Bayesian belief networks for dementia diagnosis and other applications: a comparison of hand-crafting and construction using a novel data driven technique

A Thesis

submitted to the

University of Stirling

in partial fulfilment

for the Degree of

Doctor of Philosophy

by

Lloyd Oteniya

Department of Computing Science

University of Stirling

Stirling, FK9 4LA

Scotland

2008

Abstract

The Bayesian network (BN) formalism is a powerful representation for encoding domains characterised by uncertainty. However, before it can be used it must first be constructed, which is a major challenge for any real-life problem. There are two broad approaches, namely the hand-crafted approach, which relies on a human expert, and the data-driven approach, which relies on data. The former approach is useful, however issues such as human bias can introduce errors into the model. We have conducted a literature review of the expert-driven approach, and we have cherry-picked a number of common methods, and engineered a framework to assist non-BN experts with expert-driven construction of BNs. The latter construction approach uses algorithms to construct the model from a data set. However, construction from data is provably NP-hard [45]. To solve this problem, approximate, heuristic algorithms have been proposed; in particular, algorithms that assume an order between the nodes, therefore reducing the search space. However, traditionally, this approach relies on an expert providing the order among the variables — an expert may not always be available, or may be unable to provide the order. Nevertheless, if a good order is available, these order-based algorithms have demonstrated good performance. More recent approaches attempt to "learn" a good order then use the order-based algorithm to discover the structure. To eliminate the need for order information during construction, we propose a search in the entire space of Bayesian network structures — we present a novel approach for carrying out this task, and we demonstrate its performance against existing algorithms that search in the entire space and the space of orders.

Finally, we employ the hand-crafting framework to construct models for the task of diagnosis in a "real-life" medical domain, dementia diagnosis. We collect real dementia data from clinical practice, and we apply the data-driven algorithms developed to assess the concordance between the reference models developed by hand and the models derived from real clinical data.

Acknowledgements

This research has spanned more than four years, and it has been made possible due to the support of a great number of people in so many different ways. I would like to express my thanks to everyone. There are too many names to list here, however I want to take this opportunity to give special recognition to a number of key people.

I would like to begin by acknowledging the research team for their continued support, guidance and dedication to this research project. I am very grateful for the opportunity to have worked with an extremely talented and erudite team of academics and medical professionals. First, thanks to my supervisory team: Dr Julie Cowie, Principal Investigator, and Professor Leslie Smith. Thanks, Julie, for introducing me to health decision making and for giving me this once in a lifetime opportunity. Also, thanks for providing structure, guidance and direction. I would like to thank Professor Val Belton, Strathclyde University, for her valuable input and guidance, especially during the early stages of this research. I want to thank Dr Richard Coles, Community Mental Health Team Elderly (CMHTE), Kildean Hospital, Stirling, who provided a key medical consultancy role. In addition, I would like to thank the clinical staff at CMHTE for their generosity in diligently collecting data from clinical practice. My gratitude also goes to

Kate Howie for statistical consultancy, and Professor Ken Turner for his views, comments and useful ideas regarding this research.

The majority of this research was carried out within the University, however the final stages were carried out remotely as a part-time student. I would like to thank Sam Nelson for providing an outstanding computing supporting service, particularly outside standard working hours.

Of course, this research would never have started without funding. Thanks to the Engineering and Physical Science Research Council (EPSRC), Grant Number GR/S78148/01, for providing funding.

On a more personal note, there are a number of people who have made this thesis possible in other ways. I would like to thank all fellow PhD students: Gavin Campbell, Liam Docherty Paul Godley, and Thomas Wilson for their valuable advice and for making my time more enjoyable. Special thanks goes to Alan Brown for listening to my coding rants when things were just not going to plan. In addition, I would like to thank my line managers at Halifax Bank Of Scotland (HBOS) Plc, Allan Flint and Pete Stamper, who gave me the flexibility to work from home one day a week, which immensely cut down travelling time to maximise thesis writing time. Also, thanks to colleagues, particularly Yvonne Dickson and Louise McFaddyen, who proof-read chapters.

There are three “significant others” who I am forever indebted to: Mum, Lorraine, and dad, Tony — thank you for patience and support... and everything! And finally, a special thanks to a legendary best mate and virtual brother: Bryan “can’t touch this” Wright. Your continued understanding, encouragement, patience and friendship throughout this mammoth journey, particularly in intensive work periods, during good and challenging times, is eternally appreciated.

. . . for Mum and Dad
and Bryan

Contents

List of Figures

List of Tables

Part I

Introduction and background

Chapter 1

Introduction

1.1 Introduction

Bayesian networks (BN) are a powerful framework for representing and reasoning about problems characterised by uncertainty. They are composed of two parts: 1) a qualitative part, which is responsible for expressing complex probabilistic relationships among several random variables; and 2) a quantitative part, which encodes the probabilistic uncertainty between the relationships. One of the reasons that the framework has gained popularity in recent years is due to its intuitive comprehensibility and transparency; humans can easily interpret relationships between variables from the model structure, and the outcome of probabilistic inference is explained graphically.

Before the potential value of a BN can be realised, it must first be constructed. Traditionally, BNs are hand-crafted using human-supplied expert knowledge. However, extracting expert information from human experts is a complex process, and it is fraught with many challenges, particularly during elicitation of probabilities. One gap in current literature is a concise framework for novices seeking to develop BNs using expert knowledge, which guides them through the construction process and provides tools to assist with the elicitation task.

The alternative to the hand-crafted approach is to discover the BN structure and probability distributions from a data set — the so-called data-driven approach. This is particularly useful when there is no domain expert available to provide the structure or the probabilities, when the problem is too complex to be solved entirely by hand, or when domain expert time is limited. However, this approach is no "silver bullet". The primary drawback is that this construction approach is proven to be NP-hard [45]. Because of this, exact methods for BN construction are not realistic, and therefore approximate heuristic methods are in general more superior.

Bayesian networks are well suited to health decision support problems as they can be used for classification and to represent symptom-disease relationships. In addition, since BNs are built upon rigorous probability theory, they are capable of supporting probabilistic inference with incomplete information, thus providing health professionals with a platform to perform inference and reasoning.

This thesis is about BN construction using both the hand-crafted approach and the data-driven approach, with emphasis on building the structure, and, in addition, the application of BNs to dementia diagnosis.

1.2 Problem description

Bayesian networks (BN) gained their popularity due to their inherent ability to efficiently represent and reason about problems characterised by uncertainty. However, as mentioned in Section 1.1, the BN model must first be constructed either by hand using expert knowledge, or derived automatically from a data set of cases. Both approaches are complex and present significant challenges.

Under the hand-crafted approach, vital information must be elicited from the domain expert, such as the problem variables and their relations, and, more importantly, the probability distributions that characterise the uncertainty of the relations. There are many pieces of research literature on specific aspects of BN construction, such as probability elicitation from experts (see Chapter 4). However, there is a lack of joined-up documentation covering the overall construction process for real-life applications.

The data-driven approach is a popular alternative to the hand-crated approach; it relies less on human experts, as it attempts to construct the BN model automatically from data. However, this approach has its own challenges, primarily due to the computational complexity of discovering the model, and it assumes the availability of data.

A common approach to addressing the issue of complexity involves imposing an order among the variables. Such an order has the property that a node X_i can only have node X_j as a parent node if in the ordering node X_j comes before node X_i in the order, therefore reducing the number of potential networks structure to be evaluated. However, the question arises "where does the order come from?". In general, the answer is: "from a domain expert". As mentioned above, it is not always practical for an expert to provide all the required information (the entire order). Moreover, Singh [249] notes that the quality of the resulting network structure is very sensitive to the supplied order.

However, in situations where a "good" order is available, algorithms have been developed which find good BN structures, such as Cooper and Herskovits K2 algorithm [51]. In addition, algorithms have been developed that attempt to learn a “good” order first, then discover the BN structure from the order, such as [262, 126, 154, 232]. For example, Larrañaga et al. [154] employs a Genetic Algorithm (GA) to search for an optimal order, then enters the order as input to the K2 algorithm which provides a quantitative score of the quality of the order (described in more detail in Section 7.4.2.1).

Another set of data-driven BN construction algorithms are those that are not constrained by such an order. These algorithms can be prohibitively expensive, particularly those algorithms that operate on a single solution at a time. On the other hand, however, population-based stochastic search heuristics operate on a number of candidate solutions in parallel, and appear to be promising alternatives, as they are well suited to exploring massive, high-dimensional complex search spaces, which are representative of the BN discovery problem.

Nevertheless, once the BN model is constructed, it provides domain users, in this case health profressionals, with a useful tool to support their decision making process.

1.3 Research objectives

The goal of this thesis is to investigate and review approaches for BN construction, identify development opportunities and implement improved solutions, and in doing so:

- provide practical guidance on BN construction when only expert knowledge is available

- develop a new algorithm that addresses some of the issues with the existing data-driven approaches
- demonstrate how BNs can be applied to real-world problems, such as health decision making (e.g. dementia diagnosis), using the construction techniques presented

Bayesian network construction

- Investigate in detail the two approaches for BN construction.
- Create a practical framework for hand-crafting BNs in real-life domains, and illustrate the use of the framework in hand-crafting BNs for a real-life problem.
- Review existing data-driven approaches and investigate development opportunities that: 1) suppress the requirement to specify an order among the nodes up-front; and 2) search the space unrestricted by an order efficiently; and 3) do not rely on expensive validation and repair operators.
- Compare the performance of the new algorithm with new and existing algorithms that appear in the literature.
- Compare the construction approaches and the models derived by each approach.

These objectives are achieved through 1) reviewing the two approaches for BN construction, and developing a concise set of processes and tools, which are tested during the development of BN models for a real-life problem; 2) identification of issues with existing data-driven methods, and development of new algorithms to address some of the issues; 3) empirical evaluation of the performance of the new algorithm against existing approaches; and 4) finally, a comparison of the two construction approaches and the derived models.

Application in health decision support

- Investigate whether BNs can be constructed for dementia diagnosis, and demonstrate how they can be constructed through implementation of suitable diagnostic models.
- Investigate the performance of the diagnostic models:
  - Hand-crafted models - the classification accuracy with respect to real life clinical data.
  - Data-driven models - the structure of the models with respect to the reference, hand-crafted models.

