Kati Viikki

Machine Learning on Otoneurological Data: Decision Trees for Vertigo Diseases

DEPARTMENT OF COMPUTER AND INFORMATION SCIENCES
UNIVERSITY OF TAMPERE

A-2002-8

TAMPERE 2002

Kati Viikki

Machine Learning on Otoneurological Data: Decision Trees for Vertigo Diseases

ACADEMIC DISSERTATION

To be presented, with the permission of the Faculty of Information Sciences of the University of Tampere, for public discussion in the Paavo Koli Auditorium on June 5th, 2002, at 12 noon.

DEPARTMENT OF COMPUTER AND INFORMATION SCIENCES
UNIVERSITY OF TAMPERE

A-2002-8

TAMPERE 2002

Supervisor: Professor Martti Juhola
Department of Computer and Information Sciences,
University of Tampere

Opponent: Professor Seppo Lammi
Department of Computer Science and Applied Mathematics,
University of Kuopio

Reviewers: Professor Heikki Mannila
Laboratory of Computer and Information Science,
Helsinki University of Technology

Dr. Pirkko Nykänen
National Research and Development Centre for Welfare and Health

Department of Computer and Information Sciences
FIN-33014 UNIVERSITY OF TAMPERE
Finland

Electronic dissertation
Acta Universitatis Tamperensis 189
ISBN 951-44-5390-5
ISSN 1456-954X
http://acta.uta.fi

ISBN 951-44-5378-6
ISSN 1457-2060

Tampereen yliopistopaino Oy
Tampere 2002

ABSTRACT

Expert systems may be characterised as computerised advisory systems that perform in narrow domains at a level comparable to human experts. The success of expert systems lies essentially in the knowledge embedded in their knowledge bases.

This study concerns refining and expanding the knowledge base of an otoneurological expert system ONE. ONE was developed to support decision-making for diseases involving vertigo. Its knowledge base contains descriptions or patterns for vertigo diseases in the form of weights and fitness values. The knowledge for the first version of ONE was elicited from experienced otoneurologists and the literature. In this study, machine learning is utilised in knowledge acquisition. Decision tree induction is applied to data collected on otoneurological patients in order to acquire diagnostic knowledge. Special attention is paid to data preprocessing in order to construct classifiers for real world diagnostic situations. This work produces a variable grouping method based on graph theoretic techniques. The method is useful as such, giving insight into data and, further, it can be used in feature subset selection. The knowledge acquired by decision trees is used in the refinement of ONE's knowledge base, in which fitness values learned from data and also different weighting schemes are studied. The refinement work produces a better performing knowledge base for real world situations.

Keywords: Machine learning, decision tree induction, data pre-processing, feature subset selection, expert systems, knowledge acquisition, knowledge base refinement, otoneurological data, vertigo

Acknowledgements

I want to express my sincerest gratitude to my supervisor, Professor Martti Juhola, for his guidance throughout my doctoral studies. Thanks to his encouraging and forbearing attitude I was able to finish this dissertation.

I am greatly indebted to my co-authors Professor Ilmari Pyykkö, Erna Kentala, M.D., Ph.D., and Pekka Honkavaara, M.D., Ph.D., for guiding me to the field of medical decision-making and for providing medical data for the study. I wish to thank Yrjö Auramo, Ph.D., for his contribution to this study. I also want to thank Elina Isotalo, M.D., Ph.D., and Heikki Aalto, Ph.D., for fruitful discussions.

I wish to express my deepest gratitude to Docent Erkki Mäkinen and Jorma Laurikkala, Ph.D., for their invaluable help during this work. Their support and friendship has been of great importance to me.

This study was carried out in the Department of Computer and Information Sciences, University of Tampere, during the years 1998-2002. The Department, headed by Professors Kari-Jouko Räihä, Pertti Järvinen, Seppo Visala and Jyrki Nummenmaa has provided a pleasant environment in which to work and write the dissertation. I wish to thank the entire personnel of the Department. Special thanks are reserved for Tapio Niemi, Markku Siermala, Matti J. Tapani, and Timo Tossavainen.

This work was supported financially by the Tampere Graduate School in Information Science and Engineering (TISE), the Jenny and Antti Wihuri Foundation, the Finnish Concordia Union, the Emil Aaltonen Foundation, and the Ella and Georg Ehrnrooth Foundation, all of which are gratefully acknowledged.

I also thank the reviewers of the thesis, Professor Heikki Mannila and Pirkko

Nykänen, Ph.D., for their constructive comments on the manuscript. Virginia Mattila, M.A., revised the English language of this dissertation.

I want to extend my gratitude to include all those friends with whom I have had a pleasure to share relaxing and joyful moments. I want to express my caring thanks to my parents, Matti and Leena, my brother, Jarmo, and my sister-in-law Sirkka for their continuous support during my studies. Finally, I owe my dearest thanks to Olli for caring, being patient and bringing joy to my daily life.

List of abbreviations

Abbreviation	Description
ACC	Accuracy
ANN	Artificial neural network
BPV	Benign positional vertigo
BRA	Brainstem response audiometry
CLS	Concept Learning System
CNS	Central nervous system
CONV	Confounding value
CT	Computerised tomography
DT	Decision tree
DTI	Decision tree induction
ECoG	Electrocochleography
ENG	Electronystagmography
ES	Expert system
FSS	Feature subset selection
HVDM	Heterogeneous Value Difference Metric
KA	Knowledge acquisition
KBS	Knowledge-based system
ML	Machine learning
MEN	Menière's disease
MRI	Magnetic resonance imaging
NAV	Necessary attribute value
NN	Nearest neighbour
NSAD	Nonsteroidal anti-inflammatory drug

(continued)

List of abbreviations

Abbreviation	Description
PEM	Pursuit eye movements
PONV	Postoperative nausea and vomiting
SEM	Saccadic eye movements
SUD	Sudden deafness
TDIDT	Top Down Induction of Decision Trees
TPR	True positive rate
TRA	Traumatic vertigo
VDM	Value Difference Metric
VNE	Vestibular neuritis
VSC	Vestibular schwannoma

List of original publications

This study is based on the following publications referred to in the text by their Roman numerals:

I Viikki K, Kentala E, Juhola M and Pyykkö I. Decision tree induction in the diagnosis of otoneurological diseases. *Medical Informatics & The Internet in Medicine*, 24:277–289, 1999.

II Viikki K, Juhola M, Pyykkö I and Honkavaara P. Evaluating training data suitability for decision tree induction. *Journal of Medical Systems*, 25:133-144, 2001.

III Viikki K, Kentala E, Juhola M and Pyykkö I. Confounding values in decision trees constructed for six otoneurological diseases. In Lavrač N, Miksch S and Kavšek B (eds.), *Proceedings of the Fifth International Workshop on Intelligent Data Analysis in Medicine and Pharmacology (IDAMAP-2000)*, pp. 58–60. Berlin, 2000.

IV Viikki K. A variable grouping method based on graph theoretic techniques, *Artificial Intelligence in Medicine*. (submitted)

V Viikki K, Kentala E, Juhola M, Pyykkö I and Honkavaara P. Generating decision trees from otoneurological data with a variable grouping method. *Journal of Medical Systems*. (in press)

VI Viikki K and Juhola M. Refining the knowledge base of an otoneurological expert system. In Crespo J, Maojo V and Martin F (eds.), *Medical Data Analysis*, volume 2199 of *Lecture Notes in Computer Science*, pp. 276–281. Springer, Berlin, 2001.

Contents

Chapter 1

Introduction

Expert systems (ESs) [GD93, Nik97, Tur93, Wat86] are knowledge-based systems that solve complex problems in narrow real world domains at a level comparable to that of human experts. The capacity of ESs originates from the considerable amount of domain-specific knowledge stored in their knowledge bases and from the methods utilising the knowledge. Although performance surpasses human experts in some tasks, ESs are not intended to replace humans but to assist them in various problem-solving tasks.

An expert system, like a conventional information system, is basically a piece of computer software. The development process of an expert system, however, differs somewhat from a normal software design process because of the nature of the system. A typical conventional system performs its tasks by processing information with exact algorithms. In general, requirements for the system are relatively easy to define, and its development proceeds according to a standard life cycle model such as the waterfall model [Boe81]. An expert system, on the other hand, processes uncertain or incomplete knowledge by heuristic methods. The amount and type of knowledge as well as the methods needed in problem-solving are not necessarily known when the development of the system starts. Further, users may not know their exact needs and requirements for the system. Therefore, making complete system requirements and estimating resources needed

in developing an expert system is difficult. The life cycle models presented for conventional systems as such seldom suit the development of ESs, although the same phases can be found in the life cycles of both types of systems.

The development of an expert system usually follows an evolutionary prototype approach [GD93, Nik97, Tur93] based on experimenting and iteration. The process begins with problem definition and rapid development of a usable prototype. While constructing the prototype, the requirements become clearer to the developers and users of the system. The evaluation of the constructed prototype guides the further development of the system: its refining, expanding and modifying in subsequent analysis, design, implementation and evaluation steps.

This study deals with the refinement of the knowledge base of an otoneurological expert system ONE [AJP93, KPAJ96] developed to support decision-making for diseases involving vertigo. (The diseases covered by ONE are listed in Table A1 included in the Appendix.) Vertigo is a symptom that provides challenging problems even for experienced otoneurologists [Ken96b]. ONE helps physicians to approach the problem with the right questions and to control the large amount of information needed to solve it [Ken96b].

The success of an expert system lies primarily in the knowledge stored in its knowledge base and only secondarily in its inference mechanism [Fei79]. The knowledge base of ONE contains a scoring scheme of weight and fitness values [AJP93] used by the inference mechanism [AJP93] resembling the nearest neighbour method [Mit97]. The knowledge in the first version of ONE, developed in the 1990s was elicited from experienced otoneurologists and the literature [KPAJ96]. According to previous studies, knowledge acquisition was successful: ONE outperformed another otoneurological expert system [AJ95] and also compared favorably with physicians [KAJP98]. However, certain diseases and cases with confounding values (that is, symptoms and signs not related to the current disease) have caused difficulties for ONE [AJP93, KAJP98, VJKP00], which calls for the further refinement of its knowledge base.

The continued collection of otoneurological cases in the database of ONE

enables knowledge elicitation from data using machine learning (ML) methods [Mit97] that have been developed, for example, to bypass the bottleneck of knowledge acquisition [Fei79]. Traditionally, knowledge has been elicited from experts by using laborious and time-consuming interview methods [Mic87, Qui79, Qui88]. Experts have found it difficult to verbalise their knowledge, and knowledge engineers have had problems understanding experts' knowledge and reasoning. With machine learning methods, the process of knowledge elicitation can be partly automated.

This study explores the possibilities for using machine learning in refining and augmenting the knowledge for the six largest diagnostic groups in the database of ONE: benign positional vertigo, Menière's disease, sudden deafness, traumatic vertigo, vestibular neuritis, and vestibular schwannoma. We use decision tree induction (DTI) [Qui86] to acquire diagnostic knowledge for these diseases. Decision tree induction is a mature and robust ML method well suited to medical domains due to its symbolic knowledge representation [Qui90] giving explanations for decisions. We pay special attention to the problem of feature subset selection [BL97, DL97, KJ97]. We utilise the knowledge obtained by DTI in the refinement of ONE's knowledge base. We also study methods for learning the scoring scheme from data.

This thesis consists of the present summary part and six original papers. The summary part proceeds as follows. In Chapters 2 and 3, the basics of knowledge engineering and machine learning are introduced. ONE is described in Chapter 4. The research papers are reviewed in Chapter 5. In Chapter 6, we discuss the results and outline directions for future work.

Chapter 2

Knowledge engineering

Artificial intelligence (AI) is an interdisciplinary field of study aiming at explaining and describing intelligence by using computational models that can exhibit intelligent behaviour [Nik97, Tur93, Wat86]. It originated in the 1950s with the involvement of researchers from the fields of computer science, linguistics, psychology, and philosophy. During the first decade of AI, researchers sought to build general problem solvers [NS63] but their efforts did not result in the desired goals. The shift from general problem solvers to specific ones occurred in the mid-1960s, when the developers of the first expert systems (for example, DENDRAL [LBFL80] and MYCIN [Sho76] as described in [Nik97, Tur93, Wat86]) recognised that intelligent activity requires knowledge. By developing methods that can utilise a large amount of domain-specific knowledge, practical expert systems could be built.

On the one hand, AI is a field of theoretical research studying intelligence. On the other hand, it is a set of technologies. One of them is knowledge engineering (KE) - a technology for building practical knowledge-based systems (KBS) that may be called expert systems [Nyk00]. The aim of KE, instead of simulating human intelligence, is to build advanced machine intelligence that can be based on any method or technique: A computer program does not have to solve the problem in the same way as a human does, as long as the solution obtained is

acceptable.

According to its widest definition, knowledge engineering is a subfield of AI that concentrates on building knowledge-based systems [Mer02]. A narrower definition views it as a description of the process of developing and maintaining a knowledge-based system [Tur93]. In its narrowest meaning, knowledge engineering includes knowledge acquisition, representation, validation, inference, explanation, and maintenance [Tur93]. This chapter considers knowledge engineering in its widest sense.

2.1 Knowledge

The term 'knowledge' has conventionally been used to connote the amount and comprehensiveness of what is known [Mer02]. In artificial intelligence it involves facts, assumptions, beliefs, probabilities, and heuristics concerning some application area.

