. N_E is typically used when the clusters in X have the general appearance of 'spherical clouds'.

ε_L : is the umbral epsilon, typical value being 0.01, which works as a 'cutoff' criteria, among the cluster centroids.

Norms

By varying the norm for a distance based clustering criteria, we may infer geometric and statistical properties from the data. The norm is effectively one of the parameters of fuzzy c-Means which needs to be most adequately set, in order to generate fuzzy clusters as a result with a good cluster quality; that is, compactness, distinguishability between clusters, and interpretability in terms of the characteristics of the dataset. For example, in pattern recognition, two fuzzy clusters could be generated for two geometric shapes, one of which contains the points of a cross, and the other which contains the points of a circle. The three principal norms are used for different classes of characteristics in the data sets:

> $\mathbf{N_E}$: where the characteristics are statistically independent, and variable in the same measure for clusters with a hyperspherical form
>
> $\mathbf{N_D}$: where the characteristics are statistically independent, and variable in unequal measures for clusters of a hyperellipsoid form

$\mathbf{N_M}$: where the characteristics are statistically dependent, and variable in unequal measure for clusters of a hyperellipsoid form

Typical output from Fuzzy c-Means

Below we can see a typical output generated from a fuzzy c-Means run, for which the number of clusters has been defined as 3. In the first block we see the input data, which consists of two columns of 21 cases. The scalar matrix *cc* is then shown, which is used to apply the chosen norm, for example, Diagonal, Euclidean or Mahalonobis. In this case the norm is set to 2, which corresponds to Euclidean. The MM-Clusters block then shows the successive iterations, in this case 8, with the maximum error incurred in each iteration. The termination of the algorithm can be defined as a maximum number of iterations or a minimum error. In this case the error was progressively reduced in each iteration to reach a best value of 0.0011 in iteration 8. The state at termination is given in the indicators *Fstop*, *Entropy* and *Payoff*. The cluster centres (or prototypes) at termination are then given, there being 3, each located by an *x*,*y* coordinate. The block giving the membership functions is then listed. There are three membership functions, one for each cluster, and these functions are defined over the number of cases. That is, each case has three values assigned to it, being the fuzzy grade of membership to each of the three clusters. For example, case 14 (J=14), has a grade of membership of 0.0118 with respect to cluster 1, G.O.M. of 0.9737 for cluster 2, and 0.0145 for cluster 3. It is clear that case 14 has a strong membership to cluster 2, and very weak memberships to clusters 1 and 3.

***** *** Begin Fuzzy C-Means Output *** *****
MM-Clusters

INPUT DATA

y[1][1]=	0.00	y[1][2]=	0.00
y[2][1]=	0.00	y[2][2]=	3.00
y[3][1]=	1.00	y[3][2]=	1.00
y[4][1]=	1.00	y[4][2]=	2.00
y[5][1]=	2.00	y[5][2]=	1.00
y[6][1]=	2.00	y[6][2]=	2.00
y[7][1]=	3.00	y[7][2]=	0.00
y[8][1]=	3.00	y[8][2]=	3.00
y[9][1]=	10.00	y[9][2]=	9.00
y[10][1]=	10.00	y[10][2]=	10.00
y[11][1]=	10.50	y[11][2]=	9.50
y[12][1]=	11.00	y[12][2]=	9.00
y[13][1]=	11.00	y[13][2]=	10.00
y[14][1]=	18.00	y[14][2]=	0.00
y[15][1]=	18.00	y[15][2]=	1.00
y[16][1]=	18.00	y[16][2]=	2.00
y[17][1]=	19.00	y[17][2]=	0.00
y[18][1]=	19.00	y[18][2]=	2.00
y[19][1]=	20.00	y[19][2]=	0.00
y[20][1]=	20.00	y[20][2]=	1.00
y[21][1]=	20.00	y[21][2]=	2.00

Number of cases = 21 MM-Clusters

Scalar matrix cc

0.1	0.1
0.0	0.0
0.0	0.0
0.3	0.3

MM-Clusters

Number of clusters = 3 icon = 2 exponent = 2.00

Iteration =	1	Maximum Error =	0.6119	Number of clusters = 3
Iteration =	2	Maximum Error =	0.3242	Number of clusters = 3
Iteration =	3	Maximum Error =	0.2245	Number of clusters = 3
Iteration =	4	Maximum Error =	0.3035	Number of clusters = 3
Iteration =	5	Maximum Error =	0.3529	Number of clusters = 3
Iteration =	6	Maximum Error =	0.1827	Number of clusters = 3
Iteration =	7	Maximum Error =	0.0114	Number of clusters = 3
Iteration =	8	Maximum Error =	0.0011	Number of Clusters = 3

Fstop	**1-Fstop**	**Entropy**	**Payoff**
0.957	0.043	0.112	6.648

Cluster Centres v[i][j]

V[1][1] = 10.4936 V[1][2] = 9.4918
V[2][1] = 18.9947 V[2][2] = 0.9966
V[3][1] = 1.4903 V[3][2] = 1.4886

Membership Functions

J = 1	0.0223	J = 1	0.0244	J = 1	0.9533
J = 2	0.0366	J = 2	0.0240	J = 2	0.9394
J = 3	0.0031	J = 3	0.0031	J = 3	0.9938
J = 4	0.0040	J = 4	0.0033	J = 4	0.9928
J = 5	0.0034	J = 5	0.0035	J = 5	0.9931
J = 6	0.0043	J = 6	0.0037	J = 6	0.9919
J = 7	0.0256	J = 7	0.0341	J = 7	0.9403
J = 8	0.0462	J = 8	0.0329	J = 8	0.9208
J = 9	0.9925	J = 9	0.0035	J = 9	0.0040
J = 10	0.9934	J = 10	0.0031	J = 10	0.0035
J = 11	1.0000	J = 11	0.0000	J = 11	0.0000
J = 12	0.9924	J = 12	0.0038	J = 12	0.0038
J = 13	0.9933	J = 13	0.0033	J = 13	0.0033
J = 14	0.0118	J = 14	0.9737	J = 14	0.0145
J = 15	0.0034	J = 15	0.9930	J = 15	0.0036
J = 16	0.0173	J = 16	0.9678	J = 16	0.0149
J = 17	0.0087	J = 17	0.9814	J = 17	0.0100
J = 18	0.0125	J = 18	0.9772	J = 18	0.0103
J = 19	0.0108	J = 19	0.9775	J = 19	0.0117
J = 20	0.0031	J = 20	0.9940	J = 20	0.0029
J = 21	0.0153	J = 21	0.9727	J = 21	0.0120

Number of Cases N = 21
Number de Characteristics NDIM = 2
'Default' Membership Limit EPS = 0.010
Norm for this Test ICCN = 2
Weight Exponent M = 2.00

No. of Clusters (C)	Part. Coeff. (F)	Inferior Limit (1-F)	Entropy (H)	Number of Iterations (IT)
2	0.830	0.170	0.272	11
3	0.957	0.043	0.112	8

***** *** Normal Termination of Process *** *****

Kohonen SOM (Self Organising Map)

The algorithm designed by Teuvo Kohonen [Kohonen82] falls into the family of algorithms known as 'self-organising maps'. It contains a matrix of nodes, which 'compete' to win weight and to attract the given input data. As a consequence, after successive iterations, some groups of nodes (clusters) will become more highly activated, while other nodes will become relatively disactivated. The nodes are interconnected in a typical neural architecture, and the information propagates from an input layer to a layer (or matrix) of classification nodes. In the basic version there are two node layers, one input and the other in which the classification is formed. The Kohonen architecture has demonstrated its applicability to a diversity of data domains, especially those with large volumes and many attributes. It behaves well in the presence of 'noise' and unknown values.

Description of the functionality of the Kohonen self-organising map

Kohonen made the observation that some nets of flat topology, consisting of interconnected and adaptive units, is able to modify its internal state to reflect the characteristics of a set of input signals. The Kohonen SOM is a set of processors which organise themselves in an autonomous manner, only requiring the original inputs and an algorithm to propagate changes in the net. The state of the net resides in the weights (coefficients) assigned to the interconnections between the units. It has two layers: layer one contains inputs nodes and layer two contains 'output' nodes. The modifiable weights interconnect the output nodes to the common input nodes, in an extensive manner. In other words, the point density function of the weight vectors tend to approximate to a probabilistic density function p(x) of the input vectors **x**, and the weight vectors tend to order themselves in agreement with their mutual similarity.

Terminology

The following summarises the terminology used in the explanation of the functionality of the Kohonen clustering model.

X = {x0, x1, x2,...,xN-1} represents a set of N inputs in R_m such that each xi has m dimensions (or characteristics).

m = number of input nodes

c = number of output nodes (clustering).

Wj is the vector [w0j , w1j,...,w(m-1)j]T which corresponds to the output node j, where ($0 \leq j \leq c-1$).

The output dij = (xi-Wj)T(xi-Wj) is the output node j when presented with input vector xi, where Wj is the vector which contains the weights of the m input nodes up to output node j.

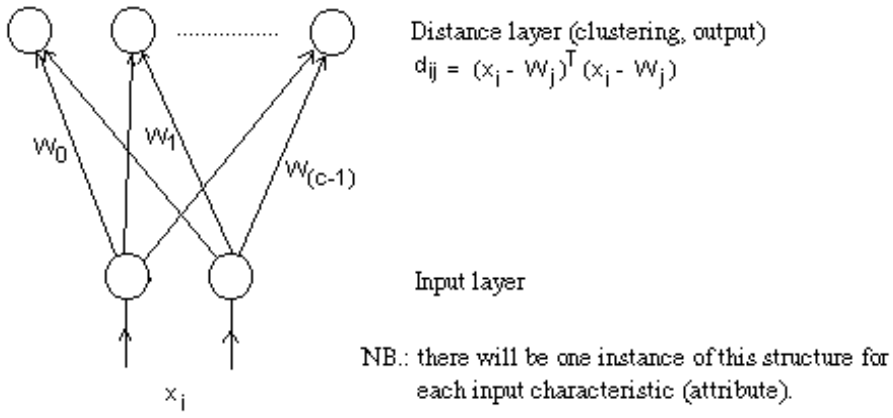


Figure 34. Neural network architecture for Kohonen 'Self Organising Map'

Basic algorithm

The global objective is to move the weights towards the cluster centres via the updating of the weights by each input value.

Step 1: initialise the weights Wj in a random fashion for all j and assign the size of the neighborhood (EN) to c/2. Assign all weight updates ΔWj to zero. The value of the learning rate (Lrate) is initialised to be between 0 and 1.

Step 2: for each input xi, select the output node j*, ($0 \leq j^* \leq c-1$), such that dij* is a minimum. Actualise Wj, using the rule:

$$Wj=Wj+Lrate*\Delta Wj,$$
$$\Delta Wj=\Delta Wj+(xi-Wj),$$

where j includes the output node j* and each of its NE neighbours to the right and left. Step 2 is repeated until there is no change in the weights.

Step 3: check if NE=0. If this is so, the algorithm terminates, otherwise it reduces NE by 1 and returns to step 2.

Fuzzy self-organising maps [Ghosh95]

The following details the modifications carried out by Ghosh in order to incorporate fuzzy techniques into the SOM, and the possible applicability of this approach to the work of the thesis.

Ghosh, in his paper [Ghosh95] first gives a review of what is understood by the measurement of the fuzziness of a fuzzy set: expressing the average presence of ambiguity in deciding if an 'element' belongs to a determined set or not. Later, he describes SOM (self organising map) type multilayer nets.

The final use to which Ghosh puts his system is for image processing. Thus, the way the NN processes and its understanding of fuzziness are orientated towards the interpretation of pixels by grade of lightness (1=totally dark, 0=totally white, intermediate scale $0 < n < 1$ indicates shades of grey).

Motivation: one problem associated with image interpretation is the 'noise' present in the image, and Ghosh proposes that incorporating a measure of fuzziness is a natural way of improving the result.

The intention is to extract homogeneous regions in the space, using a self-organising process, and only using an image representation corrupted by noise (it is not necessary to know the output classes *a priori*). In the networks organisation, under ideal conditions of an image without noise, the output status of the majority of the neurones in the output layer will be 0 or 1. Notwithstanding, due to the effect of noise, the output status of the neurones in the output layer will normally lie in [0,1] and therefore the status value will represent the grade of lightness (or darkness) of the corresponding pixel in the image.

Thus, we could consider the status of the output in an output layer as the representation of a fuzzy set "light pixels (dark)". The fuzzy measure of this set, in global terms, could be considered as the 'error or instability of the complete system', given that it reflects the deviation of the desired state of the net. Thus, when we have no *a priori* value for the output objective, we can take the grade of fuzziness as a measure of the error in the system, and back-propagate it to adjust the weights, such that the error in the system reduces with time and in the limiting case, becomes zero.

We can take the measurement of error **E** as a function which measures the fuzziness:

$$\mathbf{E} = g (\mathbf{I}) , \tag{2.79}$$

where **I** is the measure of fuzziness of a fuzzy set.

When the network stabilises, the output status of the neurones in the output layer will be 0 or 1. Neurones with an output of 0 compose one group and those with 1 will be the other group. The mathematical derivation of the update rules with different fuzzy measures follows four steps: (i) correction of weights for fuzzy index; (ii) correction of weights for entropy; (iii) correction of weights for Kosko's entropy measure; (iv) correction of weights for correlation measure. Refer to paper [Ghosh95] for complete formulae.

2.6 Classification

In this section we review some of the classification systems in the literature, namely: Klass, ID3 and C4.5. Special emphasis is given to C4.5, as this is used extensively later in Section 4.1 of the thesis for comparative benchmarking. C4.5 is discussed with consideration of possible improvements, its development, and the incorporation of fuzzy techniques into the rule induction process.

KLASS

Klass [Gibert94] is a parametric classification tool for ill-structured domains, which uses heuristics together with symbolic and qualitative information. It uses parametric criteria for distance aggregation metrics. In [Gibert94], three main application datasets are used for benchmarking, namely, 'marine sponges', 'computers' and 'stars'. Comparison is made with standard statistical methods, such as SPSS and SPAD.

Klass allows for the incorporation of partial and semantic information, performing a classification by reciprocal chained neighbours (of quadratic complexity). The end user can intervene in the rules and in the variables derived from observations. The system can work jointly with qualitative and quantitative variables.

With respect to the distance between individuals, a family of mixed distances $d^2_{(\alpha r,\beta r)}$ (i , i') is defined, together with a method for obtaining satisfactory values for the parameters (α_r,β_r).

An iterative work methodology is used to incorporate observational data and expert knowledge. Support tools are provided for interpretation of the classes, the objective being to achieve a satisfactory classification, in agreement with the objectives of the expert.

Support for interpreting the classes includes the following: measure of the difference between two classes; measure of the quality of a classification; characterisation of a classification; detection of identifier variables (class prototype); explicative power; automatic rule generation. The rules are in a format which is compatible to allow their incorporation into the knowledge base for diagnostic purposes.

Description of the ID3 rule induction algorithm

ID3 [Quinlan86] constructs classification decision trees using a 'top-down' induction method', and is the predecessor to C4.5. With reference to Figure 35, the most global concept is 'length', given that it appears highest in the tree, and 'weight' is the most specific concept, given that it appears in the lowest part (terminal nodes, or leaves) of the tree. ID3 adds the following information (in parenthesis in Figure 35) to each node: (n,m) where **n** is the number of individuals who correspond to the given branch or node, and **m** is the confidence measure for the given branch or tree.

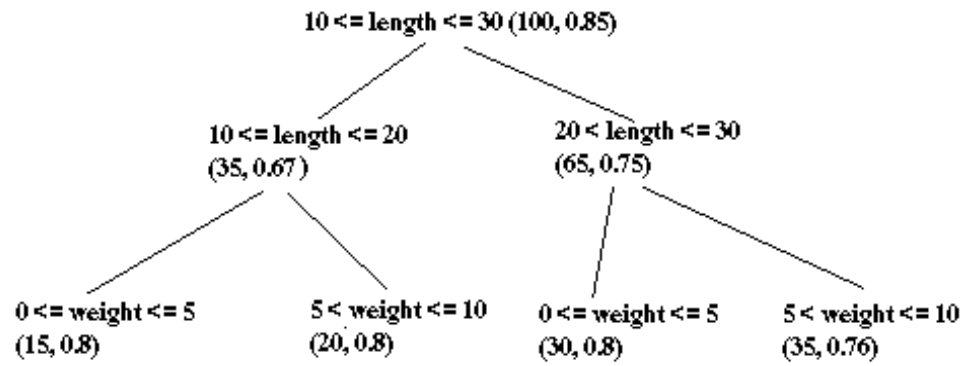


Figure 35. Example of a tree constructed by ID3, using attributes 'length' and 'weight'

ID3's objective is to construct a reasonably good decision tree (although not necessarily the best possible one), without too much computation. The type of data set with which it works would have many attributes and where the test data set has many objects. There is no guarantee of finding the optimum solution (best tree).

ID3 is an iterative algorithm. It randomly chooses a data subset from the training set (called the 'window') and it constructs a decision tree from it. This tree must correctly classify all the objects in the window. Then it tries to classify all the other objects in the complete training set using this tree. If the tree succeeds for these objects then it is correct for the complete data set, and the process terminates. Otherwise, a selection of the objects which were incorrectly classified are incorporated into the window, and the process is repeated. In this manner, the correct tree may be found after just a few iterations, for data sets with up to 30,000 objects, characterised by up to 50 attributes. The design of ID3 had an anecdotal component, given that Quinlan used a window on the subset instead of using the whole test data set, due to memory restrictions of the computers used at that time.

The C4.5 Algorithm

C4.5 [Quinlan93] is an induction algorithm which, from subsets (windows) of cases extracted from the complete training set, generates rules and evaluates their goodness using criteria which measure the precision in classifying the cases. The main heuristics (see below) that are used are the information value which a rule provides (or tree branch) calculated by 'info' and the global improvement that a rule/branch causes ('gain').

The algorithm is executed in successive iterations. In each iteration the window size is incremented in a given percentage (in proportion to the complete set). The objective is to obtain rules which correctly classify a successively greater number of cases in the complete dataset. The proposal is that it is easier to identify rules in a reduced size subset than in the complete dataset. Each iteration uses as its basis that which the previous iteration has achieved.

In each iteration a submodel is executed against the remaining cases (those which are not in the window). The incorrectly classified cases are given precedence to be included in the next window (which will be x% bigger than the previous window). In this manner, the rules continually increase their precision on the complete dataset. The inputs to C4.5 are the rows of cases with data for the selected attributes which must be representative. It is assumed that the distribution of cases and attribute values is well balanced. The output is a value which is related to all the inputs. Normally (in supervised learning) the output is provided to the training version of the model.

Induction of rules and decision trees; concept formation in the ID3 and C4.5 algorithms

ID3 [Quinlan86] and C4.5 are induction algorithms which extract structure and classes from data. Both algorithms are attributable to Quinlan. ID3 exists since the year 1986 while C4.5 was came into being in 1993, and can be considered as ID3's successor, the basic idea (windowing) of both algorithms being similar. In the case of ID3, the objective attribute is always a continuous value, while for C4.5, it is symbolic. C4.5 also constructs more compact trees than ID3.

C4.5 incorporates the following advancements with respect to ID3:

(a) instead of choosing training cases to form the window in a random fashion, C4.5 biases the selection to make the class distribution more uniform in the initial window.

(b) ID3 uses a fixed limit for the number of exceptions per cycle. C4.5 includes as a minimum, 50% of the exceptions in the next window; this results in a faster convergence towards the final tree.

(c) C4.5 terminated the construction of the tree if the precision is not improving, without having to classify all the classes.

Description of C4.5 induction process

C4.5 induces the structure of a data set in two possible forms:

(a) Tree type representation.
(b) Rule type representation.

As inputs it may receive symbolic or continuous values, although it can only associate the attributes to a symbolic objective (output), which could be, for example, a range, flag, category, and so on.

It is more concise than ID3 and generates smaller trees. The objective attribute must be symbolic (not numeric), in contrast to ID3, which allows the objective attribute to be numerical.

Tree type representation

This program generates a classifier in the form of a decision tree. The structure could be:

- a leaf, indicating a class
- a decision node, which specifies a test to be carried out on one value-attribute, with a branch and subtree for each possible result of the test.

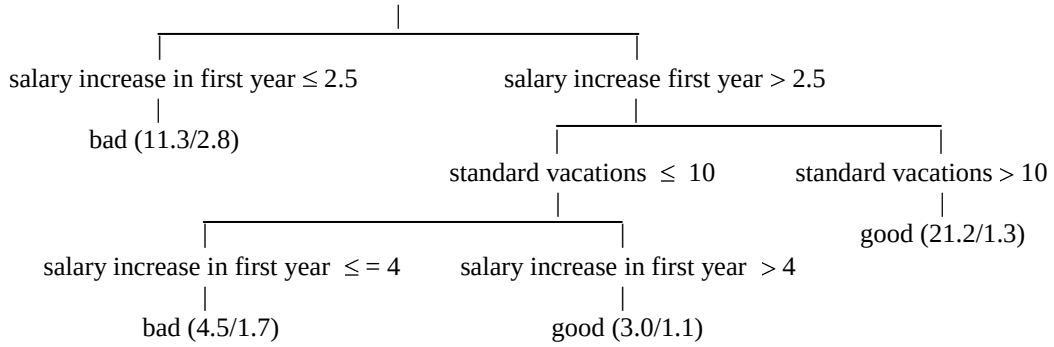


Figure 36. Example of a decision tree generated by C4.5

Interpretation of Figure 36: if the salary increase in the first year is greater than 2.5 and the standard number of days of vacation is less or equal to 10 and the salary increase in the first year is greater than 4, the employee class is 'good'. The numbers in parenthesis indicate the number of training cases associated with each leaf, and the number of cases incorrectly classified by that leaf.

C4.5 contains heuristic methods to simplify the generated decision trees, whose objective is to produce a more comprehensible structure, without losing precision on hidden cases.

It is based on the classic method of 'divide and conquer' [Hunt75], although it introduces a series of improvements with respect to Hunt's original method. The first improvement is in the evaluation of the tests/questions that are made to divide up the cases. Also, a 'benefit criteria' is introduced whose purpose is to quantify and maximise the information increase in the global system. For each candidate division, the 'benefit' is calculated and the best are chosen.

Treatment of unknown values.

For real data, it is easy that something like 30% of the cases have some value-attribute missing. An induction algorithm can incorporate heuristics to manage this type of data set. The calculation of 'benefit' can be modified in terms of the information content (see previous section) in the following manner:

$$\begin{aligned} \text{benefit}(X) &= \text{probability that A is known} \times (\text{info}(T) - \text{info}_X(T)) \\ &+ \text{probability that A is not known} \times 0 \\ &= F \times (\text{info}(T) - \text{info}_X(T)) \end{aligned} \qquad (2.80)$$

where:

info(T) measures the mean information necessary to identify the class of a case T.

$\text{info}_X(T)$ measures the expected information requirement. It is the weighted sum with respect to

the subsets.
T = set of training cases.
benefit(X) measures the information obtained by partitioning T using test X.

Partition of the test data set.

C4.5 has a probabilistic approach to partitioning. If we assign a case belonging to T (see previous section) with a known result Oi and a subset Ti, this indicates that the probability of the given case belonging to subset Ti is 1 and in all other subsets it is 0.

When the result is unknown, we are only able to make a weaker probabilistic inference. For this reason a weight is associated to each case in each subset Ti, which represents the probability that the case belongs to each subset.

Rule type representation.

A classification model in a tree form may also be represented in the form of rules. When the tree reaches a given complexity, it may become difficult to interpret. The rule form is easier to understand, although it contains the same information as that of the tree.

Example of a rule:

Rule 5:
salary increase in first year > 2.5
standard vacations > 10
→ class is good [93.0%]

Interpretation: if the salary increase in the first year is greater than 2.5 and the number of days of standard vacations is greater than 10, then the employee is a member of class 'good' with a probability of 93%.

Definitions of the calculations and criteria used in generating a tree

Once that T (the training data set) has been partitioned in accordance with the n results (outcomes) of a Test X, the forecast for the required information will be the weighted sum of the subsets:

$$\text{info}_X(T) = \sum_{i=1}^{n} \frac{|T_i|}{|T|} \times \text{info}(T_i) \qquad (2.81)$$

The quantity

$$\text{gain}(X) = \text{info}(T) - \text{info}_X(T) \qquad (2.82)$$

measures the information obtained by partitioning T in accordance with Test X. Therefore, the benefit criteria selects a test in order to maximise the information obtained. We may also interpret this result as the information mutually shared by Test X with the class.

Split Info.

$$\text{split info}(X) = -\sum_{i=1}^{n} \frac{|T_i|}{|T|} \times \log_2 \left\lceil \frac{|T_i|}{|T|} \right\rceil \qquad (2.83)$$

represents the potential information generated by dividing T in n subsets. The information 'gain' measures the information which is relevant to the classification and which has arisen due to this same division.

Therefore,

$$\text{gain ratio}(X) = \text{gain}(X) / \text{split info}(X) \quad (2.84)$$

expresses the proportion of information generated by the most promising division (which improves the classification).

Construction of the decision tree of cases Fp to Lp

{Routine 'TreeForm(Fp, Lp)'., [Quinlan93], pp136.}

(a). If all cases are in the same class, the tree is a leaf and it returns the labelled leaf for this class.

(b). For each attribute, calculate the potential information contributed by a test on the attribute (based on the probabilities that each case has a particular value for the attribute) and the information 'gain' that would result from a test on the given attribute (based on the probabilities that each case with a given value for the attribute, has to be in a given class).

(c) Based on these values, and depending on the current selection criteria, find the best attribute on which to branch. Note: this version only allows to divide on an attribute if it possesses two or more subsets with at least MINOBJS cases.

(d) Check if 'branch and test' is better than forming a leaf.

Dependencies of Programs, C4.5

The code of the suite of programs which composes 'C4.5' consists of some 31 programs of 'C' code, and some 5 'include' files. The major components are: decision tree generator, production rules generator, decision tree translator, production rules translator.

Discussion of possible areas of improvement for C4.5

C4.5 is considered as one of the best existing algorithms for rule induction from data. Notwithstanding, Quinlan himself has indicated several areas for improvement, which not only apply to C4.5, but to rule and tree induction techniques in general.

Limitations of C4.5

Geometrical interpretation.

The interpretation of the description space is geometrical, which is the basis for the 'divide and conquer' methodology. If there exist N attributes, this vector corresponds to a point in a Euclidean description space of N dimensions. If each case is described by a vector of 16 attributes, the description space will have 16 dimensions.

The space becomes more complex when we also have to consider unknown values of attributes.

Also, the resulting description space is not really Euclidean, because the distances and distance relations may alter if we reorder the discrete values along the axes. For example, when a mixture of numeric and symbolic values exist together (e.g. size, form: 10x10, square), the order of the values of 'form' will not possess any intrinsic significance.

Non-rectangular regions

The regions produced by a decision tree are hyperrectangles. Notwithstanding, it is possible that the regions of classes in the description space are not hyperrectangles. Thus, the decision tree approximates the regions with hyperrectangles.

One indication that the surfaces that delimit the regions are really not orthogonal hyperplanes, is when the percentage of incorrect classifications maintains constant while the tree size keeps increasing with the incorporation of more training cases.

Quinlan advises that in this situation, an arithmetical combination of the attributes should be sought, to be included as a new attribute. The CART system calculates weights $\{w_i\}$ which maximise the value of a division under current criteria.[Utgoff91] employ this type of tests, calculating the weights by 'hill-climbing'.

$$\sum_i w_i \text{ x } A_i > Z \qquad (2.85)$$

This test is distinct to one which simply compares an attribute with a threshold.

Ill-delimited regions

Decision trees are constructed by successive refinement. This can give rise to 'pathological' cases as a consequence of the topology of the description space. For example, regions with a low density of points may occur, which results in a wide margin in which to place the delimiting surfaces. Also, given that the work of classification is a probabilistic one, the 'objective' regions may contain a substantial number of points that do not belong to the majority class.

Region fragmentation

From the geometrical point of view, a good classifier should divide the description space into a minimum of regions, each one with a high density of points, all of the same class. Ideally this would result in few classes, few regions per class, many cases relative to the volume of the regions and total accuracy for the training cases.

Notwithstanding, in practise we may encounter 'pathological' situations in which the description space becomes highly fragmented, resulting in a proliferation of regions. Two examples of this are (i) the parity problem, in which the complexity of the concept of parity is the cause of the problem, and (ii) the presence of irrelevant attributes.

In order to combat fragmentation, Quinlan advises that as well as eliminating irrelevant attributes, we can use new derived attributes, with an enhanced information value.

Desirable improvements in C4.5

Continuous classes

Two aspects which have to be considered in both continuous and discrete classes are (i) the need to define good criteria for 'ranking' possible tests in a decision node, and (ii) the need to decide when to prune a tree to avoid 'overtraining' or 'overfitting'. Notwithstanding, solutions for continuous and discrete classes tend to be quite different. C4.5 only works with discrete classes (as output), and Quinlan has carried out some experiments with a new version of C4.5 which can treat continuous classes in [Quinlan96].

Discrete ordered attributes

If ordinal attributes are renamed in a descriptive manner, for example, assigning 1=low, 2=medium and 3=high, the descriptive symbols lose their significance of order ('low' less than 'medium'). Nevertheless, the rules and tree become a lot more readable. It would be desirable that with a mapping or appropriate transformation, the algorithm recognises the name and its numeric value, for example that 1 is equal to 'low'.

Structured attributes

Structured attributes imply a hierarchy of possible values for discrete attributes, instead of the 'flat' list which C4.5 currently uses. For example:

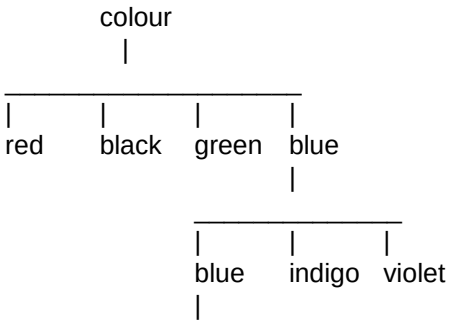


Figure 37. Hierarchy of possible values for discrete attribute 'colour'

In this manner, we can define an attribute such as 'colour' in different levels of detail. One test could use the values in any level of the hierarchy, or as a combination of levels.

Structured induction

Structured induction implies tests based on attributes whose values themselves are determined by decision trees, thus allowing a recursive definition of subconcepts. Some of the benefits of implementing a structured induction are more compact trees and the reduction of the number of training cases needed to construct a precise classifier, as demonstrated by [Shapiro87].

Incremental induction.

C4.5 is an algorithm which generates as classifier from training cases. The concept of 'partial' training does not exist; that is, to be able to continue the training run at a later time, incorporating new data. If new data is identified, there currently exist two possible actions: (i) ignore the new data or (ii) throw away the previous classifier, add the new data to the training set, and train the new classifier.

With respect to attempts at incremental induction, authors such as [Schlimmer86] and [Utgoff91] have proposed the following ideas:

Utgoff's method [Utgoff97] employs an algorithm which retains sufficient information in the tree nodes to allow their later modification in the light of new data. Utgoff's algorithm has the property that the tree produced from the new data is the same as the tree produced from training from zero with all the training cases.

A second method consists of an iterative resource constrained algorithm. It tries to find the best solution with the available resources, defined in terms of memory, time, and so on. If the training is interrupted and the resources are incremented it is able to resume from the point in which the process was interrupted.

C5.0 – improvements with respect to C4.5

In the year 2000 release of C5.0 by Quinlan, the following improvements are cited:

(i) Rulesets which occupy less memory and train faster

(ii) Decision trees: speed increase for training with same accuracy as C4.5

(iii) Boosting: technique for generating and combining multiple classifiers in order to improve predictive accuracy. Claims to improve error rates on unseen cases

(iv) New functionality: in C4.5 all errors are treated as equal, while C5.0 allows a separate cost to be defined for each predicted/actual class pair. If this option is used C5.0 constructs classifiers to minimise expected misclassification costs rather than error rates.

(v) New data types: *dates, case labels, ordered discrete values*. In addition to missing values, C5.0 allows values to be noted as not applicable. New attributes can be defined as functions of other attributes.

(vi) A *cross-reference window* enables cases to be linked to relevant parts of the classifier.

Quinlan also had previously worked on improving the use of continuous attributes in C4.5, which is summarised in [Quinlan96].

Discussion of possible areas to incorporate fuzzy techniques to improve functionality of C4.5

With reference to the previous section describing C4.5, three aspects are identified as candidates for the introduction of fuzzy techniques, in C4.5 in particular, and in induction algorithms in general.

(i). Description space and geometric interpretation

One of the problems with all induction algorithms is that in specific cases the hyperrectangles are approximations of non-rectangular regions. The grade of membership to one rectangle or another could benefit from a fuzzy interpretation. This solution would have similar applications in the case of the treatment of ill-defined and fragmented regions.

(ii). 'Intelligent' nodes in the tree

It is proposed that the tests to each node could assume fuzzy ranges, to complement the structures attributes and possibly to guide the structured induction process. Also, in the case of multiple possible tests in one node, the best test to be done in each situation could be selected in a fuzzy manner, first having confirmed that a fuzzy aspect exists in the given test. In the case of incremental induction, a use of fuzzy concepts is not proposed, as this is considered more of a programming and deterministic design problem.

(iii). 'Fuzzy attributes'

The third approach to include fuzzy concepts into decision tree induction, would be the case in which one or more inputs or outputs are defined in a fuzzy form. In this case, the induction algorithm should make use of the information provided by the similarity grade, membership grade, or grade of fuzziness, to help in the classification. It is necessary to specify some requirements for this approach to be useful and applicable:

a) The attributes really are better expressed in the fuzzy form (the concept of the problem is 'fuzzy').

b) The results of the current classification with the data set being processed is not satisfactory (e.g. after benchmarking with C4.5, ID3, neural net, Ward, Centroid, …) and we can demonstrate a significant improvement in the classification by incorporating fuzziness.

c) The standard C4.5 algorithm (or any other existing induction algorithm) cannot adequately treat problems which include 'fuzzy' information (e.g. membership grades) as part of the data.

Probabilistic interpretation of the tree generated by ID3 and C4.5

A confidence factor (probability of success) can be seen associated with the branches and leaves (terminal nodes) of the tree generated by ID3. These factors are calculated internally. If a branch or a terminal node has a confidence factor of (0.75), this implies that the corresponding branch or node would be correct in 75 of each 100 cases presented to it. What is not explicitly indicated is where the 25 cases per 100 that are not correct, are destined. At first sight they could be in the most adjacent branches to the node/branch under consideration.

Also, the probabilistic significance of the incorrectly classified cases does not affect the determinism of the tree in its classification of cases (once it has been trained). That is, the model always classifies the same cases in the same classes, in successive and equal executions. If we select the option (in Clementine) to generate the 'C' code representation (export) of the ID3 tree, it can be clearly seen that the structure is composed of a series of 'case' and 'switch' constructors, which are totally deterministic: or they follow one path or another, depending on the values of the attributes in each case given as input. The confidence factors are not used to alter the path to be followed in the tree.

Interpreting induction with a 'fuzzy' approach [Chang77]

[Chang77] gives a fuzzy definition for a decision tree. It is assumed the tree already exists, and the problem is limited to that of the search and interpretation inside the tree. This contrasts with the ID3 and C4.5 approach in which a tree is constructed (induced) from zero from a test data set, and without separate complementary information. Nevertheless, the reasoning with respect to the fuzzy interpretation within the tree is interesting in the current context, that being a tree induction in a fuzzy form which produces a fuzzy tree.

Definition 1. A fuzzy decision function f_χ in node χ is a unary function of real values in the form of a κ-tuple, with $\kappa \geq 2$,

$$f_\chi : X \rightarrow [0,1]^\kappa , \qquad (2.86)$$

where X is the input (e.g. a digitised drawing I, or a voice spectrogram S) and the κ-tuples are the labels (decision values) ν (χ_i) of the outgoing branches (χ , χ_i), i=1,.... κ, where χ_i is the ith son of node χ. A decision function of type 0-1 $f\chi$ is a fuzzy decision function that may only assume integer values 0,1:

$$f_\chi : X \rightarrow \{0,1\}^\kappa , \qquad (2.87)$$

with exactly one element of κ-tuple equal to 1.

Definition 2. A fuzzy decision tree T_r is a tree with root **r** such that each node χ that is not a leaf, possesses a corresponding κ-tuple decision function f_χ with κ ordered sons χ_1 , ..., χ_κ . A fuzzy decision tree has fuzzy type decision functions, and a 0-1 decision tree has 0-1 decision functions.

Definition 3. The decision path (χ,γ) is the path of a decision tree from node χ until node γ. Node χ has decision path (root, χ). Decision path (ω, χ ,....,γ , ζ) is the path from node ω , via nodes χ, ...γ , until reaching node ζ.

A complex decision represented by a decision tree is always composed of a number of simple decisions, each represented by a node in the tree. We assume that in practise, the time required to take a decision κ-tuple (that is, evaluate a decision function of type κ-tuple) is $O(\kappa)$.

Definition 4. The value of decision $V(\chi)$ of a decision path χ, is the product of the decision values (labels) of the branches which compose it. That is,

$$V(\chi) = \bigwedge_{\gamma \, \varepsilon \, \text{path} \, \chi} \nu(\gamma). \qquad (2.88)$$

("Product" $\bigwedge$ is a real number product in the 'prob' model or is the minimal function as defined in the 'minimax' model.

It can be demonstrated that, given the same decision tree **T** with the same decision values, each of the criteria: 0-1, max-min and fuzzy 'prob' may each reach a different decision.

Chapter 3. Development Work

Data Mining is a discipline which requires a series of steps, each one constructing a solid platform on which the following steps depend, and on which subsequently depends the quality and integrity of the whole data processing project. The first step of all is to know what data we have which enables us to describe something. We can conveniently think of data in terms of a file, which is normally defined in two dimensions: the downwise dimension is the number of cases we have, and the crosswise dimension is the number of variables we have to describe those cases. Normally the raw data is stored in a flat file or a database table, and it is usual that the variables which describe the data have different types. The first thing we normally do to the data, supposing that it has been generated by some process, or extracted randomly from a bigger data population, is to look at it, to explore it. Here is where we confront our first decision to be made about the data: in order to view data and give it meaning, we have to have previously decided the type of each of its variables. In statistics, types are often assigned more for the reason of facilitating the data processing, rather than trying to reflect the true nature of the data. For example, often a nominal variable, that is a categorical with no interpretable order between the values, is assigned values such as A, B and C, an example of which would be the variable 'postal codes'. On the other hand, a variable which does have an implicit ordering (ordinal), is assigned values such as 1, 2, 3 and 4, an example of which would be the variable 'grade of experience'. If we later wish to compare variables of different types it is essential that we spend sufficient time in assigning the initial types, otherwise meaning is lost. The same applies to variables of the fuzzy type. First we must consider if a variable is better represented in the fuzzy form, rather than as ordinal, nominal or numeric. We must justify the decision. One simple guideline is to establish how the data has been originally captured or generated. If we have used a membership function to read off a grade of membership on the y-axis with respect to some value or label on the y-axis, then we may consider this value to pertain to a fuzzy type variable. If on the other hand, the value has been assigned by selecting one of three discrete possibilites, then we may reasonably consider this value to pertain to a categorical nominal or ordinal type variable.

Once we have assigned a type to each variable and are reasonably sure that the given type is the best one for each given variable, then we can explore each individual variable. Depending on its type we can display it in different ways: numericals with a plot, categoricals with a frequency histogram or a pie chart. We can generate statistics for each variable, once again depending on its type: for numericals the maximum, minimum, mean, standard deviation, and so on; for a categorical, the mode, frequencies for each category, and so on. Variables of the same type can be analysed together and compared. Once we have terminated the exploration phase, which may involve some normalisations, elimination of missing values or adjustment of distributions, the next step would be a modelling phase. We can try to partition the dataset into clusters, or create a classificative or predictive model. The simplest algorithms which model data generally require the inputs variables to all be of the same type. Often a categorical variable (ordinal or nominal) is assigned values 1,2,3, and so on, and from then on is considered as numeric. On the other hand, we can discretize all the numeric variables, by defining numerical ranges and assigning categories to the corresponding ranges. More sophisticated algorithms are able to receive as input variables of different types. Some truly calculate distances in terms of those types, whereas others internally convert all the data to a unique form.

At the heart of being able to explore and model a dataset described by variables of different types, is the ability to measure the difference, similarity and relation between individual variables of different types. It is easy to make value-judgements between a person who is 35 years of age, and a person who is 75 years of age. It is also easy to make value-judgements between a 'small saloon' car and a 'large sportster' car. But it is more difficult to make sense out of the comparison of a 'blue small saloon' car and a car which was manufactured 8 years ago. This difficulty does not arise because any of the data is invalid, nor are the types incorrectly assigned: it is due to the comparison between types.

In Section 3.1 we will consider in detail the problems associated with data representation, data capture, and what may happen when we compare variables of different types. Section 3.1.2 presents a homogeneous representation which converts the data of any type of variable into grades of membership, thus allowing it to be processed by any algorithm as numerical input. Section 3.2.1 also continues with the theme of processing mixed data types, but this time from the point of view of a 'data fusion' process. This differs from 3.1 in that we are not just comparing variables of different types, but creating a new factor or variable which is a product in some way of two more elemental variables. This topic is associated with factor analysis and data reduction. Data reduction pretends to represent a dataset with a reduced number of highly descriptive factors, which are a result of 'amalgamating' two or more original variables. A simple example would be an insurance company customer database, in which the following three variables are defined: 'number of years as customer', 'insurance premium' and 'number of claims'. These three variables could be fused to form a 'risk factor' which indicates if the person is a good client or not, where a 'good client' for an insurance company would be someone who has been paying a premium for many years with a minimum of claims. In Section 3.1 we consider algorithms which generate 'covariance' matrices from variables. The information which the matrix tells us about the strength of relationship between variables tells us which ones to join together, and in which order. Of course, this again depends on having previously defined a way of representing the different types of variables in order to

process them together or make quantifiable comparisons between them. Being able to calculate a covariance matrix from two or more variables is fundamental in being able to 'join' those variables in a reduced number of factors, which is one of the areas of interest in the preparation of inputs to a data model. Notwithstanding, it is not always necessary to use a covariance matrix, for example in the case when we use a given model to reduce the number of factors.

Section 3.1 deals with the comparison, representation and processing of data of different types. These areas are considered here because later in Section 4.1 we need to represent and process real ICU and Apnea datasets, which consist of a mixture of binary, numeric, categorical ordinal, and nominal data types. We therefore contrast different approaches and study the behaviour of data when represented and compared in different ways. More specifically, Section 3.1.1 considers a unique algorithm approach whereas Section 3.1.3 considers separate algorithms for each data type combination. In Section 3.1.2 we define a homogeneous form of representing fuzzy variables [Nettleton99b], which is an extended version of the parametric method defined by Hathaway and Bezdek [Hathaway96].

Section 3.1.4: one of the concurrent themes of the work is the inclusion in data exploration and modelling of data captured in the fuzzy form. If we have two such variables in the same dataset and we wish to compare them in some way, we require a method which functions in a fuzzy partition space, such as a fuzzy c-Means type algorithm. In the literature, there exist diverse methods to calculate a 'fuzzy covariance matrix', which thus serves to compare two fuzzy variables, although these tend to specific to certain problems, or very complex to implement. Given this scenario, it was decided to develop a way of calculating fuzzy covariances, derived from fuzzy c-Means, and extending the idea of Gustafson and Kessel who conceived a fuzzy covariance matrix, but one whose distance measure was between a variable and the fuzzy prototype of a fuzzy set.

In Section 3.1.5 we consider how to best capture data in the fuzzy form, which involves defining a horizontal scale and which linguistic labels are placed. In the vertical dimension we have to design a membership curve for each fuzzy set, or linguistic label, which best fits the nature of the data. This involves the steepness of the curves, the extent of overlap and transition between curves, and so on. These issues have to be tackled because on them depends the quality of the data capture, and the data exploration and modelling which follows.

Section 3.2 deals with the aggregation of different types of data: Section 3.2.1 considers different fusion techniques for different data types; Section 3.2.2 details the implementation of Nettleton's version of Hartigan's 'joining' algorithm [Hartigan75] which successively reduces an initial set of variables to a reduced group of factors, which best describe the data. On the other hand, Section 3.2.3 focusses on aggregation using WOWA type operators. Aggregation is a fundamental part of the later work, which is applied to Apnea diagnosis. Also we have to solve problems with the data values, with respect to their relevance to a given output and their reliability. Another aspect is that of missing data, which often occurs in a greater or lesser percentage in real datasets. Aggregation operators provide a possible solution, especially the WOWA operator, which possesses two weighting vectors which can be used to parametrise the relevance of the variables and the reliability of the data values. One problem with using aggregation operators with weighting vectors, is the assignment of the weighting vector itself. We have developed and tested a genetic algorithm as the mechanism for learning the best weights for a given dataset. We also see how we have included a way of filtering missing values and incorporating this into the WOWA operator. One of the characteristics of the weight vectors of WOWA is that each vector is constant for all variables and data values. In the case of variables that is adequate because we can interpret each weight as representing the relevance of the corresponding variable. In the case of the data values, it would be more useful to have a weight vector, corresponding to the reliability of the data values, which varies for each variable. Thus we have modified WOWA so that it has a vector of vectors of reliability weights, one for each variable.

We have just outlined the principal areas of investigation and the approaches which have been considered. In the following sections of this Chapter we will go into the detail, dividing the work into two corresponding sections.

3.1 Representation, comparison and processing of different types of data

The type which a variable can assume is an initial consideration necessary before any exploration or modelling can be conducted on the data. Different representations are considered from a conceptual and symbolic viewpoint, and from basic statistical principles. We will see that similarity and dissimilarity can be approximated by densities and frequencies, depicted in a differing number of dimensions, and between variables of different natures. Also we consider different necessities from a processing point of view.

3.1.1 Representation and processing of different data types

In this section we consider forms of representation for different variables types, using as an example, an Intensive Care Unit (ICU) hospital admissions dataset. This dataset is outlined in this Section and given in more detail in Section 4.1 of the thesis. We also discuss considerations for developing a common approach to representing, comparing and processing a dataset composed of a mixture of these different data types, with special emphasis on fuzzy attributes, that is those which include grade of membership values and are interpreted via a membership function.

ICU Data

The ICU data consists of one record for each patient, which registers the vital clinical data, such as blood pressure, heart rate, body temperature, together with the results of a blood and urine analysis. Output variables are the prognosis, which indicates survival or not, and the number of days of stay in the ICU and the general hospital. The data contains a variety of variables types: categorical nominal and ordinal, numerical, binary, and variables adequate for fuzzy representation, such as prognosis and length of stay in ICU/hospital.

Data types

Numerical variable: (e.g. temperature, blood pressure). In the case of numerical values, we can use Bezdek's representation [Bezdek81] for the 'Fuzzy c-Means Functionals' algorithm, (fuzzy classification), which generalises a variance function between groups. This data type includes integer and floating point numbers.

(Lexical) Ordinal Categorical Variable: (e.g. type of patient, previous health state). For this type of variable, an implicit ordering exists between the categories. For example, previous health state=1 indicates a superior state of health with respect to previous health state=2, and so on.

Nominal Variable: (e.g. Conf_Inf {Y, N, unknown}, which indicates the confirmation of the presence of infection). These values are symbolic but it is not possible to establish any order among them. The 'GOM model' [Manton92] presents a fuzzy representation for this type of values. This data type is also known as 'non-ordinal categorical'.

(Binary) Nominal Variable: (e.g. respiratory failure {yes,no}, stay in ICU for 24 hours or more {yes,no}, Sex {M, F}). This type may be considered as a special case of nominal variables. In clinical records, there are often a great number of variables of this type, with a 1 or 0 response to questions about clinical conditions, concentration levels of different types in blood and urine, presence of different conditions, durations, and so on. [Bezdek81,pp86], defined a method especially for attributes which take binary values in medical data sets. Note that a variable such as 'Sex' may also be considered a special case of the Nominal Variable type (above) in which there are only two possible categorical values. This data type may also be considered as the 'non-ordinal categorical' data type.

Fuzzy Variable: (e.g. duration of stay in hospital, risk of death, prognosis of recovery). For each variable, we have to establish if it is best represented in the fuzzy form, or if it really falls into one of the crisp data types previously listed. This type is characterised by each category having a grade of membership and each case being potentially assigned to one or more of the available categories. This data type may be fuzzy numerical, fuzzy ordinal categorical or fuzzy non-ordinal categorical. In the case of ordinal values, the same representation and processing scheme is proposed as [Nettleton97] and [Aguilar91] in which fuzzy sets are represented by trapezoidal membership functions.

The definition of a homogeneous representation and processing of all data types in the fuzzy form

The search for a homogeneous representation and processing is motivated by the complexity of considering distinct methods for each different type combination of data. Some of the possible problems with this approach are: difficulties in representing numeric non-fuzzy data, different distance measures, and that generalisation may cause a loss in precision. As possible solutions to these problems, we could base our methods on existing algorithms, for example the Parmenidean Pairs of [Aguilar91] which were outlined in Section 1.2.4, or the parametric model of [Hathaway96] as extended in [Nettleton99b], and detailed later in this Section. Thus our objectives would be to refine existing solutions of representation and covariance calculation to build a good fusion algorithm, which works for any type of data.

Representation of Data: we will consider Crisp variables as a special case of Fuzzy variables. With reference to Tables 10 and 11, consider the following, in which there are three cases (objects), and two attributes to describe them: colour and size. Attribute colour has three possible values: red, green and yellow; whereas attribute size has three possible values: width, length and height. In the crisp case of colour an object can be only one colour at a time. Therefore one colour will have value 1 and the other two colours will be at value 0. In the fuzzy case, which of course reflects more truly what occurs in reality, an object will very rarely be perfectly red, but may be, for example, be tinged with green and maybe yellow. Therefore Object 1 may have a red 'reading' of 0.8, a green reading of 0.02 and a yellow reading of 0.18. These values are summarised in Table 10. There reading would have to be taken with a spectroscopic type instrument which permits discern the composition.

Table 10. Membership values for different values of variable 'colour' and of corresponding values of variable 'size'

	Colour			Size		
	red	green	yellow	width	length	height
case 1	0.8	0.02	0.18	3	5	2
case 2	0.0	1.0	0.0	8	1	2
case 3	0.0	0.0	1.0	3	4	5

Note that the colour attribute in case 1 is fuzzy but in cases 2 and 3 it is crisp. The crisp version would simply consider the predominant colour in each case. Predominant could be defined, for example, as having a percentage composition of more than 50%.

In the case of attribute size things are not so simple. Size really is a hierarchical attribute, being composed of three sub-attributes. Each sub-attribute is a numerical value, which, for example, could be measured in centimetres. How would we represent the sub-attribute width as fuzzy? One possible method would be to agree (consensus) between the majority of human experts, as to a categorisation of this attribute in the given context. For example, it could be decided that three categories best represent and describe the nature of the numeric value of width: narrow, normal and wide. This could be established after a study of the characteristics of the distribution and tendencies of the values, with appropriate axes. Together with the assignment of the number of categories, a numerical range must be also defined for each category.

For example, all objects with a width between 1 and 3 (inclusive) are narrow; between 4 and 5 are normal; and between 6 and 10 are wide. In the crisp case each object would fall into one and only one category with membership grade 1 and the other categories having membership grade 0. In the fuzzy case, we would define a membership function and would assign a corresponding membership grade for each category, to the object.

One key consideration is how the data is originally captured. The data may be originally collected as fuzzy (for example the data input process would be to write a cross on a continuous scale with a number of labels (e.g. narrow, normal, wide) assigned along it at different points. Otherwise the data could be collected as crisp numerical or categorical. Even though it has not been captured as fuzzy, it can be given as input to fuzzy c-Means, for example, to calculate membership grades of the cases to the clusters.

Looking for strength of relation between attributes – covariance: up to now, we have considered means to represent any type of data in a fuzzy form, with the objective of comparing strength of relation between attributes. We now consider a covariance type calculation which could produce a covariance matrix such as that of Table 11, calculated with the SPSS standard correlation function, with covariance option, for two sets of variable-attributes of Table 10.

Table 11. Example covariance matrix for the variables-attributes 'colour'=red and 'size' (width) of Table 10

	Colour(red)	size(width)
Colour(red)	1.0	0.66
Size(width)	0.66	1.0

The covariance values in Table 11 indicate that there is little relation between colour=red and width (0.66), for these objects. A significant positive covariance between these variable-attributes would produce a covariance value such as 1.33, for example. We note that covariance values may have a range outside [-1,1], whereas the correlation value is within [-1,1] Algorithms which process case data into covariance type values usually do so by a series of matrix and vector manipulations.

Fuzzy Representation

In the case of a *homogeneous fuzzy representation*, each variable value in the fuzzy form may be compared with every other variable value, and thus establish their covariances. A prerequisite is that the fuzzy representation must be homogeneous. Consider the real problem of the admission of patients in a hospital ICU unit. There exist several variables for which a fuzzy representation fits in well with the decision making process made by the medical experts. For example, in the case of the variables '*Type of patient'* and *'Probable infection on admission to the ICU',* where the value which is assigned to each individual is determined via an intuitive process.

One case which is difficult to categorise could have a grade of membership to each of the possible classes of approximately 0.50, and would correspond to some symptoms which are not characteristics of any of the available categories. This is a situation quite frequent in the medical domain. In [Bezdek81], an example is detailed of a set of 107 data vectors with binary values, each with 11 attributes. The attributes are symptoms which are considered clinically relevant for patients who suffer from abdominal pain caused by one of (i) hernia hiatal or (ii) gallstones, with which a classification is obtained in the presence of ambiguous grades of membership.

Taking into account that the fuzzy approach may provide a common platform to treat all variable types in a homogeneous manner, our proposal considers modifying the attribute fusion algorithm detailed in [Hartigan75] so that it can be applied to fuzzy variables matrices. In the following part of this section we see how to represent the fuzzy variables in the different cases. From this representation, we can use a fuzzy covariance calculation [Bezdek81] which lets us choose, in each step, which variables to fuse.

The representation of 'hospital admissions' data

In the case of the 'admissions' data set (see Section 4.1 and Annex 2 for description and details), we would use a fuzzy representation only for those attributes chosen by the medical experts. The following attributes were proposed: 'previous health state', 'type of patient', 'infection probable on admission to the ICU', and 'Increment of Creatinine > 124 Mol/l in last 24 hours associated with Oliguria'. We emphasise that in this case, the ICU dataset was captured in crisp form, and these proposals are as a result of data analysis (see Section 4.1) and conversations with the medical expert.