These objectives are realised through hand-crafted construction of BN models for dementia diagnosis in clinical practice using only information supplied by a domain expert. Furthermore, using real-life clinical data, we demonstrate the classification accuracy of the models developed using only expert knowledge, and, in addition, illustrate the variations in the models derived from data in comparison to the expert-developed models.

1.4 Summary of contributions

The main contributions of this thesis are listed below.

Framework for hand-crafting BNs using expert knowledge Constructing BNs using information elicited from a domain expert is fraught with challenges. One issues concerns the overall process for constructing the networks. Another issues relates to bias, which arises from the innate cognitive processes that humans use during elicitation tasks, particularly probability elicitation. This makes the approach unattractive to lay users who merely want to make use of the benefits of BNs. To address these issues, we have composed a framework using a range of methods from the literature to support the hand-crafted BN construction approach.

Bayesian network models for dementia diagnosis In Chapter 3 we present the case for dementia diagnosis decision support, and we propose the BN formalism as the underlying decision engine. The hand-crafted construction approach is employed initially, as no data is available. This also serves to demonstrate the tools and methods for hand-crafting real-life BN models, which are described in detail in Chapter 4.

Dementia data set Dementia syndrome and pathology data for 164 patients, organised into a coherent database.

New search algorithm for BN construction, and comparison with existing techniques Exact methods for BN construction from data are, in general, not computationally tractable (see Section 7.1), therefore approximate methods are used as an alternative. Heuristic algorithms are typically proposed for the problem of BN construction from data, however the problem remains computationally complex. Some of the existing algorithms require an order among the variables to reduce the size of the search space, however it is known that this order-based approach impacts on the quality of resulting structures, and the order may need to be specified by an expert (who may not always be available). However, order-based algorithms such as the K2 (see Section 7.4.1.1) are known to find good BN structure when a good order is supplied. To that end, two-tiered algorithms emerged that seek to: 1) discover a good order then 2) input the good order into an order-based search algorithm. The drawback of early implementations of this two-tiered paradigm is the overhead in repeatedly searching for good orders with one algorithm and then evaluating the order with another. To address these issues we have derived a new algorithm for BN construction from data.

Comparison of approaches and techniques for BN model construction Compare the two construction approaches as well as the models derived by each approach. Like many approaches in Computing Science (and other domains), there is no single best approach to BN construction. We provide an empirical evaluation of the construction approaches, as well as a discussion on the pros and cons — this adds value by empowering others to decide on the best approach for their problems.

1.5 Thesis organisation

This thesis is divided into five parts. Part I introduces the core topics of the thesis: Bayesian networks (BN) and dementia diagnosis. Part II and Part III give a full treatment of the two BN construction approaches, with application to dementia diagnosis. Part IV is concerned with comparing both the models derived using each approach and the construction techniques associated with each approach. Finally, we conclude in Part V.

Part I sets the scene for the whole thesis by introducing the two core themes, namely Bayesian networks (BN) and dementia diagnosis. The motivation for a decision support tool to assist with dementia diagnosis in clinical practice is discussed, and we propose the BN methodology as the decision support engine.

Chapter 2 reviews Bayesian networks. It includes an overview of the value of BNs in problems characterised by uncertainty, as well as the underlying theory, which is centered on factorising the full joint probability distribution into a set of set of local, compact conditional distributions. Notation commonly used to describe BN structures is included, as it is referred to in subsequent chapters. The last part of the chapter introduces the notion of BN construction. Finally, a number of examples are provided where BNs have been applied to "real-world" problems.

Chapter 3 reviews the application area, namely dementia diagnosis in clinical practice. It includes an introduction to the dementia syndrome, as well as a brief overview of the causal pathologies investigated in this research. The impact of dementia on the individual, family and carers is described. In building a case for a decisions support, we highlight some of the pitfalls and barriers in dementia diagnosis that currently exist in clinical practice.

Part II focuses on the first approach to BN construction: the hand-crafted approach. The concept of the hand-crafted approach is provided first followed by a detailed description of current methods to support model construction under this approach. The details of two novel hand-crafted BN models (for dementia diagnosis) are presented to illustrate how the tools and methods are used. Thereafter, we present the results of applying clinical data from clinical practice to the models in order to evaluate their classification accuracy.

Chapter 4 explains the expert-driven approach to BN construction. It relies on eliciting from human experts qualitative information to create the structure, and quantitative information to furnish the local probability distributions. In this chapter we offer a generic process that facilitates model construction and tools for probability elicitation. Issues with eliciting probabilities from humans are well known. We identify these issues and provide a framework using methods from the literature to assist with the task. We feel that it is important to offer a framework to support the construction process because the overall BN construction process is complex, and there is no single one approach suitable for use by BN novices wanting to develop BN models for real-life problems. Note that methods described in this chapter are not specific to BN construction; they can be applied to other decision support techniques, particularly those underpinned by probabilistic information.

Chapter 5 illustrates how the development process and construction framework described in Chapter 4 can be used to hand-craft BN models for real-life problems. An example is provided from the medical domain — dementia diagnosis.

Chapter 6 evaluates and demonstrates the validity of the models developed through experimentation. A number of performance measures are defined that seek to measure classification accuracy of the diagnostic models against real life clinical data.

Part III is concerned with the second approach to BN construction, namely model discovery from data. This approach does not rely solely on human expert knowledge, but instead uses data to discover the BN model. However, the approach still has its own challenges, the primary challenge being the size of the search space. The data-driven concept and its issues are described. A new set of algorithms are proposed for BN discovery from data, and their performance is evaluated.

Chapter 7 reviews the data-driven approach and reviews the most important previous research in the field. The methods for "learning" from data are categorised into methods that use conditional independence test and search and score methods, where the BN problem is expressed as an optimisation problem. This research focuses on search and score methods. The challenges and issues associated with this approach are reviewed, and a new algorithm is proposed to address some of these issues.

Chapter 8 explains the development of new algorithms for BN discovery from data, which are based on the binary variant of Particle Swarm Optimisation (PSO). The development of the algorithms takes into account several issues faced by existing algorithms in previous research. These issues include removing the need to supply an order among the variables as input to the algorithm, as well as removing the need to employ expensive repair and validation operators when a search in the entire space of solutions is conducted.

Chapter 9 demonstrates empirically the performance of the new binary PSO algorithms for BN discovery from data. Additionally, the chapter shows that binary PSO can be used for BN discovery from data and that some

> of the issues with existing algorithms can be addressed using this approach — in particular, our approach does not require order information to derive a ‘good’ model. In addition, for some of the problem data sets used, we demonstrate that our technique outperforms (with statistical significance) the defined metrics in comparison to other algorithms in the literature. For the purpose of comparing approaches and derived models, experimentation is conducted using the dementia clinical data.

Part IV, composed of Chapter 10, pulls together the hand-crafted approach and the data-driven approach. In part II and Part III, two approaches are presented for BN construction. In this chapter we compare the differences in the models derived by each approach, and then we evaluate the two approaches against each other.

Finally, Chapter 11 makes up Part V. This chapter concludes and provides a summary of the research conducted, major contributions of this thesis, and makes recommendations on future work.

Chapter 2

Bayesian networks

This chapter introduces and reviews Bayesian networks (BNs). We begin by introducing the BN concept in Section 2.1, followed by a description of the theory of BNs in Section 2.3. Formal mathematical notation used to describe and qualify BNs is defined in Section 2.2. To harness the power of BNs in both representation of probabilistic knowledge and inference, they must first be constructed. The notion of BN construction is introduced in Section 2.4, and an overview of real-world BN applications is provided in Section 2.5.

2.1 Introduction

A BN enables intuitive representation of complex domains that are characterised by uncertainty. In addition, the BN framework supports probabilistic reasoning, in particular computation of posterior probabilities based on available evidence. In other words a BN model enables inference of future uncertain events based on prior related known events.

Jensen [132] provides a more formal definition of a BN, which is composed of 4 elements:

1. A set of nodes, or vertices, that represent the domain variables. Nodes represent discrete or continuous variables[1]. Associated with each node is a set of exhaustive, mutually exclusive states (or values).

2. A set of directed edges, referred to as connecting arrows or arcs, which represent dependency relationships between the variables [193]. The nodes, together with the edges form the structure of the BN model.

3. The edges between variables must respect Directed Acyclic Graph (DAG) conditions. In other words, all the edges in the graph are 'directed' (in that they point in a specific direction from one node to another) and the edges do not form 'cycles'. In this context, a cycle is defined as a path from a node, say X_i, back to itself. These two criterion are referred to as DAG constraints.

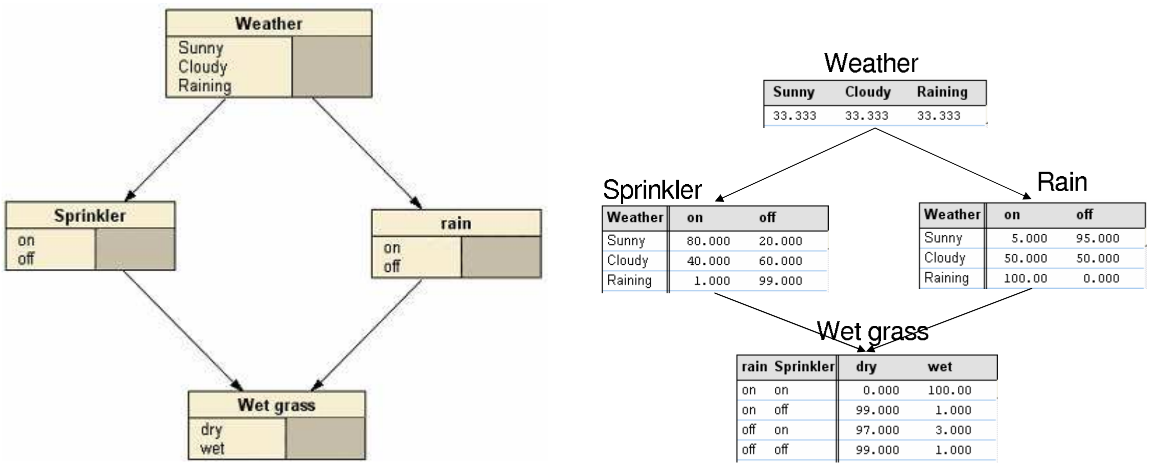
4. A set of local probability tables, one table per variable, which quantitatively encodes the strength of each dependency relation. On one hand, a variable's local probability is unconditioned if the variable has no parents, hence captures the prior probability distribution of the variable. However, if, on the other hand, a variable has one or more parent(s) then the local distribution across the states of the variable is conditioned on all possible values of the parent set.