Knowledge can be categorised in various ways. On the basis of its use, knowledge can be characterised as declarative, procedural, or metaknowledge [Nyk00, Tur93]. Declarative knowledge describes facts and events as well as their relationships in a domain. Procedural knowledge describes how objects participate in activities and events. It tells how objects behave. Metaknowledge is knowledge about the knowledge itself. For example, the methods and strategies used to guide and control problem solving belong to metaknowledge. It can also express the level of certainty and comprehensiveness of knowledge.

Another categorisation includes deep and shallow knowledge [Nyk00, Tur93]. Deep (scientific) knowledge consists of models and theories concerning phenomena and mechanisms of the domain. On the other hand, shallow (experiential) knowledge consists of heuristics and experiences. Heuristics are useful rules of thumb derived from experience. They express connections between observations and conclusions.

Yet another means of classifying knowledge is to characterise it as tacit or explicit [Nyk00]. A part of intelligent activity is based on knowledge ordinarily

inaccessible to consciousness. Accordingly, this type of knowledge is out of the reach of explication, and it is characterised as tacit. Explicit knowledge, on the other hand, can be explicated, for example, by verbalising it.

Experts typically operate in problem solving or decision making tasks. Knowledge needed to perform these tasks is called expertise [Tur93]. It is formed by education and practical experiences. The systematic part of expertise consists of theories, models, and exact methods of the domain. Experience-based knowledge is private and less generally applicable than other forms of expertise.

2.2 Expert systems

Expert systems (ESs) [Nik97, Tur93, Wat86] are knowledge-based systems that use domain-specific knowledge and general inference mechanisms to solve complex real world problems. Expert systems are not truly intelligent – they lack common sense and inference from it. Expert systems may be characterised as computerised advisory systems that provide relevant material to be interpreted and utilised for humans in various problem solving and decision making situations. They are repositories of expertise sharing knowledge.

In order to be considered an expert system, a computer program has to have certain characteristics in addition to the capability to perform at the level of human experts in the domain [Nik97]. These characteristics include the system architecture and the approaches used to achieve the desirable performance. The main architectural components of an expert system are the knowledge base, the knowledge acquisition mechanism, the inference engine, and the user interface [Nik97]. The knowledge base contains the domain knowledge coded using some knowledge representation formalism. The knowledge base is constructed and updated by the knowledge acquisition mechanism. The inference mechanism contains algorithms for the manipulation of the knowledge stored in the knowledge base. It contains the control, search and inference methods needed in problem-solving. These methods are typically heuristic rather than exact. They can cope with incomplete and uncertain knowledge. The expert system interacts with the

user via the user interface. In addition to the above four components, an expert system usually has an explanation mechanism that justifies the decisions.

The life cycle of an expert system resembles the life cycle of a typical conventional system [Boe81] with its analysis, definition, design, implementation, testing, and maintenance phases. However, the standard conventional life cycle models do not fully model the development process of an expert system, at the beginning of which the requirements for the system are not necessarily known. Therefore, evolutionary prototype approaches [GD93, Nik97, Tur93] based on experimentation and iteration are used in developing expert systems.

The next section deals with knowledge acquisition, which is the most important, but also the most difficult, phase in the construction of an expert system.

2.3 Knowledge acquisition

The process of transferring the concepts, facts, and strategies needed in problem-solving from knowledge sources to the expert system is called knowledge acquisition [Nik97, Tur93]. It consists of two parts: knowledge elicitation and knowledge representation. Knowledge acquisition usually continues during the whole life cycle of the expert system.

The most common knowledge elicitation method is interview [Nik97, Tur93]. In its simplest form, an interview is a conversation between the knowledge engineer and the expert; the other forms are variations of this basic arrangement. At the beginning of the knowledge acquisition process, interviews are typically unstructured and unfocused becoming structured and focused on specific issues after the common knowledge of the domain is elicited.

Tracking methods [Tur93] such as protocol analysis attempt to trace the problem-solving process of an expert. Protocol analysis [Nik97, Tur93] is a formal method to find out what information and methods the expert uses in her reasoning. In this method, the expert is asked to 'think aloud' while performing a task. The expert's 'thinking' is recorded or documented, and the result of this is called a protocol. The knowledge engineer later analyses, interprets, and transforms the

protocol into knowledge that is reviewed by the expert.

In some domains it may be possible to observe an expert at her work. Observation gives a knowledge engineer a realistic picture of the difficulty of the problem solving situation.

Knowledge acquisition by manual methods has turned out to be expensive and time-consuming. The productivity of these methods is often low and the results may be unreliable [Qui88, Tur93, Wat86]. One reason for this is the paradox of expertise [Wat86]: The more skilful the expert is, the greater the difficulties she has in articulating her knowledge. This also applies to the complexity of the application domain. When the complexity grows, the proportion of knowledge unreachable to the awareness of the expert grows, too [Mic87]. Behaviour at expert level is typically automatised. The problem-solving process contains unconscious and routine activities which are difficult to become aware of. For an expert, many things are self-evident, and thus, she does not mention them. An interesting characteristic in experts' behaviour is that they do not necessarily use the formal models and methods they have learned in their reasoning [Wat86]. When they are asked to describe their reasoning, they state what they should do according to the learned methods - not what they actually do. Naturally, the knowledge engineer's limited capability to follow the expert's 'train of thought' may also cause difficulties. The engineer should have enough domain knowledge in order to be able to understand the expert's reasoning.

To widen the bottleneck of knowledge acquisition, semi-automatic and automatic tools have been developed. These can be classified, for example, into conceptualisation, task specific, model-based, or refinement tools [Nik97]. Machine learning methods [Mit97] provide an alternative or addition to the manual knowlegde acquisition methods. With these methods, the process of knowledge acquisition can be partly automated. The expert does not need to articulate her knowledge explicitly but to select example cases from the domain, from which a machine learning system learns the domain knowledge. Then, the expert evaluates the learned knowledge. Machine learning is discussed in Chapter 3.

In order to be manipulated by a computer, the knowledge has to be represented in a suitable form. The most common representational formalism is rules represented as if-then statements [Nik97, Wat86]. Other formalisms include frames and semantic networks. These formalisms represent the knowledge in a symbolic form. In hybrid expert systems [Nik97], combining the symbolic approach with, for example, a connectionist approach, the knowledge can also be in a sub-symbolic form, such as in the weights of a neural network.

Chapter 3

Machine learning

The ability to learn is one of the most essential characteristics of intelligent behaviour. Machine learning is a subfield of AI studying computational methods that can improve performance on some task by learning [Mit97]. The aims of machine learning research may be cognitive, technical, or theoretical [CMM83]. Cognitive aims seek to modell human learning at some level. Automating the process of knowledge acquisition for knowledge-based systems is an example of a technical aim. Theoretical analysis considers, for example, characteristics of learning methods such as their scope and limitations. Like AI, machine learning is an inherently interdisciplinary field. Statistics, for example, is widely utilised in the field of ML.

Machine learning methods can be classified on the basis of various criteria such as the underlying learning strategy, representation of knowledge, or application domain [CMM83]. Langley and Simon [LS95] find five major paradigms in the field of machine learning: neural networks, instance-based learning, genetic algorithms, inductive learning, and analytic learning. These paradigms share the common goal of improving the performance of some task, which is typically achieved by finding and exploiting regularities in training data. Empirical evaluation methods using a separate test set are employed to show that learning has resulted in the improved performance. Differences between the paradigms derive

from the learning algorithms and the representational formalism for the learned knowledge.

Knowledge acquisition and data mining [FPSS96] are important application areas of ML. ML methods have been utilised in a wide variety of application domains such as credit card fraud detection, handwritten character recognition, speech recognition, marketing analysis, quality control in manufacturing, airline seating allocation, food and chemical formula optimisation, reengineering of business process, parsing the Japanese language, diagnosis of mechanical devices, and automatic classification of celestial objects [All94, HSO99, LS95, Mit97, WRL94]. The medical domains in which ML has been used are diagnosis of acute appendicitis [PEJ96], diagnosis of dermatological diseases [DOMK+01, WW00], diagnosis of female urinary incontinence [LJ98], diagnosis of thyroid diseases [FNI91, QCHL87], finding genes in DNA [SDFH98], identification of chest X-ray reports supporting acute bacterial pneumonia [WCFCH01], integration of western and eastern medicine [Phu97], outcome prediction of patients with severe head injury [PMLP97], prediction of metabolic and respiratory acidosis in children [BKP+00], as well as relating clinical and neurophysiological assessment of spasticity [ZSB+97], among many others.

In this chapter, we discuss the machine learning issues important for our work. We first consider classification as a learning task. Then, we desribe the process of applying ML methods. Finally, we introduce the machine learning methods used on the vertigo data.

3.1 Classification as a learning task

Medical decision-making can be seen as classification: A physician classifies the symptoms of a patient to a certain disease group on the basis of her knowledge. Thus, learning classification models for the six otoneurological diseases is the learning task in this study.

General classification models can be learned by analysing data. Training data consist of cases (examples, instances, or objects) that are described using vectors

of attributes (variables, or features). Attributes may be qualitative (measured with a nominal or an ordinal scale) or quantitative (measured with an interval or a ratio scale) [Agr96, Sha96]. In supervised learning, cases usually belong to one of the mutually exclusive classes, and the class information is utilised in learning. Attributes are adequate for the classification task, if all the cases having identical attribute vectors belong to the same class [Qui86].

The classification model constructed is, on the one hand, a description of the training data, and on the other hand, a classification rule that can be applied to new cases if the training data form a representative sample from the object space.

3.2 Process of applying machine learning

Langley and Simon [LS95] identify the following main phases in the process of applying ML methods: formulating the problem into a form suitable for the ML methods selected, determining the representation, collecting the training data, evaluating the learned knowledge, and fielding the learned knowledge (that is, taking it into use). In the next section, we consider in greater detail data pre-processing, which involves, among other things, the tasks of selecting the attributes and examples used in learning. The evaluation of learned knowledge is discussed in Section 3.2.2.

3.2.1 Pre-processing

Training data form the basis for the learning process. The learning method can find only the concepts included in the training data, and thus, selected attributes and instances should cover different situations appearing in the problem domain. Constructing the training data is an essential part of applying ML techniques and it assumes the involvement of both a knowledge engineer and a domain expert.

Feature subset selection

Determining the representation of learning data [LS95], that is the attributes used to describe training cases, is an important task. Although a larger attribute set usually carries more information than a smaller one, it does not necessarily produce better learning results. Increasing the number of attributes probably increases the amount of noise and missing data. Additional, possibly redundant or irrelevant attributes may also interfere with the generalisation capability of ML methods [KJ97]. Therefore, a carefully selected subset of attributes may produce better results. Generated models may be simpler, easier to understand, and more accurate. However, finding of a good subset may be a difficult and tedious task due to the wealth of attributes available, which calls for automated methods.

Various methods have been developed for feature subset selection (FSS) [BL97, DL97, KJ97]. These methods follow embedded, filter, and wrapper approaches for selecting attributes [BL97]. In the first approach, the selection of attributes is embedded in the learning algorithm itself. Examples of such algorithms are ID3 [Qui83], C4.5 [Qui93], and CART [BFOS84]. In the filter approach, useful attributes are filtered before the actual learning, and the filtering is based on the characteristics of the training data [BL97, DL97, KJ97]. The simplest way is to choose attributes having the strongest associations with the class attribute. Another way is to find a minimal set of attributes that are adequate for the discrimination of classes [BL97, KJ97]. Further, algorithms with an embedded selection capacity can be used as filters for other learning methods, like decision trees as filters for nearest neighbour classification [BL97, KJ97]. In the wrapper approach [KJ97], the subset selection method is wrapped around the learning algorithm. The FSS method guides the search for attribute sets and uses the learning method to evaluate these sets. In this approach, heuristics and biases of the learning algorithm and also the interaction between the learning algorithm and the training set are considered [KJ97]. The aim is to find an optimal feature subset with respect to a particular learning algorithm and a domain.

In addition to the above three explicit FSS approaches, there exist implicit

weighting methods for defining the usefulness of attributes [BL97]. Tuning the weights of neural networks with algorithms such as backpropagation [Mit97] as well as memory-based reasoning, using the Value Difference Metric (VDM) with a weighting scheme [SW86], are examples of these.

Data transformation

Many ML algorithms make assumptions about the scales of attributes. ID3, for example, requires categorical attributes and a binary class attribute [Qui83], whereas a nearest neighbour classifier with the Euclidean distance function works best with quantitative attributes [WM97]. Sometimes, ordinal attributes can be used like quantitative ones by assigning consecutive numerical scores to the ordinal categories [Agr96, Qui93], and nominal attributes can be transformed into binary dummy attributes [Agr96, Wei85]. If learning requires a discrete feature base, quantitative attributes can be coded as qualitative ones. A frequently used and perhaps the simplest method for discretisation is equal width interval binning [DKS95]. Discretisation results in loss of information, but this is not necessarily harmful for learning results. In an empirical comparison of discretisation methods [DKS95], discretisation prior to learning significantly improved the accuracy of induction methods in some tasks.

Selection of training cases

In its extremes, training data may be a 'raw' sample extracted from a large real world database or a set of instances carefully chosen by an expert. If the sample is large enough, it gives a statistically reliable picture of the task to be learned, but may not be optimal for learning purposes. Data may contain redundant cases, and rare cases essential for the problem solving may not be presented. In training data selected by an expert, every instance has a special meaning for the concept to be learned.

Behind the current theory of ML, there is an assumption that distributions of training and test data are similar [Mit97]. Good results on some distribution do

not necessarily guarantee good results on other distributions, and, thus, learning is most reliable when the distribution of the training data is as close as possible to the real world distribution [Mit97]. However, this assumption must often be violated in practical situations due, for example, to unavailability of rare cases. Further, a highly imbalanced class distribution may cause troubles for ML methods aiming at the maximal predictive accuracy [KM97, KHM98]. Balancing the class distribution, for example, by removing cases from the large classes [KM97] or by generating artificial cases of the small classes [KM97, Swi96] violates the original distribution. Other approaches to balance class distribution include weighting the cases and assigning different missclassification costs [BL97, KM97, KHM98].