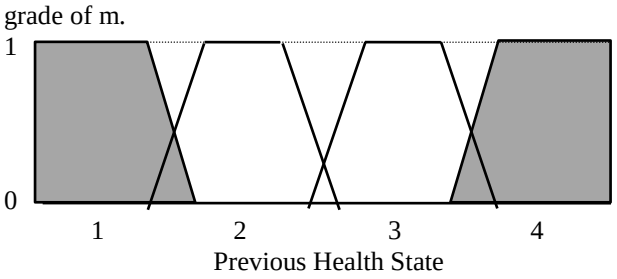


Figure 38a. Representation of input variable 'Previous Health State'

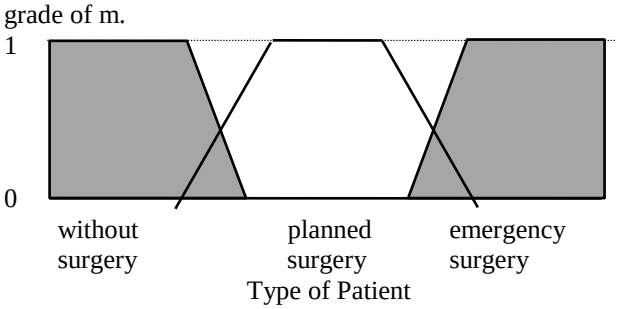


Figure 38b. Representation of input variable 'Type of Patient'

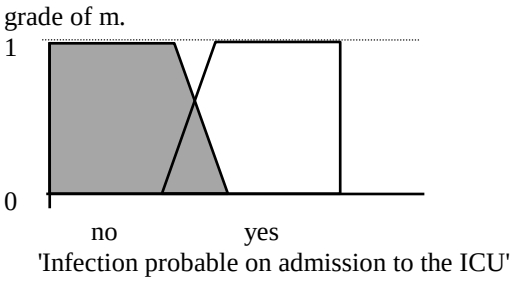


Figure 38c. Representation of input variable 'Infection probable on admission to the ICU'

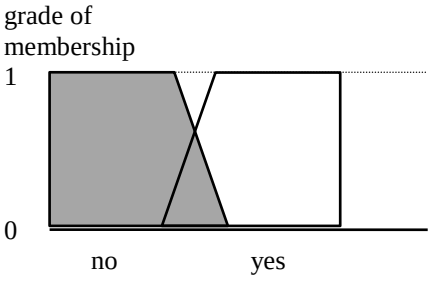


Figure 38d. Representation of input variable 'Increment of Creatinine > 124 Mol/l in last 24 hours associated with Oliguria'

Scalar representation for questionnaire responses

A questionnaire for data capture of admissions data would use continuous scales on which the doctor indicates with a cross (for example) in the place where s/he thought appropriate as the response to that question. For example:

¿ Existence of coma or profound stupor in the moment of admission to the ICU ?

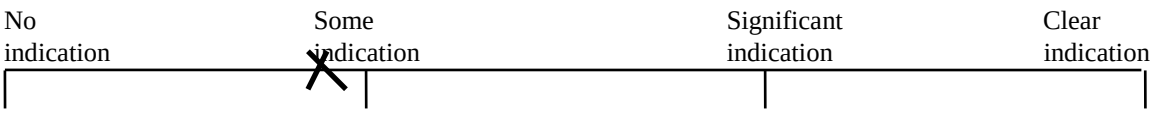


Figure 39. Example of a continuous scale with four linguistic labels

In the example of Figure 39 we can see a scale of four linguistic labels which the medical expert must assign himself (using his own terminology). The membership function decides if the distance along the scale is linear (equidistant between discrete points), logarithmic, or some other function.

The main advantages of this approach is the greater precision and information value that this representation offers. The main disadvantages or problems to be resolved are: the need to define the linguistic labels (together with the expert); the necessity to find a best possible definition for the membership function (a simplification would be to use a trapezoidal or triangular form); and that the definition depends on the subjectivity of the medical expert in each case. In practise, the method must result in crisp decisions as output. There exist many representations for fuzzy rules which produce crisp states as output from a fuzzy process, one of the most referenced being [Takagi85].

The capture of crisp and fuzzy data

For real data, it is vital to design and test different representation methods. In the case of the medical data, we have many binary variables for which a complex question requires a yes/no answer (a). There are also multiple choice responses to a question (b). Or we can have a question which requires a response indicated on a scale (c). For example:

(a) Question (to the doctor): Is it probable that the patient has had any infection prior to admission to the ICU?
Yes __ No __

(b) Question (to the doctor): What is your evaluation of the previous health state of the patient?
1 __ 2 __ 3 __ 4 __

(c) Increment of Creatinine > 124 Mol/l in the last 24 hours associated with Oliguria?

No _____ Yes

The capture of data in the fuzzy form is evident in example (c), while questions (a) and (b) require just one category and capture data in the 'crisp' form. The underlying membership function is crucial to the conversion of the input representation to a membership grade. This can be done jointly with the domain (medical) expert and verified with real test data, contrasting against the experts' opinion. To evaluate a membership function, we can use some cases with high membership to a class, and other cases which lie ambiguously between one class and another.

On the definition and successive refinement of membership functions - fuzzification

Now we consider the interpretation of the input values with membership grades for each attribute and linguistic label or fuzzy category. There exist diverse methods for defining membership functions. We consider three methods, which permit a successive refinement of the membership functions and enable us to fine tune the interpretation of the input values.
(i) Initially the membership functions for the fuzzy attributes may be decided in consensus with the medical domain experts. We could define simple triangular and trapezoidal functions, whose design issues include: gradient of the slopes, overlap between linguistic labels, and the percentage of a label which is horizontal (100% membership). We only consider symmetrical trapezoids and triangular forms.
(ii) Once we have reasonable values from **(i)** we can generate the linguistic labels using a 'parmenidean pairs' technique [Aguilar91]. For this method to work, we initially define the design criteria for a set of five trapezoidal linguistic labels (width, slope, overlap), and the membership functions are automatically generated from this initial definition.
(iii) We may consider that the trapezoidal forms are really an approximation of a non-linear membership function. There exist automatic interpolation methods for eliciting such non-linear membership functions from the initial data. Such a method is Chen and Otto's method [Chen95]. Finally, we can bias the membership function and aggregate data values using a method such as Yager's OWA [Yager88] or Torra's WOWA (Weighted Ordered Weighted Average) [Torra97a] to convert a vector of input data, interpreted by a membership function, into just one aggregated value.

To summarise, we are considering methods **(i)** and **(ii)** to initially define and subsequently refine the membership functions. Each membership function is 'made to measure' for each input attribute, and a trial and error method is cycled until coherent results are obtained. The representation method is implemented as fuzzy rules which, given an input value, constitute membership functions which produce a membership grade of that value to the defined linguistic labels for the attribute. For crisp values, the rules simply give a membership grade of 1 or 0. As the membership grades may be considered as weights, the crisp values are not altered. For the fuzzy values it is of course essential that the data capture is of a fuzzy form. In the next section we see the data capture method used.

3.1.2 An approach for the Homogeneous Fuzzy representation of variables of different types

In order to process data of different types it is necessary to have a uniform representation for those types. One way of achieving this is by considering the crisp type as a special case of the fuzzy type. Using membership functions constructed from trapezoids, triangles or straight lines, we can then define any data type on the same axes, and save it in a common numerical format. In [Hathaway96], Hathaway and Bezdek define a parametric representation method. This method allows us to represent the following types of variables: real, intervals, linguistic labels (triangular and trapezoidal), in a simple and natural manner. In [Nettleton97] a representation was detailed for fuzzy data using trapezoidal functions and *parmenidean pairs*. An example of a parmenidean pair for the variable 'length of stay in hospital' would be {short,long} from which we could generate the following five linguistic labels: {very short, short, medium, long, very long}. The parametric representation of [Hathaway96], as described in Section 2.2.4 of the thesis, has been extended by Nettleton to include parmenidean pairs (see Section 1.2.4 of the thesis), as can be seen in Figure 40c.

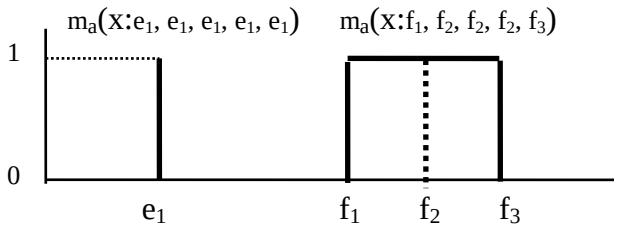


Figure 40a. Representation of Real and Interval Fuzzy Variables

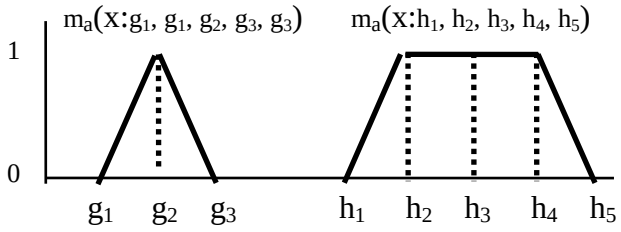


Figure 40b. Representation of Triangular and Trapezoidal Fuzzy Variables with Symmetrical Form

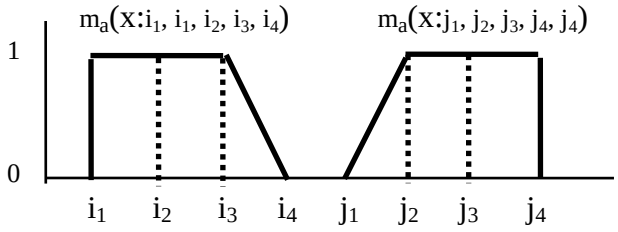


Figure 40c. Representation of 'Parmenidean Pair' Fuzzy Variables (the 3 intermediate labels can be represented with the Trapezoidal form)

The representation is defined by a parametric form in which we can define any of the types of variables given in Figure 40, using the definition $m_a(x: p_1, p_2, p_3, p_4, p_5)$. The parameters which are not used in a given variable are duplicated in a symmetrical manner. Forms 39a(right) and 39b(left) have value assignments which allow us to distinguish the rectangular form from the triangular form.

For example, a real number would be represented by (1.1, 1.1,1.1,1.1,1.1), an interval by (0.5,1.5,1.5,1.5,2.5), a fuzzy value with triangular form as (2.6, 2.6, 2.7, 2.8, 2.8) and a fuzzy value with trapezoidal form as (-0.1.0.1,0.2,0.3,0.5). These values form a matrix with which we can calculate covariances, correlations, and so on.

In the following example of Figures 41a, 41b and Table 12, we use only three forms: the trapezoidal form of Figure 39b (right), and the two trapezoidal forms of Figure 40c, which are sufficient to construct membership functions for Parmenidean pair type fuzzy sets.

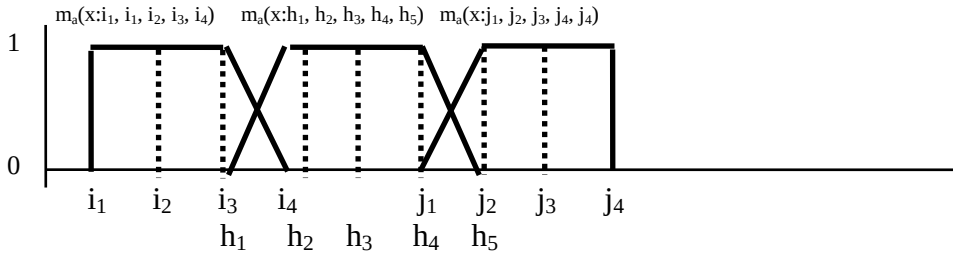


Figure 41a. Fuzzy sets represented by the data in Table 12.

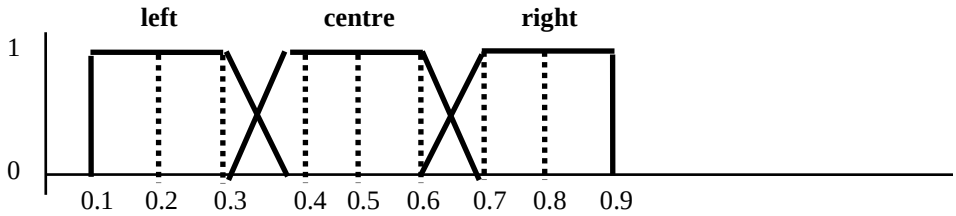


Figure 41b. Fuzzy set data points represented by the data in Table 12.

With reference to Figure 41b, given the relative positions on the x-axis of the membership functions, adn following the rules described previously, we assign the following: $m_a(x:i_1, i_1, i_2, i_3, i_4) = m_a(x:0.1, 0.1, 0.2, 0.3, 0.4)$; $m_a(x:h_1, h_2, h_3, h_4, h_5) = m_a(x:0.3, 0.4, 0.5, 0.6, 0.7)$; $m_a(x:j_1, j_2, j_3, j_4, j_4) = m_a(x:0.6, 0.7, 0.8, 0.9, 0.9)$. It is essential to note that a data reading may only have a non zero membership grade for a maximum of two fuzzy sets, which in the case of Figures 41a and 41b, limits the possibilities to left and centre, or centre and right. The permutations of these possibilities are given in Table 12.

Table 12. Fuzzy data test set for different combinations of the fuzzy sets depicted in Figures 41a and 41b.

	Fuzzy label 1 (weights representing form of membership function)					Fuzzy label 2 (weights representing form of membership function)					Input (data value)	outputs					
	w_{11}	w_{12}	w_{13}	w_{14}	w_{15}	w_{21}	w_{22}	w_{23}	w_{24}	w_{25}	i_1	o_1	o_2	o_3	Fuzzy label 1	Fuzzy label 2	Forms
x_1	0.3	0.4	0.5	0.6	0.7	0.6	0.7	0.8	0.9	0.9	0.50	0	1	0	$m_a(x:h_1, h_2, h_3, h_4, h_5)$	$m_a(x:j_1, j_2, j_3, j_4, j_4)$	Central trapezoid, right trapezoid
x_2	0.1	0.1	0.2	0.3	0.4	0.3	0.4	0.5	0.6	0.7	0.20	1	0	0	$m_a(x:i_1, i_1, i_2, i_3, i_4)$	$m_a(x:h_1, h_2, h_3, h_4, h_5)$	Left trapezoid, central trapezoid
x_3	0.3	0.4	0.5	0.6	0.7	0.6	0.7	0.8	0.9	0.9	0.80	0	0	1	$m_a(x:h_1, h_2, h_3, h_4, h_5)$	$m_a(x:j_1, j_2, j_3, j_4, j_4)$	Central trapezoid, right trapezoid
x_4	0.1	0.1	0.2	0.3	0.4	0.3	0.4	0.5	0.6	0.7	0.40	0	1	0	$m_a(x:i_1, i_1, i_2, i_3, i_4)$	$m_a(x:h_1, h_2, h_3, h_4, h_5)$	Left trapezoid, central trapezoid
x_5	0.3	0.4	0.5	0.6	0.7	0.6	0.7	0.8	0.9	0.9	0.60	0	1	0	$m_a(x:h_1, h_2, h_3, h_4, h_5)$	$m_a(x:j_1, j_2, j_3, j_4, j_4)$	Central trapezoid, right trapezoid
x_6	0.1	0.1	0.2	0.3	0.4	0.3	0.4	0.5	0.6	0.7	0.33	1	0	0	$m_a(x:i_1, i_1, i_2, i_3, i_4)$	$m_a(x:h_1, h_2, h_3, h_4, h_5)$	Left trapezoid, central trapezoid
x_7	0.3	0.4	0.5	0.6	0.7	0.6	0.7	0.8	0.9	0.9	0.63	0	1	0	$m_a(x:h_1, h_2, h_3, h_4, h_5)$	$m_a(x:j_1, j_2, j_3, j_4, j_4)$	Central trapezoid, right trapezoid
x_8	0.1	0.1	0.2	0.3	0.4	0.3	0.4	0.5	0.6	0.7	0.37	0	1	0	$m_a(x:i_1, i_1, i_2, i_3, i_4)$	$m_a(x:h_1, h_2, h_3, h_4, h_5)$	Left trapezoid, Central trapezoid
x_9	0.3	0.4	0.5	0.6	0.7	0.6	0.7	0.8	0.9	0.9	0.67	0	0	1	$m_a(x:h_1, h_2, h_3, h_4, h_5)$	$m_a(x:j_1, j_2, j_3, j_4, j_4)$	Central trapezoid, right trapezoid

The values in Table 12 are in a format which allows input of the representation of the fuzzy sets, together with the real data reading. Columns w_{11} to w_{25} represent the weights which define the forms of the two fuzzy sets for which the corresponding real reading, denoted by i_1, has a non-zero membership grade, and o_1 to o_3 are the outputs. $\{w_{11}, w_{12}, w_{13}, w_{14}, w_{15}\}$ and $\{ w_{21}, w_{22}, w_{23}, w_{24}, w_{25}\}$ represent the structure of the two membership functions for which i_1 has a non-zero membership grade. Overall, the objective has been to indicate the symmetric form and dimensions of the membership functions (trapezoidal, triangular and rectangular). The outputs $\{o_1, o_2, o_3\}$ indicate for each case, the fuzzy set for which it has the highest membership grade, and correspond to the fuzzy sets labelled 'left', 'centre' and 'right' as depicted in Figure 41b. Thus the mechanism may be considered a defuzzification technique which produces a 'crisp' output result, from fuzzy inputs.

Processing – pattern matching

The form of the data in Table 12, suggests that we have a pattern matching problem. This form of representation has the advantage that all the aspects of the fuzzy value can be reflected, while that one of its disadvantages is the number of data items needed. For one *parmenidean pair* {low, high} and intermediate labels {fairly-low,medium,fairly-high} we need 25 data values, five for each linguistic label. In each row of the example data in Table 12, we have two fuzzy sets, each with five data values, and each representing one linguistic label. This is because a data reading can only have a non zero membership grade to two fuzzy sets at a time. This data, if presented in a form which indicates its underlying structure, can be given to a neural network, which will identify the classes to which each case belongs, in a *crisp* manner. Although we are using a probabilistic algorithm to process the data, we say the result is *crisp* because each case can only belong to one cluster.

3.1.3 Comparison between different data types

Often in statistical analysis of data, too little time is spent on assigning types to variables which best relate to the nature of the data and therefore any subsequent analysis may be falsely based. This is even more so when we come to comparing different variable types. We discover that there are many ways to compare, for example, categorical with numerical variables. We can use point density diagrams, overlap, consideration of different borderline or extreme cases, and so on. The work in this section focuses on the definition of the calculations of correlation between different types of variables. We do this in order to make it possible to create a correlation matrix which would then be given as input to

the Hartigan fusion algorithm to 'join' the variables into a reduced number of factors. As an initial simplification, the ordinal and non ordinal categorical types have been considered as one type. The following algorithms have been coded and tested as 'C' programs: (A) Comparison of variables of type *Integer* or *Float* with variables of type *Integer* or *Float*; (B) Comparison of variables of type *Non-Ordinal Categorical* with variables of type *Non-Ordinal Categorical.* The tests with input data were validated by passing the same data through SPSS, in the first case with standard correlation, and in the second case with chi-squared. The results were exactly the same.

The following algorithms were developed on paper and 'dry-run' through also on paper: (C) Comparison of variables of type *Non-Ordinal Categorical* with variables of type *Integer or Float*; (D) Comparison of variables of type *Fuzzy Ordinal Categorical* with variables of type *No-Ordinal Categorical*; (E) Comparison of variables of type *Fuzzy Ordinal Categorical* with variables of *type Fuzzy No-Ordinal Categorical*

The following algorithms were not developed, but it is considered that they are a natural succession from the algorithms already defined: (F) Comparison of variables of type *Fuzzy Ordinal Categorical* with variables of type *Fuzzy Ordinal Categorical*

We say that (E) will be a simple variant of (D) and (C) , while (F) will be a variant of (D) . To introduce the difference between Ordinal and Non-Ordinal Categorical variables, one could calculate the additional information of the explicit and known order of the classes in Ordinal variables.

The final results of all the correlations will have to be normalised (which is trivial) and calibrated (redistributed) to ensure homogeneous values which can be compared one with the other. One approach is to run benchmarks with data at extremes (max, min) and chosen intermediate points, and introduce coefficients or scaling factors when necessary to calibrate in each case. For example, if the correlations between *Ordinal Categorical* and *Fuzzy Ordinal Categorical* are more heavily weighted toward 1 (density at the 1 end of the scale), while the comparison of *Integer or Float* and *Fuzzy Ordinal Categorical* are more heavily weighted toward 0, then the latter would be seen to be penalised and less often chosen in the selection of fusion pairs, due to their lower mutual correlation values. A compensative value would 'calibrate', in this sense, the distributions of the two variables.

(A) Comparison of variables of type **Integer or Float** with variables of type **Integer or Float**

This comparison involves a standard correlation of numeric variables, which has been implemented as a function in 'C' code. If we enter the data as seen in columns 3 and 4 of Table 13a, it will produce the correlation matrix as can be seen in Table 13b. The results were cross checked with SPSS (stats-correlate-bivariate-Pearson) cross product deviations & covariances)

Table 13a. Corresponding values for categorical and categorical (ordinal) variables 'sex' and 'diag'(nosis), and numerical variables 'age' and 'fio2' (clinical data)

sex	diag	age	fio2
M	5	25	65
F	18	50	80
F	5	23	75
M	50	40	85
F	18	73	60
M	21	65	70
F	30	48	60
M	21	35	45
M	21	39	55

Table 13b. Correlation matrix for variables 'age' and 'fio2'

	age	fio2
age	1.000	-0.244
fio2	-0.244	1.000

(B) Comparison of variables of type **Non-Ordinal Categorical** with variables of type **Non-Ordinal Categorical**

The method used consists of 3 steps, each step depending on the output of the previous step: (i) calculation of a Contingency Matrix; (ii) calculation of the Chi-Squared value; (iii) calculation of the Coefficient of Cramer. The functions corresponding to the 3 steps have been implemented in 'C' code.

(i) We can use an independence test in a contingency table [Cuadras80], pp227. First we define the Chi-Square distance as:

$$\chi^2 = n\left(\left(\sum_{i,j} \frac{f^2_{ij}}{f_i f_j} \right) - 1 \right) \qquad (3.1)$$

where **i** indicates value i of the first variable, **j** indicates value j of the second variable, and **n** is the number of values or cases. f^2_{ij} is the frequency of value i of the first variable, with respect to value j of the second variable, in the given dataset. For example, if value i of the variable 'sex' is 'M', and value j of the variable 'diagnosis' is 5, then the number of times when sex='M' and diagnosis=5 in the given dataset is f^2_{ij} . f_i is the sum of the frequencies for each possible value of the first variable, and f_j is the sum of the frequencies for each possible value of the second variable. f can be considered, in general, as the 'contingency table', of which an example is shown in Table 14.

For example, 'sex' is a binary variable which we consider as categorical with 2 possible classes; and 'diag' which is also a categorical variable which, in the test data of Table 13a, can be seen to have 5 possible classes. The third column, 'age', a numerical variable, is used in later examples.

First we run through the cases seen in Table 13a, columns 'sex' and 'diag', to calculate the relative frequencies, which are shown in Table 14, and which are contained in the contingency matrix. In Table 14, the columns correspond to the different values of the variable 'diag''which exist in the dataset, while the rows correspond to the different attribute-values for the categorical variable 'sex'. The values in the table are the frequencies, or the number of cases which correspond to each value-attribute pair. For example, 'diag'=21 and 'sex'=M occurs 3 times in the dataset.

Table 14. Relative frequencies for the cases of Table 13a.

		variable 'diag'					sum of occurrences for each
		5	18	50	21	30	attribute-value of 'sex'
variable	M	1	0	1	3	0	5
'sex'	F	1	2	0	0	1	4
		2	2	1	3	1	

(ii) In the next step we use can calculate the Chi-Square value, which has the following substeps: calculate the total number of cases, calculate the sum for each column, calculate the sum for each row, calculate the number of values for each attribute, then loop for all elements in the matrix.

$$((0.1 + 0.2 + 0.6 + 0.125 + 0.5 + 0.25) - 1) = 9 \times (1.775 - 1) = \frac{6.975}{9}$$

(iii) As the final step we calculate Cramer's Contingency Coefficient as:

$$\frac{6.975 / 9}{2} = 0.3875$$

Observations:
In practice Chi-Square may be regarded as a measure of fit as well as a test statistic. In this view, Chi-Square is a measure of overall fit of the model to the data. It measures the distance (difference, discrepancy, deviance) between the sample covariance (correlation) matrix and the fitted covariance (correlation) matrix. In the example (above), we may consider the first row of Table 14 (sex='M') as the sample covariance matrix, and the second row of Table 14 (sex='F') as the fitted covariance matrix. Thus we are looking for a fit (relation or distance) between the values of 'diag' for 'sex'='M' and the values of 'diag' for 'sex'='F'. Chi-Square is a badness-of-fit measure in the sense that a small Chi-Square corresponds to good fit and a large Chi-Square to bad fit. Zero chi-square corresponds to perfect fit. Thus in the

example above, the Chi-Square value calculated in step (ii) is 6.975 / 9 = 0.775, which for the data in Table 14 shows a reasonably good fit, and shows a significant dependence between the values

(C) Comparison of variables of type **Non-Ordinal Categorical** with variables of type **Integer or Float**

In order to correlate this combination of variables types, we need to choose an algorithm which allows the comparison of real numbers with symbols (categories). There are two main possibilities for 'mixed' variables such as these: (i) use own method based on simple frequencies; (ii) choose standard algorithm from those available. Table 16 shows some values for two typical categorical and numeric variables.

Processing the values of Table 13a, columns 'sex' and 'age', will produce a correlation matrix such as that of Table 15, given the low correlation between these two variables.

Table 15. Results produced from correlation of 'age' with 'sex'

	age	sex
age	1.0	0.3
sex	0.3	1.0

As modus operandi, first of all we can study each category of the categorical variable in turn and then calculate the maximum, minimum and median of the numerical variable. In the case of sex, we can see the results of this in Tables 16 and 17, as follows:

Table 16. Example 1: values of the numeric variable 'age' for each of the categories of the categorical variable 'sex'

sex=M	sex=F
25	23
35	48
39	50
40	73
65	

An example of a perfect correlation would be that all the cases of sex=M have age less or equal to 50 years and all the cases of sex=F have age greater than 50 years.

Table 17. Basic statistics for the numeric variable 'age' for each category of the categorical variable 'sex' (ref. Table 16, example 1)

	max	min	mean	range	% overlap of ranges	total number of points	total number of points in overlap	% of points in overlap
M	65	25	40.8	40	100%	5	5	100%
F	73	23	48.5	50	80%	4	4	50%

In Table 17, the mean of both categories is equal to 0.9, which is equal to the percentage of ranges which overlap. This implies that there is almost no distinction for M,F with respect to age. The correlation would be 1 - 0.9 = 0.1, which is a very small correlation.

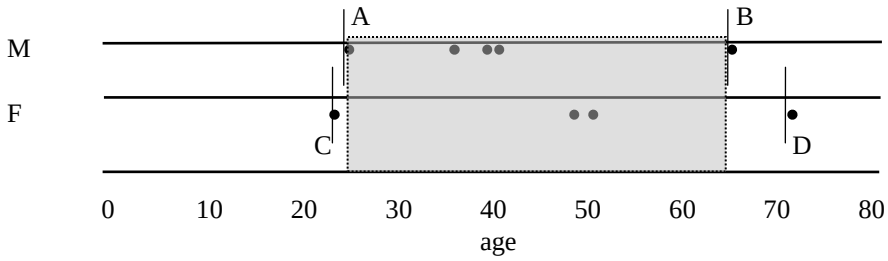


Figure 42. Graphical representation of point density used to identify degree of overlap of values of the numerical variable 'age' with respect to the categories of the categorical variable 'sex' (ref. Table 16, example 1)

In Figure 42 we see the range of values for sex=M is from points A to B, whereas the range of values of sex=F is from C to D. The overlap of the two ranges is from points A to B, as indicated by the shaded area. The last column of Table 17 indicates the calculated overlap percentage which takes into account density, for the selected data points.

Now in Table 18 we consider a slightly different distribution to illustrate a greater correlation between the two variables, age and sex.

Table 18. Example 2: values of the numeric variable 'age' for each of the categories of the categorical variable 'sex'

sex=M	sex=F
25	45
35	50
45	55
48	75
50	

The basic statistics of the variable age with respect to the variable sex can be seen in Table 19. We see from the 'overlap' value (column 5) that the correlation is much greater than for the values of Tables 16 and 17.

Table 19. Basic statistics for the numeric variable 'age' for each category of the categorical variable 'sex' (ref. Table 18, example 2)

	max	min	mean	range	% overlap of ranges	total number of points	total number of points in overlap	% of points in overlap
M	50	25	40.6	25	20%	5	3	60%
F	75	45	56.25	30	17%	4	2	50%

In Table 19, the mean of both categories is equal to ((20+17)/2)/100 = 0.185, which is equal to the percentage of ranges which overlap. This implies that there is a much greater distinction for M,F with respect to age, than in the previous example of Table 17. The correlation would be 1 - 0.185 = 0.815.

Nevertheless, this method still does not take into account the density of the points, that is, not only how much of the respective ranges overlap, but how many points overlap. In order to establish this, we could use a weighting factor such as the number that overlap divided by the total number of points.

In Figure 43, we can see a graphical depiction range overlap and range point density. The range of values for sex=M is from points A to B, whereas the range of values of sex=F is from C to D. The overlap of the two ranges is from points C to B, as indicated by the shaded area. The last column of Table 19 indicates the calculated overlap percentage which takes into account density, for the selected data points.

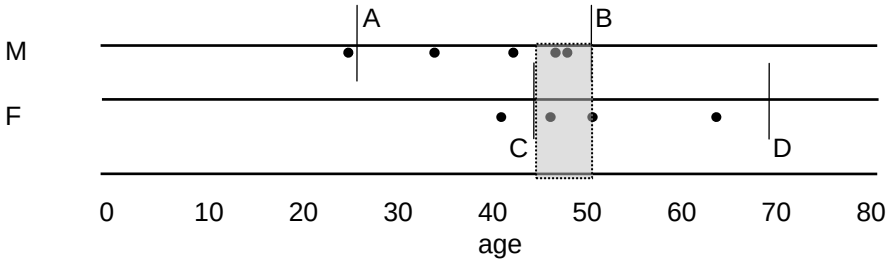


Figure 43. Graphical representation of point density used to identify degree of overlap of values of the numerical variable 'age' with respect to the categories of the categorical variable 'sex' (ref. Table 18, example 2)

This implies that, although the overlap of Figure 43 is much smaller (5 compared to 25) than that of the data of Figure 42, it still possesses a mean of 55% of the cases (points), which indicates a greater "density". Therefore this situation has to be reflected in the correlation calculation.

The following three correlation calculation methods are proposed:

(i) We define the values of the cases in the overlap as $\{v_1, v_2, \ldots, v_n\}$, then the sum of the values in the overlap will be

$$S = \sum_{i=1}^{n} v_i$$

If the number of points (cases) in the overlap is $N_{overlap}$, the number of points (cases) outside the overlap is $N_{no\ overlap}$, and the % overlap is defined as $P_{overlap}$, then the correlation value C will be:

$C = (S / (N_{overlap} + N_{no\ overlap})) * P_{overlap}$

If there is a very wide range, this would keep the values comparable.

(ii) Mean of the values in the overlap. That is, $C = S / N_{overlap}$.

(iii) Calculate a coefficient relating the relative size of the overlap with the number of cases in the overlap. If $P_{overlap}$ is the % overlap, and P_{cases} is the % of cases in the overlap, then $C = P_{overlap} \times P_{cases}$.

For example:

> First attribute-value, sex=M. If the overlap is 20% and the % of cases in the overlap is 60%, then $P_{overlap} = 0.20$ and $P_{cases} = 0.60$, which gives $0.20 \times 0.60 = 0.12$
>
> Second attribute-value, sex=F. If the overlap is 17% and the % of cases in the overlap is 50%, then $P_{overlap} = 0.17$ and $P_{cases} = 0.50$, which gives $0.17 \times 0.50 = 0.085$
>
> This produces a mean value (of the two attribute values) of $(0.12 + 0.085) = 0.205/2 = 0.1025$

We note that in order to compare ordinal variables we also need to incorporate the magnitude.

(D) Comparison of variables of type **Fuzzy Ordinal Categorical** with variables of type **No-Ordinal Categorical**

In processes (A) to (C) all the variables types compared have been crisp (non-fuzzy). Namely, these are: *numeric with numeric*, *categorical with categorical*, and *categorical with numeric*. In process (D) we incorporate a fuzzy variable type, which is considered as a category with a range.

For example, the ICU dataset variable '**Mac_Cabe'** has 3 categories: we can ask what meaning would it have to compare or calculate the correlation between the fuzzy **Mac_Cabe** data, consisting of three membership grades per data item, and the crisp categorical variable '**Sex'**, as can be seen in Table 20.

Table 20. Membership grades of values of categorical ordinal variable 'Mac_Cabe' with respect to values of categorical (non-ordinal) variable 'sex'

Mac_Cabe			Sex
1	2	3	
0.1	0.2	0.9	M
0.9	0.7	0.1	F
0.2	0.8	0.2	F
0.5	0.5	0.5	M
0.7	0.5	0.1	F
0.1	0.8	0.7	M
0.1	0.1	0.2	F
0.8	0.8	0.9	M
0.1	0.9	0.1	M

If we consider 'Mac_Cabe' as 'crisp', taking the category in each case as the one with the highest grade of membership, this results in the values shown by Table 21.

Table 21. (Crisp) values of categorical ordinal variable 'Mac_Cabe' with respect to values of categorical (non-ordinal) variable 'sex'

Mac_Cabe	Sex
3	M
1	F
2	F
2	M
1	F
2	M
3	F
3	M
2	M

Use Chi-Squared, whose definition has been given previously as formula (3.1).

But we are no longer dealing with simple frequencies. We could sum the membership grades and divide by the number of cases. Then we will have M matrices where m = Nº of categories of the fuzzy value. Then one average matrix. But we have to take care not to lose any information.

Table 22. Confusion matrix for variables 'Mac_Cabe' and 'sex'

		Sex	
		M	F
Mac_	1*		
Cabe	2		
	3		

* = Σ membership grades / number of categories

We are looking for consistency in the values. For example, if Mac_Cabe is {0.1,0.2,0.9} for sex='M' in one case, we would expect it to be similar in another. Therefore it needs to be within fuzzy ranges for categories? (only for numeric values). Therefore compute differences and calculate averages. The less the difference, the more the closeness (relation). Can also use for 2 fuzzy values and one fuzzy value with one numeric value.

Table 23. Membership grades of values of categorical ordinal variable 'Mac_Cabe' corresponding to value of categorical variable 'sex' = 'M'

Mac_Cabe			Sex='M'
1	2	3	
0.1	0.2	0.9	M
0.5	0.5	0.5	M
0.1	0.8	0.7	M
0.8	0.8	0.9	M
0.1	0.9	0.1	M

Table 24. Membership grades of values of categorical ordinal variable 'Mac_Cabe' corresponding to value of categorical variable 'sex' = 'F'

Mac_Cabe			Sex='F'
1	2	3	
0.9	0.7	0.1	F
0.2	0.8	0.2	F
0.7	0.5	0.1	F
0.1	0.1	0.2	F

(E) Comparison of variables of type **Fuzzy Ordinal Categorical** with variables of type **Fuzzy Non-Ordinal Categorical**

In this type of comparison we consider two test cases of 'closeness' and 'farness' of a 'fuzzy ordinal categorical' data type (Mac_Cabe) compared with a 'non-ordinal categorical' data type (Sex). In each case example values for the respective membership grades of each category of the first variable are compared with the membership grades of the first category of the second variable, as in Table 25 (below). Then the same respective grades of membership of each category of the first variable are compared with the membership grades of the second category of the second variable, as in Table 26. We note that the cases of the first variable are different in Table 25 and 26, given that they correspond to the crisp categories of the second variable. It follows that first the means of the columns of membership grades are calculated, followed by the standard deviations for each case and membership grade. Then the average standard deviation is calculated for each column, as can be seen in the last row of Table 25. This process is repeated for Table 26. Next the average of the standard deviations is calculated for each of the categories of the second variable. This average is then multiplied by 2 and subtracted from one to give a correlation type value.

That is:

The objective is to compare the membership grades μ of each category of the first variable, $\boldsymbol{v}_1$, of type fuzzy ordinal categorical, with the membership grades of each category of second variable, $\boldsymbol{v}_2$, of type fuzzy non-ordinal categorical, in order to produce a 'correlation' value between the two.

Let σ^{μ}_{ij} be the mean of the membership grades of category **i** of $\boldsymbol{v}_1$, with respect to category **j** of $\boldsymbol{v}_2$. Let σ^{μ}_{j} be the mean of the membership grades of category **j** of $\boldsymbol{v}_2$.

Then, the standard deviation of the membership grade μ for each category **i** and case **k** of $\boldsymbol{v}_1$will be

$$\delta^{\mu}_{ik} = \mu_{ik} - \sigma^{\mu}_{ij}$$

and the standard deviation of the membership grade μ for each category **j** and case **k** of $\boldsymbol{v}_2$ will be

$$\delta^{\mu}_{jk} = \mu_{jk} - \sigma^{\mu}_{j}$$

The average standard deviation for each category **i** of $\boldsymbol{v}_1$ will be

$$\delta^{\mu}_{i} = \left(\sum_{k=1}^{n} \delta^{\mu}_{ik} \right) / \; n$$

The average standard deviation for category **j** of $\boldsymbol{v}_2$ will be

$$\delta^{\mu}_{j} = \left(\sum_{k=1}^{n} \delta^{\mu}_{jk} \right) / \; n$$

The average of the averages of the standard deviations for each respective category of the variables, where nc_1 is the number of categories of $\boldsymbol{v}_1$ and nc_2 is the number of categories of $\boldsymbol{v}_2$, will be:

$$\rho = \left(\left(\sum_{i=1}^{nc1} \delta^{\mu}_{i} \right) + \left(\sum_{j=1}^{nc2} \delta^{\mu}_{j} \right) \right) / \; (nc_1 + nc_2)$$

The resulting correlation will be:

$$C = 1 - (\rho \times 2)$$

The following tests are designed to validate the coherence of the mechanism: (i) test for numbers which have no correlation (random, nearly all different) and (ii) test for numbers which have a very close correlation (nearly all the same). For reasons of clarity, the standard deviations and averages are calculated below in a separate table for each attribute-value of $\boldsymbol{v}_2$, that is sex='M' and sex='F'

(i) The case where the variables are considered as being 'close together':

In the case where the two variables can be considered 'close' to each other, we see in the first two columns of Table 25 the membership grades for the variable Mac_Cabe of each case to the two possible categorical values of this variable. In the third column we see the membership grades for the first category of the variable 'Sex'. It is clear that there is a relation between columns 1 and 3 in that all values of column 1 are high when all values of column 3 are low. Also all values of columns 2 and 3 are low, although to a lesser extent in the case of column 2. This is reflected in the mean standard deviations which are closer for columns 1 and 3.

Table 25. Mean and standard deviation of membership grades of the categorical variable 'Mac_Cabe' for 'sex' = 'M', when there exists a 'close' correlation between the membership grades.

	Mac_Cabe		Sex='M'	
	0.9	0.1	0.1	
	0.8	0.2	0.2	
	0.9	0.1	0.3	
	0.7	0.3	0.1	
	0.8	0.4	0.1	
Means	0.82	0.22	0.16	
Stddev	0.08	0.12	0.06	
	0.02	0.02	0.04	
	0.08	0.12	0.14	
	0.12	0.08	0.06	
	0.02	0.18	0.06	
	0.064	0.104	0.072	Average = 0.08

In the case of Table 26, we see a similar situation as for Table 25, except that the magnitude of columns 1 and 2 are reversed.

Table 26. Mean and standard deviation of membership grades of the categorical variable 'Mac_Cabe' for 'sex' = 'F', when there exists a 'close' correlation between the membership grades.

	Mac_Cabe		Sex='F'	
	0.1	0.8	0.1	
	0.2	0.8	0.3	
	0.2	0.9	0.2	
	0.1	0.9	0.1	
Means	0.15	0.85	0.175	
Stddev	0.05	0.05	0.075	
	0.05	0.05	0.125	
	0.05	0.05	0.025	
	0.05	0.05	0.075	
	0.05	0.05	0.075	Average = 0.0583rec

The average of the averages of the standard deviations is (0.08 + 0.058)/2 = 0.069, multiplied by 2 gives 0.138. Finally, (1- 0.138) gives 0.862 .

(ii) The case where the variables are considered as being 'far apart':

In the case where the two variables can be considered 'far apart' from each other, we see in the first two columns of Table 27 the membership grades for the variable Mac_Cabe of each case to the two possible categorical values of this variable. In the third column we see the membership grades for the first category of the variable 'Sex'. It is clear that there is little relation between columns 1 and 3 given that there are random differences between the values of column 1 and the values of column 3.

Table 27. Mean and standard deviation of membership grades of the categorical variable 'Mac_Cabe' for 'sex' = 'M', when there exists a 'far' correlation between the membership grades.

	Mac_Cabe		Sex='M'	
	0.9	0.1	0.9	
	0.1	0.9	0.1	
	0.9	0.9	0.1	
	0.9	0.1	0.9	
	0.1	0.9	0.1	
Means	0.58	0.58	0.42	
Stddev	0.32	0.48	0.48	
	0.48	0.32	0.32	
	0.32	0.32	0.32	
	0.432	0.48	0.48	
	0.48	0.32	0.32	
	0.384	0.384	0.382	Average = 0.384

In the case of Table 28, we see a similar situation as for Table 27, except that the standard deviations are slightly greater.

Table 28. Mean and standard deviation of membership grades of the categorical variable 'Mac_Cabe' for 'sex' = 'F', when there exists a 'far' correlation between the membership grades.

	Mac_Cabe		Sex='F'	
	0.9	0.1	0.9	
	0.1	0.9	0.1	
	0.9	0.1	0.9	
	0.1	0.9	0.1	
Means	0.5	0.5	0.5	
Stddev	0.4	0.4	0.4	
	0.4	0.4	0.4	
	0.4	0.4	0.4	
	0.4	0.4	0.4	
	0.4	0.4	0.4	Average = 0.4

The average of the averages of the standard deviations is (0.384 + 0.400)/2 = 0.392, multiplied by 2 gives 0.784. Finally, (1- 0.784) gives 0.216.

If we summarise cases (i) and (ii), we see that the 'close together' case produced a value of 0.862 and that of 'far apart' a value of 0.216. This is by virtue of simply averaging the average of the standard deviations of the membership grades of each of the categories of the two variables, and subtracting this from 1. It allows us to obtain a quantitative aggregated measure for the correlation between a fuzzy ordinal categorical variable and a fuzzy non-ordinal categorical variable.

(F) Comparison of variables of type **Fuzzy Ordinal Categorical** with variables of type **Fuzzy Ordinal Categorical**

In this case we consider that both variables are fuzzy and have categories which are orderable. In the example below, for reasons of manageability and clarity, we have assigned 5 fuzzy categories to the variable, although the number could be increased or decreased depending on the nature of the variable and the application.

Table 29. Membership grades for values of fuzzy ordinal categorical variable 'Mac_Cabe' with respect to the membership grades of the also fuzzy ordinal categorical variable previous health state, 'P_H_Stat' .

Mac_Cabe			P_H_Stat			
1	2	3	1	2	3	4
0.1	0.2	0.9	0.8	0.1	0.6	0.0
0.9	0.7	0.1	0.9	0.1	0.1	0.0
0.2	0.8	0.2	0.5	0.5	0.3	0.0
0.5	0.5	0.5	0.1	0.9	0.9	0.1
0.7	0.5	0.1	0.8	0.7	0.2	0.1
0.1	0.8	0.7	0.6	0.5	0.1	0.0
0.1	0.1	0.2	0.1	0.8	0.8	0.1
0.8	0.8	0.9	0.8	0.1	0.5	0.1
0.1	0.9	0.1	0.9	0.4	0.2	0.0

Now look for correspondences: if {0.1, 0.2, 0.9} for {0.8, 0.1, 0.6, 0.0}, then if another {0.1,0.2,0.9}, P_H_Stat should be around {0.8, 0.1, 0.6, 0.0}. This needs a lot of comparisons.

One approach would be to convert the second variable to 'crisp' leaving the first variable as fuzzy, and apply process **(D)**. Then convert the first variable to crisp leaving the second variable as fuzzy and apply process **(D)**. Finally, take the average of the two covariance calculations.

Alternatively, the proximity of each value to all other values could be calculated, which would result in a distance with which we could calculate the respective mean, standard deviation, and average.

Table 30. Example membership values for 'Mac_Cabe' and 'P_H_Stat' categories

Example:	Mac_Cabe			P_H_Stat			
	1	2	3	1	2	3	4
	0.1	0.2	0.9	0.8	0.1	0.6	0.0

Needs a 3x4 array to hold distances:

Table 31. Distances (differences) between membership values for 'Mac_Cabe' and 'P_H_Stat' from Table 30.

	0.8	0.1	0.6	0.0
0.1	0.7	0.0	0.5	0.1
0.2	0.6	0.1	0.4	0.2
0.9	0.1	0.8	0.3	0.9

If we add up all the differences in Table 31, then we have a sort of 'distance' between the two variables.

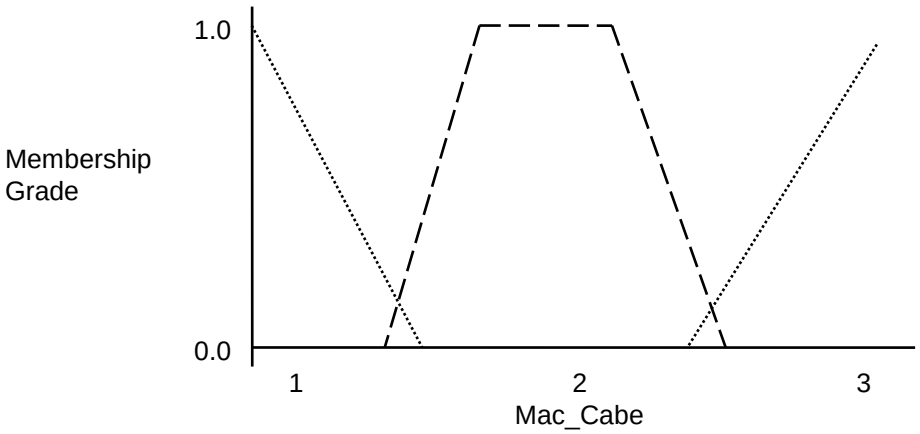


Figure 44. Trapezoidal and Triangular membership functions for fuzzy categorical variable 'Mac_Cabe'

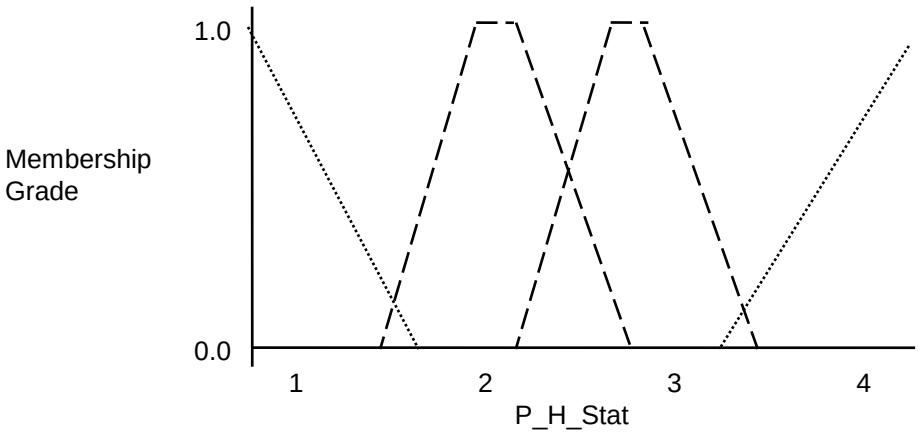


Figure 45. Trapezoidal and Triangular membership functions for fuzzy categorical variable 'P_H_Stat'

3.1.4 Fuzzy covariances – Nettleton's fuzzy covariance calculation

If the data we wish to explore and model contains two or more variables of fuzzy type, then we need to be able to calculate distances between those variables, their correlation and covariance with respect to each other, and their relative similitude and disimilitude. In order to compare one fuzzy variable with another fuzzy variable, we need a method which calculates distances in a fuzzy c-partition space. With reference to [Bezdek81] and [Gustafson90], we consider these approaches as a basis for developing a covariance calculation between two variables which are defined in the fuzzy form. We do this because we are interested in first capturing data for, say, two fuzzy variables A and B, each of which whose data values consist of grades of membership to two or more fuzzy sets. We then wish to be able to quantify the relation between A and B as a covariance type value. This is interesting because there are very few general and simple methods for calculating covariances between fuzzy variables in the literature. Also, we wish to use the technique on real data, such as the ICU dataset, and compare results to a crisp calculation, in order to observe possible improvements in diagnostic precision and in explicative value in medical terms, of the resulting identification of the strength (or weakness) of relationship between variables. We note that given that the data is input to a fuzzy c-Means type algorithm, it can be captured as crisp, and the fuzzy measure will be the calculated distance of the cases of a variable to the prototype in each fuzzy set.

(a) Considerations for representing and comparing variables in the fuzzy form

The representation of variables in the fuzzy form, and their comparison, gives rise to certain aspects which requiere consideration, and which are discussed as follows.

(i) Does it have meaning to represent variables in the fuzzy form when they are neither deterministic or probabilistic? For example, if the measurement of variable has an error factor (more or less *x*) due to the limits of precision and calibration of a machine, we could use the normal distribution of the error to construct a membership grade for the crisp value.

Consider an ordinal categorical variable such as 'duration of stay in hospital', whose possible values are, 'short', 'medium' and 'long'. There is no exact definition, not even by consensus of the medical experts. That is, it is different for distinct permutations of circumstances, medical conditions, hospitals, patient histories, and so on. Therefore, this variable is a candidate for an interpretation by grades of membership, for example, {short: 0.5}, {medium: 0.3}, {long: 0.0}. This may result useful for doctors with planning responsibilities for evaluating the assignment of resources and for the estimation of the individual patient needs.

It may have no meaning to represent a variable in the fuzzy form when their form of measurement/reading is totally precise in all cases, that is, within the permitted error range (tolerance) assigned to that variable - its margin of variation is therefore not significant.

(ii) The representation of a variable in the fuzzy form depends on its type. If it is numerical, we can establish a normal distribution and divide the resulting plot in ranges. If it is of a qualitative ordinal type, we can establish ranges based on its quantitative value, ideally advised by an expert in the data domain. If the variable is of a qualitative nominal type, we may establish a measure based on the number of cases which coincide with each value.

(iii) The covariance between two fuzzy variables (X and Y): fuzzy c-Means establishes a fuzzy prototype for each fuzzy set. Then it calculates the distance of each case from the prototype in a given cluster and so on for all clusters. It is first necessary to convert the cases of X and Y in values which are comparable, for example, via some process similar to normalisation.

If we have a sample of **n** pairs of membership grades of two fuzzy variables X and Y

$$X: x_1\ x_2\ \ldots\ x_n$$
$$Y: y_1\ y_2\ \ldots\ y_n$$

If $\overline{x} = 1/n\ \Sigma x_i$, $\overline{y} = 1/n\ \Sigma y_i$, the fuzzy covariance of the sample will be

$$S_{xy} = 1/n \sum_{i=1}^{n} (x_i - \overline{x})(y_i - \overline{y}) \qquad (3.2)$$

being

$$S_{xy} = 1/n \sum_{i=1}^{n} \overline{x_i y_i} - \overline{x}\ \overline{y} \qquad (3.3)$$

In the fuzzy context, we could interpret **x** and **y** as cluster centre $\mathbf{v}_i$ (see later in this section).

Thus we make use of the output of fuzzy c-Means (the membership grades) and in calculating the covariances it is not necessary to consider the type of the variable. This is because the pre-processing of the variables will have produced a input file in the standard form for input to fuzzy c-Means.

Thus, we interpret covariance in this context as the variation of the grade of membership of two variables in a set. The original data, given as input to fuzzy c-Means, could consist of two columns, one for each variable and one row for each case. Fuzzy c-Means produces as output two columns of data, the columns corresponding to the grade of membership of each variable, and the rows to the grades of membership of each case.

Now we consider the grades of membership for the calculation of covariance (above) using the standard covariance formula, which gives us the 'distance' between the grades of membership of each variable: intuitively, this is the distance between the weighted sum of the variation of the distance of each variable from the fuzzy prototype of each class.

(iv) Once we have the fuzzy covariances calculated, as explained in (iii,above), we may now proceed to apply the Hartigan fusion algorithm to them. Remember that we are looking for pairs of maximum covariances and fusing (joining) them to form one pair of variables as indicated in each iteration.

What problem could we have in using the standard fusion algorithm? Are we overlooking anything with respect to the interpretation of the 'fuzzy covariances'? Is it true that the highest fuzzy covariance indicates the two variables with the greatest interrelation, in each iteration?

If the fuzzy covariance between two variables is a maximum, then this indicates that the two variables in question have the smallest variation in their grades of membership to the same fuzzy cluster centre (fuzzy prototype) , as can be seen in Figure 46 (below).

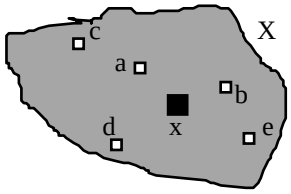


Figure 46. Variables a and b have the highest fuzzy covariances

Of the five variables {a, b, c, d, e} illustrated within the fuzzy set X, 'a' and 'b' have the smallest mutual variation between their respective distances from the cluster centre (or fuzzy prototype) 'x'.

(v) The fusion algorithm gives us three results for a fused pair of variables:

(a) The modified covariance of the new variable with respect to the remaining variables.
(b) The two variables selected to be joined are identified by a symbolic identifier.
(c) A loading matrix B gives the coefficients and factors necessary to 'go back' to the data, and fuse the data for each variable. This information indicates the fractional proportions which each original variable contributes to the new factor, in a similar manner to 'principal components'.

If the variables are fuzzy, we can interpret the new fuzzy variable produced by the fusion as a consensus between the two original fuzzy variables. What problem could arise from this? One possible consequence would be that we lose the original meaning of the data. For example, if variable A is *''grade of risk of death'* and variable B is *'duration of stay in hospital'*, we could ask what meaning would the resulting variable have? If variables A and B really do have the highest correlation between their respective grades of membership of all pairs, we could say that we have established that the most significant is between precisely these two variables.

We must not forget that it is the fuzzy cluster centre, or prototype, which we are using as a reference point and for comparison. Fuzzy c-Means generates 1 or more fuzzy sets ('c' parameter) which are essentially abstract clusters. Later, the investigator may establish that fuzzy sets 'X', 'Y' and 'Z' have grouped the cases by 'duration of stay in days', and correspond, respectively to cases of 'short', 'medium' and 'long' duration of stay.

(vi) In order to formalise the fuzzy variable C, which is the fusion of two fuzzy variables A and B, it is necessary to return to the steps used to calculate the covariance of the two fuzzy variables. In [Bezdek81], a calculation is given for a fuzzy covariance matrix and the interpretation of the distances from the centroid, but the covariance is understood as that between a variable and the centroid, as opposed to that between two variables:

The 'fuzzy scatter matrix' of cluster u_i in this case is:

$$S_{fi} = \sum_{k=1}^{n} (u_{ik})^m (\mathbf{x}_k - \mathbf{v}_i)(\mathbf{x}_k - \mathbf{v}_i)^T \qquad (3.4)$$

for the n_i points in 'crisp' cluster u_i that has centroid $\mathbf{v}_i$. Thus, if U ε M_c , the number $(\mathbf{x}_k - \mathbf{v}_i)^T C_i^{-1} (\mathbf{x}_k - \mathbf{v}_i)$ will be the squared Mahalonobis distance between $\mathbf{x}_k$ ε u_i , and its sample mean will be (sub) $\mathbf{v}_i$, where C_i^{-1} is the inverse of the sample matrix of covariances of the points in u_i. This may be interpreted as the squared *fuzzy* Mahalonobis distance between $\mathbf{x}_k$ and the fuzzy cluster centre $\mathbf{v}_i$. The memberships are distributed to minimise the '*global volume of fuzzy dispersion'* of **c** fuzzy clusters. As *modus operandi* to find an optimum solution, different norms can be applied, for example, the Euclidean (N_E) or Diagonal (N_D), once the Mahalonobis (N_M) has been tried. We could also carry out a 'benchmarking' with 'crisp' partitions to establish what improvement (if any) is caused by using a fuzzy interpretation of the data being processed. One quality (or goodness) criteria we could use would be 'clustering error', or the percentage of cases classified incorrectly with respect to the historical data.