The overall architecture of a BN model, in terms of Jensen's four components listed above, is composed of two features:

[1] This research focuses on discrete variables.

1. A qualitative feature, typically referred to as the network structure, which represents the domain variables[2]and the dependency relationships among them. Together, these components constitute a visual representation of the domain in terms of the variables and their interactions.

2. A quantitative feature, which encodes a set of local probability distributions that captures the uncertainty of the domain, characterising the impact that a variable or sets of variables have on other variables [50, 27, 216].



Sunny	Cloudy	Raining
33.333	33.333	33.333

Weather	on	off
Sunny	80.000	20.000
Cloudy	40.000	60.000
Raining	1.000	99.000

Weather	on	off
Sunny	5.000	95.000
Cloudy	50.000	50.000
Raining	100.00	0.000

rain	Sprinkler	dry	wet
on	on	0.000	100.00
on	off	99.000	1.000
off	on	97.000	3.000
off	off	99.000	1.000

(a) Weather model structure

(b) Weather model probabilities

Figure 2.1: Basic 'wet grass' BN model, adapted from [234, pp 510]. The BN structure, nodes and states are shown in (a); corresponding probability distributions for each node are shown in (b). Models created using Netica [2]

As can be seen from the example BN shown in Figure 2.1 (a), the weather BN model is composed of 4 variables: 'Weather', 'Sprinkler', 'Rain' and 'Wet_grass'. The 'Weather' node represents the possible outcomes of observations of the sky: sunny, cloudy and raining. Given the dependency relations shown in Figure 2.1 (a), it is clear that weather observations directly influence the probability of rain,

[2] We define a 'variable' as a single, identifiable entity of the domain which has 2 or more discrete attributes. Each variable is represented by a node in the BN structure.

as well as the probability that the sprinkler is on or off. Together, the nodes 'Sprinkler' and 'Rain' influence the state of the grass.

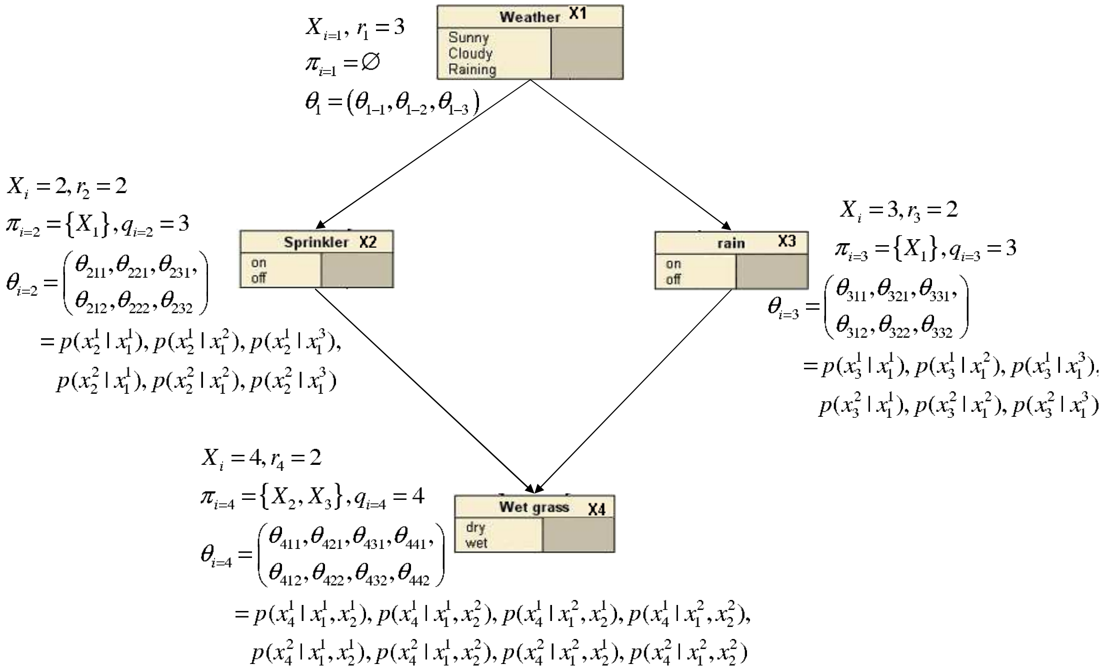
Interpreting a BN model may be likened to genealogy in that a parent-child relationship exists when there is an edge from one node to a dependent other node. For example, in Figure 2.1 (a), node 'Weather' is a parent of node 'Sprinkler', and node 'Sprinkler' is a child of node 'Weather'. This concept can be extended further to include ancestral-descendent relationships. For example, node 'Weather' is the ancestor of node 'Wet grass', and node 'Wet grass' is a descendent of node 'Weather'.

Associated with each node in the BN model is a probability table, as shown in Figure 2.1 (b). The probability that a node in the BN model is in a particular state is described by its conditional probability table. Independent nodes capture only the probability distribution across each state of the variable. On the other hand, dependent nodes have conditional probability tables that reflect the way in which the node is affected by the state of other related parent nodes.

2.2 Notation

In this section we introduce notation used by Kjrulff and Madsen [147], Lucas et al. [218] and Cooper and Herskovits [51] to describe BN models. This notation will be used throughout the thesis. In order to assist understanding, the Weather BN model described in Section 2.1 above has been annotated in Figure 2.2 to show the notation described in this section.

A BN model, Bs, is defined as a pair $(\mathcal{G}, \theta)$, where $\mathcal{G}$ is a directed acyclic graph (DAG) $\mathcal{G} = (\mathbf{V}, \mathbf{E})$, and θ is a set of local conditional probability distributions



$X_{i=1}, r_1 = 3$
$\pi_{i=1} = \varnothing$
$\theta_1 = (\theta_{1-1}, \theta_{1-2}, \theta_{1-3})$
Weather X1
Sunny
Cloudy
Raining
$X_i = 2, r_2 = 2$
$\pi_{i=2} = \{X_1\}, q_{i=2} = 3$
$\theta_{i=2} = \begin{pmatrix} \theta_{211}, \theta_{221}, \theta_{231}, \\ \theta_{212}, \theta_{222}, \theta_{232} \end{pmatrix}$
$= p(x_2^1 \mid x_1^1), p(x_2^1 \mid x_1^2), p(x_2^1 \mid x_1^3),$
$p(x_2^2 \mid x_1^1), p(x_2^2 \mid x_1^2), p(x_2^2 \mid x_1^3)$
Sprinkler X2
on
off
rain X3
on
off
$X_i = 3, r_3 = 2$
$\pi_{i=3} = \{X_1\}, q_{i=3} = 3$
$\theta_{i=3} = \begin{pmatrix} \theta_{311}, \theta_{321}, \theta_{331}, \\ \theta_{312}, \theta_{322}, \theta_{332} \end{pmatrix}$
$= p(x_3^1 \mid x_1^1), p(x_3^1 \mid x_1^2), p(x_3^1 \mid x_1^3),$
$p(x_3^2 \mid x_1^1), p(x_3^2 \mid x_1^2), p(x_3^2 \mid x_1^3)$
$X_i = 4, r_4 = 2$
$\pi_{i=4} = \{X_2, X_3\}, q_{i=4} = 4$
$\theta_{i=4} = \begin{pmatrix} \theta_{411}, \theta_{421}, \theta_{431}, \theta_{441}, \\ \theta_{412}, \theta_{422}, \theta_{432}, \theta_{442} \end{pmatrix}$
$= p(x_4^1 \mid x_1^1, x_2^1), p(x_4^1 \mid x_1^1, x_2^2), p(x_4^1 \mid x_1^2, x_2^1), p(x_4^1 \mid x_1^2, x_2^2),$
$p(x_4^2 \mid x_1^1, x_2^1), p(x_4^2 \mid x_1^1, x_2^2), p(x_4^2 \mid x_1^2, x_2^1), p(x_4^2 \mid x_1^2, x_2^2)$
Wet grass X4
dry
wet

Figure 2.2: Factorisation of Weather BN model (with notation).

that quantify the relations between the variables. The set of vertices $\mathbf{V}$, which may be discrete or continuous, represents the nodes in the BN model, and each node represents a random variable X_i in the problem domain. In Figure 2.2, four nodes are shown which correspond to the weather problem described in Section 2.1. The probabilistic dependency relations that connect the variables are represented by the set of edges, $\mathbf{E}$; these are depicted as arrows between the nodes in Figure 2.2.

Associated with each node X_i is a set of mutually exclusive events (states, or values), where the total number of states represented by a node X_i is denoted as r_i. The vector $\mathbf{x} = (x_1, \ldots, x_{r_i})$ denotes these r_i feasible instances. This can be seen in Figure 2.2, where each node has a number of states; for example, the Weather node has $r_i = 3$ states: 'sunny', 'cloudy' and 'raining'. The term X_i^k is used to refer to a node X_i being in its k-th state.

An edge, E_i, is an ordered pair $(X_i, X_j) \in \mathbf{V}$, which denotes a directed dependency link from node $X_i \in \mathbf{V}$ to node $X_j \in \mathbf{V}$. Given the edge (X_i, X_j), node X_i is said to be a parent of node X_j and X_j a child of X_i, intuitively denoted as $X_i \rightarrow X_j$. The set of parents belonging to a node X_i in a BN model B_s is denoted as $\pi_i^{B_s}$, where $\pi_i^{1,B_s}, \ldots, \pi_i^{q_i,B_s}$ denotes the values of $\pi_i^{B_s}$, and where q_i denotes the number of possible different configurations of the parents of X_i, as shown in Figure 2.2. A specific parent configuration is denoted as j_i.