In real world learning tasks, data are typically noisy, that is, attribute values and class labels may be erroneous. Noisy data result in overly complex, incomprehensible models and possibly decreased accuracy. To remove the effects of noise, methods for simplifying learned models can be used. An alternative approach is to discard the noisy cases from the training data. In general, noisy cases are in a way exceptional compared to other data. A case may be an outlier [BL87] due to an extreme value of one attribute or due to an unusual combination of normal attribute values. Whether the case is an outlier because of noise or the natural variation in the population is an important question. In some domains, removing exceptional cases that are not noisy may be harmful for learning [DVDBZ99].

Missing values

Real world data are often incomplete, that is, not all values are known. Unknown attribute values have to be taken into account in the construction of a model as well as its later usage. Some learning methods can handle incomplete data, while others require complete data. C4.5 is an example of the former case, and neural networks are an example of the latter. If the number of missing values is small, learning can be based on the complete cases only. When this is not possible, the missing values may be filled in using different methods [Wei85].

3.2.2 Evaluation

An objective way (with respect to the chosen criteria) for evaluating learning output is the use of performance measures. The most common measure used to characterise the performance of a classifier is accuracy [Lav99]. Accuracy (ACC) is calculated as the percentage of correctly classified cases:

$$ACC = 100\frac{\sum_{c=1}^{C} tpos_c}{\sum_{c=1}^{C} pos_c}\%,$$

where C is the number of classes, $tpos_c$ is the number of correctly classified cases in the class c, and pos_c is the total number of cases in c. The error rate measures the proportion of misclassified cases.

If the class distribution is highly imbalanced, the accuracy does not give the whole picture of the classifier's quality. The accuracy may be high, even though the cases of small classes are poorly identified [KM97, KHM98]. Therefore, the true positive rate is calculated separately for each class in order to obtain more detailed information about the classifier's performance. The true positive rate for class c (TPR_c) is calculated as the percentage of correctly classified cases of the class:

$$TPR_c = 100\frac{tpos_c}{pos_c}\%.$$

In the case of a multi-valued class label, the classifiers may also be constructed in the form of one class (the positive class) versus the other classes (the negative class). Then, the identification of the negative cases is measured by the true negative rate.

The data are typically divided into training and test sets. The classifier is constructed from the training set and tested with the separate test set. Descriptive performance measures are calculated from the training set and predictive performance measures from the test set. Due to the small amount of data available, a technique called N-fold cross-validation [BFOS84, Qui93] can be used instead of a separate test set to estimate the predictive performance. The data are divided into N subsets with a nearly equal size and a class distribution as close as possible to the class distribution of the original data. The classifier is constructed

from $N-1$ subsets, and the remaining subset is used as the test set. This is repeated N times such that each subset is once the test set, and, accordingly, predictive performance measures are calculated as the average of N experiments. Cross-validation can be repeated several times for random partitions. The average measures calculated from the results of different cross-validation times are fairly reliable estimates for the performance of a classifier constructed from the whole training data [Qui93].

Even if experts are not capable of fully articulating their knowledge, they can evaluate the quality and correctness of the learned models [LS95, Lav99, NCW91]. The models should be intelligible and reasonable from the viewpoint of the experts in order to be called knowledge.

3.3 Decision tree induction

Decision tree induction [Qui86] is one of the most widely used approaches for inductive inference. DT algorithms represent the learned classification models as decision trees. Most of these algoritms have their origins in Hunt's Concept Learning System (CLS) [HMS66] and ID3 [Qui79, Qui83]. Other early DT algorithms are Assistant [KBR84] (as described in [Qui86]) and CART [BFOS84]. C4.5 [Qui93] is a descendant of ID3 that addresses issues arising in real world classificaton tasks.

A decision tree is a recursive structure in which the inner nodes contain tests based on attributes and the leaves contain the class information [Qui86, Qui90]. The classification of a new instance starts from the root of the tree. The attribute assigned to the root node is examined and a branch corresponding to the attribute value is followed. This process continues until a leaf node predicting the class of the instance is encountered. The number of tested attributes depends on the classification path; it is not necessary to test all the attributes in all the paths [Qui90]. The classification paths from the root to the leaves are conjunctions of constraints set on attributes, and the whole decision tree is a disjunction of these paths. Due to the symbolic knowledge representation, decision trees are relatively

Table 3.1: A decision tree for benign positional vertigo. The **'BPV'** leaf represents the decision 'benign positional vertigo' and the **'NOT'** leaves represent the decision 'not benign positional vertigo'.

Presence of hearing loss = yes: **NOT**
Presence of hearing loss = no:
:.....Frequency of vertigo attacks <= 1: **NOT**
Frequency of vertigo attacks > 1:
:.....Injury = no: **BPV**
Injury = yes: **NOT**

easy to comprehend and scrutinise. A simple decision tree for diagnosing benign positional vertigo is presented in Table 3.1.

According to the name of the algorithm family (Top Down Induction of Decision Trees (TDIDT) [Qui86]), the construction of a decision tree starts from the root. The training cases are examined to find the best attribute. A node testing the best attribute is formed, and the training cases are further divided into subsets according to the outcome of the test. For each subset, a decision tree is constructed in the same manner. This recursive process continues until each subset represents a single decision class or until the subsets are sufficiently homogenous.

If the attributes are adequate, it is possible to construct a decision tree that classifies all the training instances correctly [Qui86, Qui90]. Usually, several DTs fit the training data, but not all the trees are equally good. The inductive bias of DT algoritms can be characterised as a preference for smaller trees deriving from the principle called Occam's razor [Mit97]. Smaller trees are more general, easier to understand, and possibly more accurate. The size of a tree can be defined as, for example, the number of leaves or nodes, or the average number of tests needed in the classification of training cases [Qui86, Qui90]. To produce simpler and smaller

trees, decision tree algorithms select the best attribute with the criterion based, for example, on information (or entropy), error, or statistical significance [Qui90]. Empirical comparisons of various splitting rules have been studied by Mingers [Min89b] as well as Buntine and Niblett [BN92]. The preference for smaller trees and the attribute selection criterion form the inductive bias of a decision tree algorithm [Mit97].

Noise results in overly complex and incomprehensible decision trees that attempt to fit the irregularities in the training data. To produce more comprehensible trees, heuristic pruning methods are used [Qui88]. These methods simplify a decision tree by replacing subtrees that hardly affect the predictive accuracy, with leaves [Qui88, Qui90, Qui96]. Common pruning techniques [Qui96] are based on cost-complexity models [BFOS84], pessimistic accuracy estimates [Qui88], or error-based models [Qui93]. An empirical comparison and a survey of pruning methods are found in [Min89a] and [BA97] respectively.

Since the early days of ID3, a variety of decision tree algorithms has been developed. As mentioned earlier, these algorithms employ different attribute selection criteria and pruning methods. Different approaches to handle noise and missing values are also used. A more sophisticated way of constructing DTs is to select attributes on the basis of various criteria depending on the node location [Bro95] and to form tests using more than one attribute [BU95, Qui96]. Incremental induction of DTs [KM96, Utg89] has also been studied, and hybrid decision trees utilising genetic algorithms have been presented [KPY$^+$01].

In this work, the See5 decision tree program [Qui97] was used. It employs the gain-ratio criterion [Qui93] for selecting the best attribute. To handle missing values, See5 uses a probabilistic approach. A case having a missing value is split into fractions according to the probabilities of attribute values in a node, and fractional cases are then sent down to corresponding branches. The constructed trees are simplified with an error-based pruning method [Qui93].

3.4 Other machine learning methods

In addition to DTI, we have used nearest neighbour (NN) classification and artificial neural networks (ANN) to classify the cases of the vertigo data. Next, we briefly describe these two methods.

Nearest neighbour classification [LS95, Mit97] is the most basic instance-based learning method [AKA91], which stores training cases and later uses them to predict the class of a new case. The knowledge is in the form of the stored instances, and generalisation occurs when a similarity or a distance function is used to find the nearest stored case(s) giving the class for a new case. Of the various distance functions, the Euclidean distance function, for example, is appropriate for quantitative attribute spaces and the Value Difference Metric (VDM) [SW86] can be used in qualitative attribute spaces. Wilson and Martinez [WM97] propose three more sophisticated distance functions for use with mixed data having both quantitative and qualitative attributes. The Heterogeneous Value Difference Metric (HVDM) [WM97] employs a normalised Euclidean distance function for quantitative attributes and a simplified, normalised version of the VDM for qualitative attributes.

Neural networks [LS95, Mit97, Swi96] represent knowledge as a multilayer network of nodes in which activation spreads from the input nodes through the nodes of the hidden layers to the output nodes. The weights associated with the connections between the nodes define the amount of spreading activation. The classification accuracy of the net is improved by tuning the weights by algorithms such as backpropagation [Mit97]. The activations of the output nodes can be transformed into the numerical predictions or discrete decisions that express the class of the input.

Chapter 4

Otoneurological expert system ONE

The main components of the otoneurological expert system ONE, as well as most expert systems, are the user interface, inference engine, knowledge base, knowledge editor, query data, and answer database [AJP93, KPAJ96]. The system was implemented in C++ language [AJP93]. In this chapter, we introduce those components of ONE which are the focus of our study: the database, the knowledge representation model, and the inference mechanism. We also describe the four versions of the vertigo data retrieved from the database of ONE and used in this study.

4.1 Database

ONE stores the patient data in the answer database, which is in the format of a relational PARADOX database [AJP93]. The database contains 170 attributes that can be divided into four categories [KPAJ95]:

- patient demographics and referring physician (Table A2 in Appendix)

- symptoms: vertigo, hearing loss, tinnitus, unsteadiness, headache, anxiety, and neurological symptoms (Table A3)
- medical history: use of ototoxic drugs, head trauma, ear trauma and noise injury, ear infections and operations, specific infections (for example, borrelia, chlamydia, and bacterial meningitis), and other diseases (Table A3)
- findings: clinical findings, otoneurological data, audiometric data, imaging data, and fistula testing (Table A3).

In addition to the attributes in the database, ONE uses in its inference 11 derived variables (Table A4) calculated from the original attributes by logical, relational, and arithmetical operations.

The use of the expert system does not require the user to answer all the questions, and, accordingly, most attributes have missing values.

4.2 Knowledge base

The knowledge base contains a description or a pattern for each disease in the form of fitness values and weights [AJP93]. The current version covers 18 diseases and disorders (Table A1) [AJ96, KAJP98]. The significance of an attribute for a disease is expressed as a weight value assigned to the attribute. Fitness values set to attribute values express the correspondence between these values and the disease.

A group of experienced otoneurologists defined the disease patterns for the first version of ONE on the basis of their knowledge and the data obtained from the literature [KAJP98]. The number of relevant attributes varies according to the diseases. Some diseases can be inferred with few attributes [KAJP98], for example, borreliosis with one attribute [AJP93]. For the six diseases in this study, the number of relevant attributes varies from 63 (vestibular schwannoma) to 78 Menière's disease (Table A5).

Weight values typically vary from 0 to 5. The weight 0 means that the attribute does not concern the disease at all. The greater the weight value is, the more

important the attribute is for the disease. Some notably large weight values are also used: for sudden deafness the attribute concerning the type of hearing loss with the weight 40, for traumatic vertigo the attribute head trauma with the weight 200, and for vestibular schwannoma the attribute concerning a tumour in the acousticus nerve with the weight 200 (Table A6). (The original disease descriptions also include negative weights. However, the negative weights can be excluded transforming the fitness values of the corresponding attribute values and, hence, for the sake of clarity, we use only positive weights.) The fitness values vary from 0 to 1.

Necessary attribute values (NAVs) are assigned to some attribute-disease combinations. In order to be diagnosed as having a certain disease, the case has to fulfil the requirements concerning the necessary attribute values set for this disease. The number of NAVs for the six diseases in this study varies from 2 (traumatic vertigo) to 16 (benign positional vertigo) (Table A5).

4.3 Inference mechanism

ONE's inference mechanism resembles the nearest neighbour method of pattern recognition [Mit97]. It transforms attribute values to scores based on the fitness values and weights. Let d be a disease and $n(d)$ be the number of attributes associated with d in the knowledge base. The score $S(d)$ for the disease d is calculated as

$$S(d) = \frac{\sum_{i=1}^{n(d)} x(i)w(d,i)f(d,i,j)}{\sum_{i=1}^{n(d)} x(i)w(d,i)},$$

where

- $x(i)$ is 1, if the value of the i^{th} attribute is known for the disease d, otherwise 0
- $w(d,i)$ is the weight value for the attribute i
- $f(d,i,j)$ is the fitness value for the value j of the attribute i.

To handle uncertainty caused by missing values, ONE generates upper and lower bounds for the score. The lower bound is calculated using the lowest fitness values for the missing values and the upper bound using the highest fitness values. The narrower the difference between the bounds is, the more reliable the inference is.

The diseases with the highest scores are the best fits and suggested by ONE. In addition to the scoring scheme, ONE uses rules that are expressed as necessary attribute values assigned to diseases. If the NAVs do not hold true, the disease is inferred to be irrelevant and the corresponding score and its upper and lower bounds are not presented.