(b) Nettleton's fuzzy covariance calculation

A 'fuzzy covariance algorithm' has been developed which allows for a covariance calculation between variables represented in the fuzzy form. As a basis, the Gustafson and Kessel fuzzy covariance algorithm has been used [Gustafson79]. This algorithm calculates a distance between values, that is points, and in order to calculate a distance between variables, it was adapted, as described in my papers [Nettleton98b][Nettleton99b], and in the following section.

Calculating fuzzy covariances

In the literature there are few examples of general purpose 'fuzzy covariance' algorithms, such as that of Gustafson and Kessel [Gustafson79]. This algorithm, however, computes covariances of variables with cluster centres, and not variables with other variables. In the following, an extension of this is described which allows for the calculation of 'fuzzy covariances' between variables. In the literature, there are also examples of fuzzy covariance algorithms designed for specific problems, such as [Nakamori97] and [Babuska96]. Even so there is still a need for more work on generic variable-variable fuzzy covariance algorithms.

In this section we describe some new methods for calculating generic 'fuzzy covariances'. All methods use the fuzzy c-partitions generated by fuzzy c-Means, which is allowed to run to termination before the methods process the resulting data matrices and vectors.

Method 1 produces fuzzy covariances from a self-contained algorithm. Methods 2 to 4 create a data matrix (j variables by k cases) of different weighted aspects of the c-partitions (cluster centres, membership grades, data values, norm coefficients), which is then passed to a standard covariance algorithm to calculate the covariances. For Methods 2, 3 and 4, the resulting C matrix is used to calculate the covariances between the variables and the membership grades. The process is repeated for each cluster i.

(i) Variation of Gustafson's Method - Method 1

A variant of Gustafson and Kessel's algorithm [Gustafson79] is used to generate a fuzzy covariance matrix, using a fuzzy c-Means type algorithm. This algorithm was first defined and benchmarked with other methods and standard data sets in [Nettleton98b].

If $\mathbf{u_i}$ is the matrix of membership grades of **n** cases to cluster **i**; $\mathbf{u_{ik}}$ is the membership grade of case **k** to cluster **i**; $\mathbf{x}_k$ is the vector of characteristics (data) of case **k**; $\mathbf{v_i}$ is the centroid of cluster **i**; **m** is a weighting factor which defines a *grade of fuzziness*; $(\mathbf{x}_k - \mathbf{v}_i)(\mathbf{x}_k - \mathbf{v}_i)$ is a distance similar to that of Mahalonobis.

We measure the grade of relation of a variable V_1 with the centroid of a cluster C_1, and then measure the grade of relation of a second variable V_2 to the centroid of the same cluster C_1. The distance is the difference between the grade of relation of V_1 to C_1 and the grade of relation of V_2 to C_1, that is, $\mathbf{d}(V_1, C_1) - \mathbf{d}(V_2, C_1)$. It follows that the calculation of the fuzzy covariances between variables in each cluster **i** is calculated with the following formula:

$$C_{fi} = \sum_{j=1}^{\rho} \sum_{k=1}^{n} (u_{ik})^m \sum_{q=1}^{\rho} \|\mathbf{d}(V_1, C_1) - \mathbf{d}(V_2, C_1)\| \quad (3.5)$$

where

$$\mathbf{d}(V_1, C_1) = (\mathbf{x}_{jk} - \mathbf{v}_i)(\mathbf{x}_{jk} - \mathbf{v}_i)$$

and

$$\mathbf{d}(V_2, C_1) = (\mathbf{x}_{qk} - \mathbf{v}_i)(\mathbf{x}_{qk} - \mathbf{v}_i)$$

ρ being the number of variables (dimensions), and **n** the number of cases.

(ii) Method 2

This method measures the relation between the membership grades and the data values of the objects (cases). If u_{ik} is the membership grade of case k to cluster i, and y_{kj} is the value of the jth variable of case k. Then C_{kj} is the product for the kth case of variable j, and C_{kq} is the product for the kth case of variable q. The first column of C_{kj} (variable 1) is loaded with the membership grades for the current cluster u_{ik}. The subsequent columns of C_{kj} are loaded with the corresponding data values in y_{kj}.

$$C_{kj} = u_{ik} \quad , j = 1$$

and

$$C_{kj} = y_{kj-1} \quad , j = 2, \rho \quad (3.6)$$

where ρ is the number of variables. The resulting matrix C has dimension **n** by ρ +1.

(iii) Method 3

This method measures the relation between the distances of the objects from the cluster centres, weighted by the norm coefficients. In method 3, y_{kj} and C_{kj} have the same meanings as in method 2. We introduce v_{ij}, which is the centre of cluster i for variable j, and cc_{jq}, which is the calculated norm for variable j and case q. The norm has been fixed for all tests as the euclidean norm.

$$C_{kj} = \sum_{q=1}^{\rho} (y_{kj} - v_{ij}) \times cc_{jq} \times (y_{kq} - v_{iq}) \quad (3.7)$$

where ρ is the number of variables. The resulting matrix C has dimensions n, the number of cases, by ρ.

(iv) Method 4

This method measures the relation between the sum of squares of the distances of the objects from the cluster centres, weighted by the norm coefficients and the corresponding membership grades. Method 4 makes the same calculation as method 3, and then does the following:

$$C'_{kj} = C_{kj} \times (u_{ik})^m \quad (3.8)$$

where **m** is a weighting factor which defines a *grade of fuzziness*, as in Method 1. The resulting matrix C is given as input to a standard covariance algorithm to calculate the covariances between the variables and the membership grades. The process is repeated for each cluster i.

Summary of Methods (i) to (iv)

Method **(i)** generates fuzzy covariances directly, whereas methods **(ii)** to **(iv)** allow us to study the different components and weights which intervene in fuzzy c-Means, and evaluate the strength of relation between them, using a standard covariance algorithm. These methods are tested and benchmarked against other algorithms in Section 4.2.

3.1.5 Improving questionnaires for Sleep Apnea diagnosis

With the objective of improving the questionnaire as a diagnostic screening tool, we have designed a study where the patient was given a general sleep questionnaire, which permits a double evaluation in a scalar and a categorical form for each question, in order to see if the scalar form extracts a greater information from the patient and thus produces a greater correlation with the AHI (apnea hypopnea index, see Section 1.2.9).

A total of 71 patients have so far been processed with this method, chosen at random in the Sleep Pathologies Center (Salamanca), and studied with respect to diverse problems: insomnia, somnolence, snoring, apneas, body movement during sleep, nocturnal choking, etc. … These patients were administered the questionnaire and were given a complete night-time polysomnogram, or a supervised night-time cardio-respiratory polygram, in the Sleep Unit. The following variables were recorded: oral-nasal airflow, snoring, thoracic and abdominal respiratory effort, body position, actimetry, electrocardiogram, pulse and oxygen saturation in haemoglobin. The AHI has determined for all patients, and this value was used to compare the predictive accuracy of the different types of questionnaire.

INHERENT PROBLEMS OF STANDARD QUESTIONNAIRES AND PROPOSED SOLUTIONS

The purpose of the questionnaire is to provide an information profile of the patient which allows a pre-diagnosis of his/her condition. This acts as a 'screening' which avoids patients entering into the sleep centre for expensive and time consuming testing, when they have a low probability of suffering from Apnea Syndrome, or have some other pathology.

The questionnaire consists of two main sections: the first records clinical data, with 15 key clinical variables: age, sex, presence of a partner, profession, work hours, education level, weight, height, neck circumference, BMI (body mass index), blood pressure, alcohol intake, cigarette intake, auto-evaluation of most important symptoms, other illnesses; the second section consists of 41 questions to which the patient responds on a five point scale {never, rarely, sometimes, frequently, always}. The questions are divided in 3 subsections: 15 general sleep questions, 16 respiratory related questions and 9 somnolence related questions. Based on this information, the doctor then gives a clinical evaluation: healthy; simple snorer; doubtful; typical apnea; other illness. We interpret this as: typical apnea; no apnea, with the corresponding grade of membership. Refer to Annex 3 for the complete scalar version of the questionnaire used.

One of the fundamental problems with the questionnaire responses is that in the general sleep and respiratory related questions, there are several key questions which rely on the bed partner as a witness. Of course if there is no bed partner, or the bed partner does not know, this eliminates some key information for the diagnosis. To improve this situation, in the case of there being a bed partner, we propose that s/he fills in the same questions separately in a different questionnaire. The responses can then be cross checked for contradictions and inconsistencies between the bed partner and the patient.

Another aspect is that the general public may respond incorrectly, untruthfully, or simply not understand the questions correctly. There are several standard techniques used in general questionnaire design which can help to identify inconsistencies or contradictions. One of the techniques consists of asking the same question several times but phrased in a different manner, throughout the questionnaire. Also we can ask a question and later its inverse, in a non-obvious manner, to detect contradictions. From this information we can derive a reliability index for the whole questionnaire for a given patient and/or a reliability grade for each individual question response.

Each patient filled in two versions of the questionnaire - one with categorical responses and the second with fuzzy scalar responses. Each patient was previously briefed as to how to fill in the questionnaires. In practice, the patients were from all types of backgrounds, educational and cultural levels. Sometimes there were errors in how the patient responded to the categorical and scalar response representations. One typical error is that the patient responds to the scalar representation as if it were categorical, placing a cross exactly on the label point in each case. Thus there was no ponderance of a grade of membership by the patient. The lesson we have learnt from this is to dedicate more time to explaining to each patient the importance of thinking about the scalar response in order that they can appreciate our objectives in doing this. For example, the added subtlety of placing a cross on the scale, for example, two thirds of the way between frequently and always, but closest to always. Of course, as the subjects are from the general public, this is not an easy task.

Finally we have a problem of data – having sufficient cases with which to test our method. Take into account that for many machine learning techniques for each N input variables we need N*10 cases. The data should also represent a homogeneous group of the population, such as professional males between 45 and 65 years of age with medium education level, living in the same geographical area, without secondary ailments. This is not our case, but instead we

have real data with which any clinic has to deal with every day, and our objective is that our data processing methods give useful and acceptable results from it.

Some questions in the questionnaire can only be responded by a partner by observing the patient while sleeping. The following give some examples of this type of question, together with other questions to be answered by the patient him/herself.

Examples of questions put to partner and patient

TO PATIENT
G11 DO YOU KNOW OR HAVE YOU BEEN TOLD THAT YOU MOVE YOUR LEGS A LOT WHILE YOU ARE SLEEPING?
1- never 2- rarely 3- sometimes 4- often 5- always

TO PARTNER
G11 DOES YOUR PARTNER MOVE HIS/HER LEGS A LOT WHILE S/HE IS SLEEPING?
1- never 2- rarely 3- sometimes 4- often 5- always

TO PARTNER
R1 DO YOUR PARTNER SNORE WHILE S/HE SLEEPS?
1- never 2- rarely 3- sometimes 4- often 5- always
TO PARTNER
R2 DOES YOUR PARTNERS SNORING WAKE YOU UP OR CAN IT BE HEARD FROM ANOTHER ROOM?
1- never 2- rarely 3- sometimes 4- often 5- always

TO PARTNER
R7 DOES YOUR PARTNER "STOP BREATHING" WHEN S/HE IS ASLEEP ?
1- never 2- rarely 3- sometimes 4- often 5- always

TO PARTNER
R8 HAVE YOU WOKEN YOUR PARTNER BECAUSE YOU THOUGHT THAT S/HE HAD STOPPED BREATHING?
1- never 2- rarely 3- sometimes 4- often 5- always

Examples of questions asked more than once in a different manner

R5 HAVE YOU NOTICED AN INCREASE IN YOUR SNORING RECENTLY?
1- no 2- <6months 3- 6-12 months 4- > 1 year

R17 HOW LONG HAS YOUR SNORING STAYED THE SAME?
1- no 2- <6months 3- 6-12 months 4- > 1 year

(a) Fuzzy Data Representation – Apnea questionnaire screening

Questionnaire screening was used as an example of comparing a crisp data representation approach with a fuzzy data representation approach. Also this problem allowed applying a complete fuzzy data representation process to a real problem: from deciding which questions to include, in what order, the form of the questions (way of asking them), the number and nature of the linguistic labels, the nature and form of the underlying membership functions. Data capture, and data processing were the two following areas which had to be addressed. Data processing was using WOWA, modified to enable the processing of missing data, and learning the weights using genetic algorithms.

Representing the linguistic labels questionnaire responses in the fuzzy form

Now we consider how to represent linguistic labels as fuzzy sets: first as simple trapezoids and then as curves which give a smoother transition between one label and the next.

Parmenidean Pairs: in general, the basic representation for parmenidean pairs is based on the use of fuzzy partitions with a trapezoidal membership function, as discussed previously in Section 1.2.4 .

From linear to non-linear membership functions

Trapezoids formed by straight lines are really approximations of real membership functions. Thus we can get closer to having a natural representation (best fit) for the linguistic labels by generating a curve in place of a straight line for the ascending and descending gradients. To achieve this, we could use an appropriate function to generate the points for the desired form of the curve. In some cases we may wish to strengthen a transition with hedges like "very" or "extremely" or weaken it with, for example, "slightly". We can perform strengthening by, for example, a sigmoid-like function, such as Zadeh's S-Function.

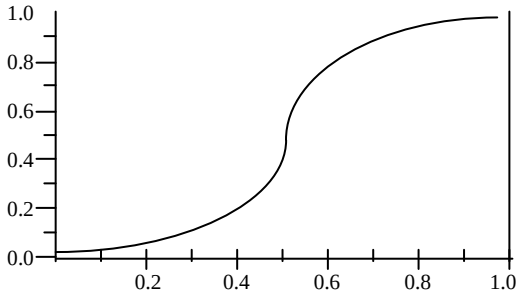


Figure 47. Zadeh's s-function can be used to customise membership transition

For linear and non-linear membership gradients, we assume a symmetrical relation between the descending membership value for the preceding fuzzy set and the ascending membership grade for the following fuzzy set (which sum to 1), as can be seen in Figure 47.

Construction of membership curves – some considerations

In Figure 48, we see a geometric construction, three segments of a membership function: segment 1 we will call the concave segment which goes from the bottom left hand corner to point 3; segment 2 we will call the linear segment which goes from point 3 to point 5; and segment 3 will be the convex segment which goes from point 5 to the upper right hand corner. As can be seen, there are seven interpolation points. The curve construction function of Microsoft Excel then uses splines to approximate the curve to the points. The points are located simply on midpoints and intersections of successively divided quadrants and diagonals. The curve in the upper right quadrant is a rotated and inverted mirror image of the curve in the lower left quadrant, the overall curve being therefore symmetrical. If we go down and right on point 2 we will make the gradient steeper and make the incoming 'hedge' into the corresponding fuzzy set occur more rapidly. Correspondingly, if we push points 1, 2 and 3 upwards a little we will make the gradient shallower and make the 'hedge' less intense. Of course, care is necessary, in the case of manual manipulation, to avoid inflexions which would create situations such as a second concave segment in the upper right quadrant.

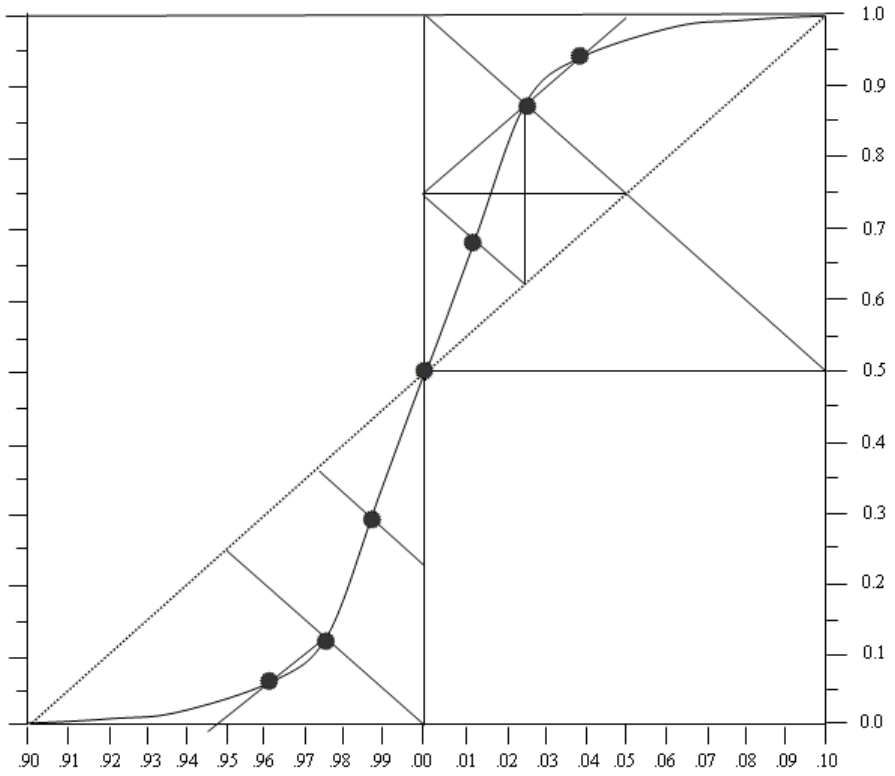


Figure 48. Construction of a membership curve

Membership curves generally tend to have the form as in Figure 48, with first a lower concave segment, followed by an upper convex segment. The concave segment in Figure 48 covers 50% of the x-axis, that is the total curve length on the x-axis. This can be reduced to 25%, for example, leaving 75% for the upper convex segment. This would result in an overall bell shaped appearance. What would not be usual, would be if the upper right segment was also concave. We can ask the question, why is this not done? In general, the transition from one state to another, where the states are ordered, naturally has a convex phase followed by a concave phase which avoids 'glitches'.

Example of fuzzy representation of a questionnaire response

For each question we design a membership function which can be overlaid on each scale to read off the grade of membership to each linguistic label.

S5. Do you fall asleep while driving on the motorway?

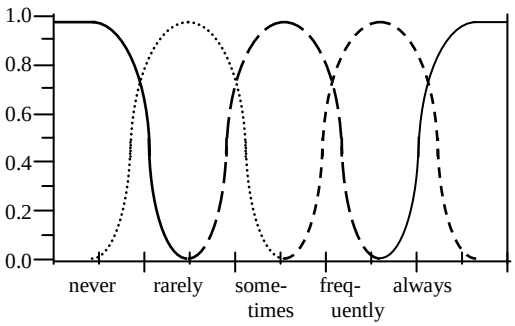
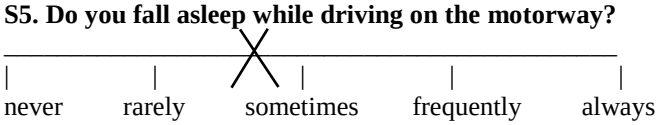


Figure 49. Example of representation for a critical question

In Figure 49 we see that the curves are formed by Zadeh's s-function. We can manipulate this type of curve as detailed previously, in order to strengthen or weaken a linguistic label. The patient draws a cross on the continuous scale (e.g. S5) to indicate his/her response to the question. In the questionnaire, this question would appear as:

S5. Do you fall asleep while driving on the motorway?



The fuzzy response would be stored as a quintuple, with a membership grade for each linguistic label. For example the response to S5 (above) could be stored as: {0:0.0, 1:0.3; 2:0.7; 3:0.0; 4:0.0}. This indicates that only linguistic labels 'rarely' and 'sometimes' have non-zero membership values, being 0.3 and 0.7 respectively. We can simply take the linguistic label with the highest membership grade, which in this case is 'sometimes'. Note that we can convert to categorical if we so desire, and that way we have both crisp and fuzzy data capture.
The membership grades of the responses can be read or by writing a computer programme which finds the corresponding point on the y-axis, for the point indicated by the response on the x-axis. Other wise, we can overlay a transparent sheet on each response line and read off the membership grade on the y-axis. Each sheet would have been drawn or created by a statistical package. We have chosen at present the latter method, which avoids dedicating time to programming and enables us to tailor one sheet of membership functions for each question.

In Figure 50 we see alternative forms of membership function to the symmetrical and equal forms of Figure 49. In Figure 50 the lower horizontal scale 'label' has four possible fuzzy sets: 'none', 'slight', 'moderate' and 'high', which refer to the incidence of apnea in the patient. The horizontal scale 'index' is simply an equidistant scale of zero to 1.0. Finally, the horizontal scale 'RDI' is the real RDI value derived from the clinical test. Thus the plot can be used to read off the corresponding membership grade on the vertical axis. The fuzzy sets correspond to the linguistic labels whose ranges are defined on the first horizontal axis 'labels'. Alternatively, or additionally, the RDI can be converted to a value between 0 and 1 on the linear horizontal scale 'Index'. The curves in Figure 50 have a meticulous design with respect to the clinical 'weight' which each linguistic label has. For example, the label 'none' occupies only 5% of the length of the horizontal axis, and has the steepest fall-off (also called 'hedge' or 'quantifier') of all the linguistic labels. On the other hand, the label 'high' has the least steep hedge and occupies approximately 50% of the horizontal scale. We also see that the label 'moderate is symmetrical whereas 'slight' is not. This is because of 'slights' interaction on the left hand side with 'none'. The two design criteria which involve medical expert judgement are (i) the amount of the scale which the linguistic label occupies and (ii) the steepness of fall-off on the left and right hand sides. The first criteria depends on the clinical range for which the incidence is defined; thus in clinical terms, apnea incidence is considered to be 'high' when the RDI is approximately 45 upwards. The second criteria depends on the fuzziness of the change from one label to the next. For example, the distinction of classification between patients who are 'none' and patients who are 'slight' is crisper than the distinction of classification between patients who are 'moderate' and those who are 'high'. The curves in Figure 50 were defined with the curve creation utility of Microsoft Word 97, which has the same automatic interpolation mechanism as Microsoft Excel 97. Once a basic curve is defined, it can be manipulated by inflexion points or by adding knots.

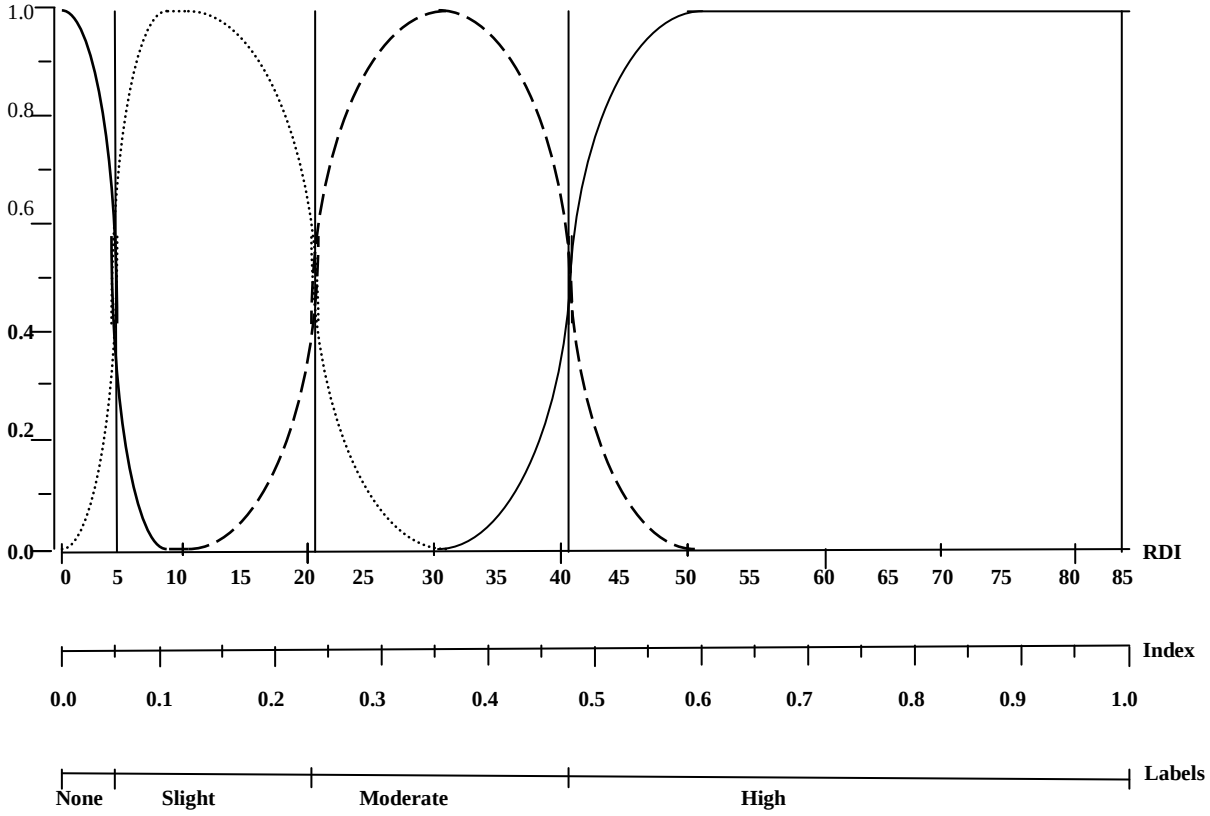


Figure 50. Example of non-symmetrical membership curves to represent output variable

3.2 Aggregation of data of different types

Data fusion may be considered a preprocessing technique which converts two or more inputs into just one output which serves as input to a subsequent data processing step. In Section 3.2.1 we look at diverse data fusion schemes in the literature, some of which use heterogenous representation formats as a solution, whereas others use generalised distance metrics, or different methods for each different type of data.

Section 3.2.2 details the implementation of Nettleton's version of the Hartigan 'joining' algorithm, which is applied later on real ICU data In Section 4.1, and other test data in 4.2, and the results compared with those of other factor analysis and variable aggregation methods.

Data fusion can also be considered as aggregation: in Section 3.2.3, we are interested in finding a data modelling method, which allows the inclusion of meta-data such as reliability and relevance information about the data. It also has to be able to give acceptable results with small datasets, because the real domains we use in Section 4 are of this nature. Also it must also be relatively easy to include enhancements, such as missing data processing and machine learning for meta-data assignment. In this section we consider methods of grouping input variables and data values. In Section 3.2.3 we consider aggregation operators, especially WOWA, which allow the inclusion of 'meta-data' as input to 'bias' the data values, on the one hand, and the variables, on the other hand.

3.2.1 Mixed Data Types - Data Fusion

Data fusion can have different interpretations: on the one hand we can aggregate data values from different data sources to give a global consensus on an output state or diagnosis; on the other hand, we can 'join' the variables themselves based on similarities and data characteristics, into a reduced number of factors. In the latter case for example, we could create a new variable C as a function of two variables, A and B; for example $C = ((A \times 0.15)/A + (B \times 0.85)/B)$. We join A and B, given that we have previously defined a relationship between these two variables, which furthermore has the proportional contributions as indicated in the previous formula, that is 15% of A and 85% of B.

We now consider different ways of processing and representing data in order to treat mixed data types in one algorithm. The state of the art has evaluated with respect to processing techniques which try to resolve the problem. It has been found to be an area in which no perfect or definitive solution exists, neither from the traditional statistics field, or from the artificial intelligence field. One of the problems in mixed data type processing is to calculate a distance measure between any two types of data; for example, the distance between the ordinal categorical attribute "duration of stay in hospital" (short, medium, …) and the numerical attribute "age" (23, 45, …). One possible approach is to convert all data to one type, for example, categorical or numeric or continuous, and use an appropriate technique to process that data type. Another approach is to maintain the original data types and devise an algorithm which can create a distance measure in terms of each data type. A third approach would be to use grades of membership, in which all the types exist as before, but also have a fuzzy interpretation. This latter approach, of fuzzy representation for all data types, has been considered in the thesis as one of the simplest to implement[Hathway96][Nettleton99b]. One of the existing systems which addressed the problem of crisp mixed data type processing is Klass [Gibert94], which has a unified distance measure calculation for categorical and numeric data. Traditional statistics has diverse techniques which separately address different data types: for numeric types there is Pearson-product-moment; for categorical data there is the Wilcoxon test, Box-Jenkins, and so on.

In Figure 51 (below) we see a depiction of the two types of algorithm, crisp or fuzzy, which we could use, together with the four different groups of data types we may encounter. The inner boxes, such as that representing 'numeric and continuous data types' are the most specific, whereas the outer boxes, such as that representing 'fuzzy and crisp mixed data types', generalise and contain the data types defined within their scope. When we refer to a 'crisp' fusion algorithm we mean an algorithm which 'joins' variables a factor reduction process, and where the decisions of which variables de 'fuse' and in which order are crisp. On the other hand, a 'fuzzy' fusion algorithm will use fuzzy decisions or a fuzzy distance measure to decide which variables to join and in which order. We are therefore referring to the mechanism of the algorithm itself. In contrast, the data types indicate the form of representation of the data itself, which may be any of the crisp data types given (below) or the fuzzy data type.

Four crisp data types are considered: integer, that is, whole numbers, {1,2,3, …N}; continuous or floating point numbers, that is {0.45, 1.71, 21.02, …}; categorical ordinal, with attribute-categories such as, {'low', 'medium', 'high'}; and categorical non-ordinal, with attribute-categories such as, {'blue', 'red', 'green', 'male', 'female', …}. A fuzzy data type, on the other hand, is typically a categorical ordinal or non-ordinal type, for which a data item may

belong to more than one of the attribute-categories, and the data value is accompanied by one or more grades of membership to the corresponding categories.

By ‘mixed data type’, we understand a dataset in which data of different types coexist, that is, integer, continuous categorical ordinal and categorical non-ordinal, but excluding the fuzzy type. Our objective is to be able to process data of ‘mixed types’ and the fuzzy type in a unified manner. Of course, we may also consider the crisp data types as a special case of the fuzzy data type, in which the membership grade is 1.0 for one and only one category, and 0.0 for the remaining categories. Thus the depiction in Figure 51, that the fuzzy data type generalises the crisp data types.

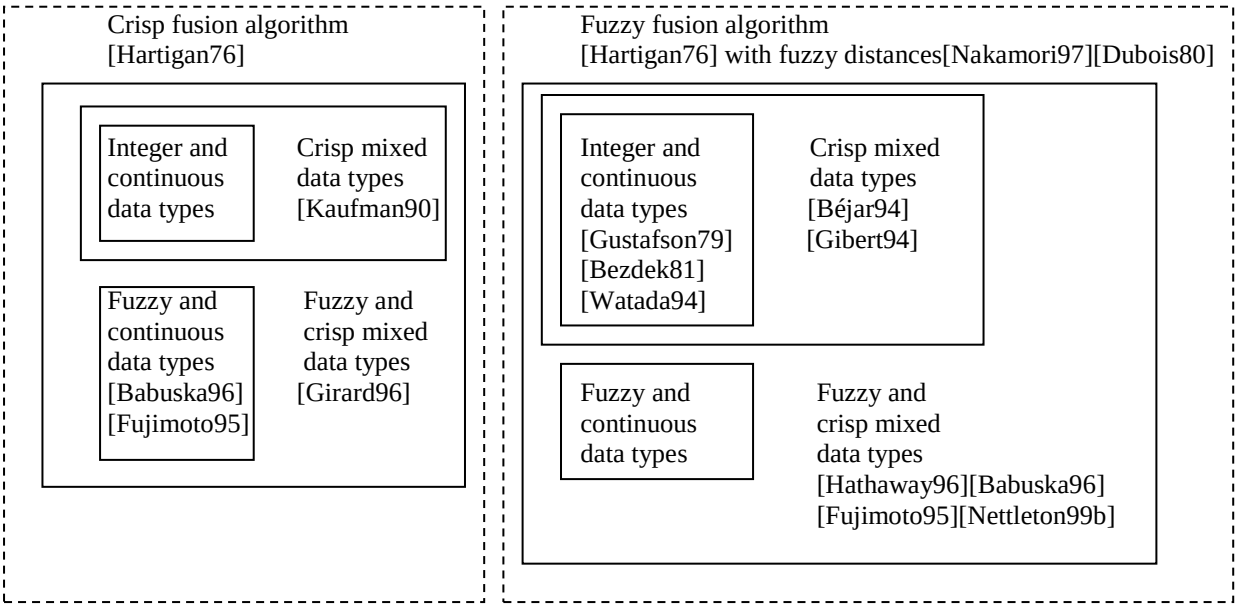


Figure 51. Relation between crisp and fuzzy data fusion algorithms, and different types of data

In the following sections, for each data type combination, we outline a candidate technique to implement the algorithm and treat the data.

(1.1) Crisp fusion algorithm with (crisp) integer and continuous data: we can use the standard Hartigan algorithm. The data is numerical and we can calculate a standard correlation matrix to give as input to the fusion algorithm.

(1.2) Crisp fusion algorithm using (crisp) mixed data types: mixed data types in this context means the processing in the same step of continuous, integer, categorical nominal and categorical ordinal data. We can use the standard Hartigan algorithm to process, but we need to pre-process the different types of data with different measures to achieve a similarity, correlation or distance measure between variables. Examples of such measures are: Chi-Squared for nominal-categorical and binary variable types; Kendall or Spearman for ordinal-categorical variable types; correlation for numeric and continuous variables. These techniques propose solutions for the comparison between pairs of variables of the same type. Another problem is to compare pairs of variables of different types. This may mean the comparison of a categorical variable with a numerical continuous variable. There actually exist various solutions, for example, the Similarity Measure of Linneo [Béjar94], the Mixed Metric of Klass [Gibert94], [Girard96], and so on. Finally we achieve a matrix of values which represent the similarity or affinity measures between all pair permutations of the variables under consideration. This matrix would then be given as input to the standard fusion algorithm.

(1.3) Crisp fusion using fuzzy and (crisp) continuous data: As in (1.1), we may use the standard Hartigan fusion algorithm. The Data is the same as in (1.1) with the added complexity of having to include and compare the continuous data types with the 'fuzzy data type'. The method to compare, for example, a continuous variable with a fuzzy variable could be to consider in the distance algorithm that the crisp data is a special case of fuzzy data with values {1,0}, or the max-min of the fuzzy range defined. The objective would be to calculate a distance/similarity matrix which would be given as input to the standard fusion algorithm. Later in this section, we consider where does the fuzzy data come from, and how it is defined. [Bilgiç97] is one of the authors who considers these type of problems. When can an attribute be

processed as ‘fuzzy’? An attribute can be processed as fuzzy when it can be represented by a membership function, which has been previously established and calculated.

(1.4) Crisp fusion using fuzzy and (crisp) mixed data types: again we can use the standard Hartigan fusion algorithm. The data is the same as in (1.2) with the added complexity of having to include and compare all the data types mentioned in (1.2) with the 'fuzzy data type'. The method to compare, for example, a categorical variable with a fuzzy variable could be that of [Girard96]. The objective would be to calculate a distance/similarity matrix which would be given as input to the standard fusion algorithm.

(1.5) Fuzzy fusion with (crisp) integer and continuous data: in this case we have the same data as in (1.1), but this time we have the fuzzy version of the fusion algorithm. This algorithm needs as input, a matrix of Fuzzy Covariances. There are various references in the literature of how to calculate a fuzzy covariance matrix [Bezdek81][Gustafson79][Watada94]. In Section 3.1.4 we enter into more detail with respect to fuzzy covariance and fuzzy covariance matrices. Once the matrix is calculated, it can be given as input to the fuzzy fusion algorithm, which is a modified version of the standard fusion algorithm, which can treat fuzzy input covariances. The selection of pairs from the correlation matrix for fusion introduces the aspect of fuzzy distance and its calculation. The method (of ranges) detailed by [Nakamori97] or the distances of [Dubois80] could be incorporated as solutions. (see later sections for details).

(1.6) Fuzzy fusion using (crisp) mixed data types: we have the same fuzzy fusion algorithm as in (1.5), but with crisp mixed data as in (1.2). We have to calculate the pair similarity measures using the different techniques depending on the type of the variable as in (1.2).

(1.7) Fuzzy fusion using fuzzy and (crisp) continuous data types: same problems apply as in (1.5) with the added requirement (as in 1.3) of comparing fuzzy types with continuous variables. There may be an intermediate step of elicitation of membership functions as in [Babuska96] and [Fujimoto95].

(1.8) Fuzzy fusion using fuzzy and (crisp) mixed data types: same problems apply as in (1.6) with the added requirement (as in 1.3) of comparing fuzzy types with categorical, ordinal, integer and continuous variables.

Problems being solved

(i) Comparison between fuzzy and (crisp) mixed data: we could use a unified scheme where fuzzy data is described by a grade of membership value in the range [0,1] , while crisp data is defined by a grade of membership value restricted to the values 0 or 1 {0,1}. We could then proceed to compare variables in terms of their respective grade of membership values.

(ii) Allows inclusion of information in fuzzy form in crisp fusion algorithm: deciding which to join next is a fuzzy decision, but which needs a crisp result. One approach would be to build and update fuzzy rules which supply the mechanism to make these decisions.

(iii) Comparison between different types of crisp data in the same matrix : (binary, nominal, ordinal, interval, ratio). [Kaufman90], p32-37 , describes an approach in which all data types are combined into one dissimilarity matrix.

Recent work in data fusion and representation and of possible approaches for the present work: fuzzy covariances, chi-squared measure, membership functions (mainly for linguistic variables) and distance measures.

Until recently, data fusion has not been a topic widely covered by authors as a specific theme. Since 2000 there exists an 'International Fusion Journal', and an 'International Fusion Conference' since 1998. [Cross95] deals with the problem of representation and establishes a definition for a fuzzy object. It is argued that one could consider that an object is a fuzzy object if it has a membership degree in the interval [0,1] for a class. A fuzzy linguistic object is an object that has at least one attribute whose value is a fuzzy set. A fuzzy object is any object that contains a possibility distribution for an attribute value. [Wangc96] creates (induction) membership functions and fuzzy rules for input values, whereas [López97] defines a language for the definition of fuzzy systems.

With respect to fuzzy covariance and fuzzy correlation matrices, [Nakamori97] considers factor analysis for fuzzy data. The correlation coefficients between measures are directly defined as interval fuzzy numbers. The fuzzy correlation matrix holds the relative fuzziness of correlation coefficients. [Watada94] considers fuzzy principal component analysis for fuzzy data, the variables being described as follows:

$$D_i = [x_{i1}, x_{i2}, ..., x_{ip}] \qquad (3.9)$$
$$x_{ij} = [x^l_{ij}, x^c_{ij}, x^u_{ij}]$$

where x_{ij} is a fuzzy number with x^l_{ij}, x^c_{ij} and x^u_{ij} which are the lower boundary, upper boundary and centre of attribute x_{ij} , respectively. Using fuzzy data , fuzzy variance- covariance matrix $V = [V^l, V^c, V^u]$ can be calculated.

[Babuska96] considers fuzzy modelling and similarity analysis applied to ecological data, in which *triangular* membership functions and matrix are defined as an initial approximation. Then fuzzy c-Means is used to establish the membership functions which best fit the data. Setnes refers to [Gustafson79], and the Takagi-Sugeno Model [Takagi85].

3.2.2 Implementation – Nettleton's version of Hartigan's 'joining' algorithm

The following describes the implementation by Nettleton of the fusion algorithm, and the artificial test data used to verify its functionality. In Sections 4.1 and 4.2 of the thesis, this algorithm is executed with real application (ICU) data and other test data, and is contrasted with diverse variable grouping and factor reduction techniques. The crisp (standard) version of the algorithm is first described, followed by the consideration of enabling it to process different data types and incorporate fuzzy techniques into the distance measure and mechanism.

The code of Hartigan's 'joining algorithm' is given at the end of the book [Hartigan75], in Fortran. This code was rewritten in 'C' and tested in its original form. As next step it was modified to enable joint treatment of discrete and continuous attributes. Also, the final post-processing phase could be implemented as the generation of a two dimensional plot of the two final resulting attributes which would represent the fusion of all the other attributes.

'Crisp' fusion algorithm – input data and processed output data: The following is the output of the fusion algorithm for 'crisp' attributes. The algorithm has been coded in 'C' from the description given in [Hartigan75].

Table 32. C matrix of covariance coefficients used by Hartigan joining algorithm to fuse variables

Original variables								
HL	1.000	0.402	0.395	0.305	0.301	0.339	0.340	0.303
HB	0.402	1.000	0.618	0.135	0.150	0.206	0.183	0.143
FB	0.395	0.618	1.000	0.289	0.321	0.363	0.345	0.305
FM	0.305	0.135	0.289	1.000	0.846	0.797	0.800	0.000
FR	0.301	0.150	0.321	0.846	1.000	0.759	0.661	0.000
FT	0.339	0.206	0.363	0.797	0.759	1.000	0.736	0.778
HT	0.340	0.183	0.345	0.800	0.661	0.736	1.000	0.731
Fused variables								
FMFR	0.303	0.143	0.305	0.846				
FTFMFR	0.315	0.164	0.324	0.778				
HTFTFMFR	0.321	0.169	0.329	0.732				
HBFB	0.399	0.618						
HLHBFB	0.399							

Observations with respect to the algorithm:[Hartigan75]

1) The last fusion is not carried out (to leave just one factor), as indicated in Hartigan's description.

2) The covariances of the other variables are not reduced or modified during the process.

3) The final state of the loading matrix 'B' is different to the values given by Hartigan.

> **Differences of the state of the B and C matrices with respect to Hartigan's joining algorithm [Hartigan75]:** The algorithm has been implemented in 'C' exactly the same as the specification at the end of Chapter 17 in Hartigan's book [Hartigan75]. Nevertheless, it did not initially choose the same variables to be joined as the example given in the book, with the same data. By changing the inner loop of step 2 from '1 to N' to '1 to K', it did give the same results. In the book, the author changes the order of the variables in the matrix in two different pages, and it could be that the form of representing the results is descriptive and is not equal to the state of the matrices. One improvement would be to place the final output in a 'results' matrix to show them in a clearer and simpler form.

4) The algorithm, in each iteration, chooses the correct pair of variables to fuse (join), and the covariances of the new factors (variables) correspond to those given by Hartigan. It is understood that the correct pair of variables to join at each step is the pair whose variables have the highest mutual covariances.

> **Observation:** this is the key aspect which is required to function correctly for the Hartigan joining algorithm. We have considered that the observations given in points 1 to 3 are due to the 'intuitive' description used in the textbook, or due to dependences of implementation.

Form of calculating the loading matrix B, and use of the matrix and coefficients for a posterior factorial analysis of the attributes, and recalculation of the data from the original variables Ref. [Hartigan75], pp324

We assume that:

> A is a data matrix of 2 dimensions M by N.
>
> F is a matrix of factors of 2 dimensions M by K.
>
> B is a loading matrix of 2 dimensions N by K.

The arrays F and B are factors of row and column, respectively, of matrix A.

Then:

the Kth Cluster is a sub matrix S of A.

F(I,K) = 1 if row I of A is in S; otherwise, F(I,K) = 0.

B(J,K) = S(J,K) if variable (column) J is in S; otherwise B(J,K) = 0.

$A = FB = \Sigma\{1 \leq K \leq L\}\ C(K)$, where C(K) = 0 (the Kth cluster).

Once the fusion and the B matrix have been calculated, we can quantify the set of factors, being the difference between the means of the variables in the pairs of clusters joined during the fusion [Hartigan75], pp322.

$F(8) = [V(4) - V(5)]^{1/2}$

$F(9) = [1/2V(4) + 1/2V(5) - V(6)]^{2/3}$

$F(10) = [1/3V(4) + 1/3V(5) + 1/3V(6) - V(7)]^{3/4}$

$F(11) = [V(2) - V(3)]^{1/2}$

$F(12) = [1/2V(2) + 1/2V(3) - V(1)]^{2/3}$

$F(13) = [1/3(V(1) + V(2) + V(3)) - 1/4(V(4) + V(5) + V(6) + V(7))]^{12/7}$

$F(14) = [V(1) + V(2) + V(3) + V(4) + V(5) + V(6) + V(7)]^{1/7}$

The constants 1/2, etc, ... guarantee that the sum of the squares of the coefficients is equal to unity. Now we can define the original variables in terms of the new factors which have been generated:

$V(1) = -\sqrt{2}/3\ F(12) - 1/3\sqrt{12/7}\ F(13) + \sqrt{1/7}\ F(14)$

$V(2) = \sqrt{1/2}\ F(11) - 1/2\sqrt{2/3}\ F(12) + 1/3\sqrt{12/7}\ F(13) + \sqrt{1/7}\ F(14)$

$V(3) = -\sqrt{1/2}\ F(11) - 1/2\sqrt{2/3}\ F(12) + 1/3\sqrt{12/7}\ F(13) + \sqrt{1/7}\ F(14)$

$V(4) = \sqrt{1/2}\ F(8) + 1/2\sqrt{2/3}\ F(9) + 1/3\sqrt{3/4}\ F(10) - 1/4\sqrt{12/7}\ F(13) + \sqrt{1/7}\ F(14)$

$V(5) = -\sqrt{1/2}\ F(8) + 1/2\sqrt{2/3}\ F(9) + 1/3\sqrt{3/4}\ F(10) - 1/4\sqrt{12/7}\ F(13) + \sqrt{1/7}\ F(14)$

$V(6) = -\sqrt{2/3}\ F(9) + 1/3\sqrt{3/4}\ F(10) - 1/4\sqrt{12/7}\ F(13) + \sqrt{1/7}\ F(14)$

$V(7) = -\sqrt{3/4}\ F(10) - 1/4\sqrt{12/7}\ F(13) + \sqrt{1/7}\ F(14)$

It is assumed that V(1) to V(7) are the original variables. The constants depend on the number of variables and on the proportions which each of the original variable contributes to the new factors. The original variables can be found in the vector JT.

F(8) to F(14) are the new variables (factors). The constants are derived from the formulas for V(1) to V(7), and the values of each factor correspond to their relative position in the 'loading matrix B. For example, in the formula:

$V(1) = -\sqrt{2}/3\ F(12) - 1/3\sqrt{12/7}\ F(13) + \sqrt{1/7}\ F(14)$

F(12) would assume the value of the matrix element B[1,12-7], which has the value 0.598. We must refer to the B matrix, once reduced, following Hartigan's comments [Hartigan75], pp320.

Table 33. Values in the B matrix after executing Nettleton's version of the fusion algorithm

HL	0.475	0.000	0.000	0.000	0.000	0.000	0.000
HB	0.000	0.618	0.000	0.000	0.000	0.000	0.469
FB	0.000	0.000	0.000	0.000	0.000	0.000	0.469
FM	0.000	0.000	0.214	0.214	0.214	1.000	0.000
FR	0.000	0.000	0.214	0.214	0.214	1.000	0.000
FT	0.000	0.000	0.000	1.000	1.000	1.000	0.000
HT	0.000	0.000	0.000	0.000	0.000	1.000	0.000

And the formulas of the original variables will be:

V(1) = 0.475 F(8)
V(2) = 0.618 F(9) + 0.469 F(14)
V(3) = 0.469 F(14)
V(4) = 0.214 F(10) + 0.214 F(11) + 0.214 F(12) + 1.000 F(13)
V(5) = 0.214 F(10) + 0.214 F(11) + 0.214 F(12) + 1.000 F(13)
V(6) = 1.000 F(11) + 1.000 F(12) + 1.000 F(13)
V(7) = 1.000 F(13)

Use of the B (loading) matrix: we assume that the elements of the B matrix, once the algorithm has terminated, are the coefficients of the equations of each variable, which are multiplied by the factors found in the C matrix. Thus, B_0 will be the identity matrix, in which all elements are assigned to zero except those in the descending diagonal which are assigned to 1. The value of each function is defined by a unitary covariance matrix. If we create a new variable V_{new} from three existing variables, with factors F_1, F_2 and F_3, then

$$V_{new} = B(1,1)\ F_1 + B(1,2)\ F_2 + B(1,3)\ F_3.$$

Qualitative values: the fusion algorithm does not have to consider qualitative values, given that we assume that the covariances of these values are already calculated and the fusion algorithm receives as input a covariance matrix, the same as in the case of the quantitative values.

For example, consider the qualitative attributes, *colour and texture*. As a simple similarity measure, each case is exhaustively compared with every other case, and a frequency table is built for the number of occurrences of category-pairs. For example, the category-pair *colour=blue* and *texture=smooth* could occur 35 times in a total of 100 cases, and the total of unique category-pairs could be 45 out of 100. The frequency table would therefore be used as a basis for calculating a similarity value which would measure the incidence of coinciding pairs for *colour* and *texture*.

Example similarity measure

$$d(Oi, Oj) = \sum_{k=1}^{n} dif(Oik, Cjk) \qquad (3.10)$$

In the case of qualitative (symbolic) values, the expression dif (Oik, Cjk) will be 1 if the values are equal and 0 if they are distinct. If the values are quantitative (numeric) the expression will evaluate to the absolute value of the difference between the two values.

Types of distance: three examples of distances which depend on the type of the attributes being compared, are: (i) the Minkowski metric; (ii) the Mahalonobis distance; and (iii) the χ^2 (chi-squared) distance. The similarity measure (i) is used by the 'Linneo+' system [Béjar94], and the distance of χ^2 is used by 'Klass' [Gibert94].

Using fuzzy techniques in the fusion algorithm

The fusion algorithm is susceptible to the introduction of fuzzy concepts, for example, in the aspect of distance calculations. The distances could be converted from Euclidean to fuzzy. The selection of the pairs of attributes to join and in which order, is a central aspect of the joining process.

Motivation: We assert that there are sets of attributes for which the best representation is in the fuzzy form. For example, in the medical domain we have the attribute 'grade of recovery possible'. This implies that a fuzzy treatment would give better results for prognostic and diagnostic models, once the original attributes have been fused (joined) in 'super-attributes'.

Distances: In the case of the distances, we must emphasise that we consider distance not between cases, but between attributes, and the clusters are clusters of attributes, formed by the fusion process.

Crisp version: we may initialise the algorithm as an averaging fusion algorithm, by using Euclidean distances when the variances are all equal to 1 (unity). If $\rho(I,J)$ is the correlation between variables I and J, $D(I,J) = [1-\rho(I,J)] / 2M$ is the square of the Euclidean distance between the standardised variables. The distance between clusters of variables is defined as the mean distance over pairs of variables, one for each cluster. Then we obtain exactly the same sequence of fusions over the distances, as in the previous algorithm.

Fuzzy version: one approach would be to use the fuzzy version of the Fuzzy c-Means 'Least Square Functionals' algorithm as a starting point. The definition of this algorithm has been given previously in Section 2.2.7 of the thesis.

Interpretation of attributes: the fusion algorithm looks for attributes with the highest covariance, and it generates a new attribute from the mean of the covariance of the original pair. When the joining algorithm makes the decision to join two attributes, it would join all the attributes with their respective grades of membership to each cluster. We could define a threshold below which we could not include an attribute in the process. This would imply that a given attribute could be referenced in more than one cluster, with different grades of membership. We could then select the cluster with the highest sum of grades of membership (of all the attributes in the cluster), or that with the highest sum of grades of membership for one or more attributes of greatest interest. The first sum could be interpreted as a 'coherence factor' for the cluster.

3.2.3 Aggregation using the WOWA operator

We now detail the work which has been based around the WOWA aggregation operator, and the family of operators to which it belongs, such as OWA, WM. As mentioned previously, we have chosen an algorithm which is adequate for small datasets, and which enables us to include reliability and relevance information as part of the data processing, in the form of weighting vectors. We have considered enhancements to the algorithm such as the learning of a weight vector using machine learning techniques, processing of missing data, and allowing a weight vector for each variable instead of just one static weight vector. The last enhancement may be important when we have very different variable types and data distributions within the same data set.

Discussion and comparison of aggregation methods

[Torra98c] gives a summary and comparison of integration methods for numerical information, from more specific operators such as WM and OWA, to generalised operators such as the Fuzzy integral. For non-numerical data, such as categorical values, we assume an appropriate representation method in a numerical form. This also applies to fuzzy data values, in which membership grades would be represented appropriately in order to process them as numerical data. The first group of operators are the arithmetic mean, the weighted mean and the OWA operator. The WM and OWA are a lineal combination of the values according to a vector of weights. In contrast to these operators, an operator which allows the consideration of fuzzy measures is the Choquet integral.[Torra98c] states that the WM, OWA and WOWA aggregation operators are Choquet integral for particular fuzzy measures. Another group of aggregation operators not related to the previous, are the weighted min and weighted max. Their difference lies in the weights: at least one of the weights must be one, and the addition of the weights can be greater than one. The Sugeno integral generalises the weighted Min and weighted Max in the same way that the Choquet integral generalises the OWA and the WM. Finally, the fuzzy integral can be considered a generalisation of the Choquet integral and the Sugeno integral. The fuzzy integral is defined over a tuple known as a t-conorm system for integration, and an operation $-_{\Delta}$, based on one of the elements of this tuple. See Section 2.3.1 for definitions of the Choquet, Sugeno and fuzzy integrals.

In the work on the Apnea applications, and the data capture and processing using the patient questionnaire, we decided to use WOWA as the mechanism which would create a model from the data and produce a diagnosis for each patient. We chose WOWA for several reasons, the first being that it provided two weighting vectors which we could interpret as 'relevance' and reliability' for the variables and data values, respectively. It also has a robust interpolation mechanism, a Bernstein Polynomial, which maintains the original characteristics of the curves which the weights represent, that is, points of inflexion, and concave or convex segments. WOWA also provides a simpler solution than Choquet or Sugeno integrals, given that WOWA requires $2 \times n$ parameters, whereas Choquet requires 2^n-2 parameters. WOWA requires 2 vectors each of n parameters, where $n \equiv$ the number of elements to aggregated. The Choquet integral requires 2^n-2 parameters, given that a fuzzy measure is a function of $P(X) \rightarrow [0,1]$. Thus, for large values of n, the measure requires many parameters and the interpretation is complex. Given that the number of examples in the Apnea data application, as described in Sections 4.3 and 4.4, is relatively small (less than 200 cases), WOWA is favourable for learning the parameters (or weights) given the smaller number to be learnt. For example, 10 variables gives $2^{10}=1024$ parameters, whereas 20 variables results in $2^{20} = 1048576$ parameters ! Thus for $<$ 200 cases, as with the Apnea application, it is more reasonable to learn $2\times20=40$ parameters rather than 1048574 !

The WOWA aggregation operator is therefore apt for processing small data sets (less than 200 cases) with a wide dimensionality of problem (between 10 and 20 variables). It also has an advantage with respect to typical 'data mining' algorithms such as neural networks and rule induction, which tend to require greater volumes of data (cases) in order to be able to produce a reasonable model of the data and therefore give a decent predictive or classificative precision. We can also have control over the process by being able to use different interpolation functions, and by varying the weights in the two weighting vectors. The small number of cases and wide dimensionality of problem also makes the WOWA more appropriate for this application in comparison to the Choquet Integral.