Each node's probability distribution is captured in a (conditional) probability table, which is denoted by θ_i. Examples of such probability tables are shown in Figure 2.1(b). A row in the table captures the probability distribution across the states of X_i for each configuration of the parents of node X_i. In other words, the table records the probability P of a variable (X_i) being in each of its k^{th} states for each of the j^{th} unique parent configurations, which is denoted as θ_{ijk}; this is shown in Figure 2.2. More formally:

$$P(X_i^k | \pi_i^j, \theta_i) = \theta_{X_i^k | pa_i^j} \equiv \theta_{ijk}$$

2.3 Theory

Having introduced the concept of BNs in Section 2.1, we now turn our attention to the theory that underpins BNs. We start by giving an introduction to the full joint probability distribution model (FJPD) in Section 2.3.1. The FJPD is a simple model for representing probabilistic knowledge; however, the approach is limiting as it requires probabilities for each and every possible combination of the variables.

A BN is a graphical representation that expresses the FJPD more compactly. This is achieved by factorising the FJPD into a set of compact and concise local distributions. The underlying theory is described in Section 2.3.2.

2.3.1 Joint probability representation

Representation of probabilistic knowledge of a domain $\mathbf{X} = X_1, \ldots, X_n$ can be achieved by defining the full joint probability distribution (FJPD) across the set of domain variables, where the FJPD is defined as $P(\mathbf{X}) = P(X_1, \ldots, X_n)$. In other words, the FJPD is the probability of all combinations as defined by the values of all the variables [10]. An example of a domain consisting of 3 binary variables, $\{X, Y, Z\}$, is given in Table 2.1.

X	Y	Z	$P(X, Y, Z)$
$\neg X$	$\neg Y$	$\neg Z$	$P(0.005)$
$\neg X$	$\neg Y$	Z	$P(0.030)$
$\neg X$	Y	$\neg Z$	$P(0.150)$
$\neg X$	Y	Z	$P(0.350)$
X	$\neg Y$	$\neg Z$	$P(0.400)$
X	$\neg Y$	Z	$P(0.030)$
X	Y	$\neg Z$	$P(0.020)$
X	Y	Z	$P(0.015)$

Table 2.1: Full joint probability distribution for 3 binary variables.

The FJPD approach for representing probabilistic knowledge and reasoning therein, is limiting. Pearl [216, pp 78] notes three limitations:

1. An arbitrary set of n binary variables requires a probability table consisting of 2^n probabilities, where n is the number of domain variables — clearly, the number of probabilities required grows exponentially in n. Despite today's advances in memory capacity, such a representation raises significant computation challenges for any real-world application, notwithstanding the

complexities associated with evaluating inference queries on large probability distributions.

2. The complexity inherent in the FJPD approach is counter intuitive to the heuristic generally adopted by humans. Humans base probabilistic judgements on a small number of propositions, especially conditional statements such as the likelihood of a disease given a set of symptoms, as opposed to a complex conjunction of all possible propositions.

3. The mechanism used by the FJPD to represent probabilistic knowledge and perform probabilistic inference lacks psychological meaningfulness. While a pure numerical representation can produce coherent probability measures for each propositional sentence, it often leads to computations that a human reasoner would not use.

By introducing conditional independence assumptions, the complexity of the FJPD representation can be reduced and the issues presented above addressed. The reduction in complexity is achieved by factorising the FJPD into a set of direct probability distributions. The factorisation process is described in detail in Section 2.3.2.

2.3.2 Factorisation of the joint probability

In Section 2.3.1, we define a full joint probability distribution for a domain, $\mathbf{X}$, as $P(X) = P(X_1, \ldots, X_n)$. A BN structure ($Bs$) for $\mathbf{X}$ represents a set of conditional (in)dependence relationships between the variables in $\mathbf{X}$. Accordingly, the structure Bs for $\mathbf{X}$ provides a compact, concise, graphical factorisation of the joint probability distribution $P(\mathbf{X})$ [61].

Factorisation of the joint probability model is achieved using the following algorithm:

1. Apply an ordering to the variables, such that X_i is a candidate parent of X_j if X_i appears before X_j in the order.
2. Apply the chain rule of probability to factorise the joint probability into a set of conditionally independent probabilities. Therefore, each distribution in the chain corresponds to $P(X_i|\text{parents}(X_i))$. In other words, $P(X_i)$ is dependent only on the values of its local parents.
3. For each variable in the chain (derived in step 2), prune the parents by making conditional independence assumptions — see Section 2.3.2.1.

The chain rule of probability, in the general case, is defined as [216]:

$$\begin{aligned} P(\mathbf{X}) &= P(X_1, \ldots, X_n) \\ &= P(X_1)P(X_2|X_1)\ldots P(X_i|X_1, \ldots, X_{i-1})\ldots P(X_n|X_1, \ldots, X_{n-1}) \\ &= P(X_i|\pi(X_i)) \end{aligned} \tag{2.1}$$

This process is more easily understood by way of an example. Consider the variables in the weather problem, described in Section 2.1, above.

Assuming that an order (C, S, R, W) has been applied to the variables, then variable C (the current weather) can be a parent of any variable following C in the list. In the same way, S can be a parent of R and W. However, S cannot be a parent of C, nor R be a parent of S or C, and so on. Using the order specified, the full, joint distribution over the variables in the domain, $P(\mathbf{X})$, is defined as $P(\mathbf{X}) = P(C, S, R, W)$.

Using Equation 2.1, the local factorised distributions for the weather model are defined as:

$$\begin{aligned} P(\mathbf{X}) &= P(C, S, R, W) \\ &= P(C) \times P(S|C) \times P(R|C, S) \times P(W|C, S, R) \end{aligned} \tag{2.2}$$

The parent set of each variable in the factorised joint probability distribution can be simplified by making conditional independence assumptions of the form: $X_i \perp\!\!\!\perp X_k \mid X_j$, in other words, X_i is conditionally independent of X_k given that the value of X_j is known. The parents of the factorised distributions $P(R|C, S)$ and $P(W|C, S, R)$ are simplified given the conditional independence relations shown in Figure 2.1, thus the final distribution is defined as:

$$\begin{aligned} P(\mathbf{X}) &= P(C, S, R, W) \\ &= P(C) \times P(S|C) \times P(R|C) \times P(W|S, C) \end{aligned} \tag{2.3}$$

The third term in Equation 2.2, $P(R|C, S)$, is simplified because the assumption is made that R is independent of S given its parents C, written as $R \perp\!\!\!\perp S \mid C$. Therefore the parent, S, is pruned, which results in the local distribution $P(R|C, S) = P(R|C)$. Similarly, the last term in Equation 2.2, $P(W|C, S, R)$, can be simplified to $P(W|S, C)$ by applying the conditional independence assumptions $W \perp\!\!\!\perp C \mid S, R$.

Such simplification (or pruning) requires an understanding of the semantics of the particular BN at hand, as well as an understanding of conditional independence in the context of BNs. We provide a description in Section 2.3.2.1.

In the general case, the factorised joint probability distribution for a BN model is defined as:

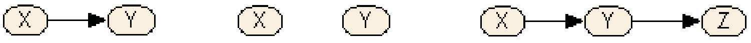
$$\begin{aligned} P(\mathbf{X}) &= P(X_1, \ldots, X_n) \\ &= P(X_1)P(X_2|X_1)\ldots P(X_i|X_1, \ldots, X_{i-1})\ldots P(X_n|X_1, \ldots, X_{n-1}) \\ &= P(X_1)P(X_2|\pi_2)\ldots P(X_i|\pi_i)\ldots P(X_n|\pi_n) \\ &= \prod_{i=1}^{n} P(X_i|\pi_i) \end{aligned} \tag{2.4}$$

where n is the number of variables in the domain, and π_i denotes the set of parents belonging to node X_i in the BN structure.

In this section we have shown how the FJPD can be represented more compactly by factorisation into local joint distributions. The benefit of the factorisation is realised in the reduction in the number of probabilities required. The savings in the number of probabilities required can be substantial. For n binary variables, the FJPD requires $O(2^n)$ probabilities; however, the factorised model requires only $O(n2^k)$, where k is the maximum number of parents feeding into a node [190].

2.3.2.1 Conditional independence in Bayesian networks

Factorisation of the full joint probability distribution is achieved by application of the chain rule of probability — see Section 2.3.2. If conditional independence



(a) Dependent (b) Independent (c) Conditional independence

Figure 2.3: Dependency, independency and conditional independence relations (a), (b) and (c)

(CI) assumptions are made, then the factorised probability distribution can be simplified further.

Conditional independence is closely coupled with the notion of information relevance, which is central to BNs. Information relevance allows successful capture of intuition about how variables interact with each other and, in the BN arena, is captured through devices known as dependence and conditional independence. These devices capture the way in which information at one node changes in response to new information presented at other nodes. For example, a proposition Y is said to be dependent on X if knowing the value of X helps predict the value of Y. In keeping with the weather example described above in Section 2.1, knowing that there are clouds in the sky (X) has a direct impact on the probability that the sprinkler (Y) is on. That is to say: $P(Y|X) \neq P(Y)$, shown in Figure 2.3(a).

The converse of dependence is independence, which implies that knowing the value of one variable has no impact on the value of another variable. For example, knowing that the sprinkler (X) is on does not have any impact on the probability that the car will not start (Y). Therefore $P(Y|X) = P(Y)$. This is shown graphically in Figure 2.3(b).

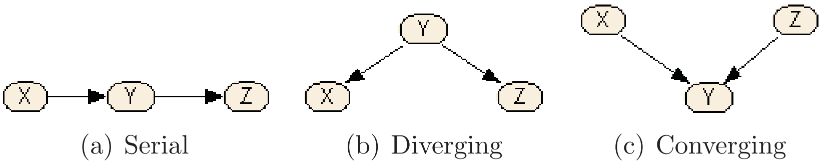
Given the respective tests $P(Y|X) \neq P(Y)$ and $P(Y|X) = P(Y)$, then determination of dependent and independent relations is trivial.

By extending the concept of independence, conditional independence relations can be formed. This is a relation in which a third node renders information at another node useless. In other words: A proposition X is said to be independent of Y, given information Z. For example, as shown in Figure 2.3(c), once Z is known, the probability of X will not be affected by discovery of Y, hence X and Y are conditionally independent if Z is known. More formally: $P(Z|X,Y) = P(Z|Y)$.