4.4 Vertigo data

The vertigo data used in this study were retreived from the ONE database. The data contain cases from the six largest diagnostic groups of the database: benign positional vertigo, Menière's disease, sudden deafness, traumatic vertigo, vestibular neuritis, and vestibular schwannoma [Ken96a]. These cases are patients referred to the vestibular unit of the Helsinki University Central Hospital.

During the development of ONE, a database of 1167 vertiginous patients was collected prospectively [Ken96a]. The otologists were able to confirm the diagnosis of 872 patients, of whom 746 belonged to the six largest diagnostic groups [Ken96a]. Because of the nature of their research, the otologists excluded cases having confounding values (that is, signs and symptoms not related to the current disease), which finally resulted in the 'original' data set of 564 cases [Ken96a] (Vertigo1 data). We started our machine learning experiments with this data set and used it in papers [I] and [II]. In the autumn of 1999, 76 new cases were added to the ONE database and the vertigo data. We studied the enlarged data set of 640 cases (Vertigo2 data) and found confounding values [VJKP00]. We wanted to examine the effects of the confounding values more thoroughly, and accordingly, 48 benign positional vertigo cases and 40 vestibular neuritis cases discarded earlier by physicians because of the confounding values, were retrieved from the database of ONE. These cases were combined with the data set of 640 cases resulting in the

Table 4.1: Distribution of the diagnosis variable in the four versions of the vertigo data.

	Vertigo1		Vertigo2		Vertigo3		Vertigo4	
Diagnosis	N	%	N	%	N	%	N	%
Benign positional vertigo	59	10.5	75	11.7	123	16.9	146	17.9
Menière's disease	243	43.1	283	44.2	283	38.9	313	38.4
Sudden deafness	21	3.7	30	4.7	30	4.1	41	5.0
Traumatic vertigo	53	9.4	56	8.8	56	7.7	65	8.0
Vestibular neuritis	60	10.6	68	10.6	108	14.8	120	14.7
Vestibular schwannoma	128	22.7	128	20.0	128	17.6	130	16.0
Total	564	100.0	640	100.0	728	100.0	815	100.0

extended data set of 728 cases (Vertigo3 data). This version of the data set was used in paper [III]. In the summer of 2000, new cases were added to the database and the vertigo data resulting in a set of 815 cases (Vertigo4 data). This data set was used in papers [V] and [VI]. Table 4.1 shows the class distributions in the four versions of the data.

The attributes in the Vertigo4 data and the corresponding numbers of missing values are shown in Table A3. The data concerning a patient's symptoms, earlier diseases and accidents as well as the use of drugs, alcohol and tobacco were acquired through a questionnaire completed by the patient. The other data derives from the examinations ordered by the physician in charge [Ken96b]; no examinations were made solely for purposes of data collection [Ken96b]. Therefore, the database and the four versions of the vertigo data contain missing information.

Chapter 5

Results

The aim of our study was to examine the possibilities for expanding and refining the knowledge base of ONE by machine learning methods. Throughout the study, the acquisition of diagnostic knowledge and data pre-processing, especially the selection of attributes and examples, were targets of our special interest. In Section 5.1, we review the work on feature subset selection for decision tree induction: the subsets formed on the basis of expert's knowledge [I], measures of association and an entropy-based approach in the evaluation of attributes [II], and a variable grouping method [IV,V]. The effects on decision trees of cases with confounding values [III] are considered in Section 5.2. The refinement of the knowledge base of ONE is discussed in Section 5.3. The results obtained by the fitness values learned from the data and different weighting schemes [VI] are studied. Finally, the six diseases as the concepts to be learned with respect to different machine learning methods are discussed in Section 5.4.

5.1 Feature subset selection

5.1.1 Subsets formed by expert

We started our studies on decision trees and the vertigo data in paper [I]. The purpose of the paper was to examine the suitability of decision tree induction for acquiring diagnostic knowledge for the six otological diseases. The possibility to use DTI to extract relevant attributes from a large set of attributes was also of interest.

We used the Vertigo1 data and five attribute sets defined by an experienced physician. The smallest attribute group contained five attributes, which were earlier defined as key questions [Ken96a]. The second group of 38 attributes had resulted in good classification accuracy with genetic algorithms [KLPJ99] and had a small number of missing values. The three remaining groups were based on the importance categories of the attributes defined by the physician. First, the physician eliminated 47 attributes as irrelevant with respect to the classification tasks in question or as having a large number of missing values. Then, she classified the remaining 123 attributes into three categories based on their importance. These categories have the 53 most important, the 57 second most important, and the 13 third most important attributes. As a result, she obtained groups of 53, 110, and 123 attributes. Further, we used six binary class attributes to distinguish the diseases, that is, six separate classification tasks were formed to separate each disease (the positive class) from the other diseases (the negative class).

Cases with benign positional vertigo, Menière's disease, traumatic vertigo, and vestibular neuritis were reliably inferred even with the group of five attributes (TPR over 85%). The level of the best decision trees with respect to the true positive rates and accuracies (both over 95%) was obtained with the group of 38 attributes in the case of TRA and with the group of 53 attributes in the case of BPV and VNE. For Menière's disease and vestibular schwannoma, the group of 110 attributes was needed to obtain the best performance (TPR over

95% and ACC over 90%; and TPR over 80% and ACC over 95% respectively). Considering these five diseases, increasing the number of attributes resulted in better classification results.

Sudden deafness was an exception. First, the group of five attributes yielded a majority classifier classifying all the cases as not having sudden deafness with TPR of 0%. Further, the best results (TPR of 71%) were obtained with the group of 38 attributes. The lower true positive rate for the group of 53 attributes can be explained by the absence of the attribute concerning the duration of hearing loss within the 38 attributes. The three largest attribute groups decreased the TPR partly due to the greater amount of confounding values, which made the cases resemble vestibular schwannoma or Menière's disease, and on the other hand due to the increasing amount of missing information [KVPJ00].

The results of paper [I] showed that decision tree induction is well suited to the modelling of the six otoneurological diseases. The trees constructed performed well with respect to the TRPs and ACCs, as well as to the evaluation made by the expert. The method was capable of picking up the useful attributes from the large sets of attributes. The number of attributes in the best decision trees ranged from 3 to 15. The most important attributes were the occurrence and duration of hearing loss, the type of hearing loss, the occurrence and duration of vertigo, the frequency of vertigo attacks, the occurrence of tinnitus, and the occurrence of a head injury in timely relation to the onset of vertigo. The attributes in the DTs agreed well with the earlier results obtained in the discriminant analysis of these six diseases [Ken96a] and with the disease patterns of ONE. However, the numbers of attributes in the DTs were notably smaller than in the patterns. Interestingly, the attributes concerning the occurrence of vertigo, hearing loss and tinnitus that do not affect ONE's inference but direct the flow of the questions in the user interface, were selected as the root attributes for several decision trees. The physician considered the decision trees to be easily explored and comprehended, as well as advantageous in gaining new information for diagnostic work [KVPJ00]. The DTs are useful as such and they can be further utilised by embedding their

knowledge in ONE [KVPJ00].

In general, the group of at least 110 attributes was needed to build the best decision trees, which proved the usefulness of attributes classified into the category of the second most important. The DTI method was able to extract good attributes from the largest attribute sets in the case of BPV, MEN, TRA, VNE, and VSC. The classification task concerning sudden deafness revealed the need for feature subset selection with the best results yielded by the group of 38 attributes.

5.1.2 Evaluation of training data

In paper [II], we examined measures of association and the entropy-based approach [Swi96] in evaluating the quality of the training data for decision tree induction. The results of paper [I], especially the majority classifier obtained with the group of five attributes for sudden deafness, and the poor classification results for two other otological data sets (the conscript data [RJH99], and the postoperative nausea and vomiting (PONV) data [HSK94, HSK95, Hon96a, Hon96b, HS98]) suggested the experiments of paper [II].

In the case of the vertigo data, the five key attributes [Ken96a] and the six binary class attributes (as in [I]) were employed. To eliminate the effects of the missing values, we used the subset of 459 complete cases from the Vertigo1 data. The conscript data was a sample of 22,252 complete cases described by 16 attributes. The cases with hearing loss represented the positive class and the cases with normal hearing the negative class. The PONV data contained 245 cases which had undergone middle ear surgery. The cases were described by eight attributes. The class attribute concerned the need for dehydrobenzperidol as a rescue drug; the cases needing the rescue drug formed the positive class.

Associations between the class attribute and the other attributes in each of the eight classification tasks were assessed with the chi-square test of k independent samples. For significant associations ($p < 0.05$), the strength of the association was determined with the phi coefficient for the 2×2 tables and with the Cramér's V coefficient for the larger tables [Pet97]. To assess the strength of the relationship,

the following classification was used for the absolute value of the phi coefficient and Cramér's V: 0.00-0.29 weak, 0.30-0.49 low, 0.50-0.69 moderate, 0.70-0.89 strong, and 0.90-1.00 very strong [Pet97].

For each task, the entropy of the class attribute was calculated. Then, the conditional entropy of the class attribute given the input vectors was calculated. Finally, the ratio of the conditional entropy to the original entropy was calculated. The ratio of the entropy values indicates the adequacy of the attributes for the classification task [Swi96]. A value near 1 means that the attributes are not adequate, and thus, the task is not learnable. A low value suggests that the attributes are adequate, and the classification task should be learnable.

In the conscript data set, the significant associations between the class attribute and the other attributes were very weak: the absolute value of the measure of association ranged from 0.01 to 0.11. The ratio of entropy values was 0.96, and accordingly, both the descriptive and predictive true positive rates (45.5% and 43.0% respectively) and accuracies (55.6% and 54.7% respectively) of the decision tree were low.

In the classification tasks concerning benign positional vertigo, Menière's disease, traumatic vertigo, and vestibular neuritis, the strength of the significant associations varied from weak to very strong (the absolute value of the measure of association varied from 0.11 to 0.96). For each task, at least one relationship of moderate strength was found. The ratios of the entropy values were low, varying from 0.00 to 0.07. The accuracies and true positive rates of decision trees were high, over 90%, except the predictive accuracy of 88.7% for Menière's disease.

For sudden deafness the strength of association varied from weak to low (the measure of association varied from 0.12 to 0.30), and for vestibular schwannoma from weak to strong (the absolute value of the measure of association varied from 0.16 to 0.79). The ratios of entropy values were low in the case of these two diseases: 0.14 for SUD and 0.12 for VSC. On the basis of the accuracies over 90%, the decision trees constructed were feasible. However, the tree for SUD was a majority classifier with a TPR of 0%. For VSC, the TPR was 66.3%, which can

be evaluated as moderate performance.

In the PONV data set, the strength of significant associations varied from weak to low (the absolute value of the measure of association ranged from 0.14 to 0.34), and the ratio of entropy values was 0.18. The descriptive and predictive TPRs of the decision tree were low (58.4% and 44.6% respectively), while the corresponding accuracies were moderate (77.1% and 68.6%).

Overall, weak or low associations resulted in low accuracies and true positive rates, whilst moderate or stronger associations resulted in high accuracies and true positive rates. Exceptions to this pattern were also found. Vestibular schwannoma had the strongest associations with the duration of vertigo, the frequency of vertigo attacks, and the duration of vertigo attacks compared to the other diseases. However, the true positive rate for VSC was notably lower (28.6% or more) than for BPV, MEN, TRA and VNE. Even if an attribute had a strong association with the class variable, it was not necessarily inserted in the constructed decision tree. For example, the decision tree for VSC tested only one attribute, the duration of vertigo attacks, and excluded the other attributes with strong relationship. Further, an attribute having a weak or low association with the class variable in the whole example set may be useful in the lower parts of a decision tree when a subset of the example cases is considered. For example, the decision trees for BPV and MEN incorporated attributes with weak or low associations.

The ratio of the entropy values was negatively correlated with the true positive rate and accuracy. The Spearman rank-order correlation coefficient was -0.883 ($p = 0.010$) for the predictive TPR and -0.881 ($p = 0.004$) for the predictive ACC. Again, exceptions to the common pattern were found. In the PONV classification task, the ratio of entropy values (0.18) was greater than in the case of SUD (0.14). Still, the true positive rates for PONV (58.4% and 44.6%) were better than for SUD (0%). In the case of sudden deafness, the class distribution was greatly out of balance (only 3.3% of the cases were positive), which partly affected the poor TPR. The majority classifier for SUD had better accuracy than the decision tree

for VSC, although the ratios of the entropy values suggested the opposite. An interesting phenomenon was observed when considering the PONV, the VSC, and the conscript classification tasks. The ratios of the entropy values suggested that the tasks for PONV (0.18) and VSC (0.12) are learnable, whereas the conscript task (0.96) is not. However, on the basis of the predictive TPRs and ACCs the PONV task resembles the conscript task rather than the VSC task.

Although the ratio of the entropy values is low, indicating the adequacy of the attributes, the construction of a good decision tree, or some other classifier, may not be possible. First, the cases representing different classes may overlap in the attribute space, making generalisation impossible. Second, increasing the number of attributes in the training set decreases the ratio of the entropy values or keeps it unchanged [Qui86], but possibly results in worse classifiers [I, KJ97]. Other factors, such as imbalanced class distribution [KHM98], may also affect the quality of the classifier.

The results of paper [II] suggest that the measures of association and the entropy values indicate the quality of the training data, but the relationships between them and the quality of the constructed classifier are not straightforward. Hence, other approaches are also needed to guide the building of the training data. Nearest neighbour (NN) classification can be used to obtain information about the locations of the classes in the attribute space. NN results for the vertigo data are discussed in Section 5.4.

5.1.3 Variable grouping method

In this section, we first briefly describe the variable grouping method based on graph theoretic techniques and measures of association [IV]. (We assume a familiarity with the basics of graph theory and the complexity of algorithms, as given, for example, in [Eve79]). Then, we report the results of paper [V], in which the method was used to filter variable subsets for decision tree induction in the context of the vertigo data.