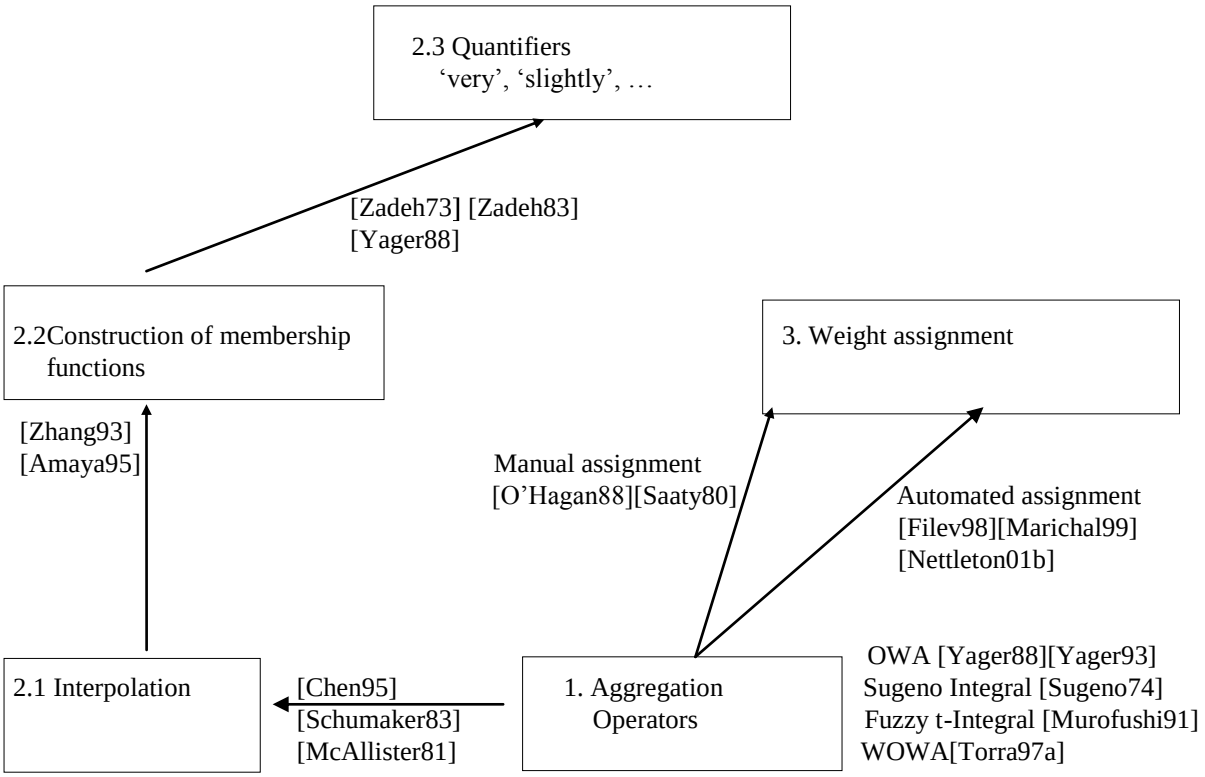


Figure 52. Scheme of different aspects of aggregation and corresponding authors

In Figure 52 we see the different aspects which go into an aggregation operator, including the different aggregation operators themselves. We have already detailed these different methods in Section 2.3; here we summarise that in the thesis we have chosen to use the WOWA operator [Torra97a], because its two weighting vectors are ideal for our application of Apnea screening, in order to represent the 'relevance' and 'reliability' meta-data. We have also tested results against the Weighted Mean and OWA [Yager88] aggregation operators. The interpolation method as based on that of [Chen95], given that is the method used by WOWA to construct the bias curves from the weights. For the construction of membership functions and the definition of quantifiers, we have used Zadeh's s-function [Zadeh73]. For the manual and automated assignment of weights we have been influenced by [O'Hagan88] and [Marichal99]. In the case of automated assignment, an original method using genetic algorithms for learning has been developed, as detailed in [Nettleton01b].

Two Approaches for Learning the Weights

Our approach to learning weights is based on using a set of examples. We assume that each consists on a set of values to aggregate and the corresponding outcome. Let the set of examples be the ones in Table 34. This is, a set of M examples, each one with N input values (the dimension of the aggregation operator is, therefore, N). In this table, a^i_j corresponds to the value supplied by the j-th source in the i-th example; and m^i corresponds to the outcome of the i-th example.

Table 34. Data examples

a^1_1	a^1_2	a^1_3	...	a^1_N	|	m^1
a^2_1	a^2_2	a^2_3	...	a^2_N	|	m^2
...						
a^M_1	a^M_2	a^M_3	...	a^M_N	|	m^M

The goal of our approach is to learn the weights that when used in conjunction with the aggregation operator returns m^j (or a similar value if no exact solution exists) when the input vector is $(a^i_1\ a^i_2\ a^i_3\ ...\ a^i_N)$. We compare two approaches, one based on the Active Set Methods and the other based on Genetic Algorithms. We review them below. In both cases, what we need is a way to determine what a good solution is. This is equivalent to define a measure of the goodness of a solution. In our case, this will be done through the accumulation, for each example $(a^i_1\ a^i_2\ a^i_3\ ...\ a^i_N \mid m^j)$, of the distance between the ideal outcome m^j and the outcome given by the aggregation operator. The distance for each example is computed through the squared difference. Therefore, in our case, the more suitable weighting vectors are the ones that minimises this expression:

$$D(\mathbf{p}) = \sum_{j=1}^{M} \left(\varsigma\left(a^j_1, ..., a^j_N\right) - m^j\right)^2 \tag{3.11}$$

Together with this objective function, a set of restrictions have to be considered when the aggregation function ς is either the weighted mean, the OWA operator or the WOWA operator. This restriction is that the parameters (the weights) have to define a weighting vector. This is, weights have to add to one and be positive. Therefore the general problem can be formalised in the following way:

$$\text{Minimize} \sum_{j=1}^{M} \left(\varsigma\left(a^j_1, ..., a^j_N\right) - m^j\right)^2$$

such that

$$\sum_{i=1}^{N} p_i = 1 \qquad\qquad \sum_{i=1}^{N} w_i = 1$$

$$p_i \geq 0 \quad \text{for all } i \qquad\qquad w_i \geq 0 \quad \text{for all } i$$

Where in the case of ς being the weighted mean, only restrictions over p apply; in the case of being the OWA operator only those over w and in the case of being the WOWA both apply.

(i) Active Set Method based approach

The problem formulated in this way is a typical optimisation problem and there exist different techniques to tackle with it. In the particular case of using a weighted mean or a OWA operator, the problem reduces to a quadratic program and there exist several algorithms that solve the problem with accuracy (see, for example, [Luenberger73], [Gill81]). For example, the ones based on active set methods.

Active set methods rely on the simplicity of computing the solution of quadratic problems with linear equality constraints. Based on this, algorithms iterate so that in each step inequality constraints are split into two groups: one with the ones that will be treated as active and considered as equality constraints; and another with the ones that are ignored. Then, the algorithm moves to an improved point moving on the surface defined by the set of active constraints. At this point constraints can be added to and removed from the active set. This process is repeated until the minimum is reached. A detailed analysis of such method applied to the learning of weighting vectors for the weighted mean and the OWA operator is presented in [Torra99b].

The application of these methods to the example introduced by [Filev98] for the OWA operator resulted to a good solution. In Table 35 the data matrix corresponding to this example is given. Using the active set method approach the resulting weighting vector for the OWA operator was:

$$w = (0.1031, 0.0, 0.2293, 0.6676)$$

while the solution in [Filev98] after 150 iterations (based on the use of the gradient technique) was:

$$w = (0.08, 0.11, 0.14, 0.67)$$

after 150 iterations. The error using active set methods was: 0.001256, while with the gradient technique due to the slow convergence was: 0.002156.

Table 35. Data matrix H and solution vector d (taken from [Filev98])

```
0.4 0.1 0.3 0.8 | 0.24
0.1 0.7 0.4 0.1 | 0.16
1.0 0.0 0.3 0.5 | 0.15
0.2 0.2 0.1 0.4 | 0.17
0.6 0.3 0.2 0.1 | 0.18
```

(ii) Genetic Algorithm based approach

An evolutive procedure is a probabilistic algorithm which maintains a population of individuals $\mathbf{P(t) = \{x_1^t,....x_n^t\}}$ for iteration **t** . Each individual represents a potential solution to the problem being considered, and is normally implemented as a data structure **S**. Each solution $\mathbf{x_i^t}$ is evaluated to give a measure of its 'aptitude'. Then, a new population is formed (iteration **t + 1**) by selection of the most apt individuals (selection step). Some members of the new population undergo transformations (modification step) using 'genetic' operators, which form new solutions. There are unary transformations $\mathbf{m_i}$ (mutation type) which create new individuals by a small change in just one individual ($\mathbf{m_i : S \rightarrow S}$), and transformations of a higher order $\mathbf{c_j}$ (crossover type), which create new individuals by the combination of parts of various (two or more) individuals ($\mathbf{c_j : S \times, ... \times, S \rightarrow S}$). After a given number of generations the program converges – with the objective that the best individual represents a solution close to the optimum.

Within the extensive literature in this field, we can highlight the following: two key authors for parameter optimisation problems are [Rechenberg73], [Schwefel81]; Fogel's evolutionary programming [Fogel66] is a technique for searching through a space of small finite-state machines; Glover's scatter search techniques [Glover97] maintain a population of reference points and generate offspring by weighted linear combinations. A matrix representation for the chromosome was introduced by Vignaux [Vignaux91], and [Koza90], [Michalewicz92] are examples of specific genetic operators to accommodate the problem to be solved. The incorporation of problem specific knowledge has been tackled by authors such as [Antonisse87], [Forrest85], [Fox91].

We can use genetic algorithm techniques to learn the weighting factors for aggregation operators such as WM, OWA and WOWA from historical data. In the case of WOWA, we could also look for interpolation functions which give best results, with an adequate parametric representation for the function in the chromosome. A genetic algorithm has as input a set of input cases and their respective outcomes (examples), a set of modifiable values (in this case the weighting factors), a set of constraints (in this case the sum of the weighting factors must be equal to 1), and an objective function, which we have defined as the minimum difference between the predicted outcome $m^{j'}$ and the real outcome m^j. We wish to find the weighting factors which best approximate the input and output data, while minimising the objective function.

Figure 53. The Basic Structure of the Evaluation routine

```
Procedure evaluate
begin
        for all (genetic) individuals do
        begin
          read weights ω and ρ from current individual i's chromosome
          total_distance ← 0
          for all data cases do
        begin
            read (data input row j)
            real_output ← read (real_output for this case)
            wowa_output ← wowa (weights, data)
            local_distance ← real_output – wowa_output
            total_distance ← total_distance + (local_distance)^2
          end
          individual(i).aptitud ← total_distance
        end
end
```

The routine in Figure 53 simply goes through all the individuals in the current population and assigns a 'fitness' score to each. The fitness score for each individual is calculated by executing the function which calculates the output (in this case the WOWA aggregator) for each of the data cases (1..j) and with the w and p weight vectors contained in the chromosome of individual **i**.

The chromosome consists of a single vector data structure which holds the w weights and the p weights. Another approach would have been to separate the w and p weights into two separate vectors, and maintain them as separate populations. This gives the possibility of converging more rapidly because the different weight types are not mixed by the crossover and mutation operations.

```
Struct chromosome
{
   int gene_vector[1..num_weights];
}
```

In the case of WOWA, gene_vector holds the w and p weights and num_weights is equal to the number of variables × 2. Likewise, in the case of OWA, gene_vector holds the w weights and num_weights is equal to the number of variables. Finally, in the case of WM, gene_vector holds the p weights and num_weights is also equal to the number of variables.

If we normalise the ω and ρ values, we guarantee that consistent WOWA's are generated:

$$\omega'_i = \omega_i / \sum \omega_i$$

$$\rho'_i = \rho_i / \sum \rho_i$$

In Table 36 the data matrix used to learn the weights for the WOWA operator using the GA method is given. Using the GA method approach the resulting weighting vectors for the WOWA operator was:

w = (0.47, 0.05, 0.11, 0.37)
p=(0.15, 0.19, 0.35, 0.31)

This was obtained using crossover in 1 point, with random and uniform mutation. The population was 150, the possible gene values ranging between 1 and 10; the crossover rate at 0.85 and the mutation rate at 0.01. The best aptitude was 0.000, run over 100 generations and best individual found after 3 generations.

If we increase the range of possible gene values to be between 1 and 100, and increase the population to 300, the following result was found, also with best aptitude 0.000, and taking 9 generations to reach it.

w = (0.07, 0.42, 0.07, 0.44)
p=(0.01, 0.51, 0.31, 0.17)

while the solution for the w and p vectors given in [Torra97a] was:

w=(0.13, 0.37, 0.37, 0.13)
p=(0.25, 0.25, 0.25, 0.25)

Table 36. Data matrix H and solution vector d (taken from [Torra97a])

```
0.7 0.6 0.4 0.3 | 0.50
0.9 0.7 0.5 0.3 | 0.60
```

The data matrix used to learn the weights for the OWA operator using the GA method is that of Table 35. Using the GA method approach the resulting weighting vectors for the OWA operator was:

w = (0.07, 0.07, 0.33, 0.53)

This was obtained using crossover in 1 point, with random and uniform mutation. The population was 150, the possible gene values ranging between 1 and 10; the crossover rate at 0.85 and the mutation rate at 0.01. The best aptitude was 0.067, run over 100 generations and best individual found after 9 generations.

If we increase the range of possible gene values to be between 1 and 100, and increase the population to 350, the following result was found, with best aptitude 0.061, and convergence in 27 generations:

$$w = (0.09, 0.01, 0.31, 0.59)$$

In Table 37 the data matrix used to learn the weights for the WM operator using the GA method is given. Using the GA method approach the resulting weighting vectors for the WM operator was:

$$p = (0.08, 0.21, 0.29, 0.38, 0.04)$$

This was obtained using crossover in 1 point, with random and uniform mutation. The population was 150, the possible gene values ranging between 1 and 10; the crossover rate at 0.85 and the mutation rate at 0.01. The best aptitude was 0.031, run over 100 generations and best individual found after 7 generations.

If we increase the range of possible gene values to be between 1 and 100, and increase the population to 300, the following result was found, with best aptitude 0.004, and convergence in 18 generations:

$$p = (0.1, 0.2, 0.3, 0.4, 0.0)$$

The values of the resulting p vector are identical to those given in [Torra99b].

Table 37. Data matrix H and solution vector d (taken from [Torra99b])

0.3	0.4	0.5	0.1	0.2	0.30
0.2	0.1	0.4	0.1	0.5	0.20
0.2	0.5	0.8	0.0	0.1	0.36
1.0	0.5	0.3	0.6	0.7	0.53
0.2	0.1	0.1	0.1	0.1	0.11
0.7	0.7	0.7	0.7	0.7	0.70
0.4	0.8	0.2	0.8	0.6	0.58
0.3	0.2	0.1	0.4	0.3	0.26
0.6	0.8	0.7	0.2	0.5	0.51
0.1	0.5	0.2	0.6	0.4	0.41

The weighted mean was the only operator which showed a significant improvement in the best solution found by varying the parameters to the GA. In the case of WOWA and OWA, diverse combinations were tried for mutation rate and crossover rate, but without improvement of the best solution found. Notwithstanding, by changing the population size and allowable values for the gene, we were able to find different weight values for the best solutions with the same aptitude.

We also tried a mutation function with Gaussian distribution instead of random and uniform, and a crossover function with 2 point crossover instead of 1. In the case of the WM, this reached the same precision as the previous solution (best aptitude 0.004) with the same weights. In the case of OWA, the solutions found were slightly worse (best aptitude 0.068 compared to 0.061). The 2 point crossover can be used to preserve order of subgroups within the chromosome, when this is significant to the solution to the problem, as in the TSP (Travelling Salesman Problem)[Lawler85], [Michalewicz96], or in this case, where the weights are ordered with respect to the data values. We think the lack of improvement for OWA in this case was due to the insufficient length of the chromosome. Although the type of problem is appropriate to show an improvement with this method, the total number of genes was insufficient to create significant subchains of genes within the chromosomes.

Reformulation for examples with a different number of variables

In order to use an aggregation operator to process data with missing values, we have several options open to us. The first option may be to fill in the missing values with a new value. This could be a flag which indicates that the value is missing, or it could be the mean of the rest of the values for that variable in the case of numeric values, or the mode in the case of categorical values. This method works better when there are a large number of cases and thus the mean and mode values are good approximations. In our case, we wish to work with a small number of cases and we cannot guarantee that the mean and mode will be accurate. Thus we choose to eliminate the missing values (not the whole case). If the cases have **N** variables, and for case **j**, two variables are missing, then we eliminate those variables for case **j** and we pass the **p** and **a** vectors with **N - 2** variables to the WOWA operator. The **w** vector remains unchanged, and we pass **N** of its weights to WOWA. This is because the **p** and **a** vectors must have the same length: for each **a** value there must exist a corresponding **p** weight. In contrast, the **w** vector is not directly related to the **p** and the **a** vectors, as it is used to construct the quantifier function. The different lengths of the **a** and **p** vectors with respect to the **w** vector make it necessary to maintain two length counters and pass these to WOWA to be used in the appropriate points of the function code. Of course, the aggregation operator performs the aggregation based only on the **N - 2** variables received

as its input. Thus, a reformulation of the aggregation operator is needed so that it can be applied to a number of variables less than **N**. This approach has its cost/benefit: on the one hand, we have the possible benefit due to a reduction in noise due to erroneous substitute values. On the other hand we have the possible cost due to information loss due to the absence to the omitted variables' value. The final decision on whether to choose one option or the other depends on the application and the nature of the data being processed, as well as the number of cases.

In the case of the WM, the reformulation is as follows:

$$D(\mathbf{p}) = \sum_{j=1}^{M} \left(\frac{\sum_{i \in I_j} a_i^j p_i}{\sum_{i \in I_j} p_i} - m^j \right)^2 \tag{3.12}$$

where I_j is a subset of {1,..., N} with the index variables of example j. In the reformulation, we divide by Σp_i in order to re-normalise the weights and assure that their sum is equal to 1. The approach for the OWA is similar and we have not detailed it here. In the case of the WOWA, the reformulation is as follows:

$$D(p) = \sum_{j=1}^{M} \left(\sum_{i \in I_j} \left(W^* \left(\frac{\sum_{j \le i : j \in I_j} p_{s(j)}}{\sum_{i \in I_j} p_i} \right) - W^* \left(\frac{\sum_{j < i : j \in I_j} p_{s(j)}}{\sum_{i \in I_j} p_i} \right) \right) a_i^j - m^j \right)^2 \tag{3.13}$$

where I_j is a subset of {1,..., N} with the index variables of example j. In the reformulation, we divide by Σp_i in the same manner as for WM and OWA, and the ordering $(\Sigma_{j \le i} p_{\sigma(j)})$ is now restricted to the subset $i \in Ij$.

Table 38. Data matrix H and solution vector d , with missing values indicated by 'M'

```
0.7 0.6 0.4  M  | 0.50
0.9 0.7 0.5 0.3 | 0.60
```

In Table 38 the data matrix with one missing value is given. A new version of WOWA, modified to process missing values as described above, was executed with this data set and with the same parameters used in the previous tests. This resulted in a best aptitude of 0.001, with convergence in 6 generations. The weight vectors were as follows:

p = (0.17, 0.39, 0.34, 0.11)
w = (0.02, 0.41, 0.29, 0.29)

If we increase the number of missing values to two, that is, in Table 38 we also make the value corresponding to row 2, column 4 equal to 'M', and rerun the WOWA modified to process missing values, the best aptitude is also 0.001, but convergence is slower in 26 generations. If we increase the number of missing values to four, that is, in Table 38 we also make row 2, column 4 and row 1 column 1 equal to 'M', and rerun the modified WOWA, the best aptitude is 0.009, achieved at the maximum (cut-off) of 100 generations. Finally, if we increase the number of missing values to six, that is, in Table 38 we also make row 1, column 2 and row 2, column 3 equal to 'M', and rerun the modified WOWA, we see a significant degradation in the performance: best aptitude is 0.300 achieved in 100 generations; the output values are 0.50 and 0.90 for rows 1 and 2, respectively; the weight vectors in this last case settle to the following:

p = (0.42, 0.04, 0.12, 0.42)
w = (0.05, 0.36, 0.41, 0.19)

In conclusion, we see that with the data set of Table 38, there is a robust handling of missing values, in which there is only a significant degradation in the overall precision when there exists a high percentage of missing values. We have to take into account that the data in Table 38 is a small artificial data set and it is relatively easy for the aggregation to find good alternative solutions for the different subsets of the values.

Advantages and Inconveniences of the two approaches

Both active set methods and genetic algorithms have been applied to learn the weights for the WM, OWA and WOWA aggregation operators. The GA learning method has been used with medium size real problems, as detailed in Sections 4.3 and 4.4 of the thesis, and was selected in preference to ASM, due to the advantages which GA has for the specific aggregation operator used (WOWA) and the application (medium size data sets for Apnea diagnosis). Notwithstanding, the two approaches are complementary, and we comment this in the following section, for several criteria of appropriateness. First we consider the WM and OWA operators, and then the WOWA.

Quality of solution found: ASM based methods are appropriate for learning the weights for the WM and the OWA operators given that for these operators the minimisation problem is a quadratic one and almost exact solutions can be found. For quadratic problems the formulation of the problem and their resolution is simple. For the WM and OWA operators, genetic algorithms are not so appropriate because the best solution they find is usually a sub-optimal one. Thus ASMs are more precise for these operators. **Computational cost:** the computation costs in terms of memory and CPU usage are larger in the case of the GA based approach than in the case of the ASM. This is due to the fact that genetic algorithms need to compute the fitness function for each of the individuals in each of the populations and this is calculated applying the aggregation function to each example. Instead, the costs of the ASM are mainly proportional to the number of variables and not to the number of examples. This is so because ASM only uses the examples once - in the initial step - to compute a N*N matrix (where N is, as above, the number of examples) and the iterative method uses this matrix but not the initial examples. According to this, the greatest difference between the costs is when the set of examples is large and the number of variables is small. In relation to the implementation, neither ASM based methods nor GA based ones are difficult to implement. **Ease of implementation:** in order to implement ASM, a general ASM algorithm is required which allows the selection or removal of active constraints (described in [Torra99b]) and the solution of linear equations. The implementation of GA has to follow the indications given in Section 3.2.3. According to the better approximation, the computational cost and the implementation difficulties, it seems that when the aggregation operator is either WM or OWA then ASM based methods are more recommendable.

WOWA operator: in the case of the WOWA operator, in contrast to WM and OWA, the complexity of ASMs increases because the function to minimise is not quadratic. This is due to the existence of the interpolation function w^* (built from the weighting vector w - one of the weighting vectors to learn) and due to the fact that this function is applied to additions of some of the p's (the other weighting vector to learn). Although there exist some optimisation techniques to find approximate solutions for non quadratic problems, for example [Luenberger73], their implementation is a difficult and non trivial task. In this case, the complexity of genetic algorithms does not increase and the main change is to use the WOWA operator in the fitness function instead of using the OWA or WM. Thus, genetic algorithms can be used to obtain sub-optimal solutions with sufficient precision more easily. Besides of these advantages, the genetic algorithm approach is also suitable because it can find several sub-optimal solutions. This is particularly adequate because initially we do not know whether the distance for the WOWA operator will be convex or not (this is not the case for the WM and OWA where D(p) is convex).

Missing values: the use of genetic algorithms presents an additional advantage for either the WM, OWA or WOWA operators. This is the case when data files include missing values or the number of variables is different in each example. In this section an alternative definition of the distance function has been presented, which takes this fact into account. Using this function is relatively simple using genetic algorithms, given that only the fitness function has to be adapted, and the rest of the program does not have to be modified. However, this is not so easy for the ASM based approach because the resulting distance is in this case more complex due to the missing values. Therefore, when the number of variables is not the same for all the examples, GA's provide a simpler solution to learn the weights.

Thus, GA's and ASM represent two different approaches to learning the weight values used by the WM, OWA and WOWA aggregation operators. Each approach has its advantages and disadvantages, some of which are summarised above, and the methods provide viable options when considering alternatives for learning weights, whose choice is influenced by the nature of the data sets being processing, the observed results and the desired precision and repeatability of the outcome.

Different ‘w’ (reliability) weight vector for each variable

This has two aspects: (i) allowing only a limited number of forms for the ‘w’ weights; (ii) the second part is the definition of a ‘w’ vector for each variable and a ‘p’ value for each variable as before. The default WOWA operator has one ‘w’ vector and one ‘p’ vector which remain constant during the processing of a given data set. In the modified version, the data set can be thought of in two dimensions as a matrix which has a ‘w’ vector associated with each column (variable), and a ‘p’ vector associated with each row. The ‘w’ and ‘p’ vectors are interpreted as the concepts of ‘reliability’ and ‘relevance’, respectively, as before.

Assignment of ω weights: the ω weight vectors depend on the characteristic curve assigned to each variable. Which characteristic curve to assign to each variable is decided by the medical expert. A ‘bias’ vector is used to hold the values which indicate which of the five possible characteristic curves is assigned to each variable. For example a bias vector, bias[i], i=1..N, where N=5 (the number of variables). The bias vector could have the following assignment: bias[1]=’M’, bias[2]=’E’, bias[3]=’H’, bias[4]=’M’, bias[5]=’E’. Thus, the content of bias[2], for example, indicates which characteristic curve is to be assigned to variable 2, which in this case will be the ‘E’ curve, which gives an even bias to all values. The characteristic curves are allowed five fixed possible forms: even bias (E) is assigned as {0.2, 0.2, 0.2, 0.2, 0.2}; low values bias (L) is assigned as{0.3,0.3,0.2,0.1,0.1}; high values bias (H) is assigned as{0.1,0.1,0.2,0.3,0.3}; high&low values bias (O) is assigned as{0.3,0.15,0.1,0.15,0.3}; middle values bias (M) is assigned as{0.1,0.25,0.3,0.25,0.1}. The index values which are stored in the bias vector can therefore be one of the values, ‘E’,’L’,’H’,’O’,’M’. This still allows for a great number of possible permutations of the values in the bias vector. The values assigned to the ‘w’ vector are then interpolated by the quantifier as in standard WOWA.

We assume that the biases (M,H,L,E,O) have been previously assigned for each variable to biases[i], i=1...N. We recall that with the ‘w’ vector we are assigning a ‘reliability’ weight, thus in the example above we are saying that extreme values for the variable *i* are not so reliable (in the given degree/scale) and their influence is diminished. On the other hand, midrange values are considered relatively more reliable and their influence is relatively potentiated.

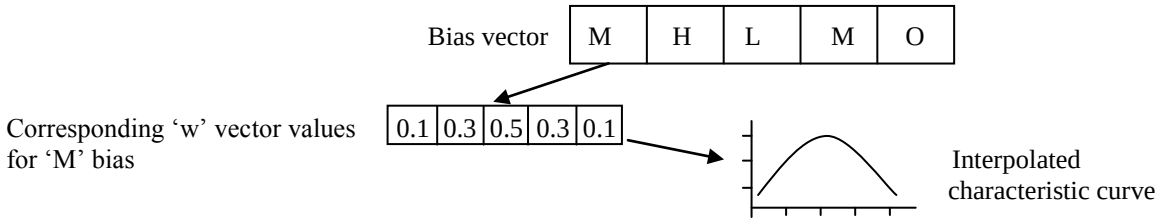


Figure 54. Bias vector as index for characteristic ‘reliability’ curves for each variable

Use of the reliability vector ω

In Figures 55a to 55e we can see each of the characteristic curves defined by the E, L, H, O and M vectors whose weight values were defined in the previous section. These vectors represent the ω vector which can be used to strengthen some responses while diminishing others, as we see in Figures 55a to 55e. For example, in Figure 55b, a response of ‘never’ will be strengthened to affect the (aggregated) outcome more than a response of ‘always’, which will have its contribution to the (aggregated) outcome diminished.

Note that we distinguish this weighting effect for the response reliability from the membership grade of the responses as detailed previously. We can say that the membership grade is reflecting the qualitative information provided by the patient, whereas the ω weighting of the responses reflects the medical experts knowledge of what responses are most expected for each question.

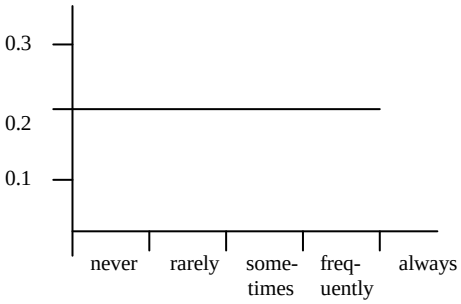


Figure 55a. Even bias vector (E)

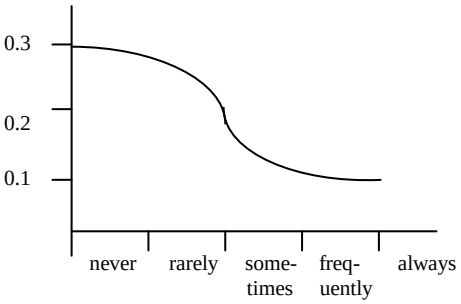


Figure 55b. Low bias vector (L)

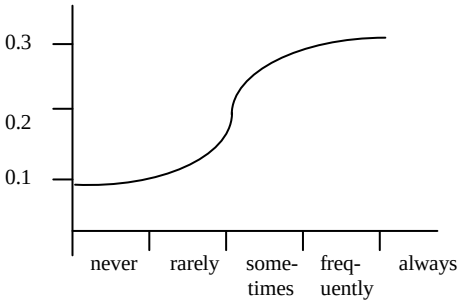


Figure 55c. High bias vector (H)

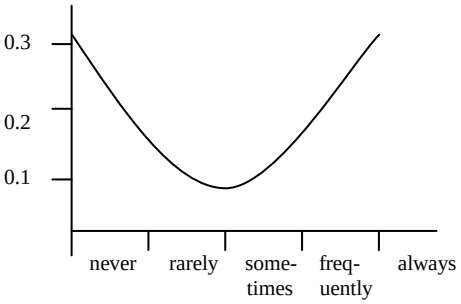


Figure 55d. High & Low bias vector (O)

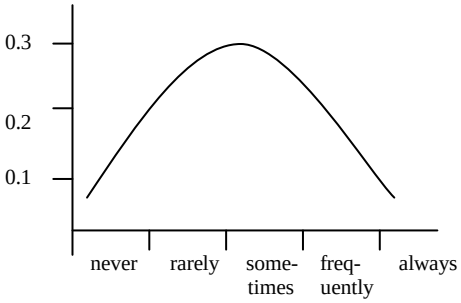


Figure 55e. Middle bias vector (M)

With reference to the description of the assignment of the values to the 'w' vector, each variable may have assigned one of five possible characteristic curves, as indicated in the 'bias' vector. Each characteristic curve is stored in a separate vector, as five value points. It is from the value points of the characteristic curves, that WOWA uses the interpolation method of Chen and Otto [Chen95], to create a continuous function curve which is used to weight all the values of each variable.

Summary of the adaptations to the WOWA aggregation operator

The following adaptations were made to the WOWA aggregation operator: (i) reformulation for examples with a different number of variables in order to enable processing of missing values; (ii) calculation and input of membership grades; (iii) interpolation function for each variable. The learning of the 'p' weights using the genetic algorithm did not require modifications to the WOWA code itself, given that the weights are assigned to/from the p and w parameters of WOWA. In contrast to the standard WOWA operator, which executes the interpolation of the values in the w-vector only once, the modified version of WOWA must execute the interpolation each time it encounters a new variable and its corresponding cases. The following is a description of this version of WOWA, called AWOWA.

AWOWA aggregates a row of values into one value, using two weighting vectors p and w. p is called the relevance vector, with one value for each value, and w is called the reliability vector, also with one value for each variable. A third vector, a, is the data vector with data for one case for each function call to AWOWA. AWOWA is called thus: AWOWA(p,w,a).

(i) Reformulation for examples with a different number of variables in order to enable processing of missing values:

let N_v be the number of variables including those with missing values; let N_{mv} be the new number of variables once those with missing values have been identified and eliminated. For each variable v_i, each of its values is tested to see if it is null (or a value which indicates that it is unassigned). For each variable v_i which has missing values, its cell in the a,w and p vectors is removed and the remaining cells moved one place to the left (that is, the vectors are compacted).

(ii) Calculation and input of membership grades:

for each fuzzy type variable, a single value, fv_i is calculated. Let n be the number of fuzzy variables, m be the number of fuzzy labels (the same for all variables), l_i be the ordinal numeric value of the linguistic label {1,2,3, ….}, and μ_i be the grade of membership corresponding to each linguistic label. Thus, for each fuzzy variable:

$$fv_i = \sum_{j=1}^{n} l_{ij} \times \mu_{ij}$$

(iii) For all variables vj, j=1,n, define an interpolation function for each variable:

assign 'w' vector weights, for each fuzzy variable fv_j . For each element i,

$$w_{ji} = bias_i$$

where *bias* is the selected characteristic function for this variable.

(iv) The function SetQ does the interpolation of w and places it in an appropriate structure, **w***.

w is a vector with **n** points to be interpolated, where **n** is the number of variables.

$$\mathbf{w}^* = SetQ_w$$

(v) The function OrderA places the 'a' data values in ascending order and moves the corresponding 'p' values to the same positions.

$$OrderA_{p,\,a}$$

(vi) Weight the interpolation of **w** which has been placed in **w*,** by vector **p**, the second weight vector:

this produces a new weighting vector Ω which will act on the data values in step (vii) below. For each row of values **j** for each case, calculate the new weighting vector, using a monotone increasing function which is represented by **w***:

$$\Omega_j = \mathbf{w^*}\left(\sum_{j\le i}^{n} p_{\sigma(j)}\right) - \mathbf{w^*}\left(\sum_{j<i}^{n} p_{\sigma(j)}\right)$$

(vii) In the last step, the function T calculates the scalar product of the two vectors, Ω and **a**, this being the final output value for the current case, and where **a** is the vector of data points.

$AWOWA = T_{\Omega, \mathbf{a}}$

Note: In order to input the membership grades into WOWA, we aggregate the non-zero membership grades using weighted mean (WM) into just one numeric value, as detailed in step (ii) above. This value is normalised and given as input data to WOWA. For the membership values of different variables to be comparable, we should use the same membership function for all variables.

Application of the adapted WOWA to Apnea diagnosis

The following gives the implementation details for the techniques described in this section to the real application of Apnea diagnosis. The results of application to the real data sets are given in Sections 4.3 and 4.4.

Learning the 'relevance' weights from historical case data using an evolutive program (genetic algorithms)

Objective Function
If O_p is the diagnosis predicted by the aggregation function WOWA, and O_r is the normalized AHI value (apnea hypopnea index, see Section 1.4.7), that is the real diagnosis, then the objective is to minimize the square of the sum of the differences between the O_p and the O_r for all patient cases, as defined in formula (3.14) below. The square is used in order that the negative and positive errors do not compensate each other.

$$\text{Min } \Sigma(O_p - O_r)^2 \qquad (3.14)$$

We now outline how a genetic algorithm technique is used to learn the p (relevance) weighting factors from historical data values. A genetic algorithm has a set of input and output data (examples), a set of modifiable values (in this case the weighting factors), a set of constraints (in this case the sum of the weighting factors must be equal to 1), and an objective function, which in our case is to minimize the difference between the predicted diagnosis and the real diagnosis. We wish to find the weighting factors which best approximate the input and output data, while minimizing the objective function.

Example:

Input (data): I = {5, 6, 4, 6, 1, 5, 2, 5, 2, 2, 1, 2, 2, 5, 9, 7, 2, 2, 8}, one value for each variable.

Input (relevance weights to be learned): initial values of p = {0.90, 0.20, 0.85, 0.25, 0.50, 0.90, 0.62, 0.90, 0.63, 0.68, 0.55, 0.67, 0.61, 0.93, 0.74, 0.63, 0.64, 0.27, 0.94}, one value for each variable.

Output: O_p = [0,1] (predicted diagnosis value) O_r = [0,1] (real diagnosis value)

and, O_r = AHI / max (AHI)

Set of Constraints
(i) Values of ρ between 0 and 1, with precision of 2 decimal points; (ii) sum of ρ values equal to 1; (iii) values of ρ must be normalized.

The 'evaluate' procedure, as detailed previously in Figure 53, simply goes through all the individuals in the current population and assigns a 'fitness' score to each. The fitness score for each individual is calculated by executing the function which calculates the diagnosis (WOWA aggregator) for each of the patient cases (1..j) and with the ρ weight vector contained in the chromosome of individual **i**.

To diagnose a patient, WOWA is called thus:

$$\sum_{j=1}^{n} (A_i, V\rho, V\omega_j),$$

where A_i is the data vector for patient **i** , $V\rho$ is the variable weight vector for all variables, and $V\omega_j$ is the data value vector for variable **j**.

Manual assignment of the 'relevance' and 'reliability' weights

The relevance and reliability weights are assigned on the one hand by the medical expert, based on current clinical literature in the Apnea diagnosis field, on his own knowledge and experience, and taking into account the type of patients (the mix) which exists in the Salamanca Sleep Clinic with whom we collaborated for the study. In the previous section we have seen a method for 'automatically' generating the relevance weights by applying a 'genetic' learning algorithm to historical case data. In Section 4.4 the results of diagnosis using the WOWA aggregation operator with 'automatically' assigned 'relevance' weights is compared for precision with the results of diagnosis using the WOWA aggregation operator with 'relevance' weights assigned 'manually' by the medical expert.

In the case of the demographic and clinical data variables, such as age, sex, height, weight, neck circumference, it is easier for the medical expert to evaluate their reliability and relevance. In the case of alcohol and tobacco consumption, this depends on the truthfulness and accuracy of the patient. Blood test results could be used as corroborative evidence to the verbal responses. Also, relevance and reliability estimates in the clinical literature can be used. The questionnaire questions themselves, by their nature, can be classified by relevance and reliability. On the other hand, they again depend on the truthfulness and accuracy of the patient. Some questions will be more susceptible to untruthfulness or inaccuracy than others. As a data post-processing phase, inconsistencies can be identified in responses, and an overall judgment of the reliability of a patient's questionnaire can be made, which may 'dampen' or 'potentiate' the general reliability values.

Statistical assignment of the 'relevance' and 'reliability' weights

We have discussed manual weight assignment by the medical expert, as well as automated weight assignment by directly learning the weights from the data, using a genetic algorithm. In Section 4.3.2, a third method for assigning the weights is attempted, which uses diverse clustering and classification/prediction methods to establish a ranking for relevance. In this sense we understand relevance as the degree of correlation of an input variable, such as 'neck circumference', with the output value, such as 'apnea diagnosis'. A 'by-product' of the clustering and classification/prediction models is the identification of the most significant variables for a data model. This analysis is one of the principal areas of the data exploration phase of any data mining project. Of course, we can save a lot of time if a 'domain expert' gives us guidance in choosing an initial subset of candidate variables, based on experience and intuition. Contrastable clustering methods, such as K-means, Kohonen SOM and Condorcet can give different rankings of significance. This is also true for classification/prediction methods such as Logistic Regression, Linear Regression, C4.5 Rule Induction and a BackPropagation Neural Network. If we poll the methods we should see a general consensus on the ranking of the variables. If the degree of consensus is quantified as an index in some way, we could say that the greater the value of the index, the greater the consensus, and the greater the reliability (of its relevance). Conversely, the smaller the value of the consensus index, the lesser the reliability. Thus we can obtain a value for relevance and a value for reliability, for each of the variables, by purely statistical means.

Chapter 4. Application and Results

This Chapter details the application of the methods and algorithms described in Chapters 2 and 3, to artificial and real data sets, with emphasis on some real medical domain problems. The real domains included are: ICU prognosis and Apnea syndrome screening. This allows us to establish their behavior as data exploration and modelling tools, and also allows us to evaluate and contrast the results with other methods from the statistics and data mining literature.

In Section 4.1 there is an extensive analysis of a real hospital ICU dataset, using statistical, data mining and experimental techniques, the latter developed by the author. The techniques include the use of the Hartigan 'joining algorithm' with crisp and fuzzy covariances as input, and the use of fuzzy c-Means to cluster data and give indications of relation between variables and the cluster centres. In Section 4.2 we apply four variants of the fuzzy covariance algorithm [Nettleton98b] to artificial datasets to generate a fuzzy covariance matrix given as input to the Hartigan 'joining algorithm'. The objective is to identify and rank the most significant variables in each dataset. The benchmark results are compared with C4.5 and a Neural Network applied to the same data. In Section 4.3 diverse clustering and classification techniques are used to establish the reliability and relevance of the variables in a dataset of Apnea cases from the Hospital Clinic of Barcelona. OWA and WOWA aggregation techniques are then applied to the same Apnea case dataset, the input data being captured by questionnaire in a crisp form, and the output being a binary valued diagnosis. There is also a comparison of the results of using weights and variables assigned by the medical expert, with the results using weights and variables assigned by statistical and machine learning methods. Finally, in Section 4.4 we apply the WOWA aggregation operator to diagnose Apnea cases using a dataset collected by the Hospital of the Santisima Trinitat, Salamanca. In this case, the data was captured in both crisp (categorical) and fuzzy (continuous scale) form, using a specially designed questionnaire.

4.1 Icu prognosis data - Hospital Parc Tauli

In this section of work, we analyse a data set of real 'hospital admissions' ICU (Intensive Care Unit) data prepared for statistical studies of diagnosis and prognosis. A complete inventory of the variables is given in Annex 2. The data set is complex, given the number of variables, but the quality, in statistical terms, is good; that is, a representative distribution of the cases is covered, there are few unknown or erroneous values, and so on. There are some 1100 cases, with 100 variables for each case. The first 17 variables are vital signs and other general data about the patient. The rest of the variables are derived from a urine sample, a blood sample, observation and initial diagnosis. The first 24 hours of admission of the patient to the ICU are the most critical, and the situation of the patient at the end of this period is detailed by a series of variables in the dataset. Given that the ICU is a critical unit in the hospital, with a concentration of expensive and limited resources, it is very important to be able to prioritise admissions and anticipate their needs, in the short and medium term.

The objective of the work in this section with respect to the ICU dataset is to find both most significant features and factors which relate the input variables to three output variables: 'duration_icu', 'duration_hos', and 'vital_state_icu'. 'duration_icu' is the number of days which a patient stays in the ICU unit. 'duration_hos' is the number of days which a patient stays in the general hospital from time of admission to release from the hospital. This includes the time spent in the ICU unit. The third output variable, 'vital_state_icu' indicates the vital state of the patient on leaving the ICU unit, which can be 'alive' or 'mortality'. 'duration_icu' and 'duration_hos' are important for several reasons: first, the prognosis of the patient, as the recuperation time indicates possible complications or additional needs for hospital resources. The duration variables are also important for hospital planning, in order to evaluate the possible load on the hospital's capacity to attend patients, while maintaining a minimum quality of service. 'vital_state_icu' is a direct prognosis of the survival of the patient, in which the first 24 hours is a critical time period. As the first Data Mining phase, which is the exploration step, we wish to discover factors and features which show tendencies and relations between the 100 or so input variables and a given output variable. In the modelling phase which follows we try different modelling techniques to create models which allows us to predict these variables, using as input the best factors and features identified in the exploration phase. Common Data Mining algorithms are first tried on the data: C4.5, ID3, Kohonen SOM, feedforward neural nets, and statistical correlation. Then we try specific techniques which have been considered in the thesis: Hartigan 'joining' algorithm, fuzzy covariances and fuzzy c-Means. We compare these less conventional techniques with the standard algorithms and evaluate the hypothesis that a fuzzy approach can complement or improve data exploration and modelling in the case of the ICU data.

One of the key areas of analysis of the data is that of factor selection, or data reduction, in which subsets of variables are identified which have the greatest relation to a given output variable, for example, 'duration of stay in ICU', or 'vital state'. We distinguish 'factor selection' from 'feature selection' in that 'features' are characteristics of the data, such as 'increase in mortality for renal failure cases when $FIO_2 = 1$. On the other hand, 'factors' are inputs to the data model,

specific variables, such as FIO_2 . Some variables are more easily identifiable, such as 'Age', 'Sex' and 'Blood Pressure', while others are clinical technical indicators, such as 'FIO_2', a binary flag which indicates if the level of FIO_2 (Fractional Inspired Oxygen as measured by a Pulsoximeter) has been greater than 0.50 during the first 24 hours, or 'Mech_Ven', a numerical value which indicates the number of hours of mechanical ventilation during the first 24 hours.

With respect to the selection of 'reduced sets' of 'most significant inputs', we try to identify and select a manageable number of variables, which will then be used as inputs to classificative and clustering models, and further data exploration. For example, if the total number of variables is 100, such as in the ICU dataset, we would consider a manageable number of variables for the subset to be between 10 and 20. In practise, for the ICU data, we have chosen 15 or 17, first by ranking on a 'significance factor', which may vary depending on the modelling/analysis method employed, and of course, the criteria used. Then we try to identify a cut-off point, beyond which the variables' significance drops quickly or indeed our confidence in a given variable diminishes. In the case of selection of variables by a medical expert, this may also vary the number chosen as a minimal set, depending on the criteria, output variable, distribution and characteristics of the cases in the dataset, and so on.

As a study of the field, different techniques were tried for clustering and classifying data with different data types. Among the techniques tested for clustering were Kohonen Neural Nets, Hartigan Joining algorithm and fuzzy c-Means. For classification/prediction the following techniques were tried: feed-forward neural networks and tree induction, represented by ID3 and C4.5. The first objective of these tests is to determine that the quality of classification and prediction with the given data can be improved by fuzzy processing and representation techniques. A second objective is to explore the given data set and thus better understand its interrelations and nature. Thus it will create a basis for formulating a working methodology for developing new algorithms.

4.1.1 Data Exploration

The following section represents the first phase of a Data Mining project, that is, the exploration of the data in order to establish distributions, data quality, and a general understanding of the data set, using simple statistical and graphics display tools. Refer to Annex 2 for a complete list of ICU variables.

Table 39. Selected variables with possible values and the distribution of those values within the dataset

Variable	Description	Possible values	% Distribution
COMA_ADM	Presence of coma or profound stupor at time of admission to ICU	0	76.27
		1	23.73
TIPO_ADM	Type of Patient {1=Emergency Surgery, 2=Planned Surgery,3=Without Surgery}	3	54.68
		2	29.74
		1	15.07
		0	0.51
MALIG	Malign Neoplasm part of actual problem?	0	82.38
		1	17.62
ICU_SER	Service at the time of admission to the ICU. {0=Medical, 1=Surgery}	0	50.2
		1	49.8
LINES	Number of lines at 24 hours after admission	4	16.5
		5	15.89
		3	15.78
		2	14.26
		6	13.44
		7	8.55
		1	6.25
		8	3.46
		0	2.44
		9	1.63
		10	1.53
A_R_FAIL	Acute renal failure	0	87.98
		1	12.02
P_H_STAT	Previous health state	1	49.29
		2	32.79
		3	16.09
		4	1.83
COPD	Chronic pathologies {1=yes, 0=no}	0	84.42
		1	15.58
OSF	Number of organ systems failing, calculated by computer program	0	49.59
		1	28.62
		2	13.85
		3	4.07
		4	3.46
		5	0.31
		6	0.10

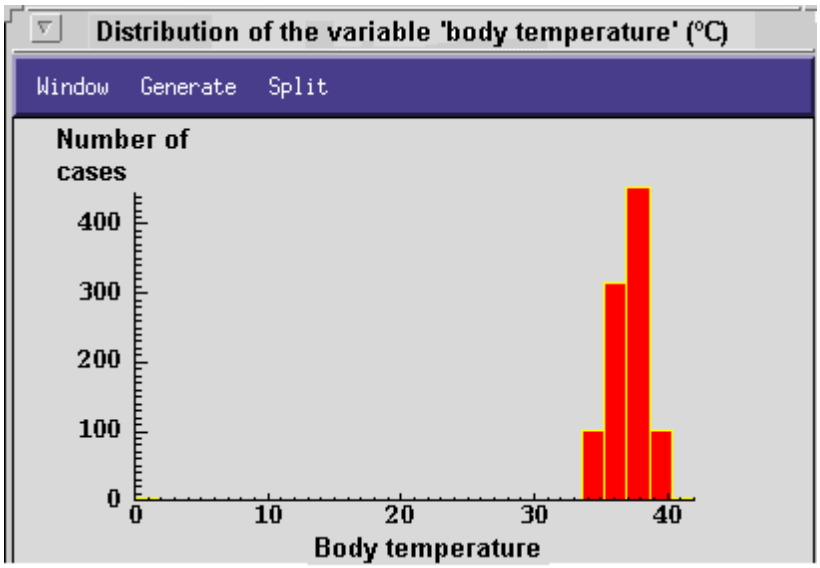


Figure 56. Distribution of the variable 'body temperature'

We see from Figure 56 that 'body temperature' has a very distinctive profile and range for the human body. The physician knows that when the values go outside this range it indicates a possibly life-threatening situation for the patient.

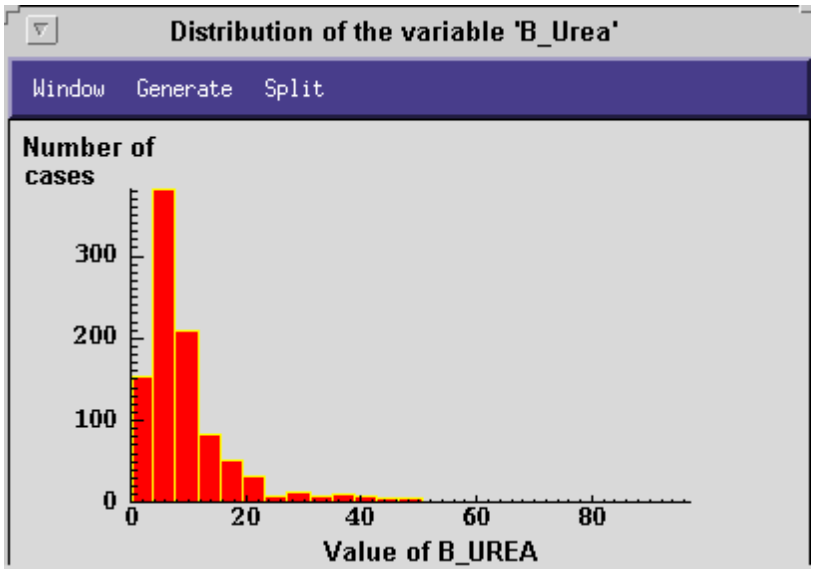


Figure 57. Distribution of the variable 'blood urea'

In Figure 57 we see the characteristic distribution for the variable 'blood urea', in which the x-axis indicates the concentration of urea in the blood. It peaks between 6 and 10 units and rapidly drops off after 20 units.

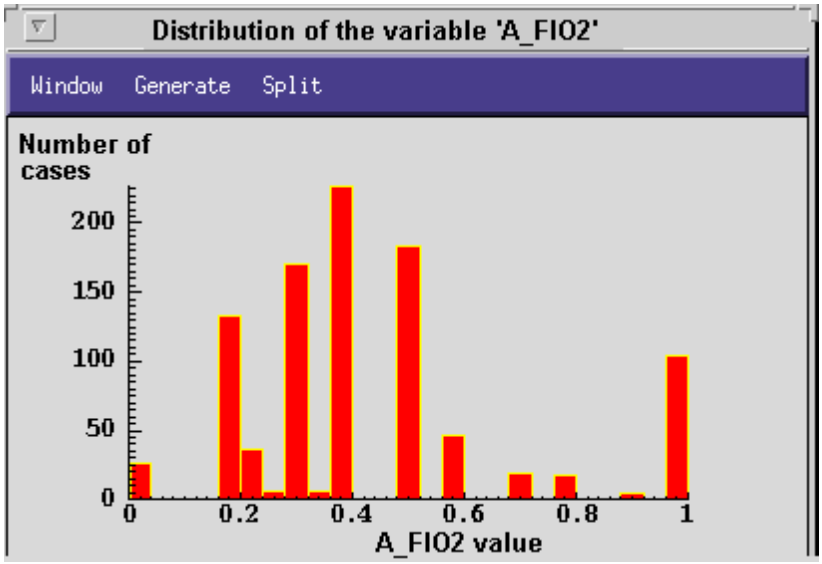


Figure 58. Distribution of the variable 'a_fio2'

In Figure 58 we see the distribution of the variable 'a_fio2', which has a roughly normal distribution between the values of 0.2 and 0.6, and a concentration of cases at the extreme point 1.

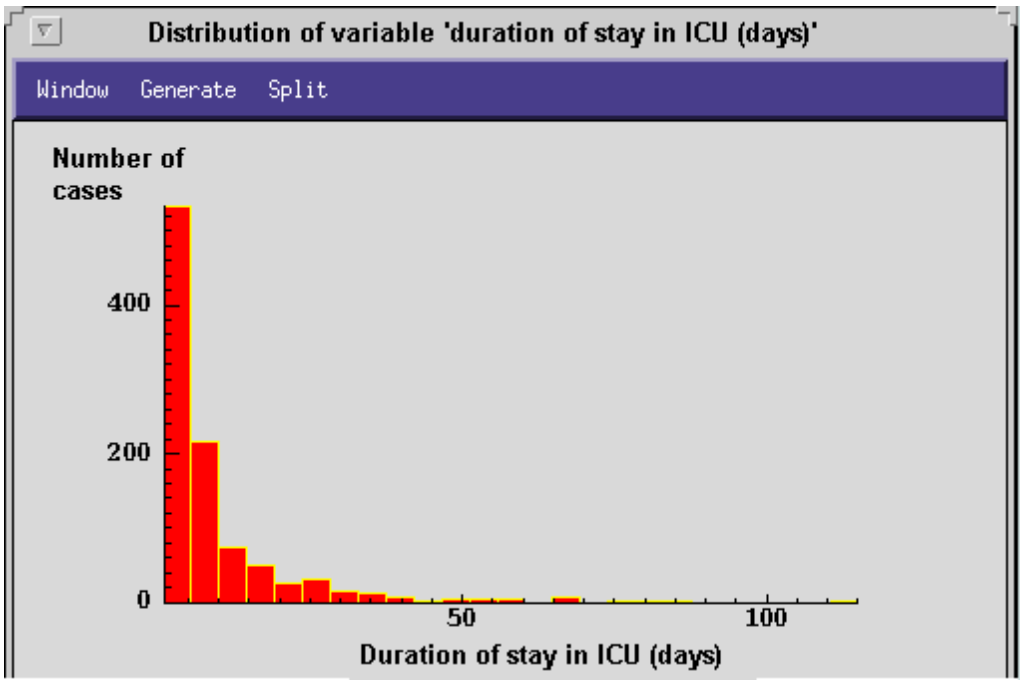


Figure 59. Distribution of the variable 'duration_ICU' in days

In Figure 59, we see a characteristic distribution for the variable 'duration_icu', which measures the time in days in which a patient is in the ICU. We see a heavy weighting for short stay, that is 5 days or less, whereas the frequency for length of stay practically reduces to zero after 40 days. 'duration_icu' is one of the variables for which we try to create predictive models, and relate the other (input) variables to it.

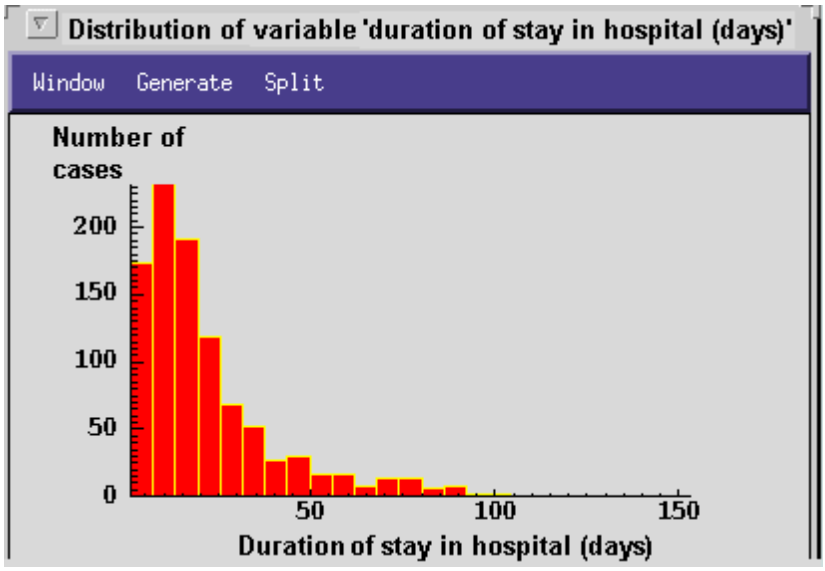


Figure 60. Distribution of the variable 'duration_hospital' in days

In Figure 60, we see the characteristic distribution for the variable 'duration_hos', which measures the time in days in which a patient is in the Hospital. We see a peak at 10 days, and the distribution tailing off until 50 days, where it stays constant until reducing to zero at 90 days. 'duration_hos' is another of the variables for which we try to create predictive models, and relate the other (input) variables to it.