Determination of conditional independence relationships is an important task to accomplish as it enables pruning of parents in the local factorised probability distributions, as discussed in Section 2.3.2. However, it is more complex to evaluate conditional independent relations.

Pearl's *d*-separation (direction-dependent separation) criterion [216, pp 116] is used to evaluate conditional independence relations. The three possible configurations of variables shown in Figure 2.4 constitute the basic building blocks of BNs [131], and they are used by Pearl's d-separation criterion to determine whether conditional independence relations hold.

The three configurations are as follows:



(a) Serial (b) Diverging (c) Converging

Figure 2.4: Families of edge connections in a BN structure (a), (b) and (c)

Serial connection Information about X provides relevant information about the probability of Y, and knowledge at Y provides relevant information about Z. However, if the state of Y is known, then information about X

provides no additional information about Z. It is said that Y 'blocks' information flow X from Z. For example: an observation of cloudy weather (X) would increase the belief about rain (Y), which, in turn, would increase the probability of the grass being wet (Z). However, knowing that it is raining (Y), a weather observation of cloud (X) is irrelevant to the probability that the grass is wet (Z).

Diverging connection Information about the state of X impacts on the probability of Z through Y. However, if information is known about Y, then any new information about X does not impact on Z. It is said that Y blocks information flowing from X to Z. Therefore, information can flow between the variables unless the state of Y is known. For example: If the current weather conditions are raining (Y), and then discover that the grass is wet (X), realising that the grass is wet (X) does not provide any additional information to the 'rain' node (Z).

Converging connection Information about X does not provide relevant information about Z unless information about Y is known — Y depends on both X and Z. In other words, information may be transmitted from X to Z only if information about the state of Y or one of its descendants is known. For example: If information about the state of the lawn (Y) is available, then knowing the state of the sprinkler (X) will modify the probability of rain. In other words, knowing that the lawn is wet and that the sprinkler is on changes the belief that the lawn was made wet by the rain. However, if information about the state of the lawn is unavailable, observing rainfall does not impact the belief that the sprinkler is on. Converging connections in BN are important as they represent a common reasoning pattern known as ‘explaining away’. In other words, Y has two causes, X and Z; if you know the state of Y, then knowing the state of X will change the probability of the other cause, Z.

A test of conditional independence is achieved by applying Pearl’s *d*-separation criteria to the three basic node configurations described above. For example, two nodes (X) and (Z) are *d*-separated, and hence independent, if there exists an intermediate variable Y such that either:

- The connection between X and Y is serial or diverging and the value of Y is known
- The connection between X and Y is converging and Y and its descendants are known.

2.4 Construction

As mentioned in Section 2.1, a BN model is composed of two parts — 1) a qualitative structure; and 2) quantitative probabilities — which together provide an efficient platform for modelling relationships and executing probabilistic inference. However, the BN model must first be built, thus the problem concerning construction of the network remains. The BN structure and probabilities could be provided by either a domain expert or induced automatically from data.

The topic of BN construction, specifically the structure, is central to this research. The expert driven approach to BN learning is treated in part II of this thesis, and structure learning from data is treated in part III.

2.5 Applications

The idea of graphical models for representing uncertainty can be traced back to the 1921's when Sewal Wright establish a method for representing causal models for the analysis of crop failure [283]. This time period was ruled by hard data and quantitative analysis, therefore the method did not receive buy-in from the statistics community [216, 131]. During the 1960's, statisticians began to realise that the decomposition properties of statistical tables is best expressed in graphical terms, which reignited research around methods for representing and reasoning with problems characterised by uncertainty.

As research evolved, efficient mechanisms for representing, encoding, storing and manipulating probabilistic information were realised, and BNs emerged. Bayesian networks have a presence in many diverse domains, including both academic research and industry. A common use of BNs is to provide decision support for problems characterised by uncertainty. This may be explained by their natural ability to graphically model uncertain domains, and provision a rigorous probabilistic reasoning [111]. Examples of real-life applications include: ecology [177, 173, 7], bioinformatics [81], sociology [9] and transport [248, 34]. Other applications include aquaculture [98], learning and development [92] and geology [228]. Bayesian networks have gained an increasing popularity in the medical domain, especially in handling the uncertain knowledge involved in establishing diagnoses of disease [218]. Examples include monitoring of patients in intensive care [11], medical decision support in breast cancer diagnosis [146, 58], radiology [28], therapeutics [200] and biomedical informatics [270].

2.6 Summary

In this chapter we have described the fundamental core concepts of Bayesian networks. It started by giving a basic overview of the BN framework, and then moved to the more theoretical aspects of the framework. The reader's attention is drawn to the value of BNs in problems characterised by uncertainty, noting that their graphical structure of dependency relations makes interpretation easier, and the conditional probability tables enable efficient probabilistic inference. In order to realise the potential of BNs, they must first be constructed, thus we introduce the notion of BN construction techniques, which is a major theme of our research.

This thesis investigates the application of BNs for the purpose of dementia diagnosis. Now that we have presented the background to BNs, the topic of dementia is explored in the following chapter.

Chapter 3

Decision support for dementia

3.1 Introduction

Dementia is the application area of choice in this research. The purpose of this chapter is to introduce dementia and, by understanding the complexity of the diagnostic problem and analysing current diagnostic practice, justify the need for dementia decision support. In addition, this chapter explores current dementia diagnosis clinical practice, which, through identification of gaps, provides motivation for a new approach.

This chapter is organised as follows: a description of dementia is provided in Section 3.2, and the impact that a diagnosis of dementia has on the individual, family and carers, as well as wider implications, is provided in Section 3.3. We provide an overview of current diagnostic clinical practice in Section 3.4. In addition, we highlight gaps and issues in dementia diagnosis in clinical practice, which provides motivation for a decision support tool in dementia diagnosis, as described in Section 3.4.1.2. Treatment management aspects of dementia are discussed in Section 3.5. A summary of the chapter is provided in Section 3.7.

3.2 What is dementia?

Cognitive changes	*Behavioural changes*	*Emotional changes*	*Physical changes*
• Short term memory impairment • Language disorder - receptive and expressive dysphasia • Preservation of thoughts and actions, accompanied by repetitive speech • Fragmented thought process - speech becomes disordered • Persecutory ideas and delusions	• Social withdrawal • Difficulty in carrying out purposeful tasks - activities of daily living • Socially inappropriate behaviour and self neglect • Wandering and restlessness • Aggression and violence	• Shallowness of mood • Lack of emotional responsiveness and consideration for others • Irritability and hostility	• Weight loss • Malnutrition • Incontinence • Emergence of primitive reflexes

Table 3.1: Selection of common, clinical features associated with dementia. Adapted from [66].

Dementia is a syndrome characterised by the presence of multiple acquired cognitive deficits, which typically progress in severity over time. The term dementia is accurately reserved for conditions considered irreversible. A number of different pathologies can lead to the syndrome of dementia, either singly, or in combination. The most common pathologies leading to dementia are more prevalent in aged populations. A selection of clinical features most commonly associated with dementia are summarised in Table 3.1.

In older people, Alzheimer's Disease and a variety of cerebrovascular related dementias — collectively often referred to as "vascular dementia" — are by far the most common dementia causing pathologies [195, pp 1]. Other pathologies that cause a significant care burden, although less common, include dementia with Lewy Bodies (DLB), the linked dementia of Parkinson's Disease, and a range of primary neurodegenerative conditions affecting the frontal and temporal lobes ("frontotemporal dementias"). A number of rarer dementia processes including Creutzfeldt-Jakob Disease (CJD), Huntington's Disease, and extremely

Disease	*Clinical features*
AD	• Memory impairment • Cognitive decline from a previous higher level of functioning • Deficits in 2 or more areas of cognition • Cognitive deficits of sufficient magnitude to interfere with occupational and/or social functioning • Onset between ages 40 and 90, most often after age 65 • Absence of systemic disorder or other brain disease that in and of themselves could account for the progressive deficits in memory and cognition
VaD	• Memory impairment • Cognitive decline from a previous higher level of functioning • Deficits in 2 or more areas of cognition • Cognitive deficits of sufficient magnitude to interfere with occupational and/or social functioning • Cognitive deficits not due to physical effects of stroke alone • Cerebrovascular disease • Relevant cerebrovascular disease by brain imaging
DLB	• Cognitive decline from a previous higher level of functioning • Cognitive deficits of sufficient magnitude to interfere with occupational and/or social functioning • Prominent or persistent memory impairment may not occur in early stages, but usually evident with progression • Cognitive decline from a previous higher level of functioning • Two of the following: a) fluctuating cognition with pronounced variations in attention and alertness b) recurrent visual hallucinations, typically well formed and detailed c) spontaneous motor features of Parkinsonism • Repeated falls
FTD	• Behavioural disorder, such as early loss of personal and social awareness, early sign of disinhibition and hyperorality • Affective symptoms, such as depression, anxiety, excessive sentimentality and bizarre somatic preoccupation • Speech disorder, such as progressive reduction of speech and stereotypy of speech • Spatial orientation and praxis preserved • Physical decline, such as early primitive reflexes, early incontinence, rigidity, tremor and low and labile blood pressure

Table 3.2: Summary of common clinical features associated with AD, VaD, DLB and FTD.

infrequent neurological disorders are also occasionally encountered, as are dementias associated with abuse of alcohol, and with Human Immunodeficiency Virus (HIV). This thesis focuses on the four most common dementias associated with aged populations: Alzheimer's disease (AD), Vascular dementia (VaD), dementia with Lewy bodies (DLB) and Fronto-temporal dementia (FTD). The core clinical features associated with each disease are summarised in Table 3.2.

3.3 Impact of dementia

The dementia syndrome can affect people of any age, but it is most common in older people. In a recent study conducted by Knapp et al. [148], the prevalence of dementia in the year 2005 was reported as 1 in 14 people over 65 and 1 in six people over 80. Overall, the number of people with dementia in the United Kingdom (UK) in the year 2005 was 683,597, 1.1% of the entire UK population, which is expected to rise to 1,735,087, 2.7% of the entire UK. This rise is due to the overall aging population.