Table 5.1: Variable grouping algorithm.

Algorithm	VG
Input:	A data set D, a list of the measurement scales of its variables, and threshold values.
Output:	Variable groups, and the corresponding degrees of variables, cliques, and independent sets.

1. Compute the association matrix M inducing a graph G.
2. Find the connected components (variable groups) of G with the depth-first search.
3. For each connected component
   3.1 Find the cliques.
   3.2 Find the independent sets.

Association matrix

The variable grouping method presented in Table 5.1 takes as its input a data set D and a list of the measurement scales of its variables. The degrees of associations between the variables are calculated using appropriate measures of association and stored in the association matrix. The matrix can be seen as an undirected graph in which vertices represent variables and the degrees of significant associations reaching some threshold value establish connections between them. The time complexity for computing the association matrix for the data set depends on the number of variables and cases as well as on the measures of association used in the matrix. The time complexity for measures of association typically varies from $O(n)$ to $O(n^2)$, where n is the number of cases. Hence, the time complexity for computing the association matrix typically varies from $O(v^2n)$ to $O(v^2n^2)$, where v is the number of variables.

Variable groups

Related variables form variable groups. Two variables may be related to each other directly or via other variable(s). The structure formed by related variables corresponds to a connected component of the graph induced by the association matrix. The connected components are found by traversing the graph using the depth-first search. The time complexity for finding the components is linear on the size of the graph, that is, the total number of nodes and edges, but quadratic on the number of variables.

The variable groups found provide insight into the data set. They can be further utilised in feature subset selection. From the viewpoint of FSS, the structures formed by independent variables in a variable group are of special interest. A maximum independent set [Eve79] is the largest structure in which all the elements are not directly connected to each other. Thus, a maximum independent set contains the maximum number of variables without including correlated variables and redundant information. Alternatively, we can choose from a variable group a dominating variable with the largest number of associated variables, which thus widely 'represents' the information of the entire group. It should be noted that neither a maximum independent set nor a dominating variable is necessarily unique.

The threshold value for establishing connections between the vertices can be used to regulate the density of the graph. Decreasing the threshold value makes the graph denser, and accordingly results in larger connected components. In the selection of the threshold value, rules for the interpretation of association measures (found, for example, in [Pet97]) can be employed. The value used should be reasonable from the viewpoint of the application area in question.

The algorithm outputs the variable groups found with detailed information about them: the degrees of variables, the maximum cliques and the maximum independent sets. The output of the algorithm can be utilised by inputting it to a system that forms variable subsets by selecting from each variable group one of the dominating variables or one of the maximum independent sets, for example.

The filtered variable subsets are, in turn, inputs for a model forming method.

Experimental results

The variable grouping method was applied to the Vertigo4 data and the PONV data in paper [V]. The attribute groups found by the method were reasonable in the case of both data sets. However, the groups found in the PONV data could not been utilised in DT construction due to the nature of the classification task, and thus, we present here results for the vertigo data only.

The relationships between the 111 attributes of the Vertigo4 data were assessed using the Spearman rank-order correlation coefficient. The four threshold values of 0.30, 0.50, 0.70, and 0.90 corresponding low, moderate, strong, and very strong associations [Pet97] respectively, were used to generate four collections of variable groups. The numbers of groups (with two or more variables) in the collections were 23 (9), 57 (21), 84 (18), and 109 (2) respectively. The largest groups in the collections contained 61, 20, 5, and 2 attributes respectively. In an experienced physician's opinion, the variable groups were sensible. The most interesting collection, containing the Menièreformic attribute group, was obtained with the threshold value of 0.5. The five attribute sets used in the decision tree construction were based on this collection. These sets contained

1. all the variables (set A)
2. a maximum independent set of variables from each variable group (set B)
3. a dominating variable from each group (set C)
4. the variables concerning the frequency and duration of vertigo attacks from the vertigo group and a dominating variable from the other groups (set D)
5. a maximum independent set from the vertigo group and a dominating variable from the other groups (set E).

In the case of the attribute sets D and E, the inclusion of certain vertigo attributes from the vertigo group was based on the domain knowledge.

The best true positive rates varied from 68.3% (sudden deafness) to 90.1% (Menière's disease). Overall, the reduced variable sets B-E produced as good or even better predictive true positive rates for all diagnoses but sudden deafness. One of the reduced variable sets yielded the best true positive rate for all diagnoses excluding sudden deafness. The accuracy of the original tree was 80.7%, and for the trees B-E, it varied from 73.7% to 80.9% with a median of 80.1%. The original tree had 36 attributes and 58 nodes. The number of attributes in the trees constructed using the reduced attribute sets varied from 29 to 33 (median 31) and the number of nodes from 51 to 57 (median 52.5).

The use of the variable grouping method enabled the construction of the simpler decision trees with respect to the number of attributes and nodes in the trees. In general, the classification power of the reduced trees reached the level of the original tree. The reduced trees incorporated 'new' attributes whose information was lost when all the variables were used in the DT construction.

5.2 Confounding values in decision trees

We found confounding values, for example hearing loss symptoms of BPV cases, when studying the Vertigo2 data [VJKP00]. In paper [III] we continued the work with CONVs. We used the Vertigo3 data, the Vertigo2 data enlarged with the 48 BPV and 40 VNE cases having CONVs, in order to model situations with CONVs. The decision trees were constructed using the group of 110 attributes [I] and the binary class labels. The new trees were compared with the corresponding original trees of paper [I] generated from the Vertigo1 data.

The true positive rate decreased for all the diseases, except sudden deafness, which had TPR of 56.7%, namely 4.3% more than in the case of the Vertigo1 data. For Menière's disease (TPR of 95.1%), traumatic vertigo (TPR of 87.5%), and vestibular schwannoma (TPR of 78.1%), the decrease in the true positive rate was less than 10%. The largest decreases of 26.8% and 17.7% were found for benign positional vertigo and vestibular neuritis, whose TPRs were 71.5% and 80.6% respectively. These large decreases can be explained by the confounding

values, which made BPV and VNE cases resemble cases with Menière's disease. Overall, classification accuracies diminished slightly, varying from 91.7% to 98.1%. The reductions in the true positive rates and accuracies agreed with the results obtained by the linear discriminant analysis [Ken96a] and ONE.

The confounding values increased the sizes of trees. The number of attributes in the original trees varied from 3 to 15 (median 5.5) and in the new trees from 5 to 18 (median 10). The increase in the number of attributes varied from 2 to 15. This was largest in the case of BPV (from 3 to 18) and VNE (from 4 to 9). Again, sudden deafness was an exception with the attributes decreased from 7 to 5. New attributes occurring in the decision trees were sensible. New decision trees incorporated CONVs in a reasonable way into the reasoning process, especially the tree for BPV. They showed that otoneurological test results, being of minor value in the classification of these six diseases on the basis of our previous studies [Ken96a, KVPJ00], are important when cases with confounding values are dealt with.

The confounding values naturally make the classification tasks more difficult. Despite the decreased true positive rates and accuracies, the decision trees constructed are valuable. They simulate better the real life situation with patients having confounding signs and symptoms, and reveal attributes that are important for the classification of cases with CONVs.

The results of paper [III] show that it is possible to model situations with confounding values when enough data are availabe. The inclusion of cases with confounding values is essential when real world classifiers are constructed.

5.3 Refinement of the ONE knowledge base

In this section, we first present the inference results obtained by the original version of ONE. Then, the knowledge base augmented with the knowledge learned from data is considered.

Table 5.2: True positive rates (%) calculated on the basis of the best fit suggested by ONE.

Diagnosis	Vertigo1	Vertigo2	Vertigo3	Vertigo4
Benign positional vertigo	74.6	61.3	45.5	43.8
Menière's disease	70.8	67.1	67.1	67.1
Sudden deafness	57.1	53.3	53.3	58.5
Traumatic vertigo	92.5	92.9	92.9	90.8
Vestibular neuritis	36.7	35.4	29.5	32.5
Vestibular schwannoma	49.2	49.2	49.2	50.0

5.3.1 Inference results of original ONE

The inference results obtained using the original version of ONE (that is, the scoring scheme with weight and fitness values defined by otoneurologists and the rules expressed as necessary attribute values) for the four version of the Vertigo data are presented in Tables 5.2 and 5.3. In Table 5.2, the true positive rates (TPRs) with respect to the best fit suggested by ONE are shown. For the Vertigo1 data, the TPRs varied from 36.7% (vestibular neuritis) to 92.5% (traumatic vertigo). The addition of new cases (from Vertigo2 to Vertigo4) caused a dramatic decrease in the TPR for benign positional vertigo and a slight decrease for Menière's disease, traumatic vertigo, and vestibular neuritis.

When the three best fits obtained by ONE were examined (Table 5.3), the true positive rates for the Vertigo1 data varied from 63.3% (vestibular neuritis) to 100.0% (sudden deafness and traumatic vertigo). The inclusion of the second- and third-best fits notably increased the true positive rates for all the diagnoses except benign positional vertigo and traumatic vertigo. Enlarging the data set decreased the TPRs, especially for BPV, SUD and VNE.

Our studies [VJKP00,III] both on confounding values and decision trees led us to examine how confounding values and necessary attribute values (NAVs) influence the inference of ONE. Confounding values usually relate to NAVs: There

Table 5.3: True positive rates (%) calculated on the basis of the three best fits suggested by ONE.

Diagnosis	Vertigo1	Vertigo2	Vertigo3	Vertigo4
Benign positional vertigo	79.7	65.3	48.8	46.6
Menière's disease	98.8	96.8	96.8	95.5
Sudden deafness	100.0	83.3	83.3	82.9
Traumatic vertigo	100.0	100.0	100.0	100.0
Vestibular neuritis	63.3	60.0	51.4	51.7
Vestibular schwannoma	95.3	95.3	95.3	94.5

may be an attribute value (a sign or a symptom) caused by some other factor than the current disease. This attribute value does not fit the current disease and, thus, the correct diagnosis is rejected by ONE. An example of this is a BPV patient with hearing loss symptoms. Hearing loss is not part of the clinical picture of BPV, but there are BPV patients who have these symptoms due, for instance, to ageing or noise injury [Ken96a].

In Table 5.4, misclassification results for the Vertigo4 data calculated on the basis of the best fit suggested by ONE are shown. For each disease, the following data are presented:

- the number of cases misclassified
- the number of cases whose correct diagnosis was rejected because of NAVs
- the number of cases that would have been classified correctly if the pure best score without NAVs had been used.

For particularly large numbers of BPV and VNE cases (63 and 53 respectively), the correct diagnosis was rejected due to the necessary attribute values. The use of the pure score improved the situation slightly in the case of BPV.

In Table 5.5, the respective numbers are shown for the three best fits suggested by ONE and for the three best scores. The use of the pure score without NAVs

Table 5.4: Misclassification results for the Vertigo 4 data calculated on the basis of the best fit obtained by ONE.

Diagnosis	Incorrect diagnosis (N_i)	Correct diagnosis rejected due to NAVs (N_r)	Correct diagnosis on the basis of the best score (N_s)
BPV	82	63	3
MEN	103	13	3
SUD	17	7	1
TRA	6	0	0
VNE	81	53	0
VSC	65	5	1

was advantageous, especially for BPV and VNE. In the case of BPV, 22 earlier misclassified cases were correctly re-classified in terms of the three best scores. For vestibular neuritis, the respective number was 30. These numbers correspond to the percentages of 15.1 and 25.0 respectively of the total number of cases in these diagnostic groups.

The above results led us to abandon the use of NAVs and to examine the possibilities of learning the fitness values from data in paper [VI] whose results are presented in the following section.

5.3.2 Learning fitness values from data

In [VI], the effects of weight and fitness values on the scores calculated by ONE were examined. Four different scoring systems formed on the basis of knowledge acquired from experts and the fitness values 'learned' from data were used. (One may wonder whether such a simple operation as calculating the fitness values on the basis of relative frequencies is learning, even though it results in better performance. Our future aim will be to find more sophisticated methods for extracting the fitness values from data.) Neither the upper and lower bounds of

Table 5.5: Misclassification results for the Vertigo4 data calculated on the basis of the three best fits obtained by ONE.

Diagnosis	Incorrect diagnosis (N_i)	Correct diagnosis rejected due to NAVs (N_r)	Correct diagnosis on the basis of the three best scores (N_s)
BPV	78	63	22
MEN	14	13	8
SUD	7	7	3
TRA	0	0	0
VNE	58	53	30
VSC	6	5	2

the score nor the necessary attribute values were employed in these experiments.

The 815 cases in the Vertigo4 data were divided into training and test sets. From each diagnostic group, approximately 70% of the cases were randomly selected (within different data collection versions) for the training set. The remaining 30% of the cases formed the testing set. The fitness values were calculated from the training data within the six diagnostic groups. For each attribute i, the frequency distribution was calculated. The fitness value $f(d,i,j)$ was calculated as the proportion of the frequency $fr(d,i,j)$ of the value j to the highest frequency $fr(d,i,h)$ of the distribution:

$$f(d,i,j) = \frac{fr(d,i,j)}{fr(d,i,h)}.$$

The test cases were diagnosed using four different weight and fitness value systems:

- Experiment 1: The weights and fitness values defined by the otoneurologists
- Experiment 2: The weights defined by the otoneurologists and the fitness values learned from the training data
- Experiment 3: For the relevant attributes defined by the otoneurologists,

the weight values of 1 and the fitness values learned from the training data

- Experiment 4: All the 170 attributes were associated with the weight 1, except irrelevant ones (with respect to the classification task) and those whose values were missing from all the training cases of the diagnostic group. The fitness values were learned from the training data.