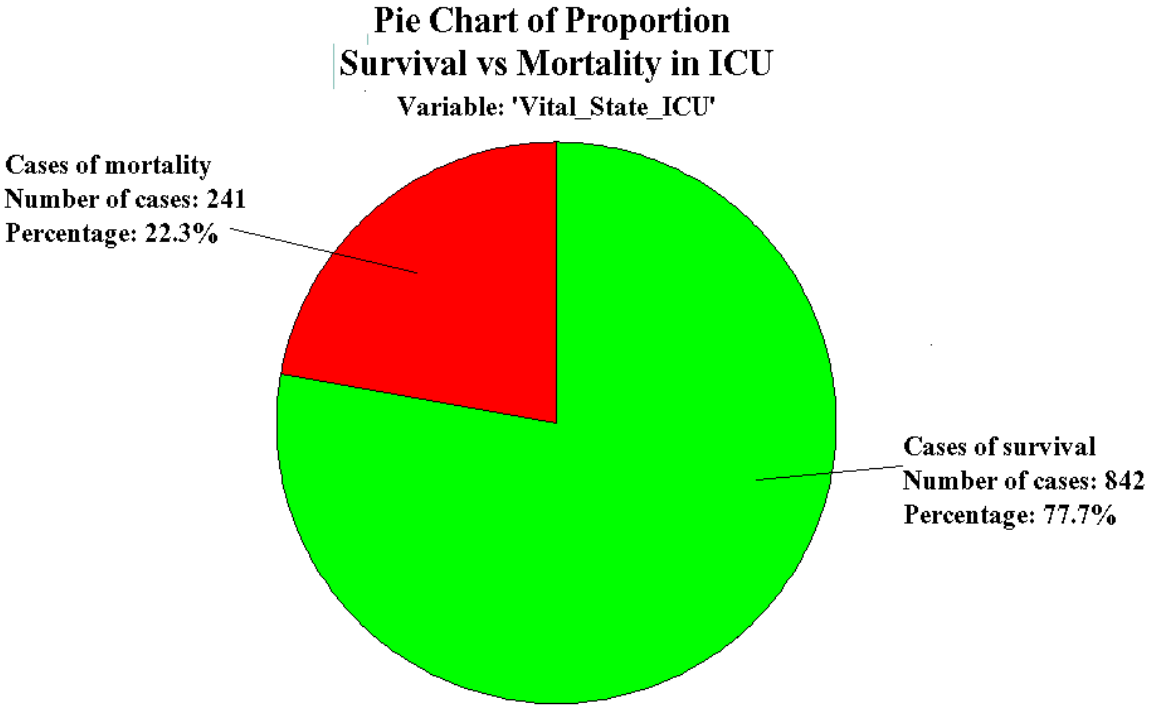


Figure 61. Distribution of the variable 'vital_state_icu'

Figure 61 (above) shows, for the variable 'vital_state_icu', the proportion of cases of mortality with respect to the proportion of cases of survival, in the ICU environment. The ratio of survival to mortality is approximately 3 to 1.

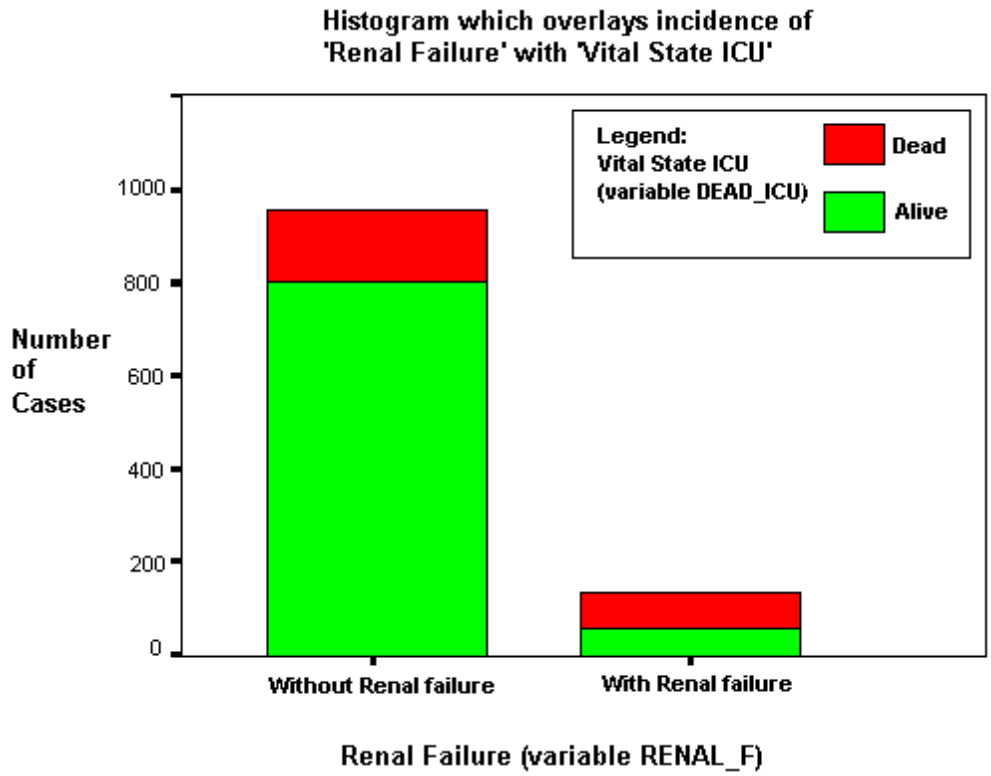


Figure 62. Distribution of the variables 'renal failure' and 'vital_state_icu'

In Figure 62 we see an example of 'overlay', which is a common technique in the data exploration, in order to identify relationships between pairs of variables. Figure 62 is a histogram of the binary variable 'renal_failure' on the x-axis, plotted against the number of cases on the y-axis. The proportion of cases corresponding to the two possible values of the variable 'dead_icu' is 'overlayed' (indicated by different colours) for each respective variable-value of the variable 'renal_f'. In Figure 62 we see that for the cases represented by the left bar, for which there is no renal failure, the mortality rate is about 15%. Contrastingly, as shown in the right bar when renal failure occurs, the mortality rate is greater than 50%. As it is important to contrast the initial 'findings' with clinical knowledge, this result was confirmed as valid by the medical expert.

The following statistics (below) have been generated for some of the most significant numerical variables of the ICU dataset. From these statistics, we can see that the data represents a difficult problem to model, given that even between the variables which have been selected as the most representative of the problem, there is a poor statistical correlation. As a possible way forward with this type of dataset, we could think that a good model would not use the variables in their present form as input, but would requiere the creation of derived factors which are composed of two or more basic variables. Another way forward would be to identify homogeneous clusters (set of cases) in the data, for which higher correlation exists between variables: for example, patients with renal failure, could be a cluster for which a model can be created, and for which a higher correlation exists between the inputs and the proposed output, for example, 'duration_icu'.

Statistics for field : BODY_TEMP
- Minimum = 0
- Maximum = 42
- Occurrences = 982
- Mean = 36.889
- Standard Deviation = 2.6645
- Correlation (Pearson Product-Moment) for field :
  - A_FIO2 = 0.129 (Low positive correlation)
  - DURATION_ICU = 0.077 (Low positive correlation)
  - B_UREA = 0.070 (Low positive correlation)
  - DURATION_HOS = 0.053 (Low positive correlation)

Statistics for field : B_UREA
- Minimum = 0
- Maximum = 97
- Occurrences = 982
- Mean = 10.081
- Standard Deviation = 10.138
- Correlation (Pearson Product-Moment) for field :
  - A_FIO2 = 0.157 (Low positive correlation)
  - DURACION_ICU = 0.157 (Low positive correlation)
  - T_CORPORAL = 0.070 (Low positive correlation)
  - DURACION_HOS = 0.050 (Low positive correlation)

Statistics for field : A_FIO2
- Minimum = 0
- Maximum = 1
- Occurrences = 982
- Mean = 0.44379
- Standard Deviation = 0.24205
- Correlation (Pearson Product-Moment) for field :
  - DURATION_ICU = 0.169 (Low positive correlation)
  - B_UREA = 0.157 (Low positive correlation)
  - BODY_TEMP = 0.129 (Low positive correlation)
  - DURATION_HOS = 0.022 (Low positive correlation)

Statistics for field : DURATION_ICU
- Minimum = 1
- Maximum = 115
- Occurrences = 982
- Mean = 9.1792
- Standard Deviation = 11.740
- Correlation (Pearson Product-Moment) for field :
  - DURATION_HOS = 0.580 (Medium positive correlation)
  - A_FIO2 = 0.169 (Low positive correlation)
  - B_UREA = 0.157 (Low positive correlation)
  - BODY_TEMP = 0.077 (Low positive correlation)

Statistics for field : DURATION_HOS

Minimum	=	1
Maximum	=	153
Occurrences	=	982
Mean	=	21.845
Standard Deviation	=	19.559

Correlation (Pearson Product-Moment) for field :

DURATION_ICU	=	0.580 (Medium positive correlation)
BODY_TEMP	=	0.053 (Low positive correlation)
B_UREA	=	0.050 (Low positive correlation)
A_FIO2	=	0.022 (Low positive correlation)

4.1.2 Benchmarking of C4.5 algorithm on the ICU dataset.

In test group A, all 100 variables are used as input, the output is the binary value 'vital state icu', which may assume values '0' (alive) or '1' (not alive). The algorithm used to create the predictive model in test group A is C4.5. Within test group A, subgroup A1 uses the original data distribution with respect to the output variable {0's: 77.51% of the training set; 1's: 22.39%}. On the other hand, subgroup A.2 has its distribution altered (by replicating the cases whose outcome is '1') with respect to the output variable {0's: 50% of the training set; 1's: 50%}.

In test group B, a reduced set of 15 variables was used, chosen by the medical expert and by statistical correlation analysis. The output variable is again 'vital state icu' considered as a binary value '1' or '0', and the modelling algorithm is C4.5.

4.1.2.1 Test group A: using all 100 variables as input; 'vital_state_icu' as output; C4.5 as modelling algorithm

Test subgroup A.1 The following group of tests used C4.5 to classify the cases in terms of the variable 'vital_state_icu', which can assume two values: 0=survival of patient, 1=non survival of patient. The 'basic mode' of C4.5 was used, in which 'windowing' and 'pruning' options have default values. The distribution of cases is: {0's: 77.51% of the training set; 1's: 22.39%}. In this group of tests, all 100 available descriptive variables were used as input.

Table 40. Prediction results for different training set percentages

% Train	Precision (%correctly classified)		
	{0,1}	**0**	**1**
10	81	95	25
20	84	93	42
30	87	96	52
40	86	94	52
50	82	88	55
60	83	91	54
70	83	92	49
80	83	92	51
90	82	91	56

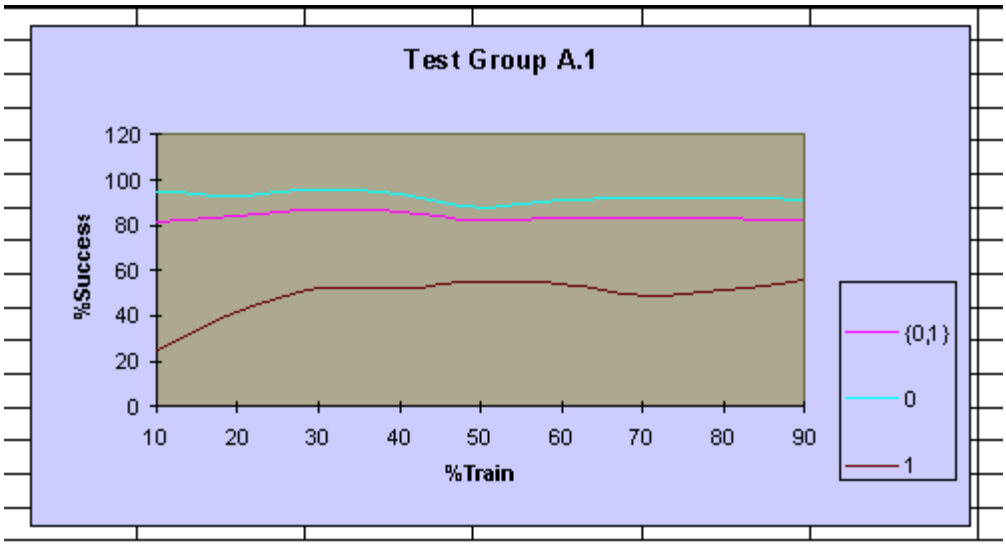


Figure 63. Prediction results for different training set percentages

Figure 63 shows a plot of the data given in Table 40. It shows that there is not a great influence of the percentage of training cases with respect to the total number of cases, on the precision. Each training set is extracted from the complete data set by random case selection, and is then validated for quality, that is, the random distribution of cases in general and the distribution profiles (seen in histograms) of values of key variables. We also carry out the elimination of

any outliers (extreme values) which could distort the training set. The complete dataset had been previously treated to filter out erroneous data and missing values. We take into account that, for 1000 cases, a 10% sample, well chosen and representatively distributed, allows C4.5 a good generalisation to the other 90% of the cases. Notwithstanding, the precision with 'vital_state_icu' = 1 stays consistently below 60% precision (y-axis of Figure 63) and reaches a peak with 30% of the dataset used for training (x-axis of Figure 63)

Test subgroup A.2 This group of tests was carried out with the same conditions as those of A.1, with the exception that a 'boost' function was used to equal the number of example cases (0's and 1's), in order to prediction 'vital_state_icu' {0,1}.

Table 41. Prediction results for different training set percentages using 'boost'

% Train	Precision (%correctly classified)		
	{0,1}	**0**	**1**
10	81	95	33
20	84	93	52
20*	83	91	56
30*	81	88	57

*expert mode: with windowing and pruning activated.

In Table 41 we can see the improvement in precision for 'vital_state_icu'=1, with a smaller percentage of training cases, due to the use of a 'boost' function. The 'boost' function ensures an even distribution of the values of the classification variable , in this case 'vital_state_icu', in the training set. In the original dataset 77% of the cases have 'vital_state_icu'=0, while only 23% of the cases have 'vital_state_icu'=1. This avoids the induction algorithm giving a proportional bias to 'vital_state_icu'=0, the 'boost' replicates the cases where 'vital_state_icu'=1 until they occupy 50% of the training dataset. The other alternative would be to proportionately reduce the cases where 'vital_state:icu'=0, but this solution is not so good because it may result in information loss of the cases eliminated, whereas if we duplicate the cases which have the lesser proportion, we will not lose any cases.

4.1.2.2 Test group B: using reduced set of 15 variables as input; 'vital_state_icu' as output; C4.5 as modelling algorithm

For this group of tests, the C4.5 algorithm was used in 'simple mode', that is with default windowing and pruning assignments. We again predict the 'vital_state_icu' variable which can assume the values {0,1}. Distinct from test group A is that a reduced set of variables was used as input to train the model. The 15 reduced variables were chosen by a joint selection by the medical expert and by statistical correlation analysis.

Table 42. Prediction results for different training set percentages using reduced variables as inputs

% Train	Precision (%Correctly classified)		
	{0,1}	**0**	**1**
10	83	92	54
20	88	97	56
30	87	94	63
40	85	93	56
50	85	90	65
60	84	94	51
70	86	93	56
80	84	91	58
90	85	96	54

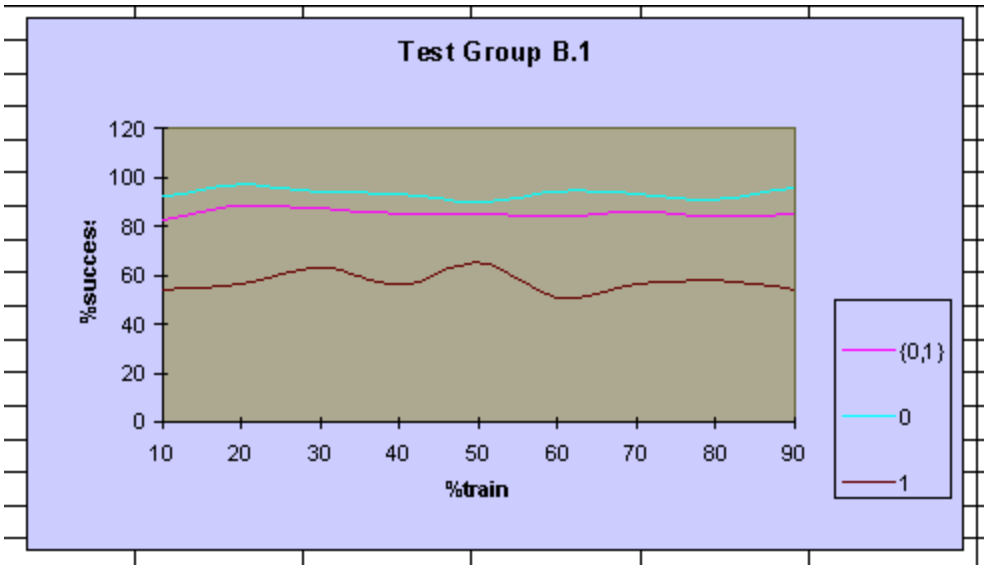


Figure 64 . Prediction results for different training set percentages using reduced variable set as inputs

Figure 64 shows a plot of the data given in Table 42. In a similar manner to the test Group A, it shows that there is not a great influence of the percentage of training cases with respect to the total number of cases, on the precision. We observe that the precision for {0} and {0,1} is fairly constant while the precision for {1} has two slight peaks for training set size equal to 30% and 50%, the latter providing the best result for {1}.

4.1.2.3 Classification tests for the variable 'duration_icu'

In the following tests, we contrast the selection of an input variable set and definition of category ranges by automatic means (statistical and data mining) (Steps (i) and (ii)), with the selection of the input variable set by a medical expert (Step (iii)). In Step (i), a neural network was first trained to select a reduced set of input variables, using a 'sensitivity analysis' which weights the inputs with respect to their internal activation level in the neural network, and with respect to the output layer. Then a categorisation for 'short stay', 'medium stay' and 'long stay' was derived from the distribution histogram of the variable 'duration_icu'. This was used by the C4.5 algorithm to train a rules model for the data. The rules can be seen below, together with respective observations and initial conclusions.

As a second step **(ii)** , the high values outliers were eliminated from the dataset, where the cutoff was defined at 32.35 days, assigned by visual and statistical inspection of the values for 'duration_icu'. The C4.5 model was retrained for this data. This showed an improvement in accuracy for all 3 categories.

In step **(iii)** , a new input set of variables was used, selected by a medical expert, and the ranges for the categories ('short', 'medium' and 'long'), were redefined, for the same data, and the C4.5 retrained. This resulted in a significant improvement for the 'long' category, but no improvement for 'short' and 'medium'.

Steps **(i)** to **(iii)** illustrate a typical 'data mining' methodology to develop models for complex data, which has a directed but also 'trial and error' nature.

Step (i) – Initial test using 3 value categorisation

In this first step we categorise 'duration_icu' into 3 possible output values; then a neural network is used to select a reduced set of variables as input. Finally these inputs are fed into a C4.5 algorithm to create a predictive model.

The variable 'duration_icu' is categorised as follows: **short** is assigned to cases whose duration in the ICU is 10 days or less; **medium** is assigned to those cases whose duration is greater than 10 days and less or equal to 20 days; **long** is assigned to those cases whose duration is greater than 20 days. These ranges were initially assigned with the consensus

of the medical expert and by graphical and statistical inspection of the distribution of the variable, and by further inspection were then refined from 10 to 9.19, and from 20 to 20.42, respectively, for the range limits.

In Table 43 we see the precise distribution within the data set of the assigned ranges for each label category.

Table 43. Distribution by occurrences of label values (classes) in the data and their ranges (derived from a distribution histogram of the variable 'duration_icu')

Category	%of cases which correspond to the category in the dataset	Category Range (days of stay in ICU)	
short	74%	< 9.19	
medium	14%	>= 9.19	<20.42
long	12%	>= 20.42	

A predictive model was generated using a dynamic neural network, that is one which varies its architecture dynamically during the training to find that which has the best predictive success. The neural network, as a byproduct, performs a 'sensitivity analysis' which, based on the activation strength of the input neurons, calculates their relevance to the overall predictive result, and presents this in the form of a 'ranking'. In Table 44 we can see the precision of the resulting model.

Table 44. Results statistics for a neural network using all inputs, NN architecture of 116-2-2-1 and a total of 757 test cases (70% of total dataset)

		Correct.	
Results:	short	82%	(452 cases)
	medium	55%	(72 cases)
	long	0%.	

Global precision: 69% correct.

In Table 45 we see the result of the 'sensitivity analysis' which has produced a 'ranking' of the inputs in terms of their relative strength of contribution to the predictive result. For example, we see that the variable 'MECH_VEN' has been identified as that which most contributes to the output, with a relative strength of 0.070. On the other hand, the variable 'MAP' is that which contributes least to the output, with a relative strength of 0.0124. Observe that we have cut off the list at 27 variables, from a total of 100. The remaining 73 variables would be discarded, and only the 27 most significant used for the reduced inputs model.

Table 45. Results of 'Sensitivity Analysis': significance ranking of variables relative to 'duration_icu' represented as a qualitative variable (relative strength of first 27 variables are given)

Ranking	Variable	Relative Strength	Ranking	Variable	Relative Strength
1	MECH_VEN	0.070	16	H_RATE	0.0149
2	PAO2	0.030	17	S_SODIUM	0.0145
3	B_UREA	0.027	18	WBC	0.0141
4	HEMA_F	0.023	19	SBP_ADM	0.0138
5	CONF_INF	0.022	20	CERE_DIS	0.0136
6	C_REN_F	0.021	21	S_CREA	0.0135
7	ARTER_PH	0.020	22	SEX	0.0133
8	S_HCO3	0.018	23	SHOCK	0.0132
9	OSF	0.016	24	A_RES_R	0.0131
10	BODY_T	0.016	25	1TYPE_ADM	0.0129
11	PROB_INF	0.016	26	SBP	0.0127
12	S_GLUCOS	0.016	27	MAP	0.0124
13	S_H_RATE	0.0159			
14	PEEP	0.0157			
15	AGE	0.0154			

In Table 46 we see the results for the NN model retrained only with the 15 most significant inputs listed in Table 45. We observe that, compared with results of the model using all 100 inputs as given in Table 41, there is no improvement in predictive accuracy. This may be due to the training time, an excess of overall inputs, or an inadequate definition of the input variables and their types. Alternatively, this may indicate some problem with the data set itself, such as lack of generalization between train and test sets.

Table 46. Results statistics for NN retrained with reduced inputs (architecture 36-2-2-1)

		Correct.	
Results:	short	82%	(450 cases)
	medium	50%	(66 cases)
	long	0%.	

Global precision: 68% success.

Once we have the reduced set of variables, we can now use the C4.5 algorithm to create a predictive model using them as input, in order to predict the variable 'duration_icu' as 3 categories

In Table 47 we see the results of the predictive model, trained using the C4.5 in basic mode, which implies using only automatic and default parameter settings (off) for windowing, pruning, and so on. We observe a reasonable global precision but a poor result for 'medium' and 'long' categories.

Table 47. Results statistics of training using C4.5 algorithm in basic mode (automatic/default parameter settings)

	Correct
global:	66%
short:	82%
medium:	29%
long:	14%

In Table 48 we see the results of the predictive model, trained using the C4.5 in expert mode, which implies activating windowing, pruning and significance test options. We observe a worsening of the results for 'medium' and 'long' categories, and the 'short' category probably being used as 'default'.

Table 48. Results statistics of training using C4.5 in expert mode with windowing, pruning and significance test

	Correct
global:	66%
short:	89%
medium:	11%
long:	9%

With reference to the rules (below), and Tables 43 to 48, we observe that among the rules which have been generated, there are some 'nuggets' which are precise and also correspond to a significant number of cases (as % of the training set, which was approx. 300 cases). For example, rules 1 and 2 for 'short' are good, that is they have a high precision (0.926 and 0.907, respectively), and have a significant number of cases assigned from the training set (17 and 185, respectively). Also we would identify rule 1 for 'medium' as (relatively) one of the most precise and significant. In the 'long' category, it is more difficult to find good rules.

Below we see the predictive rules generated by C4.5 for the variable 'duration_icu' defined as 3 categories: 'short', 'medium' and 'long'.

Rules for 'medium':
Rule #1 for medium:
if PROB_INF == 1
and MECH_VEN > 21

and BODY_T <= 37.6
and S_GLUCOS > 9.4
and PAO2 <= 186
then -> medium (13, 0.809)

Rule #2 for medium:
if AGE <= 62
and HEMA_F == 1
and MECH_VEN <= 21
then -> medium (7, 0.512)

Rules for long:
Rule #1 for long:
if MECH_VEN > 21
and PAO2 > 186
and PAO2 <= 235
then -> long (7, 0.82)

Rule #2 for long:
if WBC > 14.1
and S_SODIUM <= 121
then -> long (3, 0.63)

Rule #3 for long:
if H_RATE <= 74
and MECH_VEN > 21
and BODY_T > 37.6
then -> long (3, 0.63)

Rule #4 for long:
if AGE <= 35
and MECH_VEN > 21
and BODY_T > 38.1
then -> long (9, 0.61)

Rule #5 for long:
if MECH_VEN > 21
and B_UREA > 27.9
then -> long (4, 0.546)

Rule #6 for long:
if MECH_VEN > 21
and PACO2 <= 25
then -> long (5, 0.373)

Rules for short:
Rule #1 for short:
if PACO2 <= 19
then -> short (17, 0.926)

Rule #2 for short:
if C_REN_F == 0
and HEMA_F == 0
and MECH_VEN <= 21
and S_SODIUM > 121
then -> short (185, 0.907)

Default : -> short

Although the global precision of the model represented by the above rules is not high, there exist subgroups (individual rules with high precision) with a good quality. This implies that C4.5 has found common relations between the variables for these cases, and the resulting knowledge can be used (incorporated into a knowledge base, for example) after further verification testing of the rules on different data samples. For those cases for which there is an imprecise classification, that is, those which are not included in any specific rule, they typically fall into the 'default' class defined at the end of the rule set, which is set to 'short'. There are several possible reasons for this:

(a) There really does not exist a grouping between them, in terms of the variables presented to the modelling algorithm.

(b) A fuzzy relation exists with grades of membership to the (correct) crisp class and also to the assigned class. In these cases, a 'crisp' method would attempt to interpret this ambiguity by placing a certain percentage of the cases in another class. For example, it may correctly classify 50% of the cases in 'medium', place 5% of the cases in 'long' and the remaining cases would be placed in 'short' (the default), although these percentages cannot be directly interpreted as global grades of membership.

Step (ii) – reduction of dataset to eliminate outliers

In step (ii) we simplify the problem, only considering the part of the distribution less than 32.35, which is where we find the greatest concentration of cases (85%). This process step is also known as the elimination of 'outliers', in this case the very high values.

Table 49. Results statistics for rules generated for distribution of variable 'duration_icu' < 32.35 days

Qualitative categorisation

For duration of stay in icu:		
	short	< 9.19 days
	medium	>= 9.19 and < 20.42
	long	>= 20.42 and < 32.35

Results:

		categorisation of C4.5 model (predicted)		
(real) categorisation		short	medium	long
	short	96.26	3.74	0
	medium	47.82	52.17	0
	long	68.18	18.18	13.63

We observe from Table 49 that the precision has improved for all categories, especially 'medium', and the model continues assigning the default value as 'short'. The most difficult category to predict continues to be 'long'. Below we observe that C4.5 has extracted 2 good rules for 'short', 3 rules for 'medium' and only one for 'long'. The best rules are clearly for the 'short' category, with the reservation that this is used as the default, while the remaining rules require improvement.

Rules for long:
Rule #1 for long:
if PEEP == 1
and S_H_RATE <= 135
then -> long (3, 0.63)

Rules for medium:
Rule #1 for medium:
if B_UREA <= 8.6
and HEMA_F == 1
and MECH_VEN <= 22
then -> medium (4, 0.707)

Rule #2 for medium:
if SHOCK == 0
and MECH_VEN > 22
and WBC > 19
then -> medium (13, 0.587)

Rule #3 for medium:
if MECH_VEN > 22
and PEEP == 0
and S_HCO3 > 23
and ARTER_PH <= 7.53
then -> medium (20, 0.471)

Rules for short:
Rule #1 for short:
if AGE <= 76
and WBC <= 19
and S_HCO3 <= 23
and A_RES_R <= 31
and PAO2 <= 278
and ARTER_PH <= 7.52
then -> short (98, 0.926)

Rule #2 for short:
if C_REN_F == 0
and HEMA_F == 0
and MECH_VEN <= 22
then -> short (136, 0.905)

Default : -> short

Step (iii) – assignment of input variables and category ranges by medical expert.

In this step we compare the results achieved by models which use inputs selected by statistical and data mining methods, to those achieved by a model whose inputs are chosen by a medical expert. We will see that this set of input variables achived a significantly better accuracy for 'long duration' cases, than the models in steps (i) and (ii). 70% of the total set was used for training and 30% for testing.

The 34 variables selected by the medical expert from the initial 100 variables were the following:

AGE	ON_MECH	AIDS
COMA_ADM	SEP_SHOK	TERA_CH
CPR	FAIL_CARD	INT_VENT
C_REN_F	CERE_DIS	CREA_INC
PROB_INF	A_R_FAIL	RES_F
COMA_24H	LIMIT	CARD_F (OSF FIRST DAY)
SHOCK	CIRRHOS	RENAL_F
URINE	PEEP	HEMA_F
CONF_INF	VEN_CPAP	NEURO_F
FIO2	GCS_SAPS	HEPA_F
MECH_VEN	CARD_F (CHS)	OSF
	HEMA_MAL	

The ranges defined for 'duration_icu' by the medical expert were the following:

0-4 days:	short stay
5-14 days:	medium stay
>14 days:	long stay

Table 50. Results statistics for C4.5 retrained with inputs defined by expert

		Correct.	
Results:	short	51%	(56 cases)
	medium	33%	(130 cases)
	long	85%.	(499 cases)

Global precision: 63,25% success.

From Table 50 we see that the category which has best accuracy is the 'long duration' category. This is different from previous results in this section, in which the 'short duration' category had the best accuracy. We conclude that the difference has been caused by the different choice of input variables, all other things being equal. We may also say that there can exist distinct models with different input variable sets, which predict well for different duration categories.

The decision tree generated was the following:

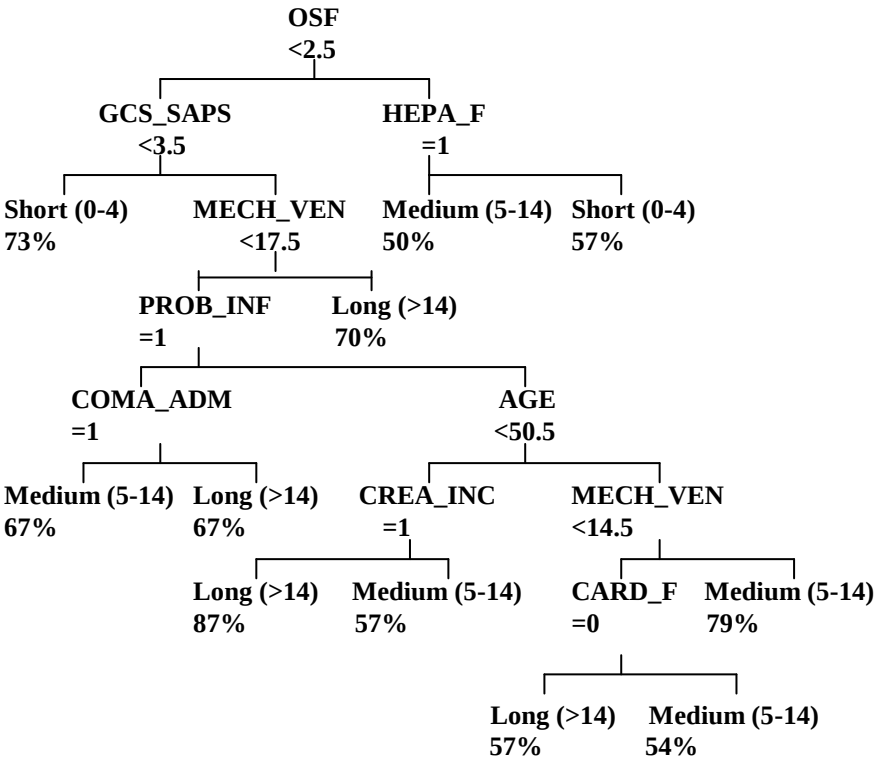


Figure 65 . Decision tree induced by C4.5 for variables selected by medical expert.

In Figure 65, we observe that the induction algorithm, as part of the induction process, has discarded the majority of the 34 input variables, and has only used the following 9: OSF, GCS_SAPS, HEPA_F, MECH_VEN, PROB_INF, COMA_ADM, AGE, CREA_INC, CARD_F. In Figure 65, the corresponding ranges are indicated between parentheses, for example, Medium (5-14), indicates a stay of between 5 and 14 days. The precision of the leaf node (decision) is indicated by the corresponding percentage, for example, Long (>14) 57%, indicates a precision of 57% for the long category and the corresponding branch stem of the decision tree.

4.1.3 Benchmarking of ID3 algorithm on the ICU dataset.

This section consists of four groups of tests (i) to (iv), using the ID3 algorithm for data modelling. In test group (i) all 100 variables as used as input, and, in contrast to test groups used in Section 4.1.2, two output variables are tried, 'duration_hosp', which is the duration of stay in days of the patient in the hospital, and 'duration_icu', which is the duration of stay in days of the patient in the ICU unit. Both variables are defined as numerical integers. The modelling algorithm used is ID3. In test group (ii) a reduced set of 15 selected variables is used as input. In test group (ii) the results of ID3 are compared with a model trained using C4.5, with the same data, but predicting 'duration_ICU' as a categorical value. In test group (iv), we try to predict variable 'b_urea' as a continuous value using ID3.

(i) The following subgroup of tests use ID3 in default mode (default windowing and pruning) to predict the variable 'duration_ICU', using all 100 variables as input.

Table 51. Results of variation of training set size on error rate - prediction of 'duration_ICU' using all variables as input

				Error.			
% Train	Min	Max	Mean	Absolute Mean	Standard Deviation	Linear Correlation	Ocurr. Test
10	-23	111	2.06	7.16	12.71	0.09	983
20	-45	110	1.19	7.08	12.85	0.25	875
30	-74	108	0.41	6.47	12.33	0.43	768
40	-75	109	1.76	6.62	12.31	0.38	652
50	-75	109	0.79	7.01	14.04	0.32	545
60	-58	60	1.26	6.46	12.1	0.46	423
70	-60	45	0.49	6.46	11.45	0.50	322
80	-74	59	-1.26	6.84	12.51	0.40	205
90	-61	59	-0.9	7.45	13.41	0.31	101
Training with Neural Network.							
30	-11	108	1.01	6.75	11.75	0.32	768

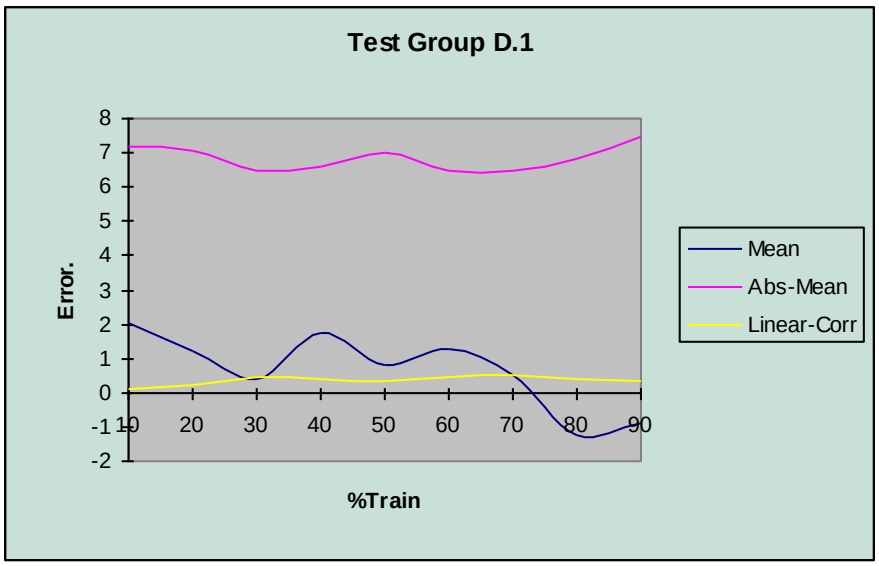


Figure 66. Results of variation of training set size on error rate - prediction of 'duration_ICU' using all variables as input

From Table 51 and Figure 66, we can see that with the ICU data, progressive increase of the training set size with respect to the total dataset has little effect on the overall precision. We see a local minimum at 30%, and the x-axis is intersected (zero error) at 74%. In Table 52 we also see the result of executing a neural network with the same inputs and data in the case of test percentage equal to 30%. The result is slightly worse than ID3 (6.75 compared to 6.47) for the absolute mean of the error, and in the case of linear correlation (0.32 compared with 0.43). For standard deviation, the NN shows a reduction relative to ID3, being 11.75 compared to 12.33 for ID3. This could indicate that the NN is achieving a greater 'smoothing' effect on the error, although the aggregate error is slightly greater. This could be convenient if what we wish is a more stable model whose error can be maintained between two predefined upper and lower limits.

(ii) The following subgroup of tests uses ID3 in default mode (default windowing and pruning) to prediction the variable 'duration_icu' with a reduced set of 15 input variables, selected by statistical methods (correlation analysis, tree pruning, …) and medical expert advice.

Table 52. Results of variation of training set size on error rate - prediction of 'duration_ICU' using reduced set of variables as input

			Error.				
% Train	Min	Max	Mean	Absolute Mean	Standard Deviation	Linear Correlation	Ocurr. Test
10	-46	110	1.47	7.45	13.05	0.10	983
20	-44	93	0.35	7.66	13.33	0.22	872
30	-76	94	0.97	7.09	13.01	0.32	768
40	-46	108	1.08	6.53	12.32	0.42	652
50	-60	109	0.13	6.93	12.83	0.42	545
60	-45	95.4	1.40	6.42	12.15	0.41	423
70	-62	49	-0.36	6.8	11.9	0.46	322
80	-38	47	-0.29	5.79	9.86	0.52	205
90	-28	40	-1.22	5.85	9.51	0.60	101
Test with Neural Network.							
30	-25	103	1.59	5.76	10.292	0.55	768

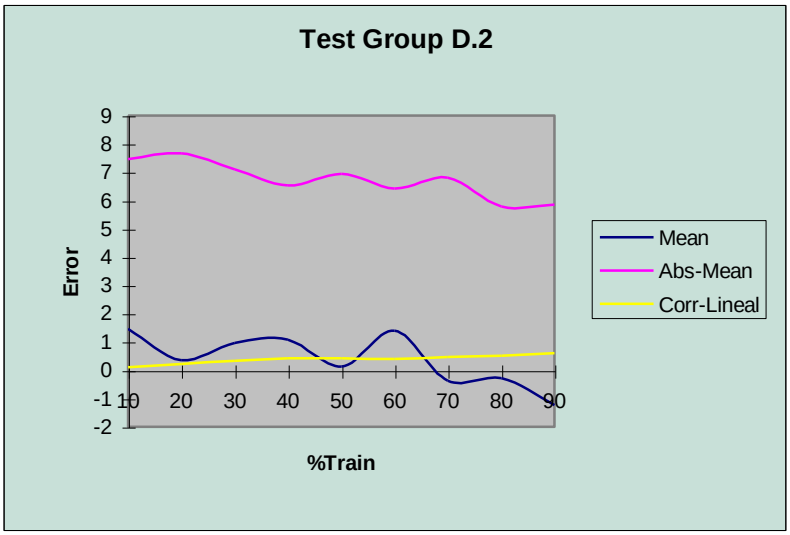


Figure 67. Variation of training set size of error rate - prediction of 'duration_ICU' using reduced set of variables as input

From Table 52 and Figure 67, we can see that with the ICU data, progressive increase of the training set size with respect to the total dataset, as in the case of test subgroup (i), has little effect on the overall precision. We see a local minimum this time at 50%, and the x-axis is intersected (zero error) at 67%. In Table 52 we also see the result of executing a neural network with the same inputs and data in the case of test percentage equal to 30%. In contrast to test subgroup (i), in which all the 100 variables were used as input, the result for the reduced variables as input is slightly better than ID3 (5.76 compared to 7.09) for the absolute mean of the error, and in the case of linear correlation (0.55 compared with 0.32). For standard deviation, the NN also shows a reduction relative to ID3, being 10.29 compared to 13.01 for ID3. This could indicate that the NN is benefiting from the reduction in 'noise' due to the reduced number of variables. Compared with test subgroup (i), (ii) also shows an overall improvement in terms of absolute mean of the

error, standard deviation and linear correlation. This again would be expected due to the increased quality of information input to the model (less inputs and greater correlation of each input to the output).

(iii) Comparison between ID3 and C4.5 algorithms: having tried to predict 'duration_icu' as an integer variable (duration in days), we now try to discretise it and predict it as a categorical ordinal variable. We say that it is categorical ordinal, because although it is defined in terms of three symbolic categories, these categories can be ordered with respect to each other, from shortest to longest.

The categorisation of the variable 'duration_icu' followed the same process as in Section 4.1.2.3. (step(i)), in which '0' indicates 'short stay', which is less than 10 days duration in the ICU, '1' indicates 'medium stay', which is defined as between 10 and 20 days duration in the ICU, and '2' indicates 'long stay', which is defined as more than 20 days duration in the ICU.

The following are the results of training a model using the C4.5 algorithm, with windowing and pruning activated.

{0,1}	correct	78%
	incorrect	22%
{0}	correct	89%
	incorrect	11%
{1}	correct	38%
	incorrect	62%
{2}	correct	41%
	incorrect	59%

The results show a good precision for 'short stay', and inadequate precisions for 'medium stay' and 'long stay'. We note that the majority of the cases are less than 10 days duration, and thus the other categories have a relative lack of example cases (although we 'balance' the sample we still lack the diversity of the real data).

The same data was presented to ID3 as in the previous test with C4.5, with the exception that the categorical variable was interpreted as an integer variable, with values {0,1,2}. This is possible because the categorical variable is ordinal, and therefore it has meaning to predict it as a numeric integer result.

Statistics for the Error Rate of the ID3 model

Min	-2
Max	2
Mean	0.14
Absolute Mean	0.24
Standard Deviation	0.564
Linear Correlation	0.588
Occurrences	761

In this case, we can see a reasonable correlation, relative to previous tests (i) and (ii), and to the C4.5 test. The best neural network model gave a correlation of 0.55, compared to this model's 0.59. Notwithstanding, as a percentage of the max value, the absolute mean of the error (12% compared to 5.6% for the NN), and the standard deviation of the error (28.2% compared to 9.9% for the NN) are inferior results.

(iv) Composition and error rate of a data subset

In this section we ran ID3 with the same input variables as in Section 4.1.2.3 step (iii), that is, selected by a medical expert rather than by statistical analysis and data mining. Once run, we saved the output value (DUR_HOS), and the error (real DUR_HOS – predicted DUR_HOS) and then we selected a subset of the data , by choosing criteria which seem 'interesting', such as (see Figure 68), a very high incidence of tubing/ventilar (INT_VENT), mechanical ventilation (ON_MECH, VEN_CPAP, MEC_VEN).

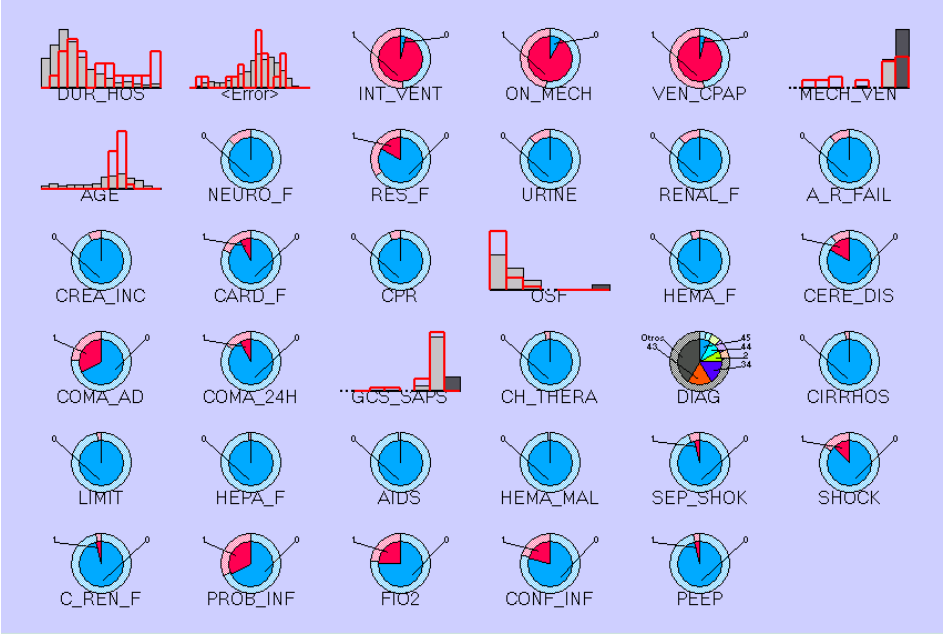


Figure 68. Graphical representation of distributions of input variables, output variable, and error in the selected data subset.

As we see from AGE, the subset also corresponds to a specific age group, from 61 to 70 years.

Figure 68 has the variables ordered by the 'chi-squared' value of the variables values in this data subset, compared with the variables values in the complete dataset. Thus a greater the difference between the distribution in this data subset and the distribution in the complete dataset will produce a greater chi-squared value. In Figure 68, after the output variable (DUR_HOS) and the error, we see the variable with the highest chi-squared ranking is INT_VENT, with a chi-square value of 0.791, followed by ON_MECH with 0.777, VEN_CPAP with 0.753, MECH_VEN with 0.251, AGE with 0.173, and so on. In order to rank all the variables with the same chi-square ranking, the numerical variables, such as MECH_VEN and AGE have been categorised by calculating quantiles. In the Figure 68 we see two types of graphical representation of the variables, histogram for the numerical variables and Pie Chart for the categorical variables. In the case of the histograms, the distribution of that variable for the total population is seen in grey, whereas the distribution for this subset of data is shown in red. Thus we can see, for example, in the case of 'organ systems failing' (OSF), that the distribution for this subset is weighted towards a lower number of OSF, with respect to the distribution of the whole dataset.

In the case of the Pie Charts, the inner area is the distribution of the categories of the given variables with respect to this subset, whereas the outer ring is the distribution in the complete dataset. Thus we can see, for example that the categorical variables INT_VENT, ON_MECH and VEN_CPAP have an incidence of approximately 50% in the complete dataset, whereas in this subset their incidence is greater than 95%. Another interesting categorical variable is DIAG (diagnostic code) which has a high incidence of diagnostic codes 43 (Surgery GI due to Neoplasm) and 34 (Craniotomy due to neoplasm).

Given that we are studying the results of the predictive model for DUR_HOS, we are especially interested in the distribution of the predicted value for DUR_HOS which can be seen above as the first variable top-left, and the error distribution, which is the variable next to it, and which represents the difference between the real DUR_HOS and the predicted value (DUR_HOS - $DUR_HOS).

In the Figure 69 below we see a 'zoom' from the previous Figure 68 on the histogram of the values for the predicted variable DUR_HOS. We see shaded the distribution of the values for the whole dataset, and in red we see the distribution for the values for this sub-dataset. We observe that there is a tendency for longer duration cases, with a peak between 60 and 65 days.

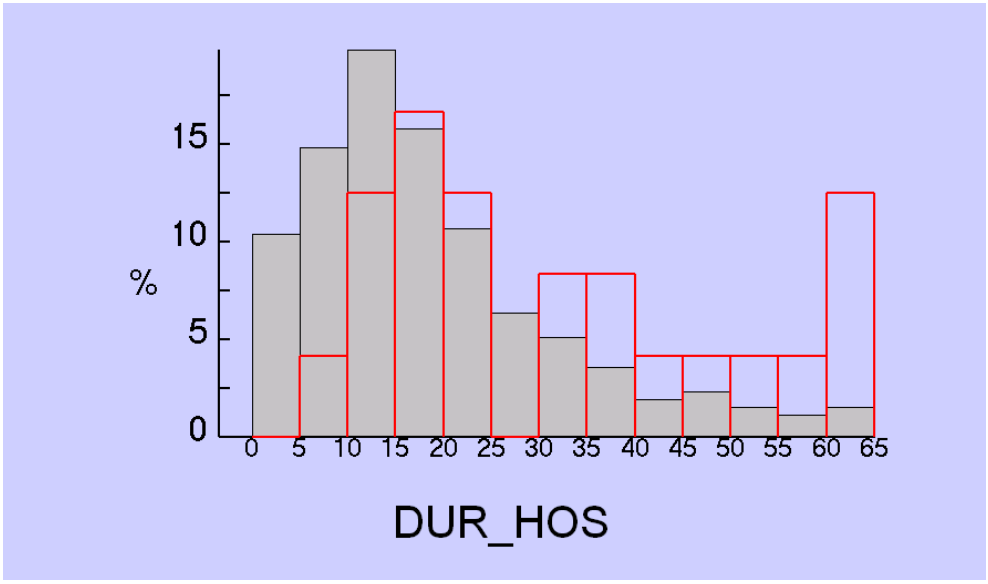


Figure 69. Histogram of the distribution of the output variable DUR_HOS (duration of stay in hospital) for the selected data subset

In Figure 70 below we see a 'zoom' from the previous Figure 68 on the histogram of the values for the error variable. As before, we see shaded the distribution of the values for the whole dataset, and in red we see the distribution for the values for this sub-dataset. We observe that there are deviations from the distribution of the complete dataset, between 3.25 and 0.0 days, between 16.25 and 19.50 days, and between –29.25 and –22.75 days.

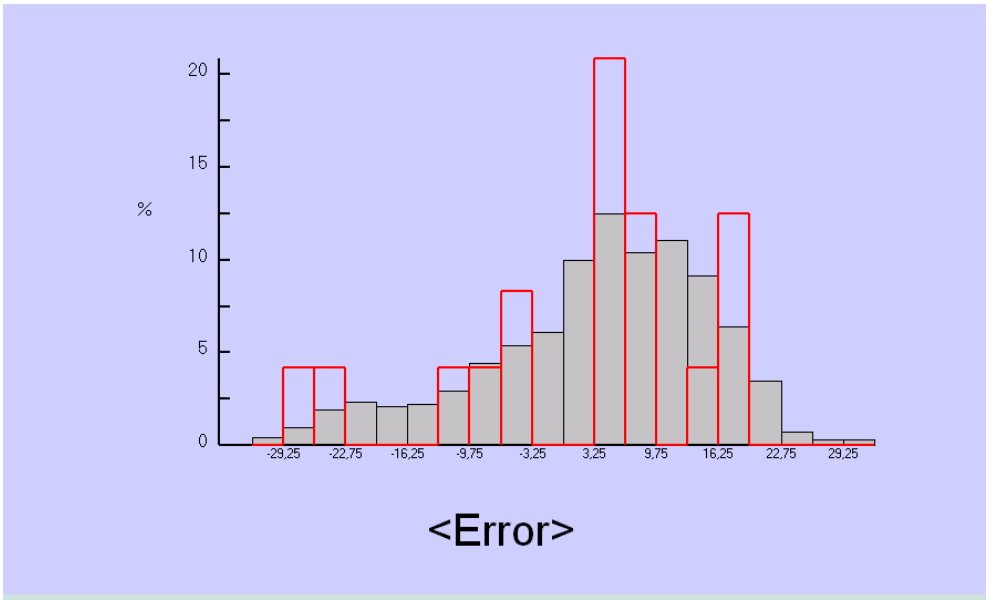


Figure 70. Histogram of the distribution of the error (real duration in hospital – predicted duration in hospital) for the selected data subset

4.1.4 Clustering with Kohonen Neural Net algorithm

The objective of applying this algorithm to the data is to try to establish 'homogeneous groupings' of the data, each of which could be trained as a separate model, and to look for trends between clusters and variables which can help in the definition of significant factors. We can also compare the results of this unsupervised learning method, with the supervised learning as represented by C4.5 and ID3.

In Figure 71 (below) we can see that the Kohonen SOM clustering algorithm has achieved a reasonable clustering of cases, in the sense that it has been able to distinguish the clusters in terms of the variable 'vital_state_icu'. The cases shown in red are the fatalities. We can see that some clusters have no fatalities (i), whereas others have a majority of fatalities (ii) and some have a mixture of fatalities and non-fatalities (iii). We could say that for case (iii) where the fatalities and non-fatalities are mixed, the clustering has failed to distinguish. Nevertheless, each generated cluster would have to be studied individually to see the characteristics and distributions of the other input variables in that cluster.

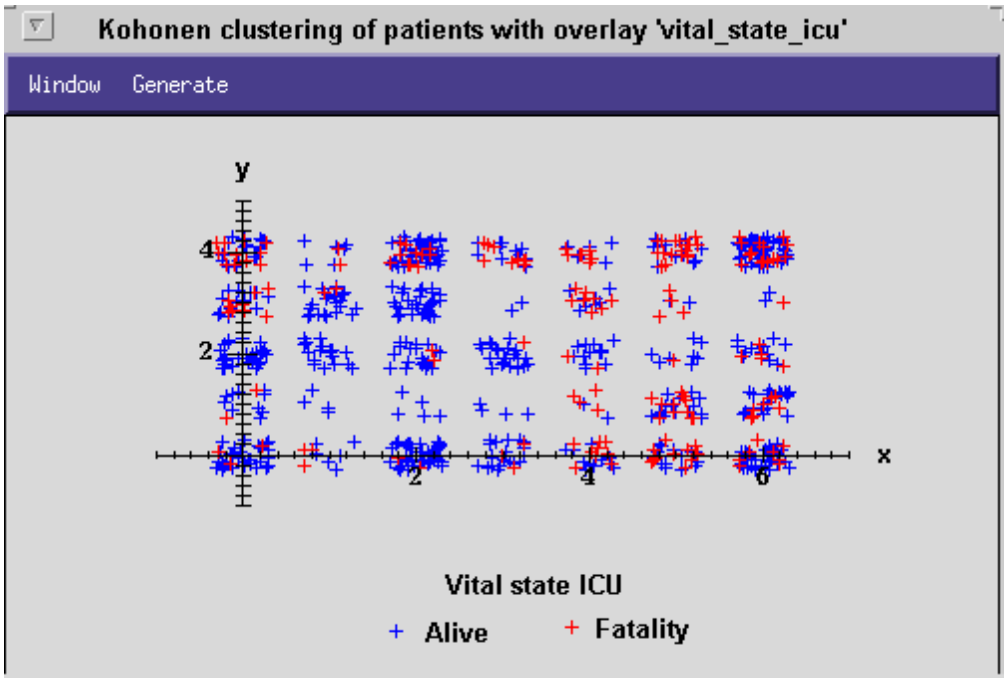


Figure 71. Clustering with reduced variable set (<u>without</u> duration_hos, duration_icu or vital_state_icu as inputs) and 'overlay' of variable 'vital_state_icu'

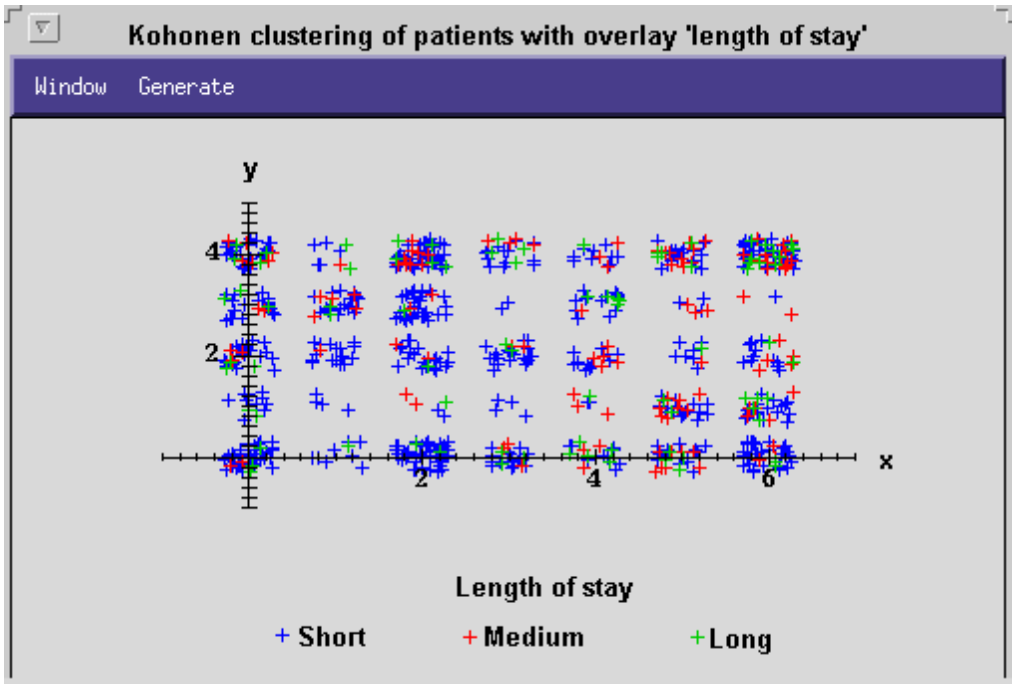


Figure 72. Clustering with reduced variables (<u>without</u> duration_hos, duration_icu or vital_state_icu as inputs) and overlay of 'duration_icu' as a discrete variable

In Figure 72, we see that, although the Kohonen SOM has created some clusters all blue ('short stay') and other clusters with a high presence of red ('medium stay') and green ('long stay'), it has not been able to clearly distinguish the medium and long stay cases. Notwithstanding, the relative frequencies have to be studied of each 'duration_icu' value in each cluster to identify more subtle differences, for example the cluster with the highest ratio of 'long stay' relative to 'medium stay'. Note that Figures 71 and 72 are of the same clustering result, but with 'overlay' of different variables.