A diagnosis of dementia has far reaching consequences for the individual, close relatives and carers. In the case of the individual, a diagnosis is often preceded with a period of frustration, which is initiated by the realisation of a decline in ability to function in daily life. In addition to frustration, Behavioural and Psychological Symptoms of Dementia (BPSD) , summarised in Table 3.3, begin to emerge as the disease progresses [14, 86]. Frequently, especially when severe, BPSD manifestations are disturbing to family members and other caregivers [151], and can pose significant danger and distress to the individual, sometimes requiring institutionalisation [71] [205, pp 67].

Notwithstanding the effect dementia has on the individual, family and carers, there is significant cost implication with regards to healthcare services. The dementia UK study [148] estimates that the cost of formal care agencies as well as the financial value of unpaid informal care provided by family and friends amounts to £17.03 billion, an average of £25,472 per person cared for per year. It is expected that costs will increase as the number of people with dementia is

BPSD symptom	*Desciption*
Wandering	
Restlessness	
Delusion	Beliefs that are not true - insisting that people are trying to inflict harm
Hallucinations	Seeing false visions or hearing false voices
Agitation/Aggression	Refusal to cooperate, refusal to receive assistance or violent behaviour
Depression/Dysphoria	Periods of sadness, low spirt
Anxiety	Nervous, worried or frightened behaviour for no apparent reason
Elation/Euphoria	Persistent, excessive and abnormally good mood. Finds humour when others do not
Apathy/Indifference	Loss of interest in surrounding environment, difficult to engage in conversation or chores
Disinhibition	Act impulsively without thinking, says things not usually said in public
Iritability/Lability	Easily irritated, impatient, easily disturbed and moods change rapidly for no specific reason
Aberrant Motor Behaviour	Pace, repetitive actions (opening/closing doors)
Night behaviour disorder	Sleep and night time behaviour disorder
Eating disorder	Appetite and eating disorder

Table 3.3: Summary of common behavioural and psychological symptoms of dementia.

set to increase, and more service provisions will be required as the number of people with dementia grows.

3.4 Dementia diagnosis in clinical practice

Diagnosis of the common dementias of old age are principally operationally defined on the basis of their differing constellations of symptoms and neuropsychological profiles. In doing so, clinicians use a variety of sources of evidence in the reasoning process, which include evidence-based clinical guidelines, often supplemented by individual consultations.

Clinical guidelines, developed by domain-specific experts, identify, summarize and evaluate the state of the art in medical evidence and current data about prevention, diagnosis, prognosis, therapy, management, risk, benefit and cost-effectiveness, and, in addition, address practical issues. Furthermore, clinical guidelines define important questions related to clinical practice, and identify possible decision options and their outcomes, thus guiding the clinician's decisions. The overriding objective of clinical guidelines is to standardise medical care, raise quality of care, reduce risk and achieve a balance between cost and medical parameters [196]. Clinical guidelines alone do not replace the knowledge and skills of the clinician ultimately responsible for patients; it is the responsibility of the clinician to make judgements and decisions in consultation with the patient, or where appropriate, close relatives or carers.

The Diagnostic and Statistical Manual (DSM), currently in its 4^{th} edition [4], lists different categories of mental disorder and the criteria for diagnosing them. The mental disorders Section of the International Classification of Diseases, version 10 (ICD-10), is another commonly-used guide which provides codes to classify diseases and a wide variety of signs, symptoms and abnormal findings. Every health condition can be assigned to a unique category and given a code. The ICD-10 guidelines are essentially "a statistical classification of diseases and other health problems, to serve a wide variety of needs for mortality and health-care data [211]"; the DSM, on the other hand, is essentially a classification of the disorders seen and treated in clinical practice. Other specific dementia-related clinical guidelines are listed below.

- Alzheimer's disease: NINDS-ADRDA, McKhann et al. [175].
- Vascular dementia:
  - NINDS-AIREN, Román et al. [231]
  - Hachinski Ischaemic Scale [102]
- Dementia with Lewy Bodies: Criteria proposed by McKeith [174].

- Frontotemporal dementia: Criteria proposed by The Lund and Manchester Groups [167].
- Scottish Intercollegiate Guidelines Network (SIGN) provide guidance on interventions for the management of behavioural and psychological aspects of dementia [195].

The clinical diagnosis of dementia should be based on a standardised system such as ICD-10 or DSM-IV [205, pp 23]. Furthermore, a careful, methodical and detailed history is an important part of the clinical assessment of someone with suspected dementia [105]. Due to the nature of the syndrome, however, consultation with the aim of obtaining detailed history information from the patient alone may be impractical, therefore, independent history should be obtained from close relatives or caregivers where available. In particular, BPSD related disorders are best known by relatives or carers, and can be detected during informant consultation.

In progressive degenerative dementias, during the very early stages cognitive deficits are usually subtle and their manifestations, though representing change from pre-morbid function in an individual, may remain within the normal range for the general population. This presents considerable challenges to early diagnosis. Cognitive screening and regular monitoring may be of assistance in tracking deterioration in cognitive and functioning abilities. Nevertheless, early diagnosis and categorisation of dementia-causing pathology is of value in planning treatment and, in some cases, initiating specific drug intervention.

Accordingly, a diagnosis of dementia can be formed by evaluation of the clinical syndrome based on diagnostic criteria, history and examination, paying particular attention to mode of onset, course of progression, pattern of cognitive impairment and presence of non-cognitive symptoms. Additionally, screening investigations, and, where appropriate, additional more specialised investigations, such as neuropsychological testing [237, 15] imaging [251, 133], blood tests [105, pp 25] and systemic investigations [105, pp 29] may be required.

There are a number of laboratory investigations which can assist in the diagnostic process, although, as yet, no definitive laboratory test for the common dementia causing pathologies is available. Histological evaluation ante mortem is not ethically justifiable.

Preliminary diagnostic studies have demonstrated the potential role of cerebrospinal fluid (CSF) in investigating dementia [137, 107], an invasive investigation made possible by lumbar puncture. In particular, studies have shown that specific biomarkers in CSF, specifically reduced levels of CSF β-amyloid and raised levels of CSF tau, can differentiate patients with AD from patients with other dementias as well as from people without dementia [273]. However, the Scottish Intercollegiate Guidelines Network (SIGN), established in 1993 by the medical Royal Colleges to develop evidence based national guidelines for NHS Scotland, does not advocate routine use of CSF markers in the diagnosis of dementia, although other countries may recommend routine CSF investigation. The justification for such invasive investigations may increase as more effective treatment options become available for some of the common dementia types.

Genetic testing can provide information relating to possible “genetic faults” (mutations) that could lead to early onset and late onset dementias, however, such testing is in its infancy, and it is not used routinely in clinical practice. Furthermore, genetic testing has far reaching implications for the individual [113]. Thus far, three genes have been identified as possible causes of the rare early onset (age ≤ 65) familial AD, namely the amyloid precursor protein (APP) gene on chromosome 21, the presenilin-1 (PS-1) gene on chromosome 14, and the presenilin-2 (PS-2) gene on chromosome 1 [105, pp 45]. The APP gene affects the production of the protein amyloid; it has been shown that an accumulation of abnormally folded amyloid in the brain has been linked to AD [108]. On the other hand, recent observations have identified a gene thought to be associated with late onset, sporadic AD [168, 110], namely the apolipoprotein ϵ gene (Apoϵ) on chromosome 19. The Apoϵ gene has three alleles, $\epsilon 2$, $\epsilon 3$ and $\epsilon 4$. The $\epsilon 4$ mutation of the Apoϵ gene increases the risk of developing AD [67, 153]. Nevertheless, many people with AD do not have the $\epsilon 4$ mutation and many people with the $\epsilon 4$ mutation do not have AD. Everyone has two copies of the APOϵ gene, one copy inherited from each parent. Those carrying two APO$\epsilon 4$ genes have the most risk of developing AD. It should be noted, however, that although carriers of two Apo$\epsilon 4$ genes have been linked to increased chance of developing AD, Apo$\epsilon 4$ is not causative of AD [105, pp 47] [106], and, in addition, not all people with two Apo$\epsilon 4$ genes will develop the disease [37]. The utility, therefore, in Apoϵ testing relates to the hypothesis that Apo$\epsilon 4$ is not a predictor of AD, rather, if AD is going to happen, Apo$\epsilon 4$ may affect when the pathology is likely to develop [67].

Integrated Care Pathways (ICP) are instruments that outline anticipated multidisciplinary care, placed in an appropriate time-frame, to help a patient with a specific condition or set of symptoms move progressively through a clinical journey [181]. Moreover, ICPs can be used as a vehicle to incorporate local and national guidelines into everyday practice, manage clinical risk, meet the requirements of clinical governance and reduce unnecessary variations in patient care and outcomes [31]. There is no "single model", however, as variations from the pathway may occur as clinical freedom is exercised to meet the needs of the individual patient within the constraints of the local service provision [181]. Furthermore, the ICP is not a prescriptive, step-by-step set of instructions, rather, it is a set of appropriate, evidence-based activities and interventions for a specific user group. The Kingshill Research Centre, UK, have developed an integrated care pathway for dementia [192]; it consists of five stages, listed below.

1. Recognition
2. Assessment
3. Management
4. Review
5. Coping with change

In addition to the Kingshill Research Centre ICP, the NHS Quality Improvement Scotland (NHS QIS) board developed an ICP [240], released in draft form in April 2007, which covers five mental health areas, including dementia. Each of the stages contained within the Kingshill Research Centre ICP, listed above, include the evidence available for that part of the care process, and a flowchart identifying best practice.

In keeping with the theme of dementia diagnosis, an abstract description of the identification and assessment of dementia by the primary care team and expert team is provided in Section 3.4.1.1 and Section 3.4.2. The remaining three stages, whilst playing an important role in the holistic view of dementia care, are not within the scope of this thesis.

3.4.1 Primary care patient journey

In the first instance of health concern, the affected individual or a carer usually presents to a Primary Care clinician — the General Practitioner (GP) — thus, the GP and the wider primary care (PC) team are the initial point of contact [21]. Therefore, the GPs and the wider PC team play an important role in the diagnosis of dementia [68].