The use of the fitness values learned from data produced the best true positive rate for benign positional vertigo (65.9%), Menière's disease (94.7%), sudden deafness (91.7%), traumatic vertigo (100.0%), and vestibular schwannoma (66.7%). The increase in the TPR was significant in the case MEN and VSC ($p = 0.000$ and $p = 0.002$ in the chi-square test respectively). (We use here the chi-square test as it is used as a goodness-of-fit test in model checking [Agr96] in order to examine whether the distributions (on classification: correct or not) produced by different scoring schemes differ.) For these two diseases, the increase was over 30%. Vestibular neuritis was an exception with the best TPR of 81.1% produced by the weight and fitness values defined by the experts.

In the case of sudden deafness, the relevant attributes defined by the otoneurologists were important. This result coincides with the results of paper [I]: Increasing the number of attributes used in the decision tree construction produced lower true positive rates for sudden deafness. The magnitude of weights was not important in the case of sudden deafness, whereas the true positive rate for traumatic vertigo decreased significantly from 100.0% to 25.0% when the weights defined by the experts were set at 1. This decrease can be explained by the attribute head trauma with a weight of 200 in the scoring scheme 2. However, the use of the largest attribute group with weights of 1 in the Experiment 4 produced a good true positive rate of 80.0% for traumatic vertigo. For Menière's disease and vestibular schwannoma, the scoring scheme 4 with the largest attribute group yielded the best results, which agrees with the results of our studies with decision trees [I, VJKP00, III].

The accuracies for scoring schemes 1, 2, 3, and 4 were 62.6%, 69.9%, 64.2% and 79.7% respectively. These results show that learning fitness values from data is

Table 5.6: True positive rates (%) obtained by ONE (ONE_{170} and ONE_{DT}), decision tree induction (DT_{170}), and nearest neighbour classification with HVDM ($1NN_{170}$) for the test set in paper [VI].

Diagnosis	N	ONE_{170}	DT_{170}	$1NN_{170}$	ONE_{DT}
Benign positional vertigo	44	65.9	70.5	68.2	54.6
Menière's disease	94	94.7	93.6	92.6	83.0
Sudden deafness	12	66.7	58.3	25.0	83.3
Traumatic vertigo	20	80.0	75.0	70.0	75.0
Vestibular neuritis	37	75.7	81.1	91.9	75.7
Vestibular schwannoma	39	66.7	66.7	76.9	61.5

useful in producing a knowledge base that performs better in real world situations with CONVs.

5.3.3 Decision trees in filtering attributes

In all our experiments concerning decision trees and the vertigo data, the number of attributes incorporated in the DTs was notably smaller than in the disease patterns of ONE defined by human experts. We were interested in whether decision trees could be employed as attribute filters for the inference mechanism of ONE. (In the following, we use the abbreviations of ONE_{170}, DT_{170}, and $1NN_{170}$ for ONE, decision tree induction, and nearest neighbour classification with HVDM all employing 170 attributes. The abbreviation ONE_{DT} is used for ONE employing attributes filtered by decision trees (DT_{170})).

We constructed decision trees for the six diseases from the training set of paper [VI] and tested them with the corresponding test set. Depending on the disease in question, the TPRs were in favour of ONE_{170} or the decision tree DT_{170} (Table 5.6), but these differences were not significant. Thus, we defined a scoring scheme for each disease with the weight 1 for the attributes of the corresponding DT and with the weight 0 for other attributes. The number of attributes in the schemes

ranged from 6 to 19. The results obtained by ONE_{DT} are presented in Table 5.6.

Regarding accuracy, no significant difference was found: ONE_{170} classified 79.7% of the cases correctly and ONE_{DT} 72.8%. For sudden deafness, the attributes filtered by the DT yielded a better TPR, but the difference was not significant. Both scoring schemes classified VNE cases equally well. For the remaining four diseases, the group of 170 attributes yielded better results. The difference was significant only in the case of Menière's disease ($p = 0.011$ in the chi-square test).

The only significant difference found for MEN suggests that DTI is not a suitable method for filtering attributes for the knowledge base of ONE. However, the accuracy did not decrease significantly and the patterns were notably simpler with respect to the number of attributes. Further, the increase in the TPR for sudden deafness was over 15%. These facts suggest that DTI could be used for attribute filtering and merits additional research. Choosing the weight values on the basis of locations of the attributes in the tree is also a possible topic for future research.

5.4 Six diseases as concepts to be learned

We used decision tree induction, the inference mechanism of ONE, and the nearest neighbour classification with HVDM [VTJP02] to classify the cases of the vertigo data. In general, the largest group of 170 attributes gave the best results. The accuracies in the test set of paper [VI] were 79.7%, 80.1%, and 80.5% for ONE_{170}, DT_{170}, and $1NN_{170}$ respectively. The corresponding true positive rates are presented in Table 5.6. Significant differences in accuracies or true positive rates were not found.

For sudden deafness, the same pattern with respect to the number of attributes and the learning results was found in the case of all three methods: the best results were obtained with an attribute set of medium size. The results obtained by 1NN [VTJP02] explain this phenomenon: when using the groups of 5, 38 and 170 attributes, the nearest case belonged to the same class for 16.7%, 50.0% and

25.0% of SUD cases respectively.

BPV, SUD and VSC were the most difficult disease to identify, and MEN the easiest one. In general, most of the misclassified cases resembled those with Menière's disease. For BPV, SUD, TRA and VSC, the numbers of cases (and the percentages within the diagnostic group) whose nearest case was MEN were 13 (29.5%), 8 (66.7%), 6 (30.0%), and 8 (20.5%) respectively when all the 170 attributes were used. These results agree with those obtained with decision trees. Accordingly, the identification of negative cases was most difficult in the case of MEN.

The similarity of results produced by the different methods is in accordance with a review of applications of rule-induction methods [LS95]. In the applications reviewed, much of the performance of the learned knowledge came from careful problem formulation as well as selection of attributes and examples rather than from specific induction methods [LS95]. Roughly equivalent performance of different methods on many domains has been shown in comparative studies [LS95]. Due to global optima that are easy to find or local optima that are nearly as good as the global ones, many ML methods may achieve reasonable performance [LS95].

We also used ANNs [JLV+99, JLV+00, JLV+01, JVL+01] to classify the vertigo data. ANNs seem to differ from DTI, ONE, and NN classifications. First, the number of training cases required by ANNs is notably larger than with the other methods. With the Vertigo1 and Vertigo2 data, we could use only the five key attributes. On a data set of 883 cases [JVL+01], including the Vertigo4 data and 68 cases representing other vertigo diseases, the number of attributes could be increased to nine. Second, ANNs are far more sensitive to imbalanced class distribution losing the small groups, such as SUD and TRA. Uniform distribution and a large enough number of training cases with respect to a network topology are well known conditions in the neural network literature for the successful application of ANNs, as explained in [Swi96].

On the Vertigo4 data, ANNs (feedforward perceptrons with the backpropaga-

tion algorithm and Kohonen networks [Koh95]) correctly classified approximately 90% of the cases with BPV, MEN and VNE [JVL+01]. TRP for VSC was 60-70%. In general, SUD and TRA cases were lost due to the small sizes of the groups. ANNs' capacity for recognising cases with BPV, MEN and VNE poses the question whether they could be used with the other three methods in order to improve classification. Unfortunately, the requirement for complete data complicates the utilisation of ANNs on the vertigo data.

Chapter 6

Discussion and conclusions

In this study, we examined the suitability of machine learning for refining and expanding the knowledge base of the otoneurological expert system ONE.

Decision tree induction proved to be a useful method for acquiring diagnostic knowledge for the six diseases. The decision trees constructed were intelligible and productive in gaining new information for diagnostic work [KVPJ00, III, V]. Further, they helped the computer scientists of the research group to get insight into the domain area and guided the refinement of the ONE knowledge base.

During the whole study, data pre-processing, especially the selection of attributes and examples, was an important issue. We found that the attributes selected by human experts generally yielded good classification results in the case of DTI as well as ONE. However, attributes classified by the experts to the category of the second most important or irrelevant, proved to be beneficial especially for the classification tasks concerning Menière's disease and vestibular schwannoma. In addition to the overall trend of increasing the number of attributes resulting in better classifiers, the need for feature subset selection also emerged. In the case of sudden deafness, the best results for DTI and ONE were obtained with groups of the 38 attributes and the 69 attributes respectively.

We examined FSS in the context of DTI by employing measures of association and the entropy-based approach to evaluate the quality of training data. Although

strong associations and low entropy values related to high accuracies and true positive rates, the relationship was not straightforward. Classes overlapping due to their nature or noise may hinder generalisation, even though attributes are adequate. The nearest neighbour method can be used to examine the locations of classes in the attribute space.

Our approach for FSS was the variable grouping method based on graph theoretic techniques and measures of association. The attribute subsets formed on the basis of the attribute groups worked well with the vertigo data, yielding simpler decision trees without reducing the classification results significantly. Moreover, it was useful as such, giving insight into the data. However, the method should be tested with other data sets and learning methods.

With respect to the selection of examples, we showed that it is possible to model situations with confounding values, if enough data are available. The results of paper [III] showed the importance of otoneurological test results that were of minor value in previous studies [Ken96a, KVPJ00]. The inclusion of cases with confounding values is essential if we want to construct classifiers that perform well in real world situations.

In the refinement of the ONE knowledge base, learning from data produced a better performing knowledge base for real world situations with confounding values. The fitness values based on the frequency distributions yielded better true positive rates for all diseases except vestibular neuritis. The increases in the TPRs were significant for MEN and VSC (over 30%). The largest attribute set yielded the best results for all the diseases except sudden deafness. We explored the possibility of using decision trees as attribute filters for the inference mechanism of ONE. Despite the decreased classification results, the attributes filtered with decision trees do merit additional research. Similarly, weight values based on the locations of the attributes in the decision trees is a possible topic for future research.

On the basis of our experiments with ONE_{170}, DT_{170}, and $1NN_{170}$, neither single learning method nor single attribute set can be ranked best. Considering

the respective accuracies of 79.7%, 80.1% and 80.5%, as well as the respective true positive rates varying from 65.9% to 94.7%, 58.3% to 93.6% and 25.0% to 92.6%, the performance of these methods can be characterised as reasonable. A hybrid system combining multiple classifiers by using a voting method or a more sophisticated one, is one of our future aims. Further, DTI and NN methods are useful additions to ONE as stand-alone methods, too. DTs help physicians to model the problem domain from different viewpoints and NN classification finds similar cases for consideration.

To describe the difficulty of diagnosing the six diseases, we refer to the study [KAJP98], in which ONE (the original version with the knowledge defined by physicians) was compared to six physicians. Five of the physicians were residents in otolaryngology and one was a senior physician working in the otolaryngology clinic. Of the 23 test patients, 17 had one of the six diseases in the present study. The physicians diagnosed correctly from 52.9% to 82.4% of the 17 cases (median 67.7%), when they were allowed to use the patients' entire medical records. ONE's classification accuracy was 70.6%. It has to be noted that the overall low percentage of correctly solved cases is due to the patient selection for the vestibular unit: typical cases, for example, of BPV and VNE are seldom evaluated in the unit but rather the cases that have also caused diagnostic difficulties for the referring physician [KAJP98].

Considering the difficulty of the domain, the refined and enlarged system seems to perform reasonably. In order to evaluate the practical clinical value of ONE with its refined knowledge base, it must be tested in clinical work. Thus, the next phase in the development of the system will be test use in the otolaryngology clinic of the Helsinki University Central Hospital.

Extending the system with machine learning capabilities is also a matter for future work. The learning component is essential for the maintenance of the knowledge. New cases occur daily and, thus, data collection is a continuous process. So far, we have enough data for ML purposes only for the six diseases discussed in this study. The disease patterns of rare diseases are still manually

crafted. Further, learning capabilities will doubtlessly be needed if the system is to be used in other otolaryngology clinics because of dissimilar patient populations and possible differences in the diagnostic process. The possibilities for integrating ONE with other health information systems in order to transfer data is also a topic for future research.

Menière's disease is already quite reliably identified. With respect to the other diseases, most difficulties in identification seem to be caused by the cases mimicking MEN. A thorough analysis of these cases may give some additional information about the possibilities for improving their identification. We are still looking for a good attribute set for sudden deafness. Nearest neighbour classification combined with a wrapper method is a possible choice for this task. The use of new derived variables, for example patterns in audiograms or hearing loss caused by noise exposure [KAJP98], might also be beneficial.

To summarise, the use of machine learning methods does not fully automate the process of knowledge acquision, but it replaces time-consuming and tedious manual knowledge extraction with somewhat easier tasks. Essential parts of applying ML are transforming the problem into a form suitable for the chosen ML method(s) and defining a good representation for training data. Domain experts have an important role in these tasks, and also in the evaluation of knowledge. The KA process is still iterative and adaptive, including crafting problem formulation and representation on the basis of the communication between system developers and experts. However, the amount of time and effort needed to acquire knowledge is notably smaller than in the manual or model-based approach, and the acquired knowledge may produce better results.

Bibliography

[Agr96] A. Agresti. *An Introduction to Categorical Data Analysis.* Wiley, New York, 1996.

[AJ95] Y. Auramo and M. Juhola. Comparison of inference results of two otoneurological expert systems. *International Journal of Bio-Medical Computing*, 39:327–335, 1995.

[AJ96] Y. Auramo and M. Juhola. Modifying an expert system construction to pattern recognition solution. *Artificial Intelligence in Medicine*, 8:15–21, 1996.