The data set of Table 53 (below), illustrates a cluster in the Kohonen SOM output shown in Figures 71 and 72, which contains a mixture of values for 'duration_icu' (qualitative version) and 'vital_state_icu'. That is to say, a cluster which has not been able to distinguish these variables. The typical methodology with this sub-dataset would be to study each variable to look for underlying traits, to which end an induction model could be trained, for example, only using the cases in this cluster. The induced rules and tree could then reveal underlying structure and relationships between the variables and the data. From simple inspection, we can see, for example, that previous health state 'p_h_state' is always equal to '1',with the exception of just one case. There is also a constancy of values for variables 'osf', 'type_admis', and 'a_r_failure'. Before reaching false conclusions, the distribution of these variables must also be established in the complete dataset (for example, 'osf' may be equal to '1' in the complete dataset, therefore it loses its significance in this cluster. Also the reverse may be true, a variable which is fairly constant in the complete dataset, or in other clusters, has a high variability in this cluster: for example, 'lines'.

Table 53. Cases corresponding to Cluster {6,4} (top rightmost cluster) of the Kohonen plot, with 'overlays' of variables 'duration_icu' and 'vital_state_icu'

coma admis	type admis	malig	icu ser	lines	a_r_ failure	body_ temp	b_ urea	p_h_ state	copd	a_ fio2	osf	vital_ state_ icu	dur_ icu	dur_ hos	stay
0	3	0	1	4	0	38.0	4.0	1	0	1.0	1	0	33	51	2
0	3	0	0	1	0	38.0	19.0	1	0	0.5	1	0	5	14	0
0	3	0	0	3	0	40.0	5.0	1	0	0.5	1	0	6	21	0
0	2	1	1	7	0	39.0	11.0	1	0	0.8	1	0	57	90	2
0	3	0	0	1	0	37.0	6.0	1	1	0.3	1	0	5	13	0
0	3	0	0	3	0	39.0	5.0	1	0	0.21	1	0	5	13	0
0	3	0	1	6	0	37.0	2.0	1	0	0.3	1	0	3	8	0
0	3	0	1	4	0	36.7	3.0	1	0	0.3	1	0	5	21	0
0	3	0	1	6	0	38.9	3.8	1	0	0.3	1	0	12	17	1
0	3	0	1	6	0	37.0	3.0	1	1	0.4	1	0	5	13	0
0	3	0	1	6	0	39.5	9.8	1	0	0.3	1	0	7	16	0
0	3	0	1	3	0	37.3	1.7	1	0	0.6	1	0	7	13	0
0	3	0	1	4	0	36.4	5.7	1	0	0.5	1	0	10	16	1
0	3	0	1	4	0	37.3	13.7	1	1	0.4	1	0	4	15	0
0	3	0	1	4	0	37.5	7.0	1	1	0.5	1	0	9	24	0
0	3	0	1	6	0	38.1	7.5	1	1	0.7	1	0	4	21	0
0	3	0	1	3	0	39.2	9.7	1	1	1.0	1	1	28	28	2
1	3	1	0	6	0	37.1	6.8	1	0	0.4	1	0	6	13	0
1	3	0	1	3	0	35.8	6.7	1	0	0.2	1	0	7	7	0
1	3	0	1	3	0	38.3	6.5	1	0	0.3	1	0	6	11	0
0	3	0	1	4	0	35.7	6.0	1	0	0.4	1	0	13	51	1
1	3	0	1	6	0	37.9	3.7	1	0	0.5	1	0	7	16	0
0	3	0	1	3	1	39.5	14.9	1	1	0.6	1	0	4	10	0
0	3	1	1	2	0	37.5	11.0	1	0	0.5	1	0	14	14	1
0	3	0	1	7	0	37.6	5.0	1	0	1.0	1	1	22	22	2
0	3	0	1	3	0	37.0	18.7	1	0	0.3	1	0	3	9	0
1	3	0	1	6	0	37.4	4.8	1	0	0.5	1	0	7	21	0
0	3	0	1	3	0	39.5	2.5	1	0	0.5	1	0	6	21	0
0	3	0	1	5	0	36.5	7.0	1	1	0.4	1	0	29	48	2
0	3	0	1	5	0	37.1	3.5	1	0	0.2	1	0	11	16	1
1	3	0	1	5	0	38.5	5.0	1	0	0.5	1	0	18	19	1
0	3	0	1	4	0	37.0	4.3	1	0	0.8	1	0	8	11	0
0	3	1	1	8	0	35.0	11.7	1	0	0.6	1	0	1	11	0
0	3	0	0	4	0	37.0	14.0	1	0	0.5	1	1	4	4	0
1	3	0	0	0	0	36.5	9.5	1	0	0.4	1	0	2	2	0
0	3	0	1	6	0	37.7	5.2	1	0	0.6	1	0	6	9	0
0	3	0	1	5	1	35.0	14.2	1	0	0.5	1	0	9	9	0
0	3	0	1	4	1	38.6	43.0	1	0	0.4	1	0	9	92	0
1	3	0	0	5	0	34.8	8.5	1	0	0.5	1	0	16	24	1
0	3	0	0	6	0	38.6	8.0	1	0	1.0	1	1	22	22	2
0	3	0	0	3	1	38.0	23.0	1	0	0.5	1	1	21	21	2
0	3	0	0	3	0	37.6	3.6	1	0	0.35	1	0	6	20	0
1	3	0	0	5	0	37.8	6.0	1	0	0.4	1	0	9	9	0
1	3	0	0	5	0	38.1	7.9	1	0	0.35	1	1	4	4	0
1	3	0	1	5	0	37.2	17.8	1	0	0.5	1	0	59	59	2
1	3	0	1	3	0	38.2	5.9	1	0	0.4	1	0	5	41	0
0	3	0	1	3	0	37.0	22.0	1	0	0.3	1	0	4	4	0
0	3	0	1	3	0	38.7	9.0	1	0	0.5	1	0	7	13	0
1	3	0	1	5	0	35.0	6.0	1	0	0.5	1	1	4	4	0
1	3	0	1	5	0	39.0	4.3	1	0	0.5	1	1	4	4	0
0	3	0	1	3	0	37.1	5.0	1	1	0.3	1	0	3	15	0
1	3	0	1	8	0	38.7	12.0	1	0	1.0	1	0	38	55	2
1	3	0	1	8	0	36.5	9.8	1	1	0.4	1	0	23	41	2
1	3	1	1	6	0	38.0	3.3	1	0	0.3	1	0	15	31	1
0	3	0	1	7	0	39.2	10.9	4	0	0.6	1	0	12	27	1
0	3	0	1	4	1	37.5	41.7	1	0	0.6	1	0	9	21	0
0	3	0	1	4	1	36.0	36.0	1	0	0.6	1	0	19	42	1
1	3	0	1	5	0	40.0	11.0	1	0	0.6	1	1	11	11	1
1	1	0	1	2	0	35.8	9.0	1	1	1.0	1	0	5	12	0
0	3	0	1	2	0	36.8	5.0	1	0	0.4	1	1	5	5	0
1	3	0	1	1	0	38.0	5.0	1	0	0.8	1	1	11	11	1
1	3	0	1	2	0	36.0	10.0	1	0	0.3	1	1	58	58	2
0	3	0	1	5	1	38.4	18.0	1	0	0.4	1	0	14	28	1
0	3	0	1	6	0	36.0	10.0	1	0	0.4	1	1	6	6	0
1	3	0	1	7	0	35.5	10.0	1	0	0.5	1	1	11	11	1
0	3	0	1	5	0	36.2	3.0	1	0	0.6	1	1	4	4	0

4.1.5 Application of Hartigan's 'joining algorithm' to the ICU data, using 'crisp' and 'fuzzy' covariances as input

Having benchmarked the C4.5, ID3 and NN algorithms against the ICU data, we now progress to test the combination of the Hartigan Joining Algorithm using as input first a 'crisp' covariance matrix, as detailed in (ii) and a 'fuzzy' covariance matrix as detailed in (iii) to indicate the grade of relation between the variables.

The fusion process starts with the generation of a covariance matrix, as detailed in (i). The following cases have been tested: standard covariances generated by SPSS from numerical data; and 'fuzzy' covariances generated by the modified version of Gustafson's algorithm [Gustafson79].

(i) Summary and comparison of fuzzy and crisp covariance results

In Tables 54 and 55 we see the fuzzy and crisp covariances, respectively, of some of the variables which describe the patients. We observe that while some of the correlations between variables maintain their respective order, such as the pairs: {acute renal failure, probable infection}, {acute renal failure, vital state}, {acute renal failure, coma24hrs}), others do not: {acute, renal, failure, cardiac failure},{vital state, probable infection}. We conclude that the fuzzy covariances, although derived from the same data, produce different results. The topology of the fuzzy partitions, membership grade values of the cases, and distance metrics norms are some of the factors which distinguish fuzzy and crisp covariances. We conclude that the fuzzy covariances provide an alternative method for grouping clinical variables, forming groups which make clinical sense and which were not formed by the crisp covariances.

The 'fuzzy covariances' of the 17 variables detailed previously were calculated using the algorithm described in Section 2.2.6 and Section 3.1.4 of the thesis. Part of the resulting matrix is shown in Table 54. We note that the fuzzy covariance of each variable with itself (the diagonal) produced a large positive number in each case, which has been assigned to '1' to improve clarity and given that it is not subsequently used in the factor analysis or the fusion process.

Table 54. Fuzzy Covariance Matrix produced for some of the 'Admissions' variables

	Acute renal failure	**Cardiac failure**	**Probable infection**	**Vital state**	**Coma 24hrs**
Acute renal failure	1*	14.867	1.432	1.286	1.807
Cardiac failure	14.867	1	1.437	1.289	1.800
Probable Infection	1.432	1.437	1	12.531	0.799
Vital state	1.286	1.289	12.531	1	0.751
Coma 24hrs	1.807	1.800	0.799	0.751	1

* The value 1 has been assigned to the diagonal and it is not used in the fusion process.

The 'crisp covariances' of the 17 variables detailed previously were calculated using the standard covariance function of SPSS. Part of the resulting matrix is shown in Table 55.

Table 55. Crisp Covariance Matrix produced for some of the 'Admissions' variables

	Acute renal failure	Cardiac failure	Probable infection	Vital state	Coma 24hrs
Acute renal failure	1	0.203	0.106	0.346	0.111
Cardiac failure	0.203	1	0.201	0.324	0.208
Probable Infection	0.106	0.201	1	0.079	-0.122
Vital state	0.346	0.324	0.079	1	0.269
Coma 24hrs	0.111	0.208	-0.122	0.269	1

(ii) Application of Hartigan's 'joining algorithm' to the ICU Data, using *'crisp'* covariances as input

The dataset was analysed in SPSS and the crisp covariances calculated for all 17 variables and a representative sample of 100 cases of 'hospital admissions'. It was necessary to convert the binary values to numeric in order for SPSS to process them; this conversion may have some statistical implications which could be evaluated at a later date. SPSS produced a correlation matrix from the raw data, and this was given as input to the fusion algorithm. We used the SPSS option: 'Statistics->Correlate->Bivariate' with a 'two-tailed Pearson Correlation Coefficient'. This generated a correlation matrix with dimension 17 x 17 (17 being the number of variables). The variables are summarised in Table 56 (below).

Table 56. Variables given as input to Hartigan's 'joining algorithm'

Short Name (as used in Figure 73)	**Name**	**Data Type**	**Description**
OS	OSF	Numeric	Number of organ systems failing, calculated by computer program.
PH	P_H_STAT	Categorical	Previous Health State. {1,2,3,4}
DI	DEAD_ICU	Binary	Vital State ICU {0=alive, 1=dead}
DH	DUR_HOS	Numeric	Calculated duration in the hospital from the moment of admission to the ICU.
AR	A_R_FAIL	Binary	Grave renal failure.
BU	B_UREA	Numeric	Concentration of Urea in the blood.
CA	CARD_F	Binary	Cardiovascular failure.
CO	COMA_24H	Binary	In coma or deep shock at 24 hours after admission.
CR	CREA_INC	Binary	Creatinine > 2.0mg/dl (176.8μMol/l) during first 24 hours.
FI	FIO2	Binary	$FIO_2 > 0.50$ during the first 24 hours
IN	IN24HRS	Binary	Stay in the ICU of 24 hours or more.
NE	NEURO_F	Binary	Neurological failure (excluding sedation)
PR	PROB_INF	Binary	Probable infection in the moment of admission to the ICU.
RE	RENAL_F	Binary	Renal failure.
RF	RES_F	Binary	Respiratory failure {1=yes, 0=no}
SE	SEXO	Binary	{1=male, 0=female}
TY	TYPE_ADM	Categorical	Type of patient {1=Emergency surgery,2=planned surgery,3=without surgery}

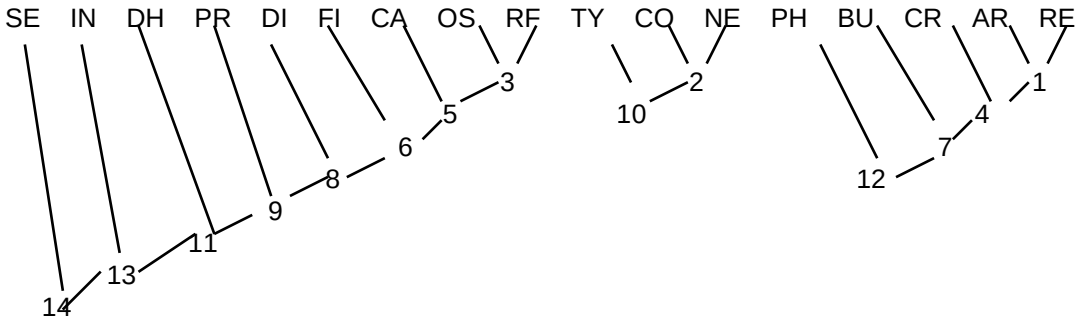


Figure 73. Tree of fusions produced by Hartigan's 'joining algorithm' with crisp covariances

The numbers in Figure 73 indicate the number of the new factor created and the order in which it has been created. We observe that the 17 original variables have been 'fused' into 3 new 'components', represented by the numbers 14, 10 and 12, respectively. Note that the first fusion is 1 on the extreme right hand side (AR and RE), followed by 2 (CO and NE), and so on. Now we can analyse the manner in which 'fusion' has grouped the variables, comparing it to ID3, C4.5, Neural Net, SPSS and the fuzzy fusion (see next section). The detailed sequence, with comments, of fusions for the 17 pre-selected variables, is the following:

Joining Sequence, 'crisp covariances': with reference to Table 56 and Figure 73, the input data was all considered as numeric; the binary values being defined as 1 or 0. It was given to SPSS which calculated the covariances between variables. This produced a matrix which was given to the Hartigan algorithm, which produced the following order of fusions:

> The first fusion produced was between 'Acute Renal Failure' and 'Renal Failure'. That there is a high correlation between these two variables is very reasonable as both refer to renal failure. We will call this *factor one*.
>
> The second fusion produced was between ' In coma or deep shock at 24 hours after admission' and 'Neurological failure (excluding sedation)'. We will call this *factor two*.
>
> The third fusion produced was between 'Respiratory Failure' and 'Number of Organ Systems failing'. We will call this *factor three*.

With these first three fusions, the algorithm has already identified the base of the three factors that it has identified among these 17 variables. It proceeds to join the other variables to these three bases, forming an 'inverted pyramid' structure.

> The fourth fusion returns to factor one and joins on 'Creatinine > 2.0mg/dl (176.8μMol/l) during first 24 hours'. It was confirmed by the medical expert that Creatinine level is associated with renal failure, because if the kidney fails, the Creatinine level may go up from its normal value of approximately 1mg/dl, to 9 or 10mg/dl, which would be a pathological level. This is due to the accumulation of Creatinine in the blood, because it is the correct functioning of the kidney which normally keeps it at a stable level.
>
> The fifth fusion goes to factor three and joins on 'Cardiovascular Failure'. This seems to follow the tendency of this factor to identify organ system failures.
>
> The sixth fusion stays with factor three and joins ' $FIO_2 > 0.50$ during the first 24 hours'. It will have to be checked with a medical expert if FIO_2 level is associated with organ system failure in general, or cardiovascular or respiratory failure in particular.
>
> The seventh fusion goes back to factor one and joins 'Concentration of Urea in the blood'. It will have to be checked with a medical expert if Urea concentration is associated with renal failure and Creatinine level.

The eighth and ninth fusions go to factor three and join 'Vital State ICU' and 'Probable infection in the moment of admission to the ICU', respectively. This seems to follow the tendency of this factor to identify global states.

The tenth fusion goes to factor two and joins 'Type of patient'. With this fusion, factor two is complete (the algorithm does not consider it again for joining.

The eleventh fusion is to factor three, joining 'Calculated duration in the hospital from the moment of admission to the ICU'. This seems to follow the tendency of this factor to identify global states.

The twelfth fusion is to factor one and joins 'Previous Health State'. With this fusion, factor one is complete (the algorithm does not consider it again for joining.

The last two fusions, thirteen and fourteen, are to factor three, and join 'Stay in the ICU of 24 hours or more' and 'Sex', respectively. Again, factor three has grouped 'general states' and with these two fusions, factor three is complete, there are no more variables and the algorithm terminates.

It could be that the last variables joined are added onto factor three because there is nowhere else to join them. To avoid this, a 'significance threshold' could be incorporated, below which the variable is discarded.

We observe that three factors have been built, one significantly more complex than the other two.

Factor one seems to be specifically for renal cases. **Factor two** seems to group neurological cases. **Factor three** appears to identify global states or temporal data.

The following summarises the composition of the factors by variables.

Factor One
'Acute Renal Failure' + 'Renal Failure' + 'Creatinine level' + 'Concentration of Urea in blood' + 'Previous Health State'.

Factor Two
'In coma or deep shock at 24 hours after admission' + 'Neurological failure' + 'Type of patient'.

Factor Three
'Respiratory Failure' + 'Number of Organ Systems Failing' + 'Cardiovascular Failure' + 'FIO_2 level' + 'Vital State ICU' + 'Probable infection in admission to ICU' + 'Duration in hospital from moment of admission to ICU' + 'Stay in ICU $\geq$ 24 hours' + 'Sex'.

Observations: one possible test for the joining sequences and placement would be to create one or two artificial random variables, with no real association with the data, and to see what the algorithm does with them. We have to take into account that the categorical and binary data has been considered as numeric, so that SPSS could calculate the covariances. Using respective algorithms to calculate covariances, respecting the types of the variables, one could study the changes in covariance and thus the changes in fusion order and grouping. In the same manner, we have to take into account that the fuzzy data has been considered as crisp. Using a fuzzy covariance calculation, we can study the changes in covariance and thus the changes in fusion order and grouping. The more we respect the natural form of the data, and we do not lose this information, the more precise will be the fusion order and grouping and more truly the real underlying nature of the data will be reflected.

(iii) Application of Hartigan's 'joining algorithm' to the ICU Data, using *'fuzzy'* covariances as input

The fuzzy covariances of the 17 pre-selected variables were calculated using the algorithm as detailed in Section 2.2.6 and Section 3.1.4 of the thesis. The resulting matrix (see Table 54) was presented as input to Hartigan's fusion algorithm. The fusion process with 'fuzzy covariances' produced four factors, one more than with '*crisp* covariances'. The initial variables chosen for factor one were 'Acute Renal Failure' and 'Cardiac Failure'. The initial variables chosen for factor two were 'Probable infection at the moment of admission to the ICU' and 'Vital state on leaving the ICU'. The initial variables chosen for factor three were 'Coma or profound sleep at 24 hours' and 'Renal Failure' and the initial variables chosen for fact or four were 'Creatinine > 2.0mg/dl (176.8μMol/l) during the first 24 hours' and 'Previous health state'.

The four final factors were:
Factor 1: 'Neurologic Failure' + 'Coma 24 hours' + 'Renal Failure'.
Factor 2: 'Sex'+'FIO_2 level' + 'Probable Infection' + 'Vital State'.
Factor 3: 'Blood Urea' + 'Patient Type' + 'Number of Organ Systems Failing' + 'Stay 24 hours' + 'Increment Creatinine' + 'Previous Health State'.
Factor 4: 'Duration of stay in hospital in days' + 'Respiratory Failure' + 'Acute Renal Failure' + 'Cardiac Failure'.

In contrast to the '*crisp* fusion', the 'fuzzy fusion' has placed organ failure states in Factors 1 and 4. 'Renal Failure' and 'Acute Renal Failure' are separated, while the *crisp* fusion grouped them. 'Hospital stay in days' has been associated with some of the organ system failures, the same as in the *crisp* factor.

4.1.6 Applying Fuzzy c-Means to the ICU data

The ICU data was prepared for input to fuzzy c-Means: 100 cases were used selected as a homogeneous random subset of the complete ICU dataset. The same subset of 17 variables was used as previously:

Variable 1: Sex
Variable 2: Type of Admission
Variable 3: Probable infection on admission to ICU
Variable 4: Coma at 24 hours after admission to ICU
Variable 5: Fio2
Variable 6: Crea_Inc
Variable 7: A_R_Fail
Variable 8: B_Urea
Variable 9: Previous health state
Variable 10: Respiratory Failure
Variable 11: Cardiac Failure
Variable 12: Renal Failure
Variable 13: Neurological Failure
Variable 14: OSF
Variable 15: Dead_ICU
Variable 16: Dur_Hos
Variable 17: In24hrs

The fuzzy c-Means processing was as follows, for number of clusters equal to three:

Number of clusters = 3, icon=1, exponent=2
Iteration=1, maximum error=0.7459
Iteration=2, maximum error=0.2889
Iteration=3, maximum error=0.4464
Iteration=4, maximum error=0.3979
Iteration=5, maximum error=0.3378

Fstop	1-Fstop	Entropy	Payoff
0.661	0.339	0.590	7862.106

Table 57. Fuzzy c-Means: cluster centres v[i][j]

Variable	*Cluster 1 , v[1][n]*	*Cluster 2, v[2][n]*	*Cluster 3, v[3][n]*
Sex, v[n][1]	0.5746	0.5472	0.6094
Type of Admission, v[n][2]	2.2262	2.3164	2.4549
Probable infection on admission to ICU, v[n][3]	0.3532	0.2440	0.1812
Coma at 24 hours after admission to ICU, v[n][4]	0.1906	0.1138	0.0247
Fio2, v[n][5]	0.5819	0.3664	0.4309
Crea_Inc, v[n][6]	0.0891	0.0991	0.0953
A_R_Fail, v[n][7]	0.0793	0.0758	0.0766
B_Urea, v[n][8]	8.4657	8.4195	9.3603
Previous health state, v[n][9]	1.4922	1.8166	1.5818
Respiratory Failure, v[n][10]	0.1847	0.2932	0.3140
Cardiac Failure, v[n][11]	0.1716	0.2961	0.1189
Renal Failure, v[n][12]	0.1007	0.1299	0.1237
Neurological Failure, v[n][13]	0.1440	0.1807	0.1457
Number of Organ Systems Failing, v[n][14]	0.6950	0.9481	0.7870
Dead_ICU, v[n][15]	0.0319	0.3414	0.1281
Dur_Hos, v[n][16]	45.3572	9.2625	22.3206
In24hrs, v[n][17]	0.9981	0.8266	0.9276

If we compare the proximity of the cluster centres between variables in Table 57 (above), we see, for example, a relation between 'Respiratory Failure', 'Cardiac Failure', 'Renal Failure' and 'Neurological Failure', given the relative proximity of the cluster centres for these variables for clusters 1, 2 and 3. Another identifiable proximity is 'Sex', 'Fio2' and 'Number of Organ Systems Failing'.

Table 58. Fuzzy c-Means: membership grades for selected cases

Case number in dataset	Membership grade cluster 1	Membership grade cluster 2	Membership grade cluster 3	Cluster assigned
1	0.0311	0.6656	0.3033	2 / 3
2	0.1308	0.5071	0.3621	2 / 3
3	0.0030	0.9748	0.0221	2
4	0.0243	0.2285	0.7472	3
5	0.0108	0.9220	0.0672	2
11	0.0440	0.0892	0.8668	3
16	0.7593	0.0884	0.1523	1
20	0.7588	0.0886	0.1526	1
41	0.4679	0.1118	0.4203	1 / 3
74	0.4667	0.1008	0.4325	1 / 3

In Table 58 (above) we see the membership grades for selected cases to each of the three clusters generated by fuzzy c-Means. In the fifth column we see the prevalent cluster assignment, given to the cluster with a clearly higher membership grade than the remaining two clusters. If there is no clear winner, this is indicated by a split on two clusters. Ideally, all cases should be clearly assigned to just one cluster. If this is not the case, this may indicate the 'c' value, the number of expected clusters, is not optimum for fuzzy c-Means, that is, the data fits best in 2, 4 or more clusters and not 3. Alternatively it can indicate a problem with the data quality or selection of input variables or data cases. Also, it may indicate the need for a better selection of the other fuzzy c-Means parameters: **m** : the bigger **m** is, the more 'fuzzy' the membership assignments will be. The norm $\|.\|_A$ may be assigned one of: N_E , the Euclidean norm; N_D , the Diagonal norm, and N_M , the Mahalonobis norm; ε_L , the epsilon threshold, which works as a 'cutoff' criteria, among the cluster centroids.

Notwithstanding, we think due to the fact we are processing real clinical data with a certain complexity, that an assignment of 73% of the cases to individual clusters, is quite reasonable. From Table 59 (below) we can see that this is the case, with 22% ambiguously assigned between clusters 2 and 3, and 5% assigned between clusters 1 and 3.

Table 59. Frequencies of memberships to clusters, for total of 100 cases

Cluster 1	Cluster 2	Cluster 3	Clusters 1, 3	Clusters 2, 3	Clusters 1, 2
9	45	19	5	22	0

Visualisation of fuzzy clusters: with reference to Figure 74, in order to obtain a graphic visualisation of the membership grades, we calculate the principal components of the membership grades as in [Kaufman90]. This was done with the data previously processed by Fuzzy c-Means, and previously commented with reference to Tables 57 to 59. The number of principal components is equal to the number of fuzzy clusters less 1, which in this example is 2. If we apply this method to one hundred admission cases we obtain the plot given in Figure 74, in which the closer to the origin is the point (case), the more complications the patient has.

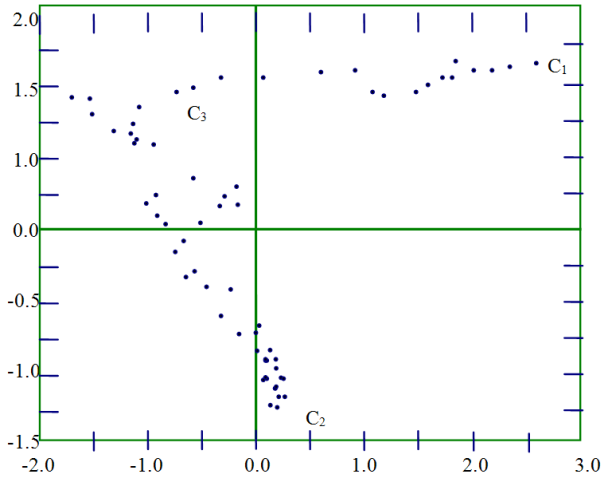


Figure 74. Principal Components of the Membership Grades of 100 patients in three fuzzy clusters

In Figure 74 we can see three tendencies, reaching the extreme points of (2.8,1.7), (0.25,-1.3) and (-1.8,1.5) which correspond to clusters C_1, C_2 and C_3, respectively. The two principal components are calculated from the membership grades generated by fuzzy c-Means.

We can conclude that the groupings have a reasonable correspondence to the factors, 'length of stay', 'number of organ systems failing' and characteristics which give positive (in general).

4.1.7 Summary of the results of the experiments of classification, prediction and factor selection for the 'hospital admissions' dataset

A series of test have been conducted in Section 4.1.2 to predict 'duration_icu' and 'duration_hos' as a categorical value using the hospital admissions dataset and the C4.5 algorithm. Then, in Section 4.1.3 we tried predicting 'duration_icu' and 'duration_hos' as a continuous value, using the same data as in 4.1.2, but this time with the ID3 algorithm. If we convert the continuous value of the predicted variable to discrete (three categories: short, medium and long) the results showed an improvement for the 'short' and 'medium' categories.

In the case of the ICU dataset, the data quality is guaranteed, given that it is a set which was collected by various hospitals in order to carry out statistical studies on ICU patients. Notwithstanding, the data is, in principal, focussed on relating the input variables to the output variable 'dead_icu', that is, a survival prognosis. Thus, relating the input variables to 'duration_hos' and 'duration_icu' were objectives assigned by the author, in consultation with the medical expert.

In terms of the models produced we can summarise the following: a good precision is found for 'duration_hos' for the category 'short stay' (< 10 days): C4.5 gives 89% and the neural net gives 82%. In the case of the objective variable 'vital_state', which is a variable which should have an accurate prediction for this data, C4.5 achieved a maximum of 96% for 'alive' and 63% for 'dead', which is a good precision por positive cases and a reasonable precision for negative cases.

We highlight we results of Sections 4.1.2.3 (iii) and Section 4.1.3 (iv), in which a different set of input variables was used, selected exclusively by the medical expert. This gave the best precision for long duration cases (85%), which is a distinct result to the other C4.5, ID3 and NN models, which gave the best precision for short duration. The other category precisions were 51% for 'short' and 33% for medium. This implies that there exist models with distinct input sets for predicting different length of stay categories.

One of the principal objectives of Sec 4.1 is to explore the relations between the variables, which is what is done with 'Hartigan'. C-Means and Kohonen. In the case of c-Means and Hartigan we do not create predictive models, but they are used to explore the data, thus in this aspect we cannot directly compare if C4.5, ID3 and NN perform better.
We also use the modelling algorithms in to identify the most significant variables with respect to 'prognosis' and 'duration of stay'. We use C4.5 and ID3 to explore the variables and discover precise rules which represent data subsets. This was achieved, and specific rules were identified and commented in section 4.1.2.3. Variables which appear in the first parts of the rules are, for example, 'age', 'mech_ven', 'c_ren_f' and 'OSF', which together have medical meaning. There are given rules which have a high precision (that is, greater than 65%) which have a significant number of corresponding cases.

The tests with Kohonen (4.1.4), c-Means (4.1.6) and Hartigan (4.1.5) are consistent when compared and with C4.5/ID3 in that they confirm the complexity in the relations between the variables of this dataset. We can also interpret a 'fuzzy' aspect, represented by the cases which have not been distinguished in a categorical form.

The results confirm that it is difficult to predict 'length of stay' of a patient from a given set of a priori variables.

In data mining projects it may occur that no algorithm gives a global good result at first, even though we have good data quality, correct variable selection, correct variable definition and representation, etc. In this case a common technique is to carry out a homogeneous segmentation, or partitioning, of the dataset, using an unsupervised clustering algorithm, such as Kohonen or Condorcet. This would be followed by training a model for each homogeneous segment. We could also try a supervised or predefined classification. In the case of the ICU data, a supervised classification could be: trauma patients, craneal trauma patients, with previous clinical history, age (paediatrics, adults, gereatrics), and so on. On trying this approach, no improvement was immediately found by created models from segments created by the unsupervised methods, or by dividing by diagnostic code.

Even so, carrying out segmentation methods would have limited comparitive use, given that the results are not directly comparable with those of c-Means or Hartigan. These latter methods should act on the whole dataset, without previous segmentation. In this manner the results are not pre-conditioned by the segmentation itself. Also, even if we do carry out a segmentation, it cannot be guaranteed to give a better result.

If we compare the use of the Hartigan 'joining algorithm' with crisp and fuzzy covariances to group variables, with other methods, we observe that SPSS Factor Analysis also found 3 factors, and Fuzzy c-Means gave lowest entropy for 2 clusters although 3 clusters had a reasonably low entropy. It has to be added that Fuzzy c-Means looks for clusters based on object grouping, rather than variable grouping.

If we compare the variable groupings with those of C4.5 as seen in the rule sets previously seen in Section 4.1, we note that in the Hartigan/fuzzy covariance groupings the clinical nature is evident. C4.5 on the other hand, groups variables in rules to produce an optimum statistical result, but the clinical justification for the groupings is sometimes not so clear.

4.2 Comparison of fuzzy covariance methods using artificial datasets

In this section of work we use contrasting techniques to evaluate and explore the significance and interaction between input variables. We can divide the work into two principal approaches: (i) generate fuzzy and crisp covariances which are used to 'fuse' variables using Hartigan's joining algorithm[Hartigan75]; (ii) execute algorithms such as C4.5, SPSS factor analysis, SPSS covariance analysis, and a neural network, directly on the input datasets to establish the significance and relations between variables. In both approaches factors are derived which can represent the data in a reduced dimensionality.

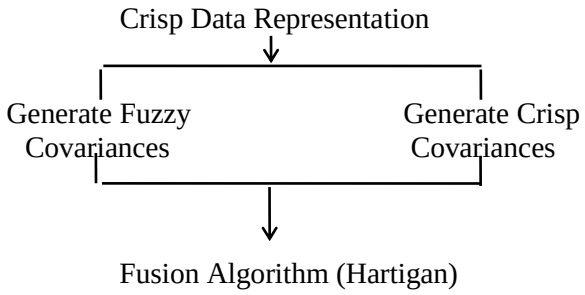


Figure 75. Processing Sequence

Four variants of the fuzzy covariance algorithm [Nettleton98b] are applied to artificial datasets to generate a fuzzy covariance matrix which is then given as input to the Hartigan 'joining algorithm'. The four variants have been previously detailed in Section 3.1.4 of the thesis. The objective is to identify and rank the most significant variables in each dataset. The benchmark results are compared with C4.5 and a Neural Network applied to the same data.

4.2.1 Test Algorithms

The following details the configurations of the test algorithms used to process the three test datasets. Fuzzy c-Means was used to generate the initial fuzzy c-partitions. The resulting membership grades, cluster and Euclidean norm coefficients were given to SPSS and methods 1 to 4, and processed as described in the previous section. The neural network and C4.5 were given the raw datasets to process.

Methods 1 to 4 Fuzzy c-Means Covariance: methods 1 to 4 as detailed in Section 3.1.4 'Fuzzy covariances – Nettleton's fuzzy covariance calculation' of the thesis, were implemented in 'C' language and were executed after fuzzy c-Means had generated the fuzzy c-partitions, for 2, 3 and 4 clusters.

SPSS - Classical Statistics: SPSS was used to represent classical statistical techniques which contrast and cross-check the machine learning algorithms. On each of the three test data sets the following functions were performed:

- Principal component factor analysis, giving the number of factors found and a factor score coefficient matrix.
- Correlation matrix of the input variables.
- Kmeans cluster analysis
- Hierarchical cluster analysis using squared Euclidean distance as the interval; cluster method was between-groups-linkage and a dendral diagram representation was generated.

Feed-Forward Neural Network: a standard 3 layer Feed-forward NN was used to generate an input strength ranking for the input variables which is then used to corroborate the results given by other methods.

C4.5 Rule Induction: standard C4.5 used to generate a pruned rule base of the input variables. The higher level the variable in the rule base, the more general it is, and the lower down the more specific. We identify where C4.5 has placed the variables in the rules and which it has pruned using its information heuristics. This information is cross checked with that of other algorithms.

Hartigan Fusion Algorithm: as previously detailed in Sections 2.4 and 3.2.2 of the thesis. It is contrasted against the factor and hierarchical analysis realised by SPSS. Also the fusion is executed with crisp and fuzzy covariances and the differences studied.

4.2.2 Test Data

In this section we present the three test datasets processed by the different test methods: Iris plant dataset; Hathaway and Bezdek dataset; and an artificial dataset.

Iris Plant Dataset: created by R.A. Fisher, Iris is one of the best known datasets in the pattern recognition literature. The data set contains 3 classes of 50 instances each, where each class refers to a type of Iris plant. One class is linearly separable from the other two; the latter are not linearly separable from each other. The predicted variable is the class of Iris plant. The number of instances is 150, and there are four numeric predictive variables and the class.

Hathaway and Bezdek Dataset: the data used is the Hathaway and Bezdek´s test data detailed in [Hathaway96]. The data is prepared for input to a supervised learning algorithm. There are six inputs i_1 to i_6 and c_1 is the output (class).

Artificial Dataset: an artificial dataset has been created with four inputs i_1 to i_4 and one output c_1, such that the first two variables have a high mutual correlation; the third variable has a medium correlation with variables 1 and 2; and the fourth variable has very little correlation with the other three variables. Column $\mathbf{c_1}$ has a crisp class definition for each object.

4.2.3 Results

In this section we first present the results of the fuzzy covariance calculations for each method, followed by the results of the aggregation and input selection algorithms. At the end of each subsection there is a cross-reference summary of all methods and datasets.

(i) Results of Fuzzy Covariance Calculations

In general there is an agreement between the methods; while the covariance values may vary, the ordering of the variables is constant. It is also interesting to observe the differences in the values to look for tendencies, related to the different combinations of input values (cluster centres, membership grades, norm coefficients, ...). As commented in Section 4.1.5 (i), we observe from Tables 60 and 61 that some of the correlations between variables do not maintain their respective order. We once again conclude that the fuzzy covariances, although derived from the same data, produce different results. The topology of the fuzzy partitions, membership grade values of the cases, and distance metrics norms are some of the factors which distinguish fuzzy and crisp covariances. With reference to Table 62, we see that the fuzzy covariance (methods 2 to 4) do produce much more similar results to the crisp covariances. We remember from Section 3.1.4 that: method 1 is based on the grade of relationship of a variable to the centroid; method 2 is based on the relation between membership grades and data values; method 3 is based on the distances of the objects between cluster centres, weighted by the norm coefficient; and method 4 is the relation between the sum of squares of the distances of the objects to the cluster centres, weighted by the norm coefficient and the membership grade. Each method, in sequence, could be considered as an enhancement of the previous method. Thus from Table 62 we conclude that the form of calculation of the fuzzy covariance is determinant in the degree of convergence with the crisp covariances and other methods.

Table 60. Fuzzy covariance matrix produced by method 1 using Iris dataset as input

	sepal-l	**sepal-w**	**petal-l**	**petal-w**
sepal-l	1.000*	232.351	160.968	301.107
sepal-w	232.351	1.000	-71.384	68.756
petal-l	160.968	-71.384	1.000	140.139
petal-w	301.107	68.756	140.139	1.000

* The value 1 has been assigned to the diagonal and is not used in the fusion process

The diagonal values which represent the fuzzy covariance of each variable with itself, have been assigned as previously commented in Section 4.1.5 (i).

Table 61. Crisp covariance matrix produced by SPSS using Iris dataset as input

	sepal-l	sepal-w	petal-l	petal-w
sepal-l	1.000	-0.1094	0.8718	0.8180
sepal-w	-0.1094	1.000	-0.4205	-0.3565
petal-l	0.8718	-0.4205	1.000	0.9628
petal-w	0.8180	-0.3565	0.9628	1.000

With reference to Table 62 (below) we can see that methods 2, 3, 4 and SPSS covariance are coinciding for all three datasets. Method 1 does not coincide with the other methods.

Table 62. Summary of covariance results: pairs of variables with first and second highest ranking covariances

Data Set	Covariance pairs	method1	method2	methods 3 & 4	SPSS Covars
Iris *	1st**	v_1, v_4	v_4, v_3	v_4, v_3	v_4, v_3
	2nd	v_1, v_2	v_1, v_3	v_1, v_2	v_1, v_3
Bezdek	1st	v_1, v_3	v_3, v_6	v_3, v_6	v_3, v_6
	2nd	v_1, v_6	v_4, v_5	v_4, v_5	v_4, v_5
Artificial	1st	v_2, v_3	v_1, v_2	v_1, v_2	v_1, v_2
	2nd	v_1, v_3	v_1, v_3	v_1, v_3	v_1, v_3

*v_1=sepal-length, v_2=sepal-width, v_3=petal-length, v_4=petal-width
**1st=pair of covariances with highest covariance value. 2nd=pair of covariances with second highest covariance value.

(ii) Results of Aggregation and Input Selection using Iris dataset

The results of this section create new variables from existing ones, and select the most relevant variables from the inputs. We consider the terms 'fusion' and 'aggregation' as synonymous, meaning the joining of two variables or factors to produce a new variable, whose output is defined as a function of the values of the two original variables. Notwithstanding, we consider that 'aggregation' can be applied to variable grouping as well as data grouping of just one variable, whereas fusion usually applies only to variable grouping. The order of selection of the variables by the different algorithms is as important as the final grouping.

Hartigan Joining Algorithm with input matrix of Fuzzy and Crisp Covariances: methods 3, 4 (Fuzzy) and SPSS Correlation (Crisp) all joined v_3 and v_4 to form v_5 and then joined v_1 with v_5 to form v_6, while method 1 differed in that it first joined v_1 and v_4 to form v_5 and then joined v_2 with v_5 to form v_6. This is summarised in Table 64.

SPSS Factor Analysis - Factor Score Coefficient Matrix - Principal Components: one factor was found by this method. The scores for each variable are as follows: sepal-length: 0.30618, sepal-width: -0.15436, petal-length: 0.34069, petal-width: 0.33152. This result indicates that petal-length is the most significant variable, followed by petal-width and sepal-length, which is summarised in Table 64.

C4.5 Heuristics - Simplified Decision Tree: below we can see that C4.5 has discarded sepal-length and sepal width and was able to classify 60 test cases with 2 errors cases using the remaining two variables. Default pruning of 25% was used, with 90 training and 60 test cases. It is clear that petal-length is the most significant variable in general terms, followed by petal-length.

petal-length ≤ 1.9 : Iris-setosa (30.0/1.4)
petal-length > 1.9 :
| petal-width > 1.7 : Iris-virginica (28.0/2.6)
| petal-width ≤ 1.7 :
| | petal-length ≤ 5.2 : Iris-versicolor (30.0/2.6)
| | petal-length > 5.2 : Iris-virginica (2.0/1.0)

Feedforward Neural Network - Input Strength: a standard feedforward NN was run until a satisfactory model was obtained. The post-process option which shows the relative contribution strengths of each input relative to the output was run. The results were: sepal-length = 4.51, sepal-width = 5.13, petal-length = 9.98, petal-width = 11.06. This indicates that petal-width is the most significant variable, followed by petal-length and sepal-width.

(iii) Results of Aggregation and Input Selection using Hathaway and Bezdek Data

Hartigan Joining Algorithm with input matrix of Fuzzy and Crisp Covariances: methods 3 and 4 (Fuzzy) first joined v_3 and v_6 to form v_7 and then joined v_4 with v_5 to form v_8. SPSS (Crisp) formed v_7 and v_8 in the same manner as Methods 3 and 4 (above), although it differed in creating an additional factor v_9 formed by joining v_2 and v_8 . Method 1 had a distinct joining sequence to the other methods, as can be seen in the following diagram:

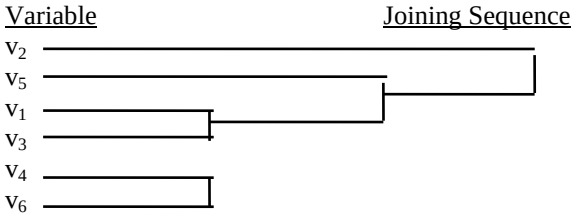


Figure 76. Joining Sequence produced for Hathaway & Bezdek data using covariance output matrix of method 1

SPSS Factor Analysis: using Principal Components to generate a Factor Score Coefficient Matrix, the results given in Table 63 (below) were produced. Principal Components has generated three factors from the six original input variables. We observe that, for example, in the case of factor C_1, v_3 is the variable which contributes most to the overall composition of the factor, followed closely by v_6 and then v_5.

Table 63. The three factors found by the SPSS factor analysis method

	C_1	C_2	C_3
v_1:	-0.29997	-0.11792	0.46278
v_2:	0.01023	0.44865	-0.60755
v_3:	0.35867	-0.21981	-0.02728
v_4:	0.13552	0.44220	0.36592
v_5:	0.19956	0.30961	0.51865
v_6:	0.33946	-0.24402	0.00509

C4.5 Heuristics - Decision Tree: C4.5 discarded all variables except variable 1 and was able to classify 3 test cases (one from each possible class) with 0 errors. Default pruning of 25% was used, with 6 training and 3 test cases. Given that C4.5 has used only one input variable, v_1, it is possible that the algorithm has been unable to find a reasonable structure from the input variables and has created a 'default' in which to categorise the data cases.

Feedforward Neural Network - Input Strength: A standard feedforward NN was run until a satisfactory model was obtained, with the option selected which calculates the relative activations of the neurons in the input layer. This post-process option effectively shows the relative contribution strengths of each input relative to the output. The results were: v_1=5.46, v_2=4.29, v_3=2.22, v_4=7.53, v_5=1.88, v_6=3.12. This indicates that v_4 is the highest contributor, followed by v_1 and then v_2. In percentage terms, this would mean that v_4 contributes 30,7% to the output layer of the model. This is calculated by dividing the input strength of v_4, that is, 7.53, by the sum of all the input strengths v_1 to v_6, that is, 24.5.

(iv) Summary of results of aggregation and input selection (ii) and (iii)

With reference to Table 64 (below) we can see that methods 3, 4 and SPSS covariance are coinciding for all three datasets. C4.5 is coinciding with method 1 and SPSS principal components is coinciding to a lesser degree with the neural network.

Table 64. Joining order and significance ranking of input variables

Data Set	Hartigan with output of methods 3 & 4	Hartigan with output of SPSS Cov	Hartigan with output of method1	SPSS principal components	C4.5 Rule Induction	Neural Net-work
Iris *	v_3 , v_4 , v_1	v_3 , v_4 , v_1	v_3 , v_4 , v_1	v_3 , v_4 , v_1	v_3 , v_4	v_4 , v_3 , v_2
Bezdek	v_3 , v_6 , v_4 , v_5	v_3 , v_6 , v_4 , v_5 , v_2	v_3 , v_1 , v_5 , v_2	v_5 , v_1 v_2 , v_4	v_1	v_4 , v_1 , v_2 , v_6
Artificial	v_1 , v_2 , v_3	v_1 , v_2 , v_3	v_1 , v_4 v_3 , v_2	v_3 , v_4 , v_1 , v_2	v_1	v_2 , v_1 , v_4 , v_3

*v_1=sepal-length, v_2=sepal-width, v_3=petal-length, v_4=petal-width

4.2.4 Summary of Section 4.2

Section 4.2 has considered the calculation of statistics such as covariances in a c-Means type fuzzy environment. Also, fusion using fuzzy covariances has been demonstrated. This section of work is limited to crisp data input and the formation of fuzzy partitions from it. As an extension, a method of fuzzy representation of the data at input time has been detailed in [Nettleton98a]. Other areas of interest are: the use of OWA operators, fuzzy data capture and representation, and studies of c-Means variants such as mixed c-Means[Pal97] and fuzzy symbolic c-Means[El-Sonbaty98]. With respect to the fuzzy covariance methods, methods 2, 3 and 4 give similar results to the crisp covariances, C4.5 and Neural Network. On the other hand, method 1 gives different variables as the most significant and the resulting covariances also have a different ordering. We conclude that method 1 requires the additional weighting and factors in the distance calculations, which are present in methods 2 to 4.

4.3 Apnea questionnaire data (Hospital Clinic, Barcelona)

In this section we apply the WOWA aggregation operator to diagnose Apnea cases using a dataset collected from patients at the Hospital Clinic of Barcelona. In this case, the data was captured in crisp (categorical) form, using a standard Apnea screening questionnaire.

In Section 4.3.1 the variable selection and weight assignment is assigned by medical expert judgement. Also, differently to the method described later in Section 4.3.2, the 'w' reliability weights are assigned by vectors of 5 values which are then interpolated to give 5 characteristic bias curves. Refer to Section 3.2.3 of the thesis, and Figures 55a to 55e, for a description of the characteristic curve definitions. A small number of Apnea cases are selected as input, which represent strongly negative, strongly positive, positive and borderline patients.

In Section 4.3.2 we see a diversity of statistical and machine learning methods based first in clustering and second in classification, to establish relevance and reliability weights for variables in a dataset. For comparison, weights are secondly assigned by the medical expert, and thirdly by a mixture of expert assignment with statistical inspection. The resulting weights in each case are assigned to the WOWA aggregation operator to produce a diagnosis for each case, and the results are discussed. The weights are assigned directly as in the standard WOWA, that is one relevance weight and one reliability weight per variable. The data set processed in this section includes all the available Apnea cases as input, with a total of 150 cases.

In Sections 4.3.1 and 4.3.2, we also note that in each method, a different set of most significant variables was chosen, by the different statistical and machine learning methods, by medical expert judgement, and by a combination of statistics, machine learning and expert judgement. In all of Section 4.3, which deals with the data provided by the Hospital Clinic of Barcelona, the data capture method has been crisp. We will later see how this contrasts with a fuzzy data capture of Section 4.4.

4.3.1 Test of Apnea diagnosis using WOWA and weights assigned by medical expert

In this section the OWA and WOWA aggregation techniques are applied to selected Apnea cases from the Hospital Clinic of Barcelona, the data being captured in a crisp form, and the output being a binary valued diagnosis. Both the OWA and the WOWA operators use reliability and relevance vectors for input variable weighting which are initially assigned by a medical expert.

(i) Objectives and problem definition

We summarise the results of applying the OWA and WOWA aggregation operators, and principal components methods to predict Apnea cases. We modified the WOWA functionality by fixing the 'w' weights to five characteristic curves, which define the weighting bias on the data values, which corresponds to the reliability of the values. Refer to Section 3.2.3 of the thesis, Figures 55a to 55e, for a description of the characteristic curve definitions.

In interpreting the aggregation results for all aggregation techniques, we need to define a threshold which indicates where 'do not admit' ends and 'admit' starts. We establish this by running known cases through and noting the values generated as output. We need a spectrum of cases, from a strongly positive case, to a strongly negative case, and a spectrum of intermediate cases ordered by degree of evidence of the apnea syndrome. This is measured clinically in terms of < 10 apneas / hour and >= 10 apneas / hour, so it is possible to assign a numeric quotient to the grade of incidence of apnea.

Table 65. Discriminant variables: example minimum set with weighting factors for aggregation

variable	description	relia-bility*	rele-vance*
age	age in years	E	0.5
sex	sex 1 or 2	E	0.7
weight	weight in Kg	M	0.7
IMC	body mass index in Kg/m^2	M	0.7
Neck circum-ference	Neck circumference in cm.	E	1
alcohol	Alcohol intake	M	0.5
HTA	Arterial hypertension mmHg	E	0.7
R1	Do you snore when sleeping or have you been told that you do?	H	0.9
R2	Does your snoring wake your partner or can it be heard from another room?	H	0.9
R11	Do you have head-ache when you wake up in the morning?	M	0.9
R13	Do you feel as if you haven't rested when you get up in the mornings?	M	0.9
S3	Do you fall asleep when at the cinema, theatre, or other spectacle?	M	1
S4	Do you sleep in meetings or in public places?	M	1
S5	Do you fall asleep while driving on the motorway?	M	1
S6	Do you fall asleep against your will during the daytime?	M	1

*the values of these columns are then converted proportionately to normalised values so that $\Sigma\rho = 1$ and $\Sigma\omega = 1$, as in Table 67 (below)

Table 66. ρ vector: each variable has a ρ weight which indicates its reliability. $\Sigma\rho = 1$

	Question Response Variable							
	R_1	**R_2**	**R_{11}**	**R_{13}**	**S_3**	**S_4**	**S_5**	**S_6**
ρ vector	0.15	0.15	0.15	0.15	0.09	0.11	0.09	0.11

In Table 67 (below) we can see five values defined for each variable. From these value points, WOWA uses an interpolation method, such as that of Chen and Otto [Chen95], to create a continuous function curve which can be used to weight all the values of each variable.

Table 67. ω vector: each variable has a vector which weights the ordered data responses for that variable, in terms of their relevance. $\Sigma\omega = 1$

	ω vector					
Variable	ω_1	ω_2	ω_3	ω_4	ω_5	Weighting Bias on:
R_1	0.20	0.20	0.20	0.20	0.20	Even
R_2	0.30	0.30	0.20	0.10	0.10	Low values
R_{11}	0.10	0.10	0.20	0.30	0.30	High values
R_{13}	0.30	0.15	0.10	0.15	0.30	High & Low
S_3	0.10	0.25	0.30	0.25	0.10	Middle values
S_4	0.20	0.20	0.20	0.20	0.20	Even
S_5	0.30	0.30	0.20	0.10	0.10	Low values
S_6	0.10	0.10	0.20	0.30	0.30	High values

(ii) Summary of results for Section 4.3.1

In Table 68 (below) we run the method using four theoretical test cases and 3 aggregation methods. Rows 1 to 3 are positive cases (admit), case 2 being *borderline*: from which we derive the % success rate of correct diagnosis of patients who have apnea syndrome; and row 4 is a strongly negative case (do not admit): from which we derive the % success rate of correct diagnosis of patients who do not have apnea syndrome.

The case data is not only weighted by the ρ and ω vectors, but also by the membership grade associated with the linguistic label of each question response.

We see that WOWA agrees with OWA and principal components for cases 1 and 3, and does not agree for the borderline case (2) and the strongly negative case (4). Principal components and OWA give positive outcomes for all four cases, thus having a good precision for positive diagnosis and low precision for negative diagnosis (high sensibility and low specificity as commented previously in Section 1.2.9 of the thesis) which is a typical result for standard statistical techniques used in the literature [Hoffstein93]. On the other hand, WOWA successfully diagnosed both the borderline case (row 2) and the negative case (row 4).