With regard to their role in diagnosis, the GP and the PC team perform a variety of functions, however, the primary judgements are: identify people suspected of having a dementing illness; determine the significance of the illness; exclude non-dementia pathologies, such as hypothyroidism and B12 deficiency; and exclude overlapping conditions, such as depression, acute confusional state and psychotic symptoms. In doing so, the GP conducts a number of investigations, including: physical examination; background history; cognitive screening tests, such as the Mini-Mental Status Examination (MMSE) [87] and the clock-drawing test [23]; and blood investigations [267]. If dementia is suspected, or if the diagnosis is unclear, the GP may choose to refer onward for more specialist diagnostic assessment at secondary care level, which can include specialist investigations such as neuropsychological evaluation [252].

Since no one single patient journey model exists, an abstract, graphical patient journey, including judgements described above, is shown in Figure 3.1. A comprehensive discussion on integrated care pathway models for dementia care can be found in [240, 192, 69].

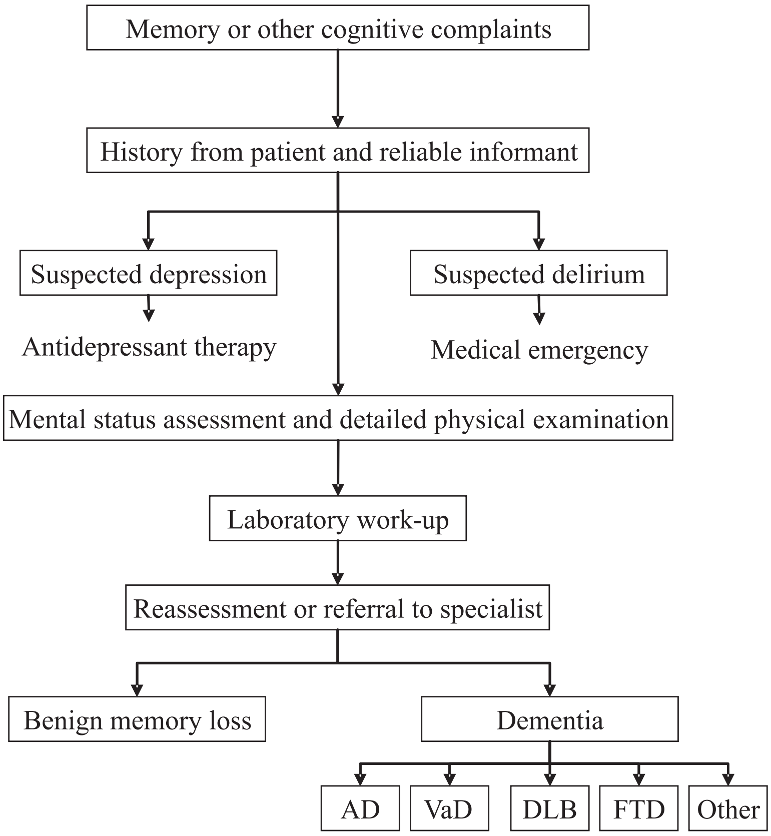

Memory or other cognitive complaints
History from patient and reliable informant
Suspected depression
Antidepressant therapy
Suspected delirium
Medical emergency
Mental status assessment and detailed physical examination
Laboratory work-up
Reassessment or referral to specialist
Benign memory loss
Dementia
AD
VaD
DLB
FTD
Other

Figure 3.1: Schematic of the diagnosis of dementia in primary care. Source: [93, pp 36]

3.4.1.1 Issues and barriers in primary diagnosis of dementia

The importance of early diagnosis in facilitating better management and quality of life for the affected individual and carers is well documented. An early diagnosis can have the effect of "reducing uncertainty about and coming to terms with the diagnosis, excluding remedial causes (however rare), planning support and avoiding crises " [129]. Early administration of anti-dementia drugs has many therapeutic effects, including improved memory, as well as boosting cognition, leading to increased participation in a range of activities of daily living [230, 261], and delaying institutionalisation [95] — Section 3.5. Other benefits of early diagnosis include introducing the individual and carers to appropriate agencies and support networks that provide respite care, and carer groups. Such support groups have the capacity to improve carer quality of life by relieving the disabling psychological distress that carers may experience, reduce carer burnout and delay institutionalisation [284, 184], as described in Section 3.3.

Since the primary care team, in particular the GP, is the first point of contact for people with dementia and their families, "they have a pivotal role in both establishing a diagnosis of dementia and in ongoing support and intervention" [129]. However, despite the benefits to be gained from early identification and diagnosis of dementia, as mentioned above, there is substantial literature which documents the difficulty GPs and the primary care team have in fulfilling this role. For example, Downs [72] and Gauthier [93, pp 32] note that GPs are generally unaware of people with dementia in their practice, and that dementia cases are identified following a medical or carer crisis, by which time the disease has progressed. Additionally, once individuals are identified as suspected dementia cases, the primary care team face a number of barriers. Some of these barriers include: difficulty differentiating normal ageing from symptoms of dementia [30]; difficulty in diagnosing due to uncertain aetiology and patholophysiology, compounded by the fact that symptoms are highly variable and non-specific [176]; lack of training to respond to the needs of people with dementia and their families [129, 182]; lack of confidence in diagnosing dementia [264]; concerns about the impact of the diagnosis on the patient [259, 220]; and lack of time [129]. These barriers pose a significant challenge to dementia care, however, literature suggests that GPs remain committed to facilitating early diagnosis of dementia [182]. It should be noted, however, that this study focused only on East Kent, England, UK, therefore, a wider study is required to determine whether this trend continues across other UK regions, but anecdotally likely to do so.

The interested reader is directed to [130, 264, 129, 72, 182, 88] for a comprehensive discussion on the issues and barriers of dementia diagnosis in primary care.

3.4.1.2 The need for dementia decision support in primary care

The diagnosis of dementia should include the use of standardised clinical guidelines, for example, SIGN [195, 93]. However, Stoppe et al. [258] report wide gaps between guidance provided in clinical guidelines and practice in primary care, which may be explained by the fact that guidelines alone have limited effect in changing clinical practice [83, 101]. Gaps occurring between guidance provided in guidelines and clinical practice, combined with the issues highlighted in Section 3.4.1.1, may contribute towards underdiagnosis or misdiagnosis.

As described in Section 3.4.1.1, the main issues relating to dementia diagnosis at the primary care level include: lack of dementia specific training; confidence in forming a diagnosis of dementia; difficulty in differentiating normal ageing from symptoms of dementia; and time. The issue relating to dementia specific training for primary care practitioners is historical, and well documented. However, in a survey of Irish GPs conducted by Cahill et al. [30], the appetite of GPs to attend dementia specific training courses appears to be increasing. Other educational packages have been introduced in recent years, such as the North of England evidence-based guidelines development project: guideline for the primary care management of dementia [80], and, more recently the SIGN guidelines discussed in Section 3.4.

The complexity surrounding dementia diagnosis and associated issues suggests that a decision support system, which is easy and quick to use, encodes specialised, expert knowledge from an Old Age Psychiatrist, clinical guidelines and a robust diagnostic inference engine, could assist physicians in forming a diagnosis of the dementia syndrome, and classification of the underlying pathology. Such a system could provide real benefits to patients and carers, and, since it is an innovative approach [63], which seeks to reduce time and assist the diagnostic task, it may gain widespread adoption by GPs. In addition, the system could be useful clinically for some patients when early intervention therapies are developed.

Existing approaches which seek to provide decision support for dementia are surveyed in Section 3.6.

3.4.2 Expert team patient journey

When the GP is unsure about the diagnosis, or the diagnosis is complicated, a patient may be referred to a specialist psychiatric service known as the *expert team*. Other people requiring referral include, but are not limited to, those where: there is associated psychopathology such as depression or psychosis; there is dangerous or severely disturbed behaviour such as violence, wandering or sexual disinhibition; or severe carer strain — see Section 3.3.

Different styles and models of service have been developed, however, guidance from government and the Royal Colleges has encouraged broadly similar levels and patterns of provision. Although there is no one single patient journey model for specialist care, good practice guidelines exist, such as: the Audit Commission's report *Forget Me Not* [48], which outlines Integrated Care Pathways (ICP) and services for older people with dementia; and *Forgetful but Not Forgotten* [205], a report produced by the Royal College of Psychiatrists which outlines good practice and ICP in old age psychiatry services working with people with dementia and their carers. Both documents include diagnostic assessment and investigation guidance.

The expert team deals with many aspects, stages and varieties of psychiatric disorder arising in old age, especially all stages of dementia, including care for people with severe dementia. Two primary roles provided by the expert team include detailed diagnostic investigations, and management of dementia. The core team typically consists of a consultant Old Age Psychiatrist and community psychiatric nurses. Other specialists from geriatric medicine, neurology and psychology, may also contribute to the assessment of dementia.

3.5 Treatment management

Early diagnosis and categorisation of dementia-causing pathology has high value in treatment management, especially in cases which require specific drug intervention [72]. In addition, once an individual has a positive diagnosis of dementia they have access to a package of care, which spans health care services, social services and voluntary services [239, 238]. It should be noted, however, that within the UK the actual services that an individual gains access to vary [148, Ch. 4 and Ch. 5]. Furthermore, pharmacological treatment available to individuals with dementia varies significantly depending on geographic location [134].

Cognitive decline, functional decline and social decline are the three cardinal features of dementia. Cognitive decline affects the individual's ability to remember, understand, communicate and reason. For some individuals, cognitive symptoms can be managed pharmacologically using agents which seek to boost cognition, the so-called *cognitive enhancers* or *anti-dementia* drugs. The National Institute for Health and Clinical Excellence, NICE, provided guidance on the use of these drugs in clinical practice — NICE-86 [197, Ch.7].

The neurotransmitter, acetylcholine (Ach), is important in the chemical basis of a number of cognitive processes including memory, thought and judgement [13]. A hallmark of AD is lower than normal level of acetylcholine in the brain, due in part to malfunction and deterioration of neurons that release Ach [217]. Ach is broken down naturally by an enzyme known as acetylcholinesterase (AChE). In 'normal' brains, however, cognition is not affected as the production and release of Ach is greater than the rate at which AchE operates. AChE activity does reduce with age [120], however, the main issue is the balance between Ach production, release, and breakdown. A simple example: if AchE activity is faster than the rate at which Ach is produced and release, or, if Ach production and release is substantially diminished, even if AchE activity is not, then the result is cholinergic dysfunction, as there is not enough Ach to stimulate Ach receptors, thus leading to cognitive and memory impairment. The role of pharmacological agents, specifically cholinesterase inhibitors, therefore, is to slow the rate at which AchE enzymes break down Ach released from remaining, undamaged neurons. This results in a boost in cognition and memory.