[AJP93] Y. Auramo, M. Juhola, and I. Pyykkö. An expert system for the computer-aided diagnosis of dizziness and vertigo. *Medical Informatics*, 18:293–305, 1993.

[AKA91] D.W. Aha, D. Kibler, and M.K. Albert. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.

[All94] B.P. Allen. Case-based reasoning: Business applications. *Communications of the ACM*, 37:40–42, 1994.

[BA97] L.A. Breslow and D.W. Aha. Simplifying decision trees: A survey. *The Knowledge Engineering Review*, 12:1–40, 1997.

[BFOS84] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees.* Wadsworth, Belmont, California, 1984.

[BKP+00] Š.H. Babič, P. Kokol, V. Podgorelec, M. Zorman, M. Šprogar, and M.M. Štiglic. The art of building decision trees. *Journal of Medical Systems*, 24:43–52, 2000.

[BL87] V. Barnett and T. Lewis. *Outliers in Statistical Data*. Wiley, Norwich, 1987.

[BL97] A. L. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97:245–271, 1997.

[BN92] W. Buntine and T. Niblett. A further comparison of splitting rules for decision-tree induction. *Machine Learning*, 8:75–85, 1992.

[Boe81] B.W. Boehm. *Software Engineering Economics*. Prentice-Hall, Englewood Cliffs, New Jersey, 1981.

[Bro95] C.E. Brodley. Automatic selection of split criterion during tree growing based on node location. In A. Prieditis and S. Russell, editors, *Machine Learning: Proceedings of the Twelfth International Conference*, pages 73–80. Morgan Kaufmann, San Francisco, California, 1995.

[BU95] C.E. Brodley and P.E. Utgoff. Multivariate decision trees. *Machine Learning*, 19:45–77, 1995.

[CMM83] J.G. Carbonell, R.S. Michalski, and T.M. Mitchell. An overview of machine learning. In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach*, pages 3–24. Morgan Kaufmann, Los Altos, California, 1983.

[DKS95] J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous features. In A. Prieditis and S. Russell, editors, *Machine Learning: Proceedings of the Twelfth International Conference*, pages 194–202. Morgan Kaufmann, San Francisco, California, 1995.

[DL97] M. Dash and H. Liu. Feature selection for classification. *Intelligent Data Analysis*, 1:131–156, 1997.

[DOMK+01] S. Dreiseitl, L. Ohno-Machado, H. Kittler, S. Vinterbo, H. Billhardt, and M. Binder. A comparison of machine learning methods for the diagnosis of pigmented skin lesions. *Journal of Biomedical Informatics*, 34:28–36, 2001.

[DVDBZ99] W. Daelemans, A. Van Den Bosch, and J. Zavrel. Forgetting exceptions is harmful in language learning. *Machine Learning*, 34:11–41, 1999.

[Eve79] S. Even. *Graph Algorithms*. Pitman, London, 1979.

[Fei79] E. A. Feigenbaum. Themes and case studies of knowledge engineering. In D. Michie, editor, *Expert Systems in the Micro-Electronic Age*, pages 3–25. Edinburgh University Press, Edinburgh, 1979.

[FNI91] J. Forsström, P. Nuutila, and K. Irjala. Using the ID3 algorithm to find discrepant diagnoses from laboratory databases of thyroid patients. *Medical Decision Making*, 11:171–175, 1991.

[FPSS96] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. The KDD process for extracting useful knowledge from volumes of data. *Communications of the ACM*, 39:27–34, 1996.

[GD93] A. J. Gonzalez and D. D. Dankel. *The Engineering of Knowledge-Based Systems: Theory and Practice*. Prentice-Hall, Englewood Cliffs, New Jersey, 1993.

[HMS66] E. B. Hunt, J. Marin, and P. J. Stone. *Experiments in Induction*. Academic Press, New York, 1966.

[Hon96a] P. Honkavaara. Effect of ondansetron on nausea and vomiting after middle ear surgery during general anaesthesia. *British Journal of Anaesthesia*, 76:316–318, 1996.

[Hon96b] P. Honkavaara. Effect of transdermal hyoscine on nausea and vomiting during and after middle ear surgery under local anaesthesia. *British Journal of Anaesthesia*, 76:49–53, 1996.

[HS98] P. Honkavaara and L. Saarnivaara. Comparison of subhypnotic doses of thiopentone vs. propofol on the incidence of postoperative nausea and vomiting following middle ear surgery. *Acta Anaesthesiologica Scandinavica*, 42:211–215, 1998.

[HSK94] P. Honkavaara, L. Saarnivaara, and U.-M. Klemola. Prevention of nausea and vomiting with transdermal hyoscine in adults after middle ear surgery during general anaesthesia. *British Journal of Anaesthesia*, 73:763–766, 1994.

[HSK95] P. Honkavaara, L. Saarnivaara, and U.-M. Klemola. Effect of transdermal hyoscine on nausea and vomiting after surgical correction of prominent ears under general anaesthesia. *British Journal of Anaesthesia*, 74:647–650, 1995.

[HSO99] M. Haruno, S. Shirai, and Y. Ooyama. Using decision trees to construct a practical parser. *Machine Learning*, 34:131–149, 1999.

[JLV+99] M. Juhola, J. Laurikkala, K. Viikki, Y. Auramo, E. Kentala, and I. Pyykkö. Neural network recognition of otoneurological vertigo diseases with comparison of some other classification methods. In W. Horn, Y. Shahar, G. Lindberg, S. Andreassen, and J. Wyatt, editors, *Artificial Intelligence in Medicine*, volume 1620 of *Lecture Notes in Artificial Intelligence*, pages 217–226. Springer, Berlin, 1999.

[JLV+00] M. Juhola, J. Laurikkala, K. Viikki, E. Kentala, and I. Pyykkö. Neural network classification of otoneurological vertigo diseases. In *CD-ROM: MIE2000/GMDS2000: Proceedings - Workshops - Posters*. Quintessenz Verlag, Berlin, 2000.

[JLV+01] M. Juhola, J. Laurikkala, K. Viikki, E. Kentala, and I. Pyykkö. Classification of patients on the basis of otoneurological data by using Kohonen networks. *Acta Otolaryngology, Supplement*, 545:50–52, 2001.

[JVL+01] M. Juhola, K. Viikki, J. Laurikkala, I. Pyykkö, and E. Kentala. On classification capability of neural networks: A case study with otoneurological data. In V. Patel, editor, *MEDINFO 2001*, pages 474–478. IOS Press, Amsterdam, 2001.

[KAJP98] E. Kentala, Y. Auramo, M. Juhola, and I. Pyykkö. Comparison between diagnoses of human experts and a neurotologic expert system. *Annals of Otology, Rhinology & Laryngology*, 107:135–140, 1998.

[KBR84] I. Kononenko, I. Bratko, and E. Roskar. *Experiments in automatic learning of medical diagnostic rules.* Jozef Stefan Institute, Ljubljana, 1984. Technical Report.

[Ken96a] E. Kentala. Characteristics of six otologic diseases involving vertigo. *The American Journal of Otology*, 17:883–892, 1996.

[Ken96b] E. Kentala. *A neurotologic expert system for vertigo and characteristics of six otologic diseases involving vertigo.* Academic Dissertation, Department of Otorhinolaryngology, University of Helsinki, 1996.

[KHM98] M. Kubat, R. C. Holte, and S. Matwin. Machine learning for the detection of oil spills in satellite radar images. *Machine Learning*, 30:195–215, 1998.

[KJ97] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97:273–324, 1997.

[KLPJ99] E. Kentala, J. Laurikkala, I. Pyykkö, and M. Juhola. Discovering diagnostic rules from a neurotologic database with genetic algorithms. *Annals of Otology, Rhinology & Laryngology*, 108:948–954, 1999.

[KM96] D. Kalles and T. Morris. Efficient incremental induction of decision trees. *Machine Learning*, 24:231–242, 1996.

[KM97] M. Kubat and S. Matwin. Addressing the curse of imbalanced training sets: One-sided selection. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 179–186. Morgan Kaufmann, San Francisco, California, 1997.

[Koh95] T. Kohonen. *Self-Organizing Maps*. Springer, Berlin, 1995.

[KPAJ95] E. Kentala, I. Pyykkö, Y. Auramo, and M. Juhola. Database for vertigo. *Otolaryngology, Head and Neck Surgery*, 112:383–390, 1995.

[KPAJ96] E. Kentala, I. Pyykkö, Y. Auramo, and M. Juhola. Otoneurological expert system. *Annals of Otology, Rhinology & Laryngology*, 105:654–658, 1996.

[KPY+01] P. Kokol, V. Podgorelec, R. Yamamoto, G. Masuda, and N. Sakamoto. Medical knowledge extraction via hybrid decision trees. In H. Kangassalo, T. Welzer, H. Jaakkola, and I. Rozman, editors, *Proceedings of the Eleventh European-Japanese Conference on Information Modelling and Knowledge Bases*, pages 378–385. Faculty of Electrical Engineering and Computer Science, Maribor, 2001.

[KVPJ00] E. Kentala, K. Viikki, I. Pyykkö, and M. Juhola. Production of diagnostic rules from a neurotologic database with decision trees. *Annals of Otology, Rhinology & Laryngology*, 109:170–176, 2000.

[Lav99] N. Lavrač. Selected techniques for data mining in medicine. *Artificial Intelligence in Medicine*, 16:3–23, 1999.

[LBFL80] R. Lindsay, B. Buchanan, E. Feigenbaum, and J. Lederberg. *Applications of Artificial Intelligence for Chemical Inference: The Dendral Project*. McGraw-Hill, New York, 1980.

[LJ98] J. Laurikkala and M. Juhola. A genetic-based machine learning system to discover the diagnostic rules for female urinary incontinence. *Computer Methods and Programs in Biomedicine*, 55:217–228, 1998.

[LS95] P. Langley and H. A. Simon. Applications of machine learning and rule induction. *Communications of the ACM*, 38:55–64, 1995.

[Mer02] Merriam-Webster, http://www.m-w.com/, 2002.

[Mic87] D. Michie. Current developments in expert systems. In J. R. Quinlan, editor, *Applications of Expert Systems: Based on the Proceedings of the Second Australian Conference*, pages 137–156. Turing Institute Press, Glasgow, 1987.

[Min89a] J. Mingers. An empirical comparison of pruning methods for decision tree induction. *Machine Learning*, 4:227–243, 1989.

[Min89b] J. Mingers. An empirical comparison of selection measures for decision-tree induction. *Machine Learning*, 3:319–342, 1989.

[Mit97] T. M. Mitchell. *Machine Learning.* McGraw-Hill, New York, 1997.

[NCW91] P. Nykänen, S. Chowdhury, and O. Wigertz. Evaluation of decision support systems in medicine. *Computer Methods and Programs in Biomedicine*, 34:229–238, 1991.

[Nik97] C. Nikolopoulos. *Expert Systems: Introduction to First and Second Generation and Hybrid Knowledge Based Systems.* Marcel Dekker, New York, 1997.

[NS63] A. Newell and H.A. Simon. GPS, a program that simulates human thought. In E.A. Feigenbaum and J. Feldman, editors, *Computers and Thought*, pages 279–293. McGraw-Hill, New York, 1963.

[Nyk00] P. Nykänen. *Decision support systems from a health informatics perspective.* Academic Dissertation, Department of Computer and In-

formation Sciences, Research Report A-2000-10, University of Tampere, 2000.

[PEJ96] E. Pesonen, M. Eskelinen, and M. Juhola. Comparison of different neural network algorithms in the diagnosis of acute appendicitis. *International Journal of Bio-Medical Computing*, 40:227–233, 1996.

[Pet97] M. A. Pett. *Nonparametric Statistics for Health Care Research: Statistics for Small Samples and Unusual Distributions.* SAGE Publications, Thousand Oaks, 1997.

[Phu97] N.H. Phuong. *Towards Intelligent Systems for Integrated Western and Eastern Medicine.* The GIOI, Hanoi, 1997.

[PMLP97] I. A. Pilih, D. Mladenić, N. Lavrač, and T.S. Prevee. Using machine learning for outcome prediction of patients with severe head injury. In *Proceedings of the Tenth IEEE Symposium on Computer-Based Medical Systems*, pages 200–204. IEEE Computer Society, Los Alamitos, California, 1997.

[QCHL87] J. R. Quinlan, P. J. Compton, K. A. Horn, and L. Lazarus. Inductive knowledge acquisition: A case study. In J. R. Quinlan, editor, *Applications of Expert Systems: Based on the Proceedings of the Second Australian Conference*, pages 157–173. Turing Institute Press, Glasgow, 1987.

[Qui79] J. R. Quinlan. Discovering rules by induction from large collections of examples. In D. Michie, editor, *Expert Systems in the Micro–Electronic Age*, pages 168–201. Edinburgh University Press, Edinburgh, 1979.

[Qui83] J. R. Quinlan. Learning efficient classification procedures and their application to chess end games. In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors, *Machine Learning: An Artificial In-*

telligence Approach, pages 463–482. Morgan Kaufmann, Los Altos, California, 1983.

[Qui86] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[Qui88] J. R. Quinlan. Simplifying decision trees. In B.R. Gaines and J.H. Boose, editors, *Knowledge Acquisition for Knowledge-Based Systems*, pages 241–254. Academic Press, London, 1988.

[Qui90] J. R. Quinlan. Decision trees and decisionmaking. *IEEE Transactions on Systems, Man, and Cybernetics*, 20:339–346, 1990.

[Qui93] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, California, 1993.

[Qui96] J. R. Quinlan. Learning decision tree classifiers. *ACM Computing Surveys*, 28:71–72, 1996.