Table 68. Input responses for 8 questions with corresponding outcomes from aggregation methods

Input									Outcomes		
Projection of crisp responses (0=never to 4=always) on normalised scale											
	R_1	**R_2**	**R_{11}**	**R_{13}**	**S_3**	**S_4**	**S_5**	**S_6**	Wowa	Owa	Principal Components
Data vector for Patient P_1	0.60*	0.60	0.60	0.60	0.60	0.60	0.40	0.40	0.53 admit	0.84 admit	1.15284 admit
Data vector for Patient P_2	0.60	0.60	0.60	0.60	0.40	0.40	0.20	0.20	0.48 do not admit	0.84 admit	1.15317 admit
Data vector for Patient P_3	0.80	0.60	0.60	0.80	0.60	0.60	0.60	0.60	0.56 admit	0.89 admit	1.15412 admit
Data vector for Patient P_4	0.40	0.40	0.60	0.60	0.40	0.40	0.20	0.20	0.45 do not admit	0.84 admit	1.15391 admit

*NB: these values are not membership grades, but are numeric equivalents of the crisp linguistic labels, that is, 1/5=0.2=never, 2/5=0.4=rarely, 3/5=0.6=sometimes, 4/5=0.8=often, 5/5=1.0=always, with some readjustment depending on the projection and on the resulting distribution.

4.3.2 Evaluating reliability and relevance for WOWA aggregation of sleep Apnea case data

In the following we use diverse clustering and classification techniques to establish the relevance and reliability of each variable, which is then given to the WOWA aggregation operator to generate an aggregated value for each patient with high correlation to the apnea diagnosis. This is also compared with expert medical assignment of the weights, and with mixed assignment, that is, expert assignment together with statistical inspection.

We describes the data used, the objectives and the method of diagnosis using WOWA operators and the problem of assigning the relevance and reliability weights. This is followed by details of the application of four unsupervised clustering techniques to identify variables used to partition the data. In the next step, supervised modelling techniques are used to generate a ranking of variables in order of significance to the diagnosis output. WOWA aggregation results are detailed using the reliability and relevance weights derived in the previous two parts, and the section terminates with a discussion of some conclusions of the approach and the results of its application to apnea diagnosis.

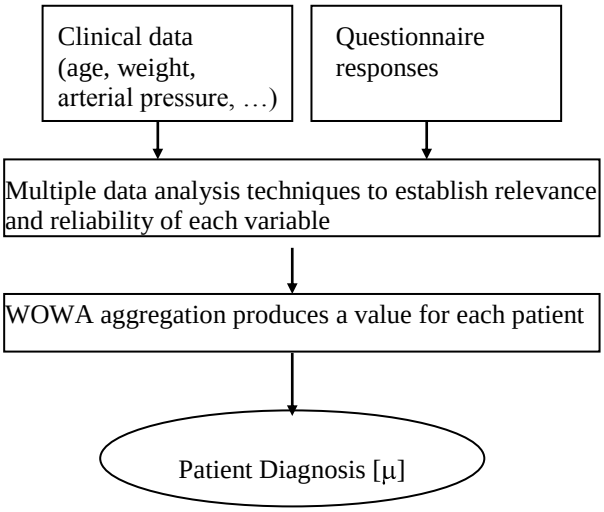


Figure 77a. Data processing of the apnea data input variables to produce a diagnosis

In Figure 77a (above) we see the data processing scheme used to establish the relevance and reliability weights, and using the WOWA aggregation operator to produce a diagnosis as the end result.

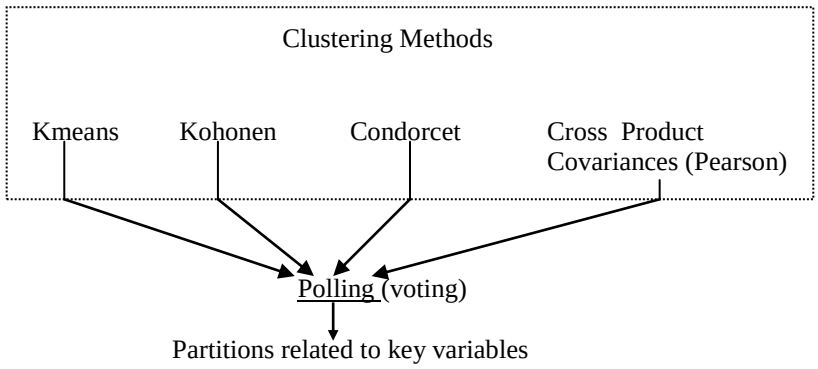


Figure 77b. Clustering Techniques determine relation of key variables to clusters

In Figure 77b (above) we see the four clustering techniques used in this section: Kohonen Net, Kmeans, Condorcet and Cross Product Covariances (Pearson). The techniques have been chosen so as to contrast the results given by significantly different approaches. The same data input to each method, which produces a clustering which upon inspection indicates the variables which best explain the grouping of the data. For example, if two clusters were produced by a method and in one cluster all the cases correspond to 'age' less than 45 years, and in the other cluster all the cases correspond to 'age' greater or equal to 45 years. Thus we could conclude that for this method, 'age' has been a

determinant variable in partitioning the dataset. In this manner, each method may indicate the same or different key variables used for partitioning, and thus there is a posterior 'polling' phase which simply conducts a 'vote' on the results of all the methods to rank the variables in frequency of appearance as key variables. For example, if we have four methods and three say that 'age' is the highest ranking key variable, and one says that 'weight' is the highest ranking key variable, then 'age' will be voted as more significant than 'weight' by simple majority (3 to 1). In the case of a 'tie', we would place both variables with an equal ranking, although, as can be seen in the results as shown in Table 69, with the given variables, there were no cases of 'ties'.

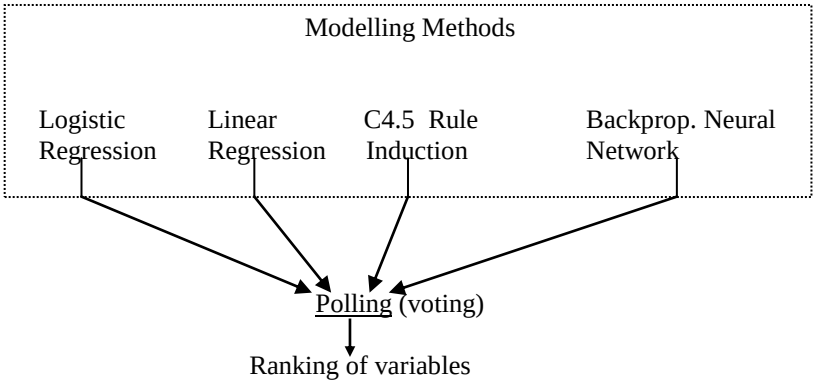


Figure 77c. Contrasting methods are polled to determine a ranking of relevance and reliability of the variables with respect to apnea diagnosis

In Figure 77c (above) we see the four classification techniques used in this section: C4.5 rule induction, back-propagation neural network, logistical and linear regression. The techniques have been chosen so as to contrast the results given by significantly different approaches. The procedure of executing the methods and polling the results is the same as that for the clustering methods explained for Figure 77b. Each modelling method produces an output which allows a ranking of significance of input variables with respect to the output, and the results are later shown in Table 70.

Apnea patient data: consists of the collected data of the standard sleep patient questionnaire, for 154 patients, captured over a 1 year period. The data set contains 68,2% positive outcomes and 31.8% negative outcomes. The questionnaire consists of two main sections: the first records clinical data (age, weight, blood pressure, etc. ..); the second section consists of 41 questions to which the patient responds. The questions are divided in 3 subsections: 15 general sleep questions {s1...s15}, 16 respiratory related questions {r1…r16} and 9 somnolence related questions{s1…s9}. Based on this information, the doctor then gives a clinical evaluation: healthy; simple snorer; doubtful; typical apnea; other illness. We simplify this to: typical apnea; no apnea.

Establishing reliability and relevance for WOWA aggregation: we consider applying diverse data analysis techniques to the variables which have been collected for apnea patients. We wish to establish, for each variable, its relevance with respect to the apnea diagnosis, and its reliability. The four clustering techniques used are: Kohonen Net, Kmeans, Condorcet and Cross Product Covariances (Pearson). These two values will be the two weights in the WOWA vectors. Also we wish to establish the most significant variables with respect to apnea diagnosis. The four classification techniques used are: C4.5 rule induction, backpropagation neural network, logistical and linear regression. The techniques have been chosen so as to contrast the results given by significantly different approaches.

WOWA Aggregation in the context of Apnea diagnosis: in [Nettleton99e], Nettleton evaluated different aggregation methods for diagnosing sleep apnea. The following aggregation methods were considered: Ordered Weighted Average (OWA) [Yager93], Weighted Ordered Weighted Average (WOWA) [Torra97a] and Principal Components. OWA uses a vector in which, for each variable, a value is assigned which indicates its relevance. In WOWA a second vector is incorporated whose values indicate the reliability of each variable. In this section we extend the previous work, calculating the weights from statistical analysis of the data.

Relevance and Reliability in the context of Apnea diagnosis: relevance is a standard data analysis objective for which we can apply diverse algorithms and interpret the results. Relevance is more straightforward to establish in statistical data analysis, than reliability. **Reliability** is influenced by different aspects. There are data aspects, such as %missing and %erroneous. Then there are application dependant aspects which, in the case of the questionnaire responses can be the truthfulness with which the patient responds (it may be that if the patient goes to sleep at the wheel

of a car, s/he does not wish to make that known, and thus tends to give a higher negative response rate to this question {s5}, than it really should have).

(i) Unsupervised Clustering and Statistical Techniques

With reference to Table 69 (below) we can see that in methods 1, 2 and 3, *partner* has influenced partitioning. Kohonen and Kmeans tend to have biased the *'g'* responses while Condorcet has used the *'s'* responses more for partitioning. There is not a clear consensus between the different clustering and statistical techniques. Methods 1, 2 and 4 used numerical representation of all the data, while method 3 used a categorical representation with ChiSquared for the significance tests. **Kohonen net:** Various architectures of net were tried: input layers of 41 neurones (questionnaire responses only), 27 (clinical data only), and 68 (questionnaire responses and clinical data). **Kmeans:** Standard SPSS Kmeans was used for 2 clusters, maximum iterations set to 100, convergence at 0.02. **Condorcet – mixed data type clustering:** A proprietary IBM algorithm based on the *Condorcet* [IBM96] distance criteria was used to generate 9 segments. All data was prepared as categorical and a chisquared measure was used to rank the variables in each segment and between segments. **Cross product covariances (Pearson):** A standard SPSS numeric covariance was used with the Pearson Product Moment option, to generate covariances between all the variables, defined as numeric.

Table 69. Clustering and statistical techniques applied to the apnea cases and the identification of key variables which distinguish the resulting partitions

	Kohonen (1)	Kmeans (2)	Condorcet (3)	Cross product covariances (Pearson) (4)
Most significant variables	partner, weight, g1, r1, g4, s5 (2 and 6 clusters)	partner, sex, g4, r6, g13, g5, s5 (2 clusters)	hta, s5, s2, s1, s6, r13, partner, g6, g7 (9 clusters)	neck, weight, age, alcohol

(ii) Supervised classification and statistical models

We contrast four techniques, each using a different algorithmic basis, with the objective of realising a consensus for the variables being evaluated. With reference to Table 70 (below) we can see that methods 1, 2 and 3 have identified *waist* as a significant variable, while methods 1, 3 and 4 have identified *g1* as significant. Other identified variables are *r2*, *partner*, *weight* and *s10*. **C4.5 rule induction:** Quinlan's standard C4.5 algorithm was used, with 25% pruning, no external test set, and no grouping. **Backpropagation neural network:** The neural network training phase generates a *sensitivity analysis* which provides a ranking of the variables with respect to the output (in this case, the diagnosis yes/no). **Logistic regression:** Standard SPSS logistic regression was used with 3 test models. Overall precision's were: 89,66%, 88% and 75%. **Linear regression:** One SPSS linear regression was generated. The R^2 value was 0.31309, the standard error was 0.51035.

Table 70. Significance ranking of input variables for different methods

	Logistic regression (1)	Linear regression (2)	C4.5 rule induction (3)	Back propagation neural network (4)
Most significant variables	neck, g1, partner, s9, s8, s7, s6, s10, waist, r12, r2, r5, r6, g2, g6	g8, partner, waist, hip, weight	r3, r2, waist, age, weight, g1	sex, r15, g10, g1, r9, r1,hta,tabacco, height, alcohol, weight, r3, r8, s7, g5, s10, r2, r5

(iii) WOWA Aggregation using the weights established in Steps (i) and (ii)

The resulting consensus from all data analysis methods (Tables 69 and 70) indicated the following 9 most significant variables and their corresponding reliability and relevance weights: {partner(0.90, 0.70), weight(0.93, 0.7), neck(0.95, 0.92), g1(0.65, 0.68), s5(0.45, 0.95), sex(1.0, 0.7), r15(0.65, 0.60), hta(0.95, 0.67), r5(0.55, 0.90)}. Question **g1** is*: "how many hours do you normally sleep?"*; question **s5** is *"do you fall asleep while driving on the motorway?"*; question **r15** is *"do you have lapses of memory or loss of concentration"* and question **r5** is *"have you noticed an increase in the intensity of your snoring recently?".* Executing the WOWA aggregation with the above input weighting vectors and the 154 patients case data rows, produced the results of Table 71. In Table 71 the output aggregated value produced by WOWA has been correlated with the binary value which represents the apnea diagnosis.

One part of the reliability weight can be calculated in terms of the consistency between methods for each variable. For example, the variable *'hta'* (arterial hypertension) may be chosen as one of the 9 most significant variables by all 8 methods, for which we assign it a reliability quotient of 1.0, while *'partner'* appears in the top 9 variables for 4 out of the 8 methods, for which we assign it a reliability quotient of 0.5. This value is pondered by the percentage of missing values in the original data for each variable, and, in the case of the questionnaire responses, the possibility that the patient does not respond correctly or truly to a given question. We limited the number of variables to 9 given that, by statistical inspection, from the tenth variable on it was though that the choices of variable were no longer sufficiently reliable to include. This is also explained given that different methods vary in the number of variables which are clearly identified, with a minimum of 5 variables in the case of linear regression, and up to 18 by the backprop. neural network, as can be seen in Tables 69 and 70.

(iv) Weights assigned by medical expert and statistical analysis

Nettleton, in [Nettleton99b] defined, jointly with the medical expert, a most significant sub-set of 15 variables for apnea diagnosis. This was: {age(1.0, 0.5), sex(1.0, 0.7), weight(1.0, 0.7), imc(1.0, 0.7), neck circumference(1.0, 1.0), alcohol(0.7, 0.5), hta(1.0, 0.7), r1(0.7, 0.9), r2(0.7, 0.9), r11(0.7, 0.9), r13(0.7, 0.9), s3(0.4, 1.0), s4(0.5, 1.0), s5(0.4, 1.0), s6(0.4, 1.0)} with figures in brackets being the respective reliability and relevance weights, also assigned by the medical expert. This is the same set of variables and weights which is used in Section 4.3.1. The above reliability and relevance values have to be prepared for input to WOWA so that they sum to 1 for each case. The values are reduced proportionately to achieve this. The normalised value is converted into the relevance and reliability weights for WOWA which respectively sum to 1 for all variables, that is , $\Sigma\rho = 1$, $\Sigma\omega = 1$, where ρ is reliability and ω is relevance.

(v) Summary of results – Section 4.3.2

With reference to Table 71 (below), which summarises the results of Section 4.3.1, in which we have evaluated assignment by statistical/machine learning methods, medical expert assignment alone and mixed assignment (expert and statistical/machine learning) methods. Overall, in Table 71 we can see a favourable result for diagnosis of positive cases and a good result for negative cases, in comparison with the methods used in the literature[Hoffstein93][Katz90]. It can be seen in Table 71 that the best results are given by the method in which the weights are assigned by statistical and machine learning methods, but are then revised by the medical expert.

Table 71. Correlation of WOWA with Apnea Diagnosis for three different weight assignment methods for reliability and relevance

Weight assignment method	Expert assignment of weights	Data analysis assignment of weights	Expert + data analysis assignment
Diagnosis of positive cases	0,75	0,78	0,81
Diagnosis of negative cases	0,65	0,61	0,67

4.3.3 Summary of Section 4.3

The work in this section has been jointly developed with medical and data analysis expertise, and an area has been chosen for which there is real room for improvement, due to the lack of precision of existing screening methods (especially for negative case prediction), and the high cost and resource requirements for sleep centre testing. Two fundamental aspects have been considered from a data analysis point of view: representation of the data and aggregation. With respect to aggregation, three contrasting methods have been proposed for aggregating the data values: Principal Components, OWA and WOWA.

We have described a method for establishing the relevance and reliability weights needed by the WOWA aggregation operator, applying them to complex data of a real medical problem. With the WOWA aggregation method to apnea diagnosis we can include relevance and reliability information in a more precise manner, to improve the success rate for correct diagnosis. The approach described in this section is previously untried in the literature of Apnea diagnosis, which has tended to focus on multiple linear regression and logistic regression models (see Tables 1 and 2, Section 1.2.9 of the thesis). The aggregation techniques have been tested on a real crisp Apnea case data set [Nettleton99c][Nettleton99e] in collaboration with the Hospital Clinic of Barcelona.

This work of Section 4.3.1 is summarised in [Nettleton99c] and has demonstrated a good precision for both positive and negative cases. The work of Section 4.3.2 is summarised in [Nettleton99e], where the data was again captured by crisp (categorical) question responses but this time processed by fuzzy aggregation algorithms, WM, OWA and WOWA. With reference to Section 4.3.1, Table 68, we observe and conclude that WOWA was the only method which correctly diagnosed the negative case (row 4) and the borderline case (row 2).

The next step on from the work in this section involves the evaluation of the aggregation of questionnaire responses captured on a fuzzy (continuous scale), as a opposed to the crisp (categorical) representation. This is detailed in the following Section 4.4, in which a specially designed questionnaire is completed by patients, and processed with the methods which have been described in this section.

4.4 Apnea questionnaire data (Hospital of the Santisima Trinitat, Salamanca)

In this section we apply the WOWA aggregation operator to diagnose Apnea cases using a dataset collected from patients at the Hospital of the Santisima Trinitat, Salamanca. In this case, the data was captured in both crisp (categorical) and fuzzy (continuous scale) form, using a specially designed questionnaire. The patients fill in two questionnaires, one in the fuzzy/continuous form and a second questionnaire with identical questions but represented in crisp/categorical form. This will enable us to compare the diagnosis using crisp and fuzzy representation methods. The work gives a novel approach for questionnaire data capture and processing where linguistic labels and subjective / uncertain inputs play an important role, and enables expert knowledge and statistical knowledge to be incorporated into the data processing.

The complete variable set was the same as that collected in Section 4.3 for the data of the Hospital Clinic, but the selected variables were slightly different, due to the criteria of a different medical expert and the use of different statistical techniques. Different types of weight assignment were tried: statistical analysis, medical expert assignment, statistical analysis and medical expert assignment. Also, the WOWA precision for diagnosing positive and negative cases was benchmarked against ID3 tree induction and a feedforward neural network. The data processing differs from the crisp Apnea data of Section 4.3, given that we also incorporate membership grade values as part of the input data. We summarise the results of applying WOWA, Neural Nets and tree induction to predict Apnea cases, using data from questionnaire responses collected in both scalar (continuous) and discrete (categorical) form.

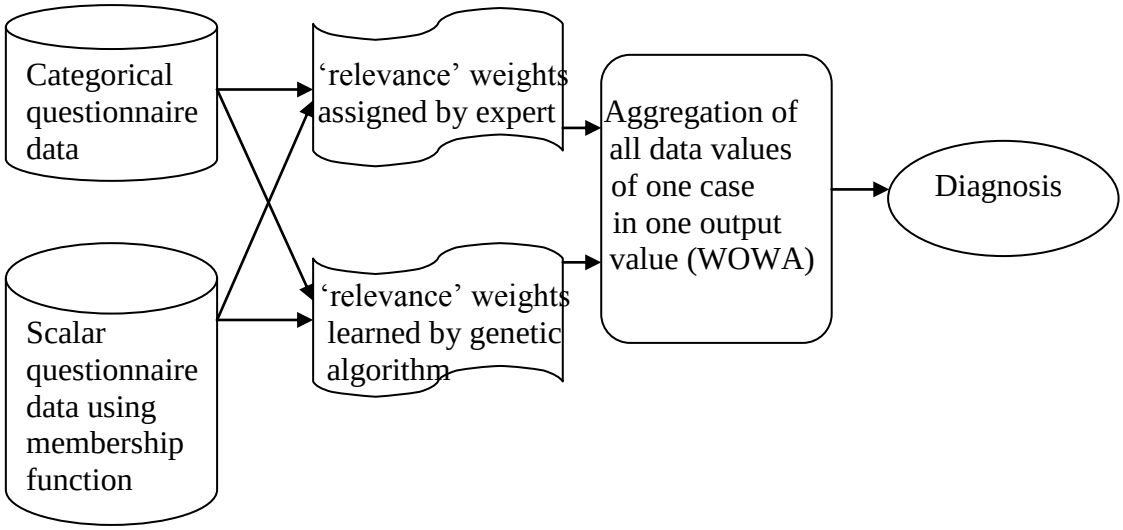


Figure 78. General data processing scheme

With reference to Figure 78, the data processing scheme allows us to compare categorical and scalar questionnaires for diagnostic accuracy, and compare results when the 'relevance' weights are assigned by the medical expert (based on personal knowledge and the literature) or learned by the genetic algorithm. The 'reliability' weights are always assigned by the medical expert (based on personal experience and a knowledge of the specific case data captured in his Clinic). This is because it was considered that 'reliability' would be more difficult to learn statistically, whereas 'relevance' is more akin to correlation analysis. Note that the scalar questionnaire makes use of a manually defined membership function to interpret the patient's response. The 'reliability' and 'relevance' values, on the other hand, are automatically interpolated into curves as part of the aggregation operator itself.

4.4.1 Test data – selected variables

The questionnaire is designed to detect diverse sleep pathologies. Thus the medical expert has selected a subset of variables with highest correlation specifically with Apnea diagnosis. Statistical details of these variables are given in Tables 72(a) to 72(c) below. In Table 72(a) we see in column 1 the variable names, starting with the clinical data: age, sex, neck circumference, body mass index, and somnolence indicator. These are followed by the selected questionnaire responses: G3 is a general question, while R1 to R12 are respiratory related questions, and S4 to S10 are somnolence related questions. The variables and the questionnaire responses have been detailed in length in Section 3.4 of the thesis. We observe that the numerical variables such as 'age' and 'neck circumference' have been categorised by the

definition of discrete bands. These bands have been defined by medical expert assignment. We see that the questionnaire response variable have been defined with five fuzzy linguistic labels. The membership curve to interpret the membership grade and boundaries for each label, has been defined previously in Section 3.1.5 of the thesis.

Table 72(a). Selected variables for apnea diagnosis and meta-data (reliability and relevance) assigned by medical expert

Variable	Relevance	Reliability (bias on expected values)	Categorisation	Observations
Age	0.65	3 (M)	1 "0-20"; 2 "21-40"; 3 "41-60"; 4 "61-80"; 5 ">80"	Sleep apnea is more frequent as age advances. They are very unusual in children.
Sex	0.60	1(E)	1 "MALE"; 2 "FEMALE"	Apneas are more frequent among males (3-4:1)
Neck Circumf-erence (cm)	0.87	4(M)	1 "<30"; 2 "30-35"; 3 "36-41"; 4 "42-48"; 5 ">48"	The neck circumference is an important predictive factor. The thicker the neck, the greater probability of apneas.
BMI (Body Mass Index)	0.8	3(M)	1 "<22"; 2 "23-26"; 3 "27-30"; 4 "31-34"; 5 ">35"	The BMI has a similar significance to that of the Neck Circumference, but is slightly less relevant. A greater BMI implies greater probability of apneas
Somno-lence	0.8	2(L)	1 "YES"; 2"NO"	Somnolence is a good indicator for sleep apnea. To evaluate the degree, we tend to use the apnea/hypopnea index per hour. The AHI correlates with the indexes and scales of excessive somnolence. Thus, the absence of somnolence practically discounts an elevated AHI, that is, above 30 per hour.
G3	0.6	H	1 "never"; 2 "rarely"; 3 "sometimes"; 4 "often"; 5 "always"	High response values indicate greater probability of apneas.
R1	0.75	E	Idem	Idem
R2	0.90	H	Idem	Idem
R6	0.85	H	Idem	Idem
R7	0.95	H	Idem	Idem
R8	0.85	H	Idem	Idem
R9	0.63	H	Idem	Idem
R10	0.80	H	Idem	Idem
R12	0.70	H	Idem	Idem
S4	0.75	H	Idem	Idem
S5	0.90	H	Idem	Idem
S7	0.85	H	Idem	Idem
S9	0.85	H	Idem	Idem
S10	0.85	H	Idem	Idem

In Table 72(b) below we see all the selected questionnaire responses used as input. These have been selected by medical expert knowledge and statistical analysis from the total of 40 questions asked to the patient in the questionnaire. There is 1 general question, 8 respiratory related questions and 5 somnolence related questions. The reliability weights play a key part here: although, given that as we have preselected these variables, they all have a relatively high relevance weight, some of the questions may not be answered truthfully, for example, S5, S9 or S10. Questions R1, R2, R7 and

R8 depend on a witness, which is usually the bed partner. In the absence of a witness, the reliability of the responses to these questions drops sharply.

Table 72(b). Description of selected questionnaire questions

Variable/ Question	Description
G3	Are you accustomed to taking a nap during the day?
R1	Do you sleep while asleep or have you been told that you do?
R2	Does your snoring wake up your partner or can it be heard from another room?
R6	Do you wake up at night with a sensation of choking?
R7	Have you been told that you "stop breathing" when you are asleep?
R8	Has your partner woken you for fear that you have stopped breathing ?
R9	How many times do you get up to go to the toilet at night?
R10	Do you sweat a lot at night?
R12	Do you wake up with a dry mouth?
S4	Do you fall asleep in meetings or in public places?
S5	Do you fall asleep when driving on the motorway?
S7	Do you fall asleep while eating?
S9	Do you fall asleep when driving you stop at a traffic light?
S10	Do you fall asleep in your workplace while doing your normal work activities?

In Table 72(c) below we see the 7 clinical data variables used as input. These variables have been preselected by the medical expert from a total of 15 variables (see Annex 3 of the thesis for a complete detail of all the variables). We observe that the mean age is 53 years, and the patients are predominantly male. The AHI index is the clinical index which indicates if the patient has Apnea or not. We categorised this as a binary variable, using the cut-off point of AHI ≥10 for positive cases, as indicated in the literature and by our medical expert.

Table 72(c). Basic Statistics of the Clinical Variables

Variable	Minimum	Maximum	Mean	Frequencies for Categorical Variables
Age	22	86	52.94	
Sex				50 male, 21 female
Neck Circumference (cm)	34	50	39.52	
BMI (Body Mass Index)	19	43	25.46	
Somnolence				40=NO, 28=YES, 3=UNKNOWN
AHI Index (output)	0	85	19	
Flag 1/0 (AHI ≥10) (output)				39 Positive cases; 32 Negative cases

4.4.2 Questionnaire responses – comparison of categorical and scalar representation of questions

In this section we evaluate the responses to the with the categorical form, and the responses to the questions with the scalar form. We compare the response frequencies to identify tendencies, differences, and improvements, if any, of the scalar form over the categorical form. With reference to Table 73 (below), we observe from the '*Sca*' columns that in general, the fact that a person tends to think of a response in a scalar form rather than categorical, depends more on the question than the linguistic label (never, rarely, …). Notwithstanding, if we study subgroups of questions (G, R, S) we can see signs of greater frequencies for the 'Sca' responses 'frequently/always' (R), and 'never/rarely'(S). In Table 73 we can also see clear tendencies for specific questions, such as S9 with a higher frequency on responses 'never' and 'rarely', and R12 for the preference for higher range values 'sometimes', 'frequently' and 'always'. We can also see an inversion of the tendency for responses to 'never' and 'rarely' when we compare categorical and scalar response frequencies (totals at bottom of respective columns).

Table 73. Summary of frequencies of categorical responses to each question (Cat) and the number of scalar questions responded as scalar (as opposed to a categorical response) (Sca)

	never		**rarely**		**Some-times**		**Freq-uently**		**always**		**(M)issing**	**TOTALS**	
	Cat	***Sca***	**Cat**	***Sca***	**Cat**	***Sca***	**Cat**	***Sca***	**Cat**	***Sca***	**Cat**	**Cat**	***Sca***
G3	13	*8*	16	*14*	20	*12*	10	*13*	20	*7*	1	**71**	***54***
R1	4	*0*	2	*1*	9	*4*	27	*16*	9	*13*	0	**71**	***34***
R2	12	*3*	4	*7*	14	*10*	23	*17*	14	*11*	2	**71**	***48***
R6	39	*9*	4	*15*	18	*10*	6	*7*	18	*3*	1	**71**	***44***
R7	37	*7*	2	*13*	15	*9*	9	*6*	15	*3*	5	**71**	***38***
R8	42	*9*	7	*13*	8	*6*	5	*6*	8	*4*	4	**71**	***38***
R9	20	*16*	26	*25*	10	*14*	12	*6*	10	*1*	3	**71**	***62***
R10	11	*6*	21	*18*	16	*18*	16	*9*	16	*3*	1	**71**	***54***
R12	14	*5*	7	*10*	18	*14*	23	*19*	18	*10*	0	**71**	***58***
S4	49	*11*	3	*12*	11	*2*	8	*5*	11	*4*	0	**71**	***34***
S5	41	*11*	5	*13*	8	*4*	7	*4*	8	*2*	10*	**71**	***34***
S7	61	*13*	6	*15*	0	*2*	3	*2*	0	*2*	1	**71**	***34***
S9	54	*10*	6	*13*	2	*3*	1	*0*	2	*0*	8*	**71**	***26***
S10	48	*7*	7	*11*	9	*7*	3	*7*	9	*4*	3	**71**	***36***
TOTALS	**445**	***115***	**116**	***180***	**158**	***115***	**153**	***117***	**158**	***67***	**39**	**994**	***594***

*mainly omitted by people who indicated that they do not drive a car.

From Table 74 (below), we can see, that although the patient can respond to all the questions in a scalar form if s/he wishes, only 31% are pondered as scalar, and only 15% have a high uncertainty response (membership grade > 0.09 for any one category). As part of the understanding of this finding, we have to take into account, that although each patient was explained how to fill in the two different types of questionnaire, and an explanatory section was included at the beginning of the questionnaire, there were, upon inspection, approximately 35% of patients who had filled in the scalar questionnaire totally with categorical responses, that is, placing a cross on the scale but exactly on the category boundary. On the other hand, there are people who when required to think introspectively in more intuitive terms, go for the categorical way of thinking as a preference. We could go as far as to say that this could reflect the type of personality – more deterministic or more reflexive thinking on the part of the patient.

Table 74. Frequency table of preference of scalar response with respect to categorical response

	Number of categorical responses	**Number of scalar responses**	**Number of scalar responses with high uncertainty**	**% of responses with high uncertainty**	**Number of responses with value missing** (scalar/cat)
G1	44	27	12	44	0
R1	54	17	8	47	0
R2	45	24	12	50	2
R6	49	22	11	50	0
R7	50	19	9	47	2
R8	48	19	12	63	4
R9	39	31	21	68	1
R10	44	27	14	52	0
R12	42	29	15	52	0
S4	54	17	6	35	0
S5	44	17	11	65	10*
S7	54	17	4	23	0
S9	48	13	3	23	10*
S10	49	18	8	44	4
TOTALS	**664**	**297**	**146**		**33**

*mainly omitted by people who indicated that they do not drive a car.

4.4.3 Learning and assignment of the weights

As commented previously, the relevance weights were assigned by two different methods: (i) learning by a genetic algorithm, and (ii) by the medical expert. The weights assigned by the medical expert can be seen in columns 2 and 3 of Table 72(a). In the case of the reliability weights, these were always assigned by the medical expert, and represent 'characteristic' curves, as previously explained in Section 3.2.3 of the thesis. The following details the values of the relevance weights learned from the categorical and the fuzzy data, respectively, and makes some observations with respect to the differences and resulting values.

The details of the execution of the genetic algorithm are as follows: the GA was run for 200 generations each dataset; a population size of 25 was used (the number of chromosomes) and the number of genes per chromosome was 19, which is, of course, equal to the number of weights and the number of corresponding variables. The crossover rate was set to 0.85 and the mutation rate was set to 0.01.

Diverse tests were run with different population sizes, generations, crossover rate and mutation rate, but the best found were those above, taking into account the memory and processing power restrictions of the PC which ran the tests. A test was also run to divide the chromosome en different sections, depending on the type of variable. Four subdivisions were tried: 'clinical data variables', and one for each questionnaire responses type, G, R or S. Crossover was only then allowed within each of these subsections, the objective being to keep the weight values of homogeneous variables together. In practise, no significant improvement was found by subdividing the chromosome in this manner, and the results published used a simple undivided chromosome structure.

Table 75. Weight values assigned by medical expert and by learning with genetic algorithm

Assignment method	Weight assignments of variables																		
	Age	**Sex**	**Neck**	**BMI**	**Somn**	**G3**	**R1**	**R2**	**R6**	**R7**	**R8**	**R9**	**R10**	**R12**	**S4**	**S5**	**S7**	**S9**	**S10**
Medical expert	0.65	0.60	0.87	0.80	0.80	0.60	0.75	0.90	0.85	0.95	0.85	0.63	0.80	0.70	0.75	0.90	0.85	0.85	0.85
Genetic algorithm learns from fuzzy data	0.05	0.47	0.05	0.05	0.42	0.42	0.16	0.47	0.05	0.37	0.21	0.21	0.16	0.37	0.32	0.47	0.16	0.11	0.16
Genetic algorithm learns from crisp data	0.11	0.47	0.05	0.05	0.47	0.21	0.16	0.21	0.16	0.37	0.11	0.21	0.26	0.16	0.37	0.21	0.42	0.32	0.47

Table 76. Agreement between different weight assignments

	Learned from categorical data	Learned from fuzzy data	Assigned by medical expert
Learned from categorical data	19*	10	7
Learned from fuzzy data	10	19	7
Assigned by medical expert	7	7	19

*number of variables assigned by one method whose weights are 'of the order of' the weights of the corresponding variables assigned by the other method.

From Tables 75 and 76 we can see that there is a significant difference in the assignment of the weights between methods and data types. There is less difference between crisp and fuzzy data types for genetic learning, but a greater difference between the learning methods and the medical expert. In particular, variables which are considered relevant by the medical expert, such as 'Neck' and 'BMI', are not considered relevant by the learning method. The medical expert assigned 0.87 and 0.80 to 'Neck' and 'BMI', respectively, whereas the learning method with crisp data assigned 0.05 to both variables, and the learning method with fuzzy data also assigned 0.05 to both variables.

On the other hand, there was an agreement by all methods for the variables, 'Age', 'Somnolence', 'R7', 'R9'and 'S4'. We take into account that the medical expert defined the weights within a restricted range, which was from 0.60 to 0.95, whereas the values of the learned weights ranged from 0.05 to 0.47. Thus we consider the minimum learned value of 0.05 equivalent to the minimum expert value of 0.60, and the maximum learned value of 0.47 as equivalent to the maximum expert value of 0.95. Also note that in order to input the weights to WOWA, they were normalised so that their sum was equal to 1. We can conclude that the GA learning process is not very precise for individual clinical variables, but tries to find an overall reasonable result. This agrees in general with the characteristics of GA's, in that a GA can find a reasonable result quickly, such as an overall diagnosis. One the other hand, a GA finds it more difficult to achieve a high precision, or specific sub-solutions such as those represented by the individual relevance weights of the variables.

4.4.4 Results: diagnosis using aggregation function

The data and the meta-information is given as input to the aggregation operator which 'fuses' all its inputs together in one single diagnosis output per patient. We have tried four variations: (i) scalar question representation; (ii) categorical question representation; (iii) 'relevance' weight assignment by medical expert; (iv) 'relevance' weights learned by genetic algorithm. From the complete Apnea dataset of 71 cases, 41 were randomly sampled for the training set and 30 were randomly sampled for the test set. The resulting diagnostic accuracy of permutations of these techniques executed against the test set is given for positive, negative and all cases in Table 77.

Table 77. Diagnostic accuracy on test dataset for positive, negative and all cases

	Positive Cases	Negative cases	All cases
Categorical question representation / weights** assigned by medical expert	0.735*	0.462	0.498
Categorical question representation / weights** learned by genetic algorithm	0.645	0.374	0.530
Scalar question representation / weights** assigned by medical expert	0.625	0.433	0.598
Scalar question representation / weights** learned by genetic algorithm	0.601	0.459	0.550

*correlation coefficients of predicted AHI with real AHI values.
**relevance weights

With reference to Table 77, we observe a typical result with greater accuracy for positive cases and lesser accuracy for the negative cases, with the expert fixed weights giving slightly better results than the genetically learned results. The results compare favourably with the literature[Hoffstein93][Young94][Ward97] for pure questionnaire based diagnosis of sleep apnea syndrome which tends to be in the order of 55% to 65% accuracy, and pure clinical data based diagnosis which is in the order of 70% to 90%. We think that, giving the genetic algorithm more evolutive time (we used only 15 generations) and a bigger population (we used 80 individuals) would give a better result for the learned weights.

4.4.5 Comparison of predictive accuracy of diagnosis using WOWA aggregation against other predictive modelling methods

In order to compare the method with other artificial intelligence predictive techniques we executed a neural network and a tree induction algorithm (ID3) against the same data, to predict the degree of apnea-hypopnea (AHI). The neural network was a standard feed-forward net with 3 layers, and the rule induction was run with unlimited tree depth and minimum of 5% of cases to form a branch. As previously, we divided the data into a random sampled 58% training set (41 cases) and 42% test set (31 cases). The results of executing against the test set are resumed in Table 78, in which we see that WOWA aggregation performs better than neural nets and tree induction overall and for positive cases. For negative cases, WOWA performs worse than tree induction and slightly better than neural nets. In general neural nets and tree induction techniques require larger data volumes in order to build models, whereas the weighted aggregation approach should produce reasonable results with much fewer cases.

Table 78. Comparison of the predictive accuracy of Neural Net, ID3 Tree Induction and WOWA algorithms with the Apnea test dataset

	Neural Net	Tree Induction	WOWA
All cases (test)	0.540*	0.548	0.598
Positive cases	0.600	0.523	0.735
Negative cases	0.450	0.625	0.462

*correlation coefficients of predicted AHI value and real AHI value

4.4.6 Summary of Section 4.4

From the questionnaire responses we can see interesting tendencies emerging of the way in which patients respond to the questions, depending on the type of the question, and the strength of the response required. In some cases a question can provoke more of a scalar response ('shades of grey') while in other cases the question provokes a more 'black or white' response from the patient. With respect to the diagnostic accuracy, we can see a promising result, achieved with few cases and a wide dimensionality of problem (19 variables). We have also been able to include three types of meta-data as part of the processing, thus adding insight which may improve the end result. From the point of view of 'medical informatics', we have learnt that careful selection of an adequate medical application is fundamental; one criteria for choosing an application is that is must allow real scope for improvement with respect to existing methods. Also, collaboration with medical experts has as a prerequisite, the need for sufficient availability of their time for initial definition of the meta-data, selection of variables and later analysis of the feedback of the results. The data quality and how representative a sample is, are also key aspects, together with the challenge of obtaining and capturing real case data in situ from the hospital environment.

Chapter 5. Conclusions

In this work we have reviewed some of the problems which exist for the representation of real data, we have considered the selection of key descriptive variables, and the aggregation and modelling of variables and data. Throughout the text of the thesis, we have contrasted the major issues and authors, current approaches, theoretic and practical historical background, the authors original development work and the application and results of that work.

We have developed and refined a collection of methods and tools which can be applied to the different steps of the Data Mining process, consisting of tools for data exploration and analysis on the one hand, and data modelling, classification and prediction, on the other hand.

In Chapter 1 we saw an initial summary of some of the current approaches and methods, and the limitations of those approaches. One of the first problems considered was that of the representation of the data. In the current literature, a diversity of different conceptual representations are evident, as well as many variants of data processing algorithms based on neural networks, rule induction, genetic algorithms, and so on.

Chapter 3 has developed several themes: the consideration of basic principles of the nature of data, data representation and processing approaches general in concept but specific in their application to two clinical datasets: ICU Prognosis and Apnea Screening. The considerations developed with respect to how to compare variables of different data types, leads on to the work on 'fuzzy covariance', which in turn leads on to the consideration of aggregation operators with three types of meta-data: 'relevance', 'reliability' and 'grade of membership'. In the field of data representation we have considered from first principles the nature of data, the different types it can have, and possible ways of comparing different types and processing it. We have refined methods for defining membership functions for data capture, and specially designed a questionnaire using these methods.

In Section 4.1, the ICU data is processed first by standard statistical and AI techniques, then by Fuzzy c-Means and finally Hartigan's joining algorithm using a new way of calculating fuzzy covariances. The results have permitted a richness of comparison between fuzzy and crisp forms of processing. The Hartigan method using fuzzy covariances extracted 4 factors, whereas the Hartigan method with crisp covariances extracted 3 factors. It was found that it was easier to find clinical meaning in these factors, rather than in the rules generated by exhaustive processing of the same data with C4.5. Also the processing of the data with fuzzy c-Means, and using principal components to process the membership grades and visualise the clusters in a two dimensional space, identified 3 clinically interpretable groupings of cases.

Section 4.2 presented the benchmarking of a set of four novel fuzzy covariance algorithms with artificial test datasets. We note that in the literature there are very few general algorithms which allow the calculation of fuzzy covariances between variables defined in the fuzzy form. The majority of algorithms are very application specific, as detailed in Chapters 1 and 2, and the few general ones are very complex. The first method proposed in Section 4.2, which measures the fuzzy grade of relation between variables and the cluster centre, gave comparable results to C4.5, for ranking of variables by relevance. The third method proposed in Section 4.2, measures the relation between distances of objects from the cluster centre, weighted by the norm coefficients, and the fourth method proposed measures the relation between the sum of squares of the distances of objects from the cluster centre, weighted by the norm coefficients and the membership grades. We found that the ranking of variables for relevance by the third and fourth methods coincided with SPSS covariances for all datasets.

The work on Apnea diagnosis provides an alternative approach to data processing for a small number of cases, including meta-data about the data. The fuzzy data capture techniques also provides a powerful tool when combined with the questionnaire screening method. In the case of Section 4.3, we found that Apnea diagnosis by WOWA aggregation using weight assignment by medical expert and data analyst, gave the best results: 0.81 correlation for positive cases and 0.67 correlation for negative cases, which is favourable to results found in the medical informatics literature. Tests with the 'bias' characteristic curve modification to WOWA, showed that WOWA gave the best results, when compared to OWA and Principal Components; for one borderline case, two positive cases and one negative case, WOWA was the only method to get all the diagnoses correct.

The final work in Section 4.4, gave best overall results for the Scalar Questionnaire method and weight assignment by the expert (as opposed to Categorical questionnaire method and weight assignment by statistical or genetic learning methods). This resulted in an overall correlation of the diagnosis of 0.598, which is favourable when compared to results in the medical informatics literature. When WOWA was compared to neural networks and tree induction, run on the same data, it gave the best overall result (positive and negative cases), being 0.598 for WOWA, compared to 0.54 for neural network and 0.548 for tree induction.

We can say, in conclusion, that our methods possess an advantage with respect to standard 'data mining' tools, because they make use of a natural way of representing the data, some mechanism for giving additional information to the processing algorithm (such as the reliability and relevance weights), and an algorithm which allows a non-deterministic grouping and distance measure, for classification or clustering, prognosis or diagnosis.

In the context of data mining, we have laid a promising and novel foundation for a data analysis and data representation toolkit which offers additional insight and dimensionality to the data. In our data mining toolkit, which would requiere the integration of the various algorithms in a single user interface, we have tools for data representation, data exploration, and data modelling. Variable selection is a major part of the first stages of data mining. We have used a new fuzzy covariance calculation together with Hartigan's joining algorithm to define a ranking of variables in terms of their relevance; also this method allows us to identify interrelations between the variables. We have been able to compare results of analysis of a complex ICU dataset using diverse standard data mining methods, and comparing this to results of applying fuzzy c-Means and Hartigan's Joining algorithm.

In the field of data aggregation, we have carried out new developments to the WOWA aggregation operator. A novel weights learning method has been applied, using genetic algorithms. A modification has been made to allow WOWA to process data effectively with missing values. We have used WOWA for a previously untried application, that of Apnea diagnosis, being especially apt for data processing with a small number of cases. Finally, we have compared WOWA with other aggregation methods, WM and OWA, and compared the genetic algorithm approach with other learning techniques such as ASM (Active Set Methods). We concluded that ASM based methods are appropriate for learning the weights for the WM and the OWA operators because in this case the minimisation problem is a quadratic one and almost exact solutions can be found. This is not the case when weights are learned for the WOWA operator. In this case, the complexity of ASMs increases because the function to minimise is not quadratic. This is due to the existence of the interpolation function w^* (built from the weighting vector w - one of the weighting vectors to learn) and due to the fact that this function is applied to additions of some of the p's (the other weighting vector to learn). The use of genetic algorithms presents an additional advantage for either the WM, OWA or WOWA operators. This is the case when data files include missing values or the number of variables is different in each example.

As possible future lines of work, the unification of these algorithms and methods into a single application would be the theme for a future project. Also, with reference to the Apnea diagnosis project in collaboration with the Hospital of the Santisima Trinitat, Salamanca, a process of collecting new cases to double the size of the train and test data sets has been proposed. This would allow us to look more closely at the reasons for weaker performance in the negative cases of the apnea data.

We hope that the information which is summarised in the work will serve as a new reference for those who need to analyse clinical data. Also, the diagnosis methods, questionnaire design and data capture create new alternatives for those who wish to use questionnaires for Apnea screening. We have been able to offer improvement for questionnaire screening by the quantification of reliability and relevance information, together with a way of capturing the natural fuzziness of the responses.

6. Annexes

Annex 1.1. Bibliographic revision of publications in the field by the author: 1996 - 2001

Since 1996 I have published 8 papers in specialised academic journals and conferences on specific technical aspects of data processing and representation, with special emphasis in the medical data domain, for diagnosis and prognosis. I have also published 4 papers in journals and conferences on more general data mining themes. In chronological order, the publications are a reasonable representation of the evolution of my work from the initial mixed data types studies in 1996-1997, to the WOWA studies from 1998-2001. There has been a constant theme throughout, of the use of medical patient data from diagnosis and prognosis, first in ICU survival analysis and later in Apnea patient screening. I have considered a complementary aspect, that of being up to date in the latest commercial data mining tools in the field, which have been used for cross checking results and benchmarking data quality in several papers.

PUBLICATIONS

[Nettleton96] Nettleton, D.F., "*Data Mining en el entorno hospitalario"*. Novatica, Spain, pp. 69-73, 1996.

[Nettleton97] Nettleton, D.F., Gibert, K. "*Fusión de atributos con técnicas fuzzy en Data Mining"*. ESTYLF '97. VII Congreso Español de Lógica Difusa, Tarragona, Spain, pp. 217-220, 1997.

[Nettleton98a] Nettleton, D.F., "*Representación, fusión e interpretación de atributos con técnicas fuzzy.* ACIA '98. 1º Congreso Catalán de Inteligencia Artificial, Tarragona, Spain, pp. 185-187, 1998.

[Nettleton98b] Nettleton, D.F., "*Fuzzy covariance analysis, aggregation and input selection for fuzzy data".* IKBS '98. International Conference on Knowledge Based Computer Systems. Mumbai, India, pp. 261-272, 1998.

[Nettleton99a] Nettleton, D.F., "*El uso de tecnología de Minería de Datos para la construcción y explotación del Data Warehouse".* Novatica, Spain, pp.52-55, 1999.

[Nettleton99b] Nettleton, D.F., "*Variable fusion using a heterogeneous representation of crisp and fuzzy medical data".* IFSA '99. Eighth International Fuzzy Systems Association World Congress, Taipei, Taiwan, Vol II, pp. 618-623, 1999.

[Nettleton99c] Nettleton, D.F. , Hernandez, L., "*Evaluating reliability and relevance for WOWA aggregation of Sleep Apnea case data".* EUSFLAT '99 - Congress of the European Society of Fuzzy Logic and Technology, Palma de Mallorca, Spain, pp. 283-286, 1999.

[Nettleton99d] Nettleton, D.F. , "*Data Mining y Márketing a través de Internet".* Márketing & Ventas, Ediciones Deusto, Sept./Oct., pp. 22-26, 1999.

[Nettleton99e] Nettleton, D.F., Hernandez, L., "*Questionnaire screening of sleep apnea cases using fuzzy knowledge representation and intelligent aggregation techniques".* IDAMAP '99. Workshop 'Intelligent Data Analysis in Medicine and Pharmacology', Washington DC, United States, pp.91-102, 1999.

[Nettleton00a] Nettleton, D.F., Fandiño, V.L., Witty, M. Vilajosana, E. "*The use of a data mining workbench for macro and micro economic modelling".* Conference 'Data Mining 2000', Cambridge University, U.K., July 5-7, pp.25-34, 2000.

[Nettleton01a] Nettleton, D.F., Muñiz, J., "*Processing and representation of meta-data for sleep apnea diagnosis with an artificial intelligence approach".* International Journal of Medical Informatics, 63 (1-2), pp.77-89, Elsevier, Sept. 2001.

[Nettleton01b] Nettleton, D.F. , Torra, V. "*A comparison of active set method and Genetic algorithm approaches for learning weighting vectors in some aggregation operators".* International Journal of Intelligent Systems, Vol. 16, Nº. 9, Wiley Publishers, Sept. 2001.

Annex 1.2. General bibliographic references

[Aczél84] Aczél, J. *"On weighted synthesis of judgements".* Aequationes Math., 27, pp. 288-307, 1984.

[Almuallim91] Almuallim, H., Dietterich, T. *"Learning with many irrelevant features".* Proc. AAAI-91, Anaheim, CA., MIT Press, Cambridge, MA., pp.547-552, 1991.

[Aguilar91] Aguilar-Martín, J. , Gibert-Oliveras, K., *"Sobre Variables Linguisticas Difusas, Paradigmas Parmenidianos y Logicas Multivaluadas"*. Primer Congreso Español sobre Tecnologías y Lógica Fuzzy. Universidad de Granada (1991) pp185-192.

[Antonisse87] Antonisse, H.J., Keller, K.S. *"Genetic Operators for High Level Knowledge Representation"*. Proceedings of the Second International Conference on Genetic Algorithms, Lawrence Erlbaum Associates, Hillsdale, NJ, 1987, pp. 69-76.

[Armengol00] Armengol, E., Palaudáries, A., Plaza, E. *"Individual prognosis of diabetes long-term risks: A CBR Approach"*. IIIA Research Report 2000-04, IIIA-CSIC, Campus UAB, Bellaterra, Catalonia, EU, 2000.

[Aymerich97] Aymerich, F., Sobrevilla, P., Gili, J., Montseny, E. *"Utilización de técnicas difusas para la detección de lesiones pequeñas de esclerosis múltiple"*. VII Congreso Español sobre Tecnologias y Logica Difusa, Tarragona, pp. 153-158, 1997.

[Babuska96] Babuska, R., Setnes, M., Kaymak, U., van Nauta Lemke, H. *"Rule based simplification with similarity measures".* Proceedings FUZZ-IEEE'96, pp.1642-1647, New Orleans, USA, Sept. 1996.

[Bajula97] Bajula, S., Pomerleau, D. *"Dynamic relevance: vision-based focus of attention using artificial neural networks".* Artificial Intelligence , Vol. 97, #1-2, Ed. Elsevier, December 1997.

[Baldwin95] Baldwin, J., Martin, T. "Fuzzy modelling in an intelligent data browser". FUZZ-IEEE '95, pp. 1171-1176, Yokohama, Japan, 1995.

[Béjar94] Béjar, J. *"Adquisición de Conocimientos en Dominios Poco Estructurados".* PhD Thesis. Departament of Computer languages and Systems, University Polytechnic of Catalunya, 1994.

[Bersini97] Bersini, H., Duchateau, A., Bradshaw, N. *"Using incremental learning algorithms to search for minimal and effective fuzzy models"*. Sixth IEEE International Conference on Fuzzy Systems, Barcelona, Vol. III, pp. 1417-1422, 1997.

[Bezdek73] Bezdek, J.C. *"Fuzzy mathematics in pattern classification".* Ph.D. dissertation, Appl. Math., Cornell Univ., Ithaca, NY, 1973.

[Bezdek76] Bezdek, J.C. *"Feature Selection for Binary Data: Medical Diagnosis with Fuzzy Sets".* Proc. 25th National Comp. Conf. (S.Winkler, Ed.), pp1057-1068, AFIPS Press, Montvale, New Jersey, (1976).

[Bezdek77] Bezdek, J.C. and Castelaz, P.F. *"Prototype Classification and Feature Selection with Fuzzy Sets".* IEEE Trans. Sysy. Man Cybern., Vol. SMC-7, no. 2, pp. 87-92, Feb. 1977.

[Bezdek78] Bezdek, J.C., Harris, J.D. *"Fuzzy partitions and relations; an axiomatic basis for clustering".* Fuzzy Sets and Systems, vol. 1, pp. 111-127, 1978.

[Bezdek80] Bezdek, J.C. *"A convergence theorem for the fuzzy ISODATA clustering algorithms".* IEEE Trans. Pattern Anal. Machine Intell., vol. PAMI-2, no. 1, pp. 1-8, January 1980.

[Bezdek81] Bezdek, J.C. *"Pattern recognition with Fuzzy Objective Function Algoritms".* S13, pp86, 'Feature Selection for Binary Data: Important Medical Symptoms'. Plenum Ress, 1981.

[Bezdek87] Bezdek, J.C., Hathaway, R.J., Sabin, M.J., Tucker, W.T. *"Convergence theory for Fuzzy c-Means: counterexamples and repairs".* IEEE Trans. Syst., Man, Cybern., vol. SMC-17, no. 5, pp. 873-877, Sept./Oct. 1987.

[Bilgiç97] Bilgiç, T., Türksen, B. *"Elicitation of membership functions: how far can theory take us?".* Sixth IEEE International Conference on Fuzzy Systems, Barcelona, Vol. III, pp. 1321-1325, 1997.

[Blum97] Blum, A., Langley, P. *"Selection of relevant features and examples in machine learning".* Artificial Intelligence , Vol. 97, #1-2, pp.245-271, Ed. Elsevier, December 1997.

[Boixader97] Boixader, Recasens, Jacas. *"Similarity-based approach to defuzzification".* Sixth IEEE International Conference on Fuzzy Systems, Barcelona, Vol. 2, pp. 761, 1997.

[Borgelt97a] Borgelt. *"Evaluation measures for learning probabilistic and possibilistic networks".* Sixth IEEE International Conference on Fuzzy Systems, Barcelona, Vol. 1, pp. 669, 1997.

[Bosc97] Bosc,P.,Pivert, O. *"On the comparison of imprecise values in fuzzy databases".* Sixth IEEE International Conference on Fuzzy Systems, Barcelona, Vol. II, pp. 707-712, 1997.

[Bowen95] Bowen, J. y Dozier, G. "*Solving Contraint Satisfaction Problems Using a Genetic/Systematic Search Hibrid that Realizes when to Quit".* Proceedings of the Sixth International Conference on Genetic Algorithms, Morgan Kaufmann, San Mateo, *CA*, (1995), pp.122-129.