[Qui97] J. R. Quinlan. See5, 1997. http::/www.rulequest.com/.

[RJH99] F. Rasmussen, M. Johansson, and H.O. Hansen. Trends in overweight and obesity among 18-year-old males in Sweden between 1971 and 1995. *Acta Paediatrica*, 88:431–437, 1999.

[SDFH98] S. Salzberg, A. L. Delcher, K. H. Fasman, and J. Henderson. A decision tree system for finding genes in DNA. *Journal of Computational Biology*, 5:667–680, 1998.

[Sha96] S. Sharma. *Applied Multivariate Techniques*. John Wiley & Sons, New York, 1996.

[Sho76] E. Shortliffe. *Computer Based Medical Consultations: MYCIN*. Elsevier, 1976.

[SW86] C. Stanfill and D. Waltz. Toward memory-based reasoning. *Communications of the ACM*, 29:1213–1228, 1986.

[Swi96] K. Swingler. *Applying Neural Networks: A Practical Guide.* Academic Press, London, 1996.

[Tur93] E. Turban. *Decision Support and Expert Systems: Management Support Systems.* Macmillan, New York, 1993.

[Utg89] P.E. Utgoff. Incremental induction of decision trees. *Machine Learning*, 4:161–186, 1989.

[VJKP00] K. Viikki, M. Juhola, E. Kentala, and I. Pyykkö. Building training data for decision tree induction in the subspecialty of otoneurology. In *CD-ROM: MIE2000/GMDS2000: Proceedings - Workshops - Posters.* Quintessenz Verlag, Berlin, 2000.

[VTJP02] K. Viikki, M. Tapani, M. Juhola, and I. Pyykkö. Nearest neighbour classification of otoneurological data. *MIE2002*, 2002. Accepted for publication.

[Wat86] D.A. Waterman. *A Guide to Expert Systems.* Addison-Wesley, Reading, Massachusetts, 1986.

[WCFCH01] W. Webber Chapman, M. Fizman, B.E. Chapman, and P.J. Haug. A comparison of classification algorithms to automatically identify chest X-ray reports that support pneumonia. *Journal of Biomedical Informatics*, 34:4–14, 2001.

[Wei85] S. Weisberg. *Applied Linear Regression.* John Wiley & Sons, New York, 1985.

[WM97] D.R. Wilson and T.R. Martinez. Improved heterogeneous distance functions. *Journal of Artificial Intelligence Research*, 6:1–34, 1997.

[WRL94] B. Widrow, D.E. Rumelhart, and M.A. Lehr. Neural networks: Applications in industry, business and science. *Communications of the ACM*, 37:93–105, 1994.

[WW00] D. West and V. West. Improving diagnostic accuracy using a hierarchical neural network to model decision subtasks. *International Journal of Medical Informatics*, 57:41–55, 2000.

[ZSB+97] B. Zupan, D. S. Stokié, M. Bohanee, M. M. Priebe, and A. M. Sherwood. Relating clinical and neurophysiological assessment of spasticity by machine learning. In *Proceedings of the Tenth IEEE Symposium on Computer-Based Medical Systems*, pages 190–194. IEEE Computer Society, Los Alamitos, California, 1997.

Appendix

Table A1. Diseases and disorders covered by the expert system ONE.

Diagnosis
Autoimmune disease
Benign paroxysmal vertigo of childhood
Benign positional vertigo
Benign recurrent vertigo
Borreliosis
Brain stem ischaemia
Chlamydiosis
CNS tumour
Epilepsy
Lues
Menière's disease
Neurovascular compression
Ototoxicity
Perilymphatic fistula
Sudden deafness
Traumatic vertigo
Vestibular neuritis
Vestibular schwannoma

Table A2. Attributes in the database of ONE.

	Attribute
1	Patient's social security number
2	Patient's name
3	Preliminary diagnosis
4	Referring physician's name
5	Referring physician's clinic
6	Referring physician's address
7	Referring physician's telephone number
8	Investigator
9	Creation date
10	Modification date
11	ONE's 1. diagnosis
12	ONE's 2. diagnosis
13	ONE's 3. diagnosis

Table A3. Attributes in the database of ONE. Number of missing values in the Vertigo4 data.

	Attribute	Missing values	
		N	%
14	Vertigo	0	0.0
15	Age at onset of vertigo symptoms	14	1.7
16	Duration of vertigo symptoms	12	1.5
17	Frequency of vertigo attacks	51	6.3
18	Duration of vertigo attack	60	7.4
19	Intensity of vertigo attack	24	2.9
20	Score for rotational vertigo	70	8.6
21	Score for floating sensation	144	17.7
22	Severity of nausea	12	1.5
23	Tumarkin-type drop attacks	17	2.1
24	Score for position-induced vertigo	33	4.0
25	Score for visually induced vertigo	34	4.2
26	Score for pressure-induced vertigo	29	3.6
27	Score for physical activity-induced vertigo	35	4.3
28	Instability or gait difficulties	10	1.2
29	Unsteadiness outside vertigo attacks	78	9.6
30	Hearing loss	3	0.4
31	Duration of hearing loss	44	5.4
32	Hearing loss of the right ear	45	5.5
33	Hearing loss of the left ear	44	5.4
34	Hearing loss during vertigo attacks	87	10.7
35	Fluctuation in hearing	221	27.1
36	Type of hearing loss	527	64.7
37	Tinnitus	3	0.4
38	Duration of tinnitus	52	6.4

(continued)

Table A3. Attributes in the database of ONE. Number of missing values in the Vertigo4 data.

	Attribute	Missing values	
		N	%
39	Severity of tinnitus	19	2.3
40	Site of tinnitus	38	4.7
41	Lightheadedness	6	0.7
42	Anxiety	58	7.1
43	Score for functional symptoms	41	5.0
44	Headache	36	4.4
45	Side of headache	775	95.1
46	Site of headache	775	95.1
47	Duration of headache	775	95.1
48	Frequency of headache	114	14.0
49	Tension neck-induced headache	178	21.8
50	Migraine	64	7.9
51	Headache during vertigo attacks	38	4.7
52	Neurological signs	31	3.8
53	Fainting	61	7.5
54	Visual blurring or double vision	87	10.7
55	Dysarthria	61	7.5
56	Cranial nerve palsy	72	8.8
57	Paresthesias or sensitivity disturbances	53	6.5
58	Use of alcohol	110	13.5
59	Use of tobacco	115	14.1
60	Use of ototoxic drugs	11	1.3
61	Use of diuretics	22	2.7
62	Use of aminoglycocides	29	3.6
63	Use of pain killers (salicylates)	35	4.3

(continued)

Table A3. Attributes in the database of ONE. Number of missing values in the Vertigo4 data.

	Attribute	Missing values	
		N	%
64	Use of painkillers (NSAD)	62	7.6
65	Use of cytostatics	29	3.6
66	Use of other vestibulotoxic drugs	29	3.6
67	Use of tricyclic antidepressives	47	5.8
68	Use of chlorpromazine	49	6.0
69	Use of barbiturates	49	6.0
70	Use of diazepam	63	7.7
71	Head or ear injury related on onset of symptoms	23	2.8
72	Brain concussion	33	4.0
73	Brain contusion	32	3.9
74	Whiplash injury	32	3.9
75	Trauma of head	35	4.3
76	Noise or pressure injury	33	4.0
77	Trauma of ear	56	6.9
78	Long-term noise exposure	55	6.7
79	Ear infections	39	4.8
80	History of ear discharge	68	8.3
81	Tympanic membrane perforation	67	8.2
82	Present ear discharge	67	8.2
83	Cholesteatoma	77	9.4
84	Ear operations	139	17.1
85	Myringoplasty	170	20.9
86	Tympanoplasty	169	20.7
87	Mastoidectomy	168	20.6
88	Radical ear operation	167	20.5

(continued)

Table A3. Attributes in the database of ONE. Number of missing values in the Vertigo4 data.

	Attribute	Missing values	
		N	%
89	Stapedectomy	168	20.6
90	Endolymphatic sac	178	21.8
91	Other ear surgery	168	20.6
92	General illnesses	27	3.3
93	Coronary heart disease	51	6.3
94	Hypertension	42	5.2
95	Arteriosclerosis	66	8.1
96	Brain ischaemia	88	10.8
97	Kidney insufficiency	48	5.9
98	Diabetes mellitus	44	5.4
99	Thyroid problems	67	8.2
100	Specific infections	806	98.9
101	Bacterial labyrinthitis	596	73.1
102	Bacterial meningitis	524	64.3
103	Viral meningoencephalitis	493	60.5
104	Borrelia	589	72.3
105	Chlamydia	815	100.0
106	Spumaretrovirus antibodies	815	100.0
107	Lues serology	815	100.0
108	Autoimmune assay	813	99.8
109	Constant vertigo after infection	815	100.0
110	Head position-induced vertigo after infection	815	100.0
111	Postural instability after infection	815	100.0
112	Gait disorders after infection	815	100.0
113	Hearing loss after infection	815	100.0

(continued)

Table A3. Attributes in the database of ONE. Number of missing values in the Vertigo4 data.

	Attribute	Missing values	
		N	%
114	Tinnitus after infection	815	100.0
115	Spontaneous nystagmus	246	30.2
116	Head-shaking nystagmus	251	30.8
117	Finger-nose test	252	30.9
118	Diadochokinesis	291	35.7
119	SEM latency right	169	20.7
120	SEM latency left	185	22.7
121	SEM accuracy right	178	21.8
122	SEM accuracy left	177	21.7
123	SEM peak velocity right	170	20.9
124	SEM peak velocity left	187	22.9
125	Posturography eyes open	202	24.8
126	Posturography eyes closed	203	24.9
127	ENG Spontaneous nystagmus	104	12.8
128	ENG Caloric asymmetry	74	9.1
129	ENG Response with 44°C right	128	15.7
130	ENG Response with 44°C left	134	16.4
131	PEM gain amplitude	193	23.7
132	PEM gain latency	225	27.6
133	Tone burst audiometry at 500 Hz right	22	2.7
134	Tone burst audiometry at 1 KHz right	23	2.8
135	Tone burst audiometry at 2 KHz right	22	2.7
136	Tone burst audiometry at 3 KHz right	798	97.9
137	Tone burst audiometry at 4 KHz right	293	36.0
138	Tone burst audiometry at 6 KHz right	753	92.4

(continued)

Table A3. Attributes in the database of ONE. Number of missing values in the Vertigo4 data.

	Attribute	Missing values	
		N	%
139	Tone burst audiometry at 8 KHz right	293	36.0
140	Tone burst audiometry at 500 Hz left	25	3.1
141	Tone burst audiometry at 1 KHz left	22	2.7
142	Tone burst audiometry at 2 KHz left	22	2.7
143	Tone burst audiometry at 3 KHz left	805	98.8
144	Tone burst audiometry at 4 KHz left	292	35.8
145	Tone burst audiometry at 6 KHz left	708	86.9
146	Tone burst audiometry at 8 KHz left	293	36.0
147	Speech perception audiometry	644	79.0
148	Decibel level of 50% speech discrimination right	658	80.7
149	Decibel level of 50% speech discrimination left	657	80.6
150	Percentage of maximum speech discrimination right	656	80.5
151	Percentage of maximum speech discrimination left	654	80.2
152	ECoG summation potential	811	99.5
153	ECoG action potential	811	99.5
154	Specific examinations	652	80.0
155	Computerised tomography (CT)	557	68.3
156	Magnetic resonance imaging (MRI)	815	100.0
157	Size of tumour	703	86.3
158	Site of tumour	702	86.1
159	Neurovascular compression	709	87.0
160	Brain atrophy	708	86.9
161	Brain infarction or hypodensity	709	87.0
162	CNS infection or MS	708	86.9
163	Other lesions in CT or MRI	717	88.0

(continued)

Table A3. Attributes in the database of ONE. Number of missing values in the Vertigo4 data.

	Attribute	Missing values	
		N	%
164	Fistula test	815	100.0
165	Brainstem response audiometry (BRA)	771	94.6
166	Unidentifiable waves in BRA	771	94.6
167	BRA I-III latency difference $>$ 0.2 ms	775	95.1
168	BRA Latency of brain wave right	780	95.7
169	BRA Latency of brain wave left	780	95.7
170	BRA Amplitude difference of the II waveform $>$ 33%	776	95.2

Table A4. Derived attributes used in the inference of ONE.

	Attribute
1	True vertigo
2	Concussion or contusion
3	Serious trauma of head
4	Symptoms began contemporaneously
5	Bilateral hearing loss or tinnitus
6	Unilateral hearing loss or tinnitus
7	Normal hearing and no tinnitus
8	Normal hearing
9	Tumour in acoustic nerve
10	Tumour elsewhere in CNS
11	Ischaemia in CT or MRI

Table A5. Number of attributes in the patterns for the six diseases.

	Number of attributes		
Diagnosis	Total	Derived	With NAVs
Benign positional vertigo	68	2	16
Menière's disease	78	2	3
Sudden deafness	69	4	5
Traumatic vertigo	72	6	2
Vestibular neuritis	72	4	11
Vestibular schwannoma	63	3	4

Table A6. Weights of relevant attributes in the patterns for the six diseases.

Diagnosis	Minimum	Maximum	Mode	Median	Mean
Benign positional vertigo	1	5	4	3.5	3.4
Menière's disease	1	5	4	3.5	3.4
Sudden deafness	2	40	4	4	4.2
Traumatic vertigo	1	200	1	3	5.6
Vestibular neuritis	1	5	4	4	3.4
Vestibular schwannoma	2	200	3	4	6.8