[Branco94] Branco, A., Silva, L., Evsukoff, A., Aragon, D. "An Expert System applied to the petroleum industry". II World Conress on Expert Systems, 1994.

[Breiman84] Breiman, L. , Friedman, J.H., Olshen, R.A. and Stone, C.J. *"Classification and Regression Trees"*. Belmont, CA: Wadsworth, 1984.

[Bouaziz96] Bouaziz, T., Wolski, A. *"Incorporating fuzzy inference into database triggers".* Research report Nº TTE1-2-96, VTT Information Technology, Espoo, Finland, Nov. 1996.

[Calvo00] Calvo, T., Mayor, G., Torrens, J., Suer, J., Mas, M., Carbonell, M. *"Generation of weighting triangles associated with aggregation functions".* Int. J. of Unc., Fuzziness and Knowledge Based Systems, 8:4, 417-451, 2000.

[Cartright91] Cartright, H.M., y Mott, G.F. *"Looking Around: Using Clues from the Data Space to Guide Genetic Algorithm Searches".* Proceedings of the Fourth International Conference on Genetic Algorithms, Morgan Kaufmann, San Mateo, CA, (1991), pp.108-114,

[Castro98] Castro, J., Castro-Sánchez, J., Espin, A., Zurita, J. *"A methodology for developing knowledge based systems".* Mathware, Vol. V, Nº 2-3, pp343-353, 1998.

[Chang77] Chang, R., Pavlidis, T. "Fuzzy decision tree algorithms". IEEE Trans. Syst., Man, Cybern., 1977.

[Chen95] Chen, J.E., Otto, K.N., *"Constructing membership functions using interpolation and measurement theory"*. Fuzzy Sets and Systems, Vol. 73:3 (1995) pp313-327.

[Chen97] Chen, P., Nasu, M., Toyota, T. *"Sequential self-reorganization method of symptom parameters and identification method of membership function for fuzzy diagnosis".* Sixth IEEE International Conference on Fuzzy Systems, Barcelona, Vol. I, pp. 433-440, 1997.

[Cordon97] Cordon, O., del Jesus, M., Herrera, F. *"Nuevos métodos de razonamiento en sistemas de clasificación basados en reglas difusas".* VII Congreso Español sobre Tecnologias y Lógica Difusa, Tarragona, pp. 109-114, 1997.

[Cross95] Cross, V. *"Fuzzy extensions to the object model".* Proceedings of the IEEE Systems, Man and Cybernetics Conference, Vancouver, Canada, October 1995.

[Cuadras80] Cuadras, C.M. "*Métodos de Análisis Multivariante"*. Vols I y II, Chapman-Hall, 1980.

[Czogala97] Czogala, E., Leski, J., Rozentryt, P., Zembala, M.. *"Entropy measure of fuzziness in detection of QRS Complex in noisy ECG signal".* Sixth IEEE International Conference on Fuzzy Systems, Barcelona, Vol. II, pp. 853-856, 1997.

[Delgado93] Delgado, M., Verdegay, J.L., Vila, M.A. *"On Aggregation Operations of Linguistic Labels".* International Journal of Intelligent Systems, Vol. 8, 351-370 , John Wiley & Sons, 1993.

[Delgado95] Delgado, M., Gómez Skarmeta, A., Martín, F. "Generating fuzzy rules using clustering based approach". Third European Congress on Fuzzy and Intelligent Technologies and Soft Computing, pp.810-814, Aachen, Germany, August 1995.

[Dement78] Dement WC et al. *"Excessive daytime sleepiness in the sleep apnoea syndrome".* In Guilleminot...Sleep apnoea syndromes. New York Alan R. Liss 1978; pp 23-46

[Demsar99] Demsar, J., Zupan, B., Aoki, N., et al. *"Feature mining and predictive model construction from severe trauma patient's data".* Workshop Intelligent Data Analysis in Medicine and Pharmacology, pp.32-41. AMIA, 99. Washington, DC, Nov. 1999.

[Devlin97] Devlin, B. "*Data Warehouse - from Arquitecture to Implementation".* Addison-Wesley, 1997.

[Dreiseitl99] Dreiseitl, S., Ohno-Machado, L., Vinterbo, S. "Evaluating variable selection methods for diagnosis of myocardial infarction". Proc. AMIA Symposium 99, Symposium Supplement of the Journal of the American Medical Informatics Association, pp.246-250, Pub. Hanley&Belfus Inc., Nov. 1999.

[Dubes88] Dubes, R., Jain, A. *"Algorithms for clustering data".* Prentice Hall, 1988.

[Dubois80] Dubois, D. *"Triangular norms for fuzzy sets".* Proc. 2nd Int. Seminar on Fuzzy Set Theory. Linz, Austria, 1980. P39-68.

[Dubois91] Dubois, D., Prade, H., Mo, X. *"Fuzzy discriminant trees".* In Fuzzy Engineering toward Human Friendly Systems (IFES '91), Vol. I, pp250-259, 1991.

[Dubois97] Dubois, D., Prade, H., Rannou, E. *"User-driven summarization of data based on gradual rules".* Sixth IEEE International Conference on Fuzzy Systems, Barcelona, Vol. II, pp. 839-844, 1997.

[Duda73] Duda, R., Hart, P. *"Pattern Classification and Scene Analysis".* New York: Wiley, 1973.

[Dunn74] Dunn, J.C. *"A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters".* J. Cybern., vol. 3, pp32-57, 1974.

[Duran96] Duran J, et al. *"Prevalence of obstructive sleep apnea in the male population of Vittoria-Gasteiz (Spain)".* Eur Respir J 1996; 9: Suppl 23, 156s

[El-Sonbaty98] El-Sonbaty, Yasser, Ismail, M.A. "*Fuzzy Clustering for Symbolic Data"*. IEEE Transactions on Fuzzy Systems, Vol. 6, Nº 2, pp. 195-204, May 1998.

[Escalada99] Escalada, G., Jaureguizar, J. *"Knowledge Based System for Real Time Physiopathological Diagnosis in a Critical Care Setting".* IIIA Research Report 99-19, IIIA-CSIC, Campus UAB, Bellaterra, Catalonia, EU, 1999.

[Filev98] Filev, D., Yager, R.R. *"On the issue of obtaining OWA operator weights",* Fuzzy Sets and Systems, 94 (1998), 157-169.

[Fisher87] Fisher, D. *"Knowledge acquisition via incremental conceptual clustering".* Machine Learning Journal, 2, pp139-172, 1987.

[Flemonds97] Flemonds WW, et al. *"Clinical prediction of the sleep apnoea syndrome".* Sleep Med Rev 1997; 1: 19-32

[Flores-Sintas97] Flores-Sintas, A., Cadenas, J., Martin, F. *"Test del algoritmo fuzzy-minimals con datos reales".* VII Congreso Español sobre Tecnologias y Logica Difusa, Tarragona, pp. 171-176, 1997.

[Fodor94] Fodor, J., Roubens, M., (1994), "*Fuzzy Preference Modelling and Multicriteria Decision Support"*, Kluwer Academic Publishers, Dordrecht, The Netherlands.

[Fodor95] Fodor, J., Marichal, J.-L., Roubens, M., *"Characterisation of the Ordered Weighted Averaging Operators",* IEEE Trans. on Fuzzy Systems, 3:2 (1995) 236-240.

[Fogel66] Fogel, L.J., Owens, A.J., and Walsh, M.J., "*Artificial Intelligence Through Simulated Evolution"*, John Wiley, Chichester, UK, 1966.

[Forrest85] Forrest, S., *"Implementing Semantic Networks Structures using the Classifier System".* Proceedings of the First International Conference on Genetic Algorithms, Lawrence Erlbaum Associates, Hillsdale, NJ, 1985, pp. 24-44.

[Fox91] Fox, B.R., McMahon, M.B. "*Genetic Operators for Sequencing Problems".* First Workshop on the Foundations of Genetic Algorithms and Classifier Systems, Morgan Kaufmann, 1991 (Ed. Rawlins, G.), pp. 284-300.

[Friedman74] Friedman, J.H., Tukey, J. *"A projection persuit algorithm for exploratory data analysis".* IEEE Transactions on Computers, Vol. C-23, Nº 9, pp881-890, 1974.

[Friedman77] Friedman, J.H. *"A Recursive Partitioning Decision Rule for non-Parametric Classification".* IEEE Transactions on Computers, pp404-408, 1977.

[Fujimoto95] Fujimoto, T.*"A study of constructing Situation Database".* Master's thesis, Department of Systems Science, Tokyo Institute of Technology, Feb. 1995.

[Fujimoto97] Fujimoto, T., Sugeno, M. *"Clustering verb, adjective, adjective-verb concepts using proximity relation".* Proc. Sixth IEEE International Conference on Fuzzy Systems, Volume I, pp. 231-234. (1997).

[Galles97] Galles, D., Pearl, J. *"Axioms of causal relevance".* Artificial Intelligence , Vol. 97, #1-2, Ed. Elsevier, December 1997.

[Gerstorfer97] Gerstorfer, E., Hellendoorn, H. *"On the Role of Fuzzy Logic in Production Planning"*. Proceedings of the Sixth IEEE International Conference on Fuzzy Systems, Volume III, pp. 1271-1275. (1997).

[Gibert94] Gibert, K., Cortés, U. *"Combining a Knowledge Based System and a Clustering Method for a Construction of Models in Ill-Structured Domains.".* Cap. V Selecting Models from Data. Lecture Notes in Statistics vol. 89. P Cheeseman and R.W. Oldford Eds. Springer-Verlag. New York, 1994. pp351-360.

[Gibert97] Gibert, K., Cortés, U. *"Weighing Quantitative and Qualitative Variables in Clustering Methods".* Journal Mathware and Soft Computing 4 (2), número especial de Gener 1997.

[Gill81] Gill, P.E., Murray, W., Wright, M. H., "*Practical Optimization"*, Academic Press, 1981.

[Girard96] Girard R., Ralambondrainy, H. *"Conceptual classification from imprecise data".* Information Processing and Management of Uncertainty in Knowledge-based Systems (IPMU'96), Vol 1., pp. 247-252, Granada, Spain, 1996.

[Glorennec94] Glorennec, P., *"Fuzzy Q-Learning and Dynamical Fuzzy Q-Learning".* Proc. of the 3rd IEEE Int. Conf. on Fuzzy Systems, Orlando, June 1994.

[Glover97] Glover, F.,*"Heuristics for Integer Programming Using Surrogate Constraints",* Decision Sciences, Vol.8, Nº.1, pp-156-166, 1997.

[Gonzalez97] Gonzalez, A., Pérez, R. *"Using information measures for determining the relevance of the predictive variables in learning models".* Sixth IEEE International Conference on Fuzzy Systems, Barcelona, Vol. III, pp. 1423-1428, 1997.

[Ghosh95] Ghosh, A. *"Use of fuzziness measures in layered networks for object extraction : a generalization"*, Fuzzy Sets and Systems, vol.72, no. 3, pp.331-348, 1995

[Grabisch95] Grabisch, M., Nguyen, H. T., Walker, E. A., *"Fundamentals of Uncertainty Calculi with Applications to Fuzzy Inference",* Kluwer Academic Publishers, Dordreht, The Netherlands, 1995.

[Greiner97] Greiner, R., Grove, A., Kogan, A. *"Knowing what doesn't matter:exploiting the omission of irrelevant data".* Artificial Intelligence , Vol. 97, #1-2, Ed. Elsevier, December 1997.

[Guilleminault92] Guilleminault C., Stoohs R., Clerk A et al. *"From obstructive sleep apnea syndrome to upper airway resistance syndrome: consistency of daytime sleepiness".* Sleep. 1992; 15: 513-516

[Gustafson79] Gustafson, D.E., and Kessel, W., *"Fuzzy Clustering with a Fuzzy Covariance Matrix",* in Proc. IEEE-CDC, Vol. 2 (K.S. Fu, Ed.), pp761-766, IEEE Press, Piscataway, New Jersey, (1979).

[Hartigan75] Hartigan, J.A. *"Clustering algorithms",* New York: John Wiley & Sons, Inc., 1975.

[Hartigan77] Hartigan, J.A. *"Distribution problems in clustering".* Classification and Clustering, ed. J. Van Ryzin, New York: Academic Press, Inc., 1977.

[Hartigan78] Hartigan, J.A. *"Asymptotic distributions for clustering criteria".* Annals of Statistics, 6, 117-131, 1978.

[Hartigan79] Hartigan, J.A., Wong, M. *"A k-means clustering algorithm: algorithm as 136".* Applied Statistics, 28, pp126-130, 1979.

[Hartigan81] Hartigan, J.A. *"Consistency of single linkage for high-density clusters".* Journal of the Americal Statistical Association, 76, 388-394, 1981.

[Hartigan85a] Hartigan, J.A. *"Statistical theory in Clustering".* Journal of Classification, 2, 63-76. 1985.

[Hartigan85b] Hartigan, J.A. and Hartigan, P.M. *"The dip test of unimodality".* Annals of Statistics, 13, 70-84, 1985.

[Hathaway96] Hathaway, R.J., Bezdek, J.C., Pedrycz, W. *"A Parametric Model for Fusing Heterogeneous Fuzzy Data".* IEEE Transactions on Fuzzy Systems, Vol. 4, Nº 3, Agosto 1996.

[Herrera95] Herrera, F., Herrera-Viedma, E., Verdegay, J.L., *"Basis for a Consensus Modeling Group Decision Making with linguistic Preferences",* presented at EUFIT'95, Aachen, Germany, August 28-31, 1995.

[Herrera96] Herrera, F., Herrera-Viedma, E., Verdegay, J.L., *"A model of consensus in group decision making under linguistic assessments"*, Fuzzy Sets and Systems, 78, 73-87, 1996.

[Herrera97] Herrera F., Herrera-Viedma E., Verdegay J. L*., "A rational consensus model in group decision making using linguistic assessments"*, Fuzzy Sets and Systems 88, 31-49, 1997.

[Hoffstein93] Hoffstein, V., Szalai J.P. *"Predictive value of clinical features in diagnosing obstructive sleep apnea".* Sleep 1993; 16: 118-122.

[Hooper97] Hooper, B., Hu, X., Jaros, G.,Baker, B. *"A fuzzy logic based decision support system for low-flow / closed-loop anaesthesia"*. Sixth IEEE International Conference on Fuzzy Systems, Barcelona, Vol. III, pp. 1615-1620, 1997.

[Hunt75] Hunt, E.B. *"Artificial Intelligence".* Academic Press, New York, 1975.

[IBM96] IBM Data Management Solutions White Paper. IBM Corp., 1996.

[IIIA96] SMASH Project. *"Systems of Multiagents for Medical Services in Hospitals".* TIC96-1038-C04-01, IIIA-CSIC, Campus UAB, Bellaterra, Catalonia, EU, 1996. (www.iiia.csic.es/Projects/smash).

[Imai00] Imai, H., Miyamori, M., Miyakosi, M., Sato, Y., *"An algorithm Based on Alternative Projections for a Fuzzy Measures Identification Problem"*, Proc. of the Iizuka Conference, Iizuka, Japan (CD-Rom), 2000.

[Irani95]. Irani, E., Slagle, J., and the Posch Group. *"Automating the Discovery of Causal Relationships in a Medical Records Database".* Knowledge Discovery in Databases. Ed. Shapiro, G., Frawley, W., 1995.

[Inuiguchi97] Inuiguchi, M., Sakawa, M., Ushiro, S. *"Mean-absolute-deviation-based fuzzy linear regression analysis by level sets automatic deduction from data".* Sixth IEEE International Conference on Fuzzy Systems, Barcelona, Vol. II, pp. 829-834, 1997.

[Jacas88] Jacas, J. *"On the generators of t-indistinguishability operator".* Stochastica, XII-1 (1988), 49-63.

[Jacas90] Jacas, J. *"Similarity relations – the calculation of minimal generating families".* Fuzzy Sets and Systems 35 (1990), 151-162, North-Holland.

[Jacas93] Jacas, J. *"Igualdades borrosas".* Arbor CXLVI, 573-574 (September-October 1993), p137-146.

[Jacas95] Jacas, J., Recasens, J. *"Fuzzy t-transitive relations: eigenvectors and generators".* Fuzzy Sets and Systems 72 (1995), p147-154.

[Juang97] Juang, C., Lin, C. *"A recurrent self-organizing neural fuzzy inference network".* Sixth IEEE International Conference on Fuzzy Systems, Barcelona, Vol. III, pp. 1369-1374, 1997.

[Keller00] Keller, A., Klawonn, F. *"Fuzzy clustering with weighting of data variables".* International Journal of Uncertainty and Fuzzy Knowledge Systems, December 2000.

[Kahraman97] Kahraman, C., Ulukan, Z. *"Continuos Compounding in Capital Budgeting using Fuzzy Concept".* Sixth IEEE International Conference on Fuzzy Systems, Barcelona, Vol. III, pp. 1451-1455, 1997.

[Katz90] Katz I., Stradling J., Slutsky A.S., et al. *"Do patients with sleep apnea have thick necks?"* American Review of Respiratory Diseases, 1990; 141: 1228-1231.

[Kaufman90] Kaufman, L., Rousseeuw, P.J. "*Finding Groups in Data, an Introduction to Cluster Analysis".* Wiley, 1990.

[Khang99] Khang, T., Phuong, N. "Using hedge algebras for constructing inference mechanism in medical expert systems". IFSA '99. Proc. 8th Int. Fuzzy Systems Association World Congress, Taipei, Taiwan, Vol I., pp.265-268, 1999.

[Khardon97] Khardon, R., Roth, D. *"Defaults and relevance in model-based reasoning".* Artificial Intelligence , Vol. 97, #1-2, Ed. Elsevier, December 1997.

[Kim96] Kim, J., Jang, W., Bien, Z. "A dynamic gesture recognition system for the korean sign language". IEEE Trans. on Systems, Man, and Cybernetics, pp. 354-359, April 1996.

[Kira92] Kira, K., Rendell, L. *"A practical approach to feature selection".* Proc. 9th Int. Conf. on Machine Learning, pp-249-256, Aberdeen, Morgan-Kaufmann Pub., 1992.

[Kivinen97] Kivinen, J., Warmuth, M., Auer, P. *"The Perceptron algorithm versus Winnow: linear versus logarithmic mistake bounds when few input variables are relevant".* Artificial Intelligence , Vol. 97, #1-2, Ed. Elsevier, December 1997.

[Knaus81] Knaus, W. , Zimmerman, J. , Wagner, D., Draper, E., Lawrence, D. *"APACHE – Acute Physiology and Chronic Health Evaluation: a physiologically based classification system".* Critical Care Medicine, 1981;9:591-7.

[Kohavi97] Kohavi, R., John, G. *"Wrappers for feature subset selection".* Artificial Intelligence , Vol. 97, #1-2, pp.273-324, Ed. Elsevier, December 1997.

[Kohonen84] Kohonen, T. *"Self-organization and associative memory".* Berlin. Springer-Verlag, 1984.

[Koza90] Koza, J.R., *"Genetic Programming: A Paradigm for Genetically Breeding Populations of Computer Programs to Solve Problems",* Report Nº. STAN-CS-90-1314, Stanford University, 1990.

[Kushida97] Kushida CA, et al. *"A predictive morphometric model for the obstructive sleep apnoea syndrome".* Ann Inter Med 1997; 127: 581-587

[Lavie84] Lavie P, et al- *"Prevalence of sleep apnoea among patients with essential hypeternsion".* Am Heart J 1984; 108: 373-376

[Lawler85] Lawler E.L., Lenstra J.K., Rinnooy A.H.G., Shmoys D.B. (Eds.)., *"The Travelling Salesman Problem. A Guided Tour of Combinatorial Optimization".* John Wiley & Sons (1985).

[Lebart85] Lebart L., Morineau A., Fénelon J.P., *"Tratamiento Estadístico de Datos".* Marcombo, 1985.

[Lee80] Lee, E.T. "*Statistical Methods for Survival Data Analysis".* Lifetime Learning Publications, Belmont, California, 1980.

[LeGall93] LeGall, J., Leveshow, S., Saulnier, F. *"A new simplified acute physiological score (SAPS II) based on a European/North American multicenter study"*. JAMA 270:2957-2963, 1993.

[Levy97] Levy, A., Fikes, R., Sagiv, Y. *"Speeding up inferences using relevance reasoning: a formalism and algorithms".* Artificial Intelligence , Vol. 97, #1-2, Ed. Elsevier, December 1997.

[Lopez97] Lopez, D., Moreno, F., Barriga, A., Sánchez-Solano, S. "*XFL: a language for the definition of fuzzy systems".* Sixth IEEE International Conference on Fuzzy Systems, Barcelona, Vol. III, pp. 1585-1591, 1997.

[López-García97] López-García, H., López-Díez, H. *"Medidas de desigualdad difusas de tipo Gastwirth: definición y estudio de condiciones de existencia"*. VII Congreso Español sobre Tecnologias y Logica Difusa, Tarragona, pp. 207-211, 1997.

[Loutchmia97] Loutchmia, D., Ralambondrainy, H. *"Inductive learning using similarity measures on lattice-fuzzy set".* Sixth IEEE International Conference on Fuzzy Systems, Barcelona, Vol. III, pp. 1307-1313, 1997.

[Luenberger73] Luenberger, D. G., "*Introduction to Linear and Nonlinear Programming"*, Addison-Wesley, Menlo Park, California, 1973.

[Lugaresi83] Lugaresi E, et al. *"Staging of heavy snoring disease. A proposal".* Bull Eur Physiopathol Respir 1983; 19: 590-594

[Magdalena97] Magdalena, L., Monasterio, F., Rivero, C. *"Hierarchical decomposition of multiobjective fuzzy controllers based on metaknowledge".* VII Congreso Español sobre Tecnologias y Logica Difusa, Tarragona, pp. 103-108, 1997.

[Manton92] Manton. K.G, Woodbury, Max. A. *"Statistical Applications using Fuzzy Sets".* John Wiley & Sons, Inc. 1992.

[Mardia79] Mardia,K., Kent,J., Bibby, J. "*Multivariate Analysis".* Academic Press, London, 1979.

[Marichal98] Marichal, J. *"Aggregation operators for multicriteria decision aid".* PhD Thesis, Universite de Liege, Belgium, 1998.

[Marichal99] Marichal, J., Roubens, M.,*"Determination of weights of Interacting Criteria from a Reference Set",* Papiers de Recherche, Faculté d'Economie de Gestion et de Sciences Sociales, Groupe d'Etude des Mathématiques du Management et de l'Economie, N. 9909, 1999.

[Martin85] Martin RJ, et al. *"Indications and standards for cardiopulmonary sleep studies".* Sleep 1985; 8: 371-379

[McLeish95]. McLeish, M., Yao, P., Garg, M., Stirtzinger, T. *"Discovery of Medical Diagnostic Information: An Overview of Methods and Results".* Knowledge Discovery in Databases. Ed. Shapiro, G., Frawley, W., 1995.

[Michalewicz92] Michalewicz, Z., Janikow, C., *"GENOCOP: A Genetic Algorithm for Numerical Optimization Problems with Linear Constraints",* accepted for publication, *Communications of the ACM*, 1992.

[Michalewicz96] Michalewicz, Z., *"Genetic Algorithms+Data Structures=Evolution Programs",* 3rd Edition, Springer, 1996.

[Murofushi91] Murofushi, T., Sugeno, M., *"Fuzzy t-conorm integral with respect to fuzzy measures: generalization of Sugeno integral and Choquet integral",* Fuzzy Sets and Systems, 42:1 57-71, 1991.

[Myoshi97] Myoshi, T., Ichihashi, H., Tanaka, F. *"Fuzzy projection persuit ID3".* Sixth IEEE International Conference on Fuzzy Systems, Barcelona, Vol. III, pp. 1301-1306, 1997.

[Nakamori97] Nakamori, Y., Watada, J. "*Factor Analysis for Fuzzy Data".* Sixth IEEE International Conference on Fuzzy Systems, Barcelona, Vol. II, pp. 1115-1120, 1997.

[Nakashima97] Nakashima, T., Morisawa, T., Ishibuchi, H. *"Input selection in fuzzy rule-based classification systems".* Sixth IEEE International Conference on Fuzzy Systems, Barcelona, Vol. III, pp. 1457-1462, 1997.

[Nakhaeizadeh96] Nakhaeizadeh, G. *"Classification as a Subtask of Data Mining: Experiences from some Industrial Projects".* IFCS-96. Fifth Conference of International Federation of Classification Societies, 1996.

[O'Hagan88] O'Hagan, M., *"Aggregating template or rule antecedents in real-time expert systems with fuzzy set logic",* Proceedings of the 22nd Annual IEEE Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, 1988, 681-689.

[Okamoto94] Okamoto, H., Umano, M., Hatono, I., Tamura, H., et al. *"Fuzzy decision trees by fuzzy ID3 algorithm and its application to diagnosis systems".* In Proc. of 3rd IEEE Int. Conf. on Fuzzy Systems, pp2113-2118, 1994.

[Olson95] Olson LG, King MT, Hensley MJ, Saunders NA. *"A community study of snoring and sleep-disordered breathing prevalence".* Am. J Respir Crit Care Med 1995;152: 711-716

[Pal97] Pal, N.R., Pal, K, Bezdek, J.C. "*A Mixed c-Means Clustering Model".* Proc. Sixth IEEE International Conference on Fuzzy Systems, Barcelona, Vol. 1, pp. 11-21, 1997.

[Partinen88] Partinen M, et al. *"Long term outcome for obstructive sleep apnoea syndrome patients: mortality".* Chest 1988; 94: 1200-1204

[Peña84] Peña, D. "*Estadística Modelos y Métodos*". Alianza, 1984.

[Pessi95] Pessi, T., Kangas, J., Simula, O. "Patient grouping using Self-Organizing Map". Proc. ICANN '95, Conférence Internationale sur les Réseaux de Neurones Artificiels, Neuronimes '95, Session 5, Medicine., Pub. EC2, 1995.

[Platt91] Platt, J. *"A resource-allocating network for function interpolation".* Neural Computation 3(2), pp213-225, 1991.

[Quinlan86] Quinlan, J.R. *"Induction of decision trees".* Machine Learning Journal 1, pp. 81-106, 1986.

[Quinlan93] Quinlan, J.R. "C4.5: *Programs for Machine Learning"*, Morgan Kaufmann, San Mateo, Calif.. 1993.

[Quinlan96] Quinlan, J.R. *"Improved use of continuous variables in C4.5".* Journal of Artificial Intelligence Research 4, pp77-90, 1996.

[Rechenberg73] Rechenberg, I. "*Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution"*, Frommann-Holzboog Verlag, Stuttgart, 1973.

[Rosenblatt59] Rosenblatt, F. *"The Perceptron. A probabilistic model for information storage and organization in the brain".* Psychological Review, 65, pp386-408, 1959.

[Rousseeuw89] Rousseeuw, P.J., Derde, M.P. , Kaufman, L. *"Principal components of a fuzzy clustering".* Trends in Analytical Chemistry, 8, pp249-250, 1989.

[Roychowdhury97] Roychowdhury, S., Shenoi, S. *"Fuzzy rule encoding techniques".* Sixth IEEE International Conference on Fuzzy Systems, Barcelona, Vol. II, pp. 823-828, 1997.

[Saaty80] Saaty, T. L., "*The Analytic Hierarchy Process"* , McGraw-Hill, New York, 1980.

[Sánchez97] Sánchez, L., Couso, I. *"Modelado a partir de ejemplos imprecisos mediante algoritmos GA-P".* VII Congreso Español sobre Tecnologias y Logica Difusa, Tarragona, pp. 121-126, 1997.

[Schlimmer86] Schlimmer, J., Fisher, D. "A case study of incremental concept induction". Proc. Fifth National Conference on Artificial Intelligence, pp. 496-501, San Mateo, CA, Morgan Kaufmann, 1986.

[Schwefel81] Schwefel, H.P., (1981), "*Numerical Optimisation for Computer Models"*, John Wiley, Chichester, UK, 1981.

[Shapiro87] Shapiro, A. *"Structured induction in expert systems".* Wokingham, U.K., Addison-Wesley, 1987.

[Sierra89] Sierra, C. "MILORD. Arquitectura multinivell per a sistemes experts en classificació". PhD thesis, Universitat Politécnica de Catalunya, 1989.

[StatLog94] StatLog, Esprit Project 5170. *"Comparitive testing and evaluation of statistical and logical learning on large-scale applications to classification, prediction and control".* CEE Esprit Program, 1991-1994.

[Strauss97] Strauss, O. "*Filtering and fusing compass and gyrometer data using guess filter".* Sixth IEEE International Conference on Fuzzy Systems, Barcelona, Vol. III, pp. 1593-1599, 1997.

[Takagi85] Takagi, T., Sugeno, M. *"Fuzzy identification of systems and its application to modelling and control".* IEEE Trans.Syst.,Man Cybern. 15(1), pp116-132, 1985.

[Torra96] Torra. *"The Weighted OWA Operator".* Fifth IEEE International Conference on Fuzzy Systems, 1996.

[Torra97a] Torra, V., *"The Weighted OWA Operator".* International Journal of Intelligent Systems, Vol. 12, 153-166. John Wiley & Sons (1997)

[Torra98a] Torra, V., *"On some relationships between the WOWA operator and the Choquet integral",* Proceedings of the Seventh Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'98) (ISBN 2-84254-013-1), 818-824, Paris, France 1998.

[Torra98b] Torra, V. *"On considering constraints of different importance in fuzzy contraint satisfaction problems"*. International Journal of Uncertainty and Knowledge Based Systems, Oct. 1998.

[Torra98c] Torra, V. *"On the integration of numerical information: from the arithmetic mean to fuzzy integrals"*. Research Report, ETSE, Universitat Rovira i Virgili, Tarragona, Spain, 1998.

[Torra99a] Torra, V. *"The WOWA operator and the interpolation function W*: Chen and Otto's interpolation method revisited".* Fuzzy Sets and Systems, 1999.

[Torra99b] Torra, V., *"On the learning of weights in some aggregation operators: the weighted mean and the OWA operators".* Mathware and Soft Computing, 6, pp.249-265, Ed. Universidad Politécnica de Catalunya, Barcelona, 1999.

[Torra99c] Torra, V. *"Interpreting membership functions: a constructive approach".* Int. J. of Approx. Reasoning, 20, pp191-207, 1999.

[Umano97] Umano, M. et al. *"Generation of fuzzy decision trees by fuzzy ID3 algorithm and its application to diagnosis by gas in oil".* Proc. 1994 Japan-USA Symposium on Flexible Automation, pp1445-1448, 1994.

[Utgoff91] Utgoff, P., Brodley, C. *"Linear machine decision trees".* COINS Technical Report 91-10, University of Massachusetts, Amherst MA, 1991.

[Verdagay97] Verdagay, J., Sancho, A. *"Empirical determination of membership functions for stimuli comparison".* Sixth IEEE International Conference on Fuzzy Systems, Barcelona, Vol. III, pp. 1327-1332, 1997.

[Vignaux91] Vignaux, G.A., Michalewicz, Z., *"A Genetic Algorithm for the Linear Transportation Problem",* IEEE Transactions on Systems, Man, and Cybernetics, Vol.21, Nº 2, pp.445-452, 1991.

[Wangc96] Wang, C., Hong, T., Tseng, S. *"Inductive learning from fuzzy examples"*. Fifth IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 1996.

[Wangh95] Wang, H., Lin, L. "A multicriteria analysis of factor selection in an uncertain system". Int. Journal of Uncertainty, Fuzziness and Knowledge-based Systems, 1995.

[Ward97] Ward Flemons, W, McNichols, Walter T. *"Clinical prediction of the sleep apnea syndrome".* Sleep Medicine Reviews, Vol. 1, Nº 1, pp 19-32, 1997.

[Watada94] Watada, J., Yabuuchi, Y. *"Fuzzy principal component analysis and its application to company evaluation"*. Proceedings of the Japan-Brazil Joint Symposium on Fuzzy Systems, Campinas and Manaus, Brazil, July 19-27, 1994.

[Yager88] Yager, R. R. "*On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decisionmaking".* IEEE Transactions on Systems, Man, and Cybernetics, Vol. 18, Nº. 1, pp183-190, January/February 1988.

[Yager93] Yager, Ronald R. *"Families of OWA operators".* Fuzzy Sets and Systems 59 (1993) pp125-148, North-Holland.

[Yager96] Yager, R. R., *"Quantifier Guided Aggregation Using OWA operators", Int. J. of Int. Systems*, 11 (1996) 49-73.

[Young94] Young T, Palta M, Dempsey J., et al. *"The occurrence of sleep-disordered breathing among middle-aged adults".* N Engl J Med 1994; 328: 1230-1235.

[Zadeh65] Zadeh, L.A. *"Fuzzy Sets".* Information Control, vol. 8, pp338-353, 1965.

[Zadeh71] Zadeh, L.A. *"Similarity Relations and Fuzzy Orderings".* Information Science, Vol. 3, pp-177-200. Elsevier Science Publishing Company, Inc. (1971).

[Zadeh73] Zadeh, L.A. *"Outline of a New Approach to the Analysis of Complex Systems and Decision Processes".* IEEE Trans. Syst., Man, Cybern., Vol. SMC-3, no. 1, pp28-44, Jan. 1973.

[Zadeh83] Zadeh, L.A. *"A computational approach to fuzzy quantifiers in natural languages".* Comps. & Maths. with Appls., vol. 9, p149-184, 1983.

[Zahan97] Zahan, S., Michael, C., Nikolakeas, S.. *"A fuzzy hierarchical approach to medical diagnosis".* Sixth IEEE International Conference on Fuzzy Systems, Barcelona, Vol. I, pp. 319, 1997.

[Zeidler96] Zeidler, J., Schlosser, M. *"Continuous valued variables in fuzzy decision trees".* In Information Processing and Management of Uncertainty in Knowledge Based Systems (IPMU'96), Vol I, pp395-400, Granada, Spain, 1996.

[Zhang93] Zhang, L. *"Structural and functional quantization of vagueness",* Fuzzy Sets and Systems, 55, 51-60, 1993.

Annex 2. Detail of all the variables of the 'Hospital Admissions' ICU data set used in Section 4.1 of the thesis

Notes: With respect to the abbreviations, MPM, SAPS and APACHE: they are a set of indices, obtained by calculation from the values of the variables which are included in each of them, and which are habitually used by the specialist doctors in 'Intensive Medicine' to evaluate the seriousness of the patients state. In the case of MAP, this is a measure of the average arterial pressure, and is defined in units of measure mm Hg.

Table A2.1. Attribute-Values for the data set 'Hospital Admissions'

VARIABLE	EXAMPLE VALUE	VARIABLE TYPE	UNITS	ALLOWABLE VALUES	DESCRIPTION
PATIENT DEMOGRAPHIC INFORMATION					
AGE	74	Numerical NN			Age in years
SEX	1	Binary		{0,1}	{1:male, 0:female}
					MPM AT TIME OF ADMISSION
COMA_ADM	0	Binary		{0,1}	Presence of coma or profound stupor at time of admission to ICU
INTOXICATION	0	Binary		{0,1}	If COMA_ADM=1, is this due to a drug overdose?
TIPO_ADM	3	Categorical No-Ord		{1,2,3}	Type of Patient {1=Emergency Surgery, 2=Planned Surgery,3=Without Surgery}
CPR	0	Binary		{0,1}	CRP previous to admission to the ICU (within 24 hours).
MALIG	0	Binary		{0,1}	Malign Neoplasm part of actual problem?
METASTAT	0	Binary		{0,1}	If MALIG=1, ¿is it a MetaStatic?
PREV_ICU	0	Binary		{0,1}	Previous admission to ICU (in last 6 months) .
TASA_H	80	Numerical NNN	Beats/ minute		Pulse at the time of admission to the ICU
SBP_ADM	115	NNN	mm Hg		Systolic blood pressure at the time of admission
C_REN_F	0	Binary		{0,1}	History of renal failure?
ICU_SER	1	Binary		{0,1}	Service at the time of admission to the ICU. {0=Medical, 1=Surgery}
PROB_INF	0	Binary		{0,1}	Probable infection at the time of admission to the ICU
MPM 24 hours after admission					
COMA_24H	0	Binary		{0,1}	In coma or profound stupor at 24 hours after admission
PRO_TIME	0	Binary		{0,1}	'Protrombin' time > 3 seconds above standard or < 25%
SHOCK	0	Binary		{0,1}	Probable shock during the first 24 hours
ORINA	0	Binary		{0,1}	Urine output < 150ml in any 8 hour period
CONF_INF	0	Binary		{0,1}	Confirmed infection at 24 hours after admission
PO2	0	Binary		{0,1}	PO_2 < 60mmHg (o < 7.98kPa) during first 24 hours).
FIO2	0	Binary		{0,1}	FIO_2 > 0.50 during the first 24 hours.

CREATIN	0	Binary		{0,1}	Creatinine > 2.0mg/dl (176.8μMol/l) during the first 24 hours.
MECH_VEN	0	Numerical NN	Hours		Hours of mechanical ventilation during first 24 hours.
SER_24H	1	Binary		{0,1}	Service at 24 hours {0=Médico, 1=Cirugía}
LINES	1	Numerical NN			Number of lines at 24 hours after admission
MPM 91: ADDITIONAL ADMISSION VARIABLES					
INT_CRAN	0	Binary		{0,1}	Effect on the inter cranial mass
ON_MECH	0	Binary		{0,1}	Receives mechanical ventilation
SEP_SHOK	0	Binary		{0,1}	Septic Shock
GI_SANGRE	0	Binary		{0,1}	Acute bleeding GI
DIS_CARD	1	Binary		{0,1}	Cardiac Disrrythmias
ENF_CARD	1	Binary		{0,1}	Isquimic coronary pathology
FALLO_CARD	0	Binary		{0,1}	Cardiac Failure
CERE_DIS	0	Binary		{0,1}	Cerebral vascular pathology
A_R_FAIL	0	Binary		{0,1}	Acute renal failure
LIMIT	0	Binary		{0,1}	Restriction on patient care by order of patient or family
CIRRHOS	0	Binary		{0,1}	Cirrhosis.
MPM 91: ADDITIONAL VARIABLES AT 24 HOURS					
EMERSURG	0	Binary		{0,1}	Emergency surgery during the first 24 hours.
LIMIT24H	0	Binary		{0,1}	Restriction on patient care during the first 24 hours..
PH_7P2	0	Binary		{0,1}	pH ≤7.2 during the first 24 hours.
PEEP	0	Binary		{0,1}	PEEP > 10cm during the first 24 hours.
PLATELET	0	Binary		{0,1}	Platelets < 50,000 o "low" during the first 24 hours.
CONT_VAS	0	Binary		{0,1}	Therapy with medicines. Continuous IV vasoactive during the first 24 hours.
SAPS					
TASA_S_H	180	Numerical NNN	Beats /minute		Pulse (heart pulsation rate)
PSS	120	Numerical NNN	mmHg		Systolic blood pressure
TEMP_CORP.	35.8	Numerical FFF.F o FF.F	ºF or ºC		Body temperature
TASA_RES	24	Numerical NN			If VEN_CPAP=0, measure rate of spontaneous respiration
VEN_CPAP	0	Binary		{1,0}	Mechanical ventilation or CPAP {1=yes, 0=no}
SALIDA_UR	1.8	Numerical FF.F	Litres in 24 hours		Urine output
B_UREA	3.7	Numerical NNN o FFF.F	mMol/l or mg/dl		Blood urine concentration
HEMATOCR	41	Numerical NN	%		Hematocrit
WBC	8.4	Numerical FFF.F	FFF.F		WBC(10^3/mm^3)
S_GLUCOS	7.8	Numerical FFF.F o FF.F	mMol/l or g/l		Glucose in Serum

S_POT	4.4	Numerical FF.F	mMol/l		Potassium in Serum
S_SODIUM	135	Numerical NNN	mMol/l		Sodium en Serum
S_HCO3	23	Numerical NN	mMol/l		Standard Serum of HCO_3
SEDADO	0	Binary		{0,1}	Glasgow Coma Scale. Is the patient sedated ? {1=yes, 0=no}
GCS_SAPS	15	Numerical NN			If SEDATED=1, estimated GCS. If SEDATED=0, actual GCS
CHRONIC HEALTH STATE					
P_H_STAT	1	Categorical		{1,2,3,4}	Previous health state
MAC_CABE	1	Categorical		{1,2,3}	MacCabe {1=without illness or nonmortal, 2=eventually mortal (<5años), 3=rapidly mortal (< 1 año).
COPD	0	Binary		{0,1}	Chronic pathologies {1=yes, 0=no}
INSULINA	0	Binary		{0,1}	Diabetes dependent on Insulin
F_CARD	0	Binary		{0,1}	Cardiac Failure
HEMA_MAL	0	Binary		{0,1}	Immune system compromised: haematological malignancy. {1=yes, 0=no}
SIDA	0	Binary		{0,1}	SIDA
TERA_CH	0	Binary		{0,1}	Chemotherapy
NSAID	0	Binary		{0,1}	NSAID
ESTEROIDES	0	Binary		{0,1}	Steroids (long term or high consumption)
DIAG	1	Categorical		{0,1,2,3,...51}	Principal Diagnostic Category consequence of admission to the ICU (see Table A2.2 for possible categories)
APACHE II					
MAP	10	Numerical NNN	mmHg		MAP
A_RES_R	90	Numerical NNN			Respiration rate (with or without ventilation)
A_FIO2	24	Numerical F.FF			Oxygenation: FiO_2
PAO2	0.2	Numerical NNN	mmHg		Oxygenation: $PaCO_2$
PACO2	79	Numerical NNN	mmHg		Oxygenation: PaO_2
A_ADO2	38	Numerical?			A-aDo2 calculated by computer
PH_ARTER	0	Numerical F.FF			Arterial pH
INT_VENT	1	Binary		{0,1}	Tubing/ventilator
S_CREA	0	Numerical FF.F o FFF.F	mg/dl or µMol/l		Serum creatinine
S_BILI	0.2	Numerical FF.FF o FFFF.F	mg/dl or µMol/l		Serum bilirubin (total)
S_ALBU	0	Numerical FF.F o FFFF.F	g/l or µMol/l		Serum albumin.
S_BUN	4	?			Serum BUN calculated by computer
CREA_INC	1	Binary		{0,1}	Creatinine increment > 124 Mol/l in last 24 hours associated with Oligury {1=yes, 0=no}
O.S.F. FIRST DAY					
RES_F	0	Binary		{0,1}	Respiratory failure{1=yes, 0=no}

CARD_F	0	Binary		{0,1}	Cardiovascular failure
RENAL_F	1	Binary		{0,1}	Renal failure
HEMA_F	0	Binary		{0,1}	Haematological failure
NEURO_F	0	Binary		{0,1}	Neurological failure (excluding sedation)
HEPA_F	0	Binary		{0,1}	Hepatic failure
OSF	0	Numerical			Number of organ systems failing, calculated by computer program
OUTPUT VARIABLES					
D_ADM	3/5/91	Date MM/DD/YY			Date of admission to the ICU.
DIA_ICU	3/5/91	Date MM/DD/YY			Date of departure from the ICU
DEAD_ICU	0	Binary		{0,1}	Vital state ICU {0=alive, 1=dead}
DUR_ICU	33	Numerical NNN			Calculated duration of stay in the ICU
DUR_HOS	6	Numerical NNN			Calculated duration of stay in the hospital from the time of admission to the ICU
DEAD_HOS	0	Binary		{0,1}	Vital state Hospital {0=alive, 1=dead}
SAL_HOS	1/3/91	Date MM/DD/YY			Date of departure from hospital
ADDITIONAL VARIABLES: **NOT PRESENT IN ORIGINAL DATA COLLECTION FORMS**					
INCLUDE	**1**	**Binary**		**{0,1}**	**Fulfils criteria for inclusion in analysis (is not coronary care, burns or coronary surgery, with minimum age of 18 years {0=no, 1=yes}**
IN24HRS	**1**	**Binary**		**{0,1}**	**Duration of stay in ICU 24 hours or more {0=no, 1=yes}**
1TYPE_ADM	**1**	**?**			
2TYPE_ADM	**0**	**?**			
3TYPE_ADM	**0**	?			

Table A2.2. Principal Diagnostic Categories Motive for Admission to the ICU

(this is the value for the categorical variable, DIAG)

CODE	CATEGORY	DESCRIPTION
	PATIENT WITHOUT SURGERY: **Respiratory failure or insufficiency of:**	
1		Asthma/Allergy
2		COPD
3		Pulmonary Edema (nocardiogenic)
4		Post respiratory stoppage
5		Inhalation/Poisoning/Toxic
6		Pulmonary Embolus
7		Infection
8		Neoplasm
	Cardiovascular failure or insufficiency of:	
9		Hypertension
10		Rhythmic disturbance
11		Congestive coronary arrest
12		Shock/hypovolemia haemorrhage
13		Pathology of the coronary artery
14		Sepsis
15		Post cardiac arrest
16		Cardiogenic Shock
17		Desecando thoracic aneurysm/ abdominal
	Trauma:	
18		Multiple trauma
19		Head trauma
	Neurological:	
20		Attack disorder
21		ICH/SDH/SAH
	Others:	
22		Drug overdose.
23		Diabetic Ketoacidosis
24		Bleeding GI
	If not one of the above specified groups, which main vital organ system was the principal reason for admission?	
25		Metabolic/Renal
26		Respiratory
27		Neurological
28		Cardiovascular
29		Gastrointestinal
	POSTOPERATIVE PATIENTS	
30		Multiple Trauma
31		Admission due to a chronic cardiovascular pathology
32		Peripheral vascular surgery
33		Coronary valve surgery
34		Craniotomy due to neoplasm
35		Renal Surgery due to neoplasm
36		Renal Transplant
37		Head trauma
38		Thoracic Surgery due to neoplasm
39		Craniotomy due to ICH/SDH/SAH

40		Laminectomy /other surgery of the spinal cord
41		Shock due to haemorrhage
42		Bleeding GI
43		Surgery GI due to Neoplasm
44		Respiratory Insufficiency after surgery
45		Perforation GI/obstruction
	If not one of the above, which principal system of vital organs was the cause of admission to the ICU?	
46		Neurological
47		Cardiovascular
48		Respiratory
49		Gastrointestinal
50		Metabolic/Renal
51		Others (specify)

Annex 3. Apnea Screening Questionnaire used in Sections 4.3 and 4.4 of the thesis

CLINICAL QUESTIONNAIRE
RESPIRATORY ALTERATIONS DURING SLEEP

Nº HISTORIAL:_____ DATE: _____/_____/_____

SURNAME:_____ NAME:_____

AGE:_____ TELEPHONE: _____

INTERVIEW CONDUCTED IN PRESENCE OF ROOM PARTNER

1 – Yes 2 – No

PROFESSION: _____

WORK HOURS: 1-Morning 2- Afternoon 3- Night 4- Rotative 5- Retired/does not work
EDUCATION LEVEL: 1- Elemental 2- Medium 3- Higher

- WEIGHT (kg):_____HEIGHT(m):_____NECK DIAMETER (cm):_____
- BWI (Kg/m_2):_____ ARTERIAL TENSION (mmHg):_____

ALCOHOL INTAKE (gr/día):_____ TOBACCO (Packs/year):_____

WHAT IS YOUR MOST IMPORTANT DISCOMFORT OR SYMPTOM

1 – Snoring 2-Somnolence during the day 3- Choking at night
4 – Other discomforts
(specify):_____

ILLNESSES or PREVIOUS BACKGROUND OF INTEREST

1. Arterial hypertension: 1-Yes ; 2-No 2. Cardiopathic ischemia: 1-Yes ; 2-No
Others:
_____
_____

INSTRUCTIONS FOR COMPLETING THE QUESTIONNAIRE

YOU WILL FIND **THREE TYPES OF QUESTION** IN THE FOLLOWING SECTIONS OF THE QUESTIONNAIRE:

In the first example you simply place a number after the question, for example, the number of hours which you normally sleep (8).

G1 HOW MANY HOURS DO YOU NORMALLY SLEEP? 8

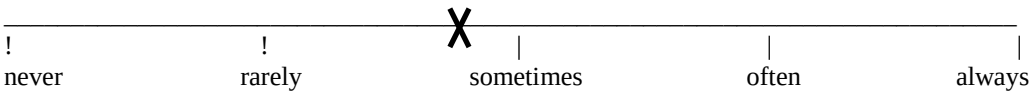
In the second example there are four possible categories and you must indicate only ONE of these categories. In this case it is **3-sometimes**.

G2 DO YOU TAKE TRANQUILIZER TABLETS IN ORDER TO SLEEP?

1- never 2 - rarely 3 - sometimes 4 – frequently

In the third example there are five possible categories placed above the continuous line. You may place a cross in any position along the line in the place which most corresponds to your opinion. In this example, a point has been marked between '**rarely'** and '**sometimes'**, but closest to '**sometimes'**.

G10 HAVE YOU NOTED A SENSATION OF PARALYSIS AT THE START OF SLEEP OR ON WAKING UP?



FOR THE QUESTIONNAIRE TO BE OF USE TO DIAGNOSE YOUR CASE, WE ASK YOU TO BE HONEST AND FRANK IN RESPONDING TO THE QUESTIONS. THE QUESTIONNAIRE WILL BE MAINTAINED IN STRICT CONFIDENCE

GENERAL SLEEP QUESTIONS

G1 HOW MANY HOURS DO YOU NORMALLY SLEEP?

G2 DO YOU TAKE TRANQUILIZER TABLETS IN ORDER TO SLEEP?

never	rarely	sometimes	often	always

G3 ARE YOU USED TO TAKING AN AFTERNOON NAP?

never	rarely	sometimes	often	always

G4 APPROXIMATE DURATION OF THE NAP (in minutes):

G5 HAVE YOU GAINED WEIGHT RECENTLY?

No	< 5Kg	5-10Kg	>10Kg

G6 SINCE WHEN HAVE YOU GAINED WEIGHT? (ONLY REPLY IF YOU HAVE GAINED WEIGHT).

< 6 months	6 months - 1 year	1 - 2 years	> 2 years

G7 WHEN YOU ARE SLEEPING, DO YOU HAVE NIGHTMARES THAT SEEM AS IF THEY ARE REAL?

never	rarely	sometimes	often	always

G8 HAVE YOU EVER HAD THIS TYPE OF NIGHTMARE WHILE AWAKE?

never	rarely	sometimes	often	always

G9 HAVE YOU NOTED, ON ANY OCASIÓN, THAT DURING A STRONG EMOTION (ANGER, LAUGHTER) YOU HAVE LOST STRENGTH, IF ONLY IN PART OF THE BODY, AND EVEN FALLEN ON THE FLOOR AS A RESULT ?

never	rarely	sometimes	often	always

G10 HAVE YOU NOTED A SENSATION OF PARALYSIS AT THE START OF SLEEP OR ON WAKING UP ?

never	rarely	sometimes	often	always

G11 DO YOU KNOW OR HAVE YOU BEEN TOLD THAT YOU MOVE YOUR LEGS A LOT WHILE YOU ARE SLEEPING ?

never	rarely	sometimes	often	always

G12 DURING THE DAY WHEN YOU SIT DOWN, DO YOU NOTE A PAIN IN THE CALF OF THE LEG WHICH GETS BETTER WHEN YOU WALK ?

never	rarely	sometimes	often	always

G13 ARE YOU FRANKLY DEPRESSED ?

never	rarely	sometimes	often	always

G14 DO YOU HAVE PROBLEMS OF INSOMNIA?

never	rarely	sometimes	often	always

G15 WHAT TYPE OF INSOMNIA PROBLEMS DO YOU HAVE ? (ONLY RESPOND IF YOU HAVE INSOMNIA)

1- difficulty in getting to sleep 2- waking up in the middle of the night 3- getting up too early 4- others

QUESTIONS RELATED TO RESPIRATORY ILLNESSES DURING SLEEP

R1 DO YOU SNORE WHILE YOU SLEEP OR HAVE YOU BEEN TOLD THAT YOU DO?

never	rarely	sometimes	often	always

R2 DOES YOUR SNORING WAKE UP YOUR PARTNER OR CAN IT BE HEARD FROM ANOTHER ROOM?

never	rarely	sometimes	often	always

R3 HAS SNORING SOMETIMES CAUSED PROBLEMS WITH THE NEIGHBORS OR WHEN YOUR HAVE SLEPT AWAY FROM HOME ?

never	rarely	sometimes	often	always

R4 WHEN DID YOU BEGIN SNORING ?

< 1 year	1 - 3 years	4 - 9 years	> 10 years

R5 HAVE YOU RECENTLY NOTICED AN INCREASE IN THE INTENSITY OF YOUR SNORING?

no	< 6 months	6 - 12 months	> 1 year

R6 DO YOU WAKE UP AT NIGHT WITH A SENSATION OF CHOKING?

never	rarely	sometimes	often	always

R7 HAVE YOU BEEN TOLD THAT YOU "STOP BREATHING" WHEN YOU ARE ASLEEP ?

never	rarely	sometimes	often	always

R8 HAS YOUR BEDROOM PARTNER EVER WOKEN YOU UP FOR FEAR THAT YOU HAVE STOPPED BREATHING ?

never	rarely	sometimes	often	always

R9 HOW MANY TIMES DO YOU GET UP TO URINATE AT NIGHT ?

never	rarely	sometimes	often	always

R10 DO YOU SWEAT A LOT AT NIGHT ?

never	rarely	sometimes	often	always

R11 DO YOU HAVE A HEADACHE WHEN YOU GET UP IN THE MORNING ?

never	rarely	sometimes	often	always

R12 DO YOU WAKE UP WITH A DRY MOUTH ?

never	rarely	sometimes	often	always

R13 WHEN YOU GET UP IN THE MORNING DO YOU HAVE THE SENSATION OF NOT HAVING RESTED ?

never	rarely	sometimes	often	always

R14 DO YOU FIND IT VERY DIFFICULT TO GET UP IN THE MORNING AND DO YOU HAVE THE SENSATION FOR A WHILE OF NUMBNESS ?

never	rarely	sometimes	often	always

R15 HAVE YOU LOST YOUR MEMORIA OR ABILITY TO CONCENTRATE ?

never	rarely	sometimes	often	always

R16 DO YOU HAVE PROBLEMS OF SEXUAL IMPOTENCE ?

never	rarely	sometimes	often	always

QUESTIONS RELATED TO DAYTIME SLEEPINESS

S1 DO YOU SLEEP WHILE WATCHING THE TELEVISION ?

never	rarely	sometimes	often	always

S2 DO YOU SLEEP WHILE YOU ARE READING ?

never	rarely	sometimes	often	always

S3 DO YOU SLEEP WHEN YOU ARE IN THE CINEMA, THEATRE, OR AT OTHER SPECTACLES ?

never	rarely	sometimes	often	always

S4 DO YOU FALL ASLEEP IN MEETINGS OR IN PUBLIC PLACES ?

never	rarely	sometimes	often	always

S5 DO YOU FALL ASLEEP WHEN DRIVING ON THE MOTORWAY ?

never	rarely	sometimes	often	always

S6 DO YOU SLEEP DURING THE DAYTIME AGAINST YOUR WILL ?

never	rarely	sometimes	often	always

S7 DO YOU FALL ASLEEP WHILE EATING ?

never	rarely	sometimes	often	always

S8 DO YOU FALL ASLEEP WHILE SPEAKING WITH ANOTHER PERSON ?
(In person or by telephone)

never	rarely	sometimes	often	always

S9 DO YOU FALL ASLEEP WHILE DRIVING WHEN YOU HAVE STOPPED AT A TRAFFIC LIGHT ?

never	rarely	sometimes	often	always

S10 DO YOU FALL ASLEEP IN YOUR WORKPLACE WHILE CARRYING OUT YOUR NORMAL WORK ACTIVITIES ?

never	rarely	sometimes	often	always