In the UK, Aricept®(donepezil HCL), Exelon ®(rivastigmine tartrate), and Reminyl ®(galantamine), cognitive enhancers which target the cholinergic neurotransmitter system, are licensed for use in the symptomatic treatment of mild to moderately severe AD [197, pp 185]. Memantine (Ebixa) is another cognitive enhancer licensed for moderate to severe AD in the UK; however, it works differently to that of the acetylcholinesterase inhibitors. Glutamate is a neurotransmitter involved in learning and memory, however, it is believed that abnormally elevated amounts of glutamate may lead to cell death, which has been observed in patients with AD. By targeting the glutamatergic neurotransmitter system, memantine seeks to boost cognition by protecting neurons against elevated levels of glutamate. The potential for combination therapy is being explored, and preliminary data suggest that memantine in combination with aricept is superior to aricept alone in the treatment of moderate to severe AD [260]. However, whether doctors will prescribe both drugs together routinely in clinical practice, especially on the NHS, is unclear.

In the UK, rivastigmine is the only pharmacological therapeutic agent licensed for the symptomatic treatment of people with VaD, DLB and FTD, although there are no studies on the use of acetylcholinesterase inhibitors or memantine for the treatment of FTD [195, pp 186].

In the year 2005, NICE reviewed their guidance on the management of early-stage and late-stage dementia, and, on the basis of a cost-benefit analysis, NICE ruled that donepezil, rivastigmine and galantamine should only be used to treat Alzheimer's once it has progressed to its moderate stages [195]. NICE has also ruled that memantine should be used only in clinical studies of people with moderately severe to severe AD. The review sparked a heated debate between Alzheimer's campaigners, drug companies and NICE, as campaigners argue that patients in the early stages of Alzheimer's should also have access to the drugs. Patients already prescribed the drugs will continue to receive them in accordance with clinical guideline criteria.

As a result of biological change in the brain due to progression of the disease process, non-cognitive features, or BPSD, may emerge — see Section 3.3. Pharmacological interventions may be instituted to help manage behavioural and psychological symptoms, particularly in cases "where the disturbance is acute, severe and/or life-threatening [205, 67]". The "Forgetful but not Forgotten" report [205, 67], however, recommends that non-pharmacological approaches should be the management of choice in the first instance. Non-pharmacological approaches to treating people with dementia adjunct to pharmacological intervention appear in the literature, including aromatherapy, reminiscence therapy, validation therapy and music therapy. Note, however, this area of dementia treatment is under-researched and requires further review [282, 271, 118].

3.6 Dementia decision support and the Bayesian network value proposition

As mentioned in Section 3.4.1.1, the first port of call for health concern is usually the Primary Care (PC) clinician — the General Practitioner (GP) — thus, the GP and the wider PC team have a key role at the initial point of contact, and therefore have a crucial role in the diagnosis of dementia [68]. It is the view of the The Audit Commission in their Forget Me Not [48] report, however, that more needs done in general practice to diagnose dementia in its early stages. The view from general practice, however, is that a number of barriers are preventing accurate and timely identification of dementia, meaning that the individual in some cases may not obtain an accurate or timely diagnosis, nor will they be referred to the appropriate secondary level, expert care team [208]. Barriers including under-resource and lack of training [264, 209], combined with the complexity associated with dementia diagnosis. These barriers are some of the key drivers that motivate the development of tools for dementia decision support, particularly in primary care.

Iliffe et al. [128] identified that primary care physicians are under-equipped to diagnose and manage the complexities associated with dementia diagnosis. To address this, they proposed a computer decision support (CDSS) tool to assist physicians in clinical practice with dementia diagnosis. The system prompts the user with questions to consider in deciding whether a diagnosis can be made, or whether specialist referral is required. The decision engine platform is a decision tree, hand-crafted and validated by a multi-disciplinary team of health professionals, and is based on clinical guidelines for dementia diagnosis. The completed CDSS is integrated into two GPs Patient Information Systems, and is deployed in some clinical practice centres. Iliffe et al. demonstrated that: 1) it is possible to construct a model for dementia diagnosis based on decision trees; and 2) the CDSS appeared useful in routine practice, although commented that further evaluation is required in primary care clinical practice.

The benefit of the approach proposed by Iliffe et al. is that the decision engine is graphical and transparent in its diagnostic output. In addition, the system appears to cover a range of issues relating to dementia diagnosis, and not just identification of the syndrome and disease. For example, multiple system modules support diagnostic reviews, management of dementia, and carer needs assessment. The primary drawback with this proposition, however, is that it does not appear to cater for co-existing pathologies. In addition, British electronic medical record disease classification codes are used to code output, which provides categorical values such as 'dementia likely' — there is no probabilistic output generated by this system. Furthermore, the diagnostic output is textual, thus the user does not benefit from the full explanation capability of a graphical model.

Another computer-based clinical decision support system is that proposed by Lindgren [164]. The system implements clinical guidelines for diagnosis of a range of cognitive diseases, including dementia. The system is driven by clinical guidelines, DSM-IV [4] (see Section 3.4), and applies if-then rules to a knowledge base to support diagnostic reasoning. The scope of the system is wide, and it covers many aspects of the differential diagnosis. For example, the tool includes screening tools for cognitive deficiencies, behavioural and psychological symptoms, functional dysfunctions, complex laboratory findings, neurological function and physical assessment.

In text-book cases of dementia, the system provides a categorisation of the types of pathologies that meet the guideline criteria, given the evidence presented. However, in atypical cases, where the system is unsure of the diagnosis, the system presents all the evidence to the physician, then the physician attempts to infer a diagnosis based on the evidence presented. This may be a potential drawback.

Other dementia decision support systems exist, including that proposed by Herrero [119]. However, this system focuses specifically on diagnosis of Alzheimer's disease, and it does not consider the other old-age dementia pathologies. Other computer-driven tools, which are not typically used in primary care clinical, are listed below.

- Mani et al. [170] and Shankle et al. [243, 241], who proposed a number of machine learning based algorithms for problems in dementia decision support. For example, in [242], Shankle et al. apply rule-based approaches and decision-trees to electronic medical records to differentiate between the onset of Alzheimer's disease (AD) and vascular dementia (VaD).

- de Figueiredo et al [65] applied a neural network to Single Photon Emission with Computed Tomography (SPECT) brain image data with the purpose of classifying individuals into: cognitively normal, demented, Alzheimer's disease or vascular dementia.
- French et al. [89] proposed artificial neural networks (ANNs) for differentiating patients with Alzheimer disease from healthy control subjects and for staging the degree of dementia.

The approaches by Mani, Shankle and de Figueiredo are propositions that produce encouraging results. However, the inputs that the models require are more aligned to specialist investigations environments that are conducted at secondary level care, therefore have negligible utility in the primary care setting. Secondly, these approaches generally focus on the Alzheimer's disease and vascular dementia, and not dementia with lewy bodies or frontotemporal dementia. The same can be said for the neural network model developed by French et al. Note, however, that the ANN results were very promising: the model correctly discriminated between the control subjects and patients with dementia in 91.1% of the cases. However, ANNs do not have have the high level of explanation transparency that is required in clinical practice.

Bayesian networks have been applied to many different domains (see Section 2.5. With regard to medicine, however, they are a natural choice [99] and have been widely used as decision support engines. A non-exclusive list includes breast cancer diagnosis [29, 36, 146], risk assessment of Mental Retardation for a particular pregnancy or infant [171], the diagnosis of muscle and nerve disease through analysis of bio-electrical signals (electromyography) [6] acute abdominal pain [263], anesthesia [11], clinical pathology [103, 138] and pulmonary embolism assisted diagnosis [166]. The popularity of BNs across various areas of medicine is due to their inherent ability to model problems characterised by uncertainty in a way that is very intuitive to medical practitioners, notwithstanding their support for reasoning through probabilistic inference [218].

Domains that are generally characterised by uncertainty, and can be modelled loosely on a 'cause-effect' basis can benefit from the maximum value proposition offered by BN. The graphical structure of the BN model loosely represents cause-effect relations, and the probability tables capture the quantitative belief about the impact that one variable has on another. Therefore, the BN paradigm offers a natural way to represent the uncertainties involved in medicine when dealing with diagnosis, prognostic prediction and treatment selection. In addition, BN formalism is not categorically mutually exclusive; rather, the output is a probability distribution across each node. This means that the user still obtains an output in situations where very little information is available or where input evidence is atypical.

3.7 Summary

This Chapter provided necessary background on dementia, the wider impact of dementia, diagnosis of dementia in clinical practice and treatment options. The review highlighted issues and barriers faced by clinicians in determining a diagnosis of dementia, specifically in primary care. A brief overview of treatment options for the pathologies considered in this research are presented. Pharmacological treatment is the treatment option of choice, however non-pharmacological approaches are sometimes used adjunct. Nevertheless, the challenge of determining a diagnosis of dementia remains.

A number of decision support systems for the general area of dementia diagnosis have been proposed. Some of the systems focused either specifically on identifying the presence of the dementia syndrome or on differentiating Alzheimer's disease from vascular dementia. Other approaches provide a very detailed offerings that encompass a wide variety of tests and investigations. Note that all these tests and investigations may not be available in primary clinical practice in the U.K. In addition, previous offerings do not appear to provide probabilistic output, and there are varying degrees of transparency in terms of output in terms of how the decision was made. This is unfortunate, as these two properties are common place in medical decision making. To that end, we propose BN for clinical decision support for dementia syndrome and dementia pathology diagnosis.

In this thesis we propose BN for dementia diagnosis. One of the central themes is concerned with how BNs are constructed. In the next Part of this thesis, we explore methods for constructing BNs. We begin in the next chapter with the first of the construction approaches, namely the hand-crafted approach.

Part II

Expert-driven Bayesian network construction

Chapter 4

Methods for expert-driven BN construction

4.1 Introduction

The benefits of using BNs for problems characterised by uncertainty are well documented [216, 218, 132]. However, BN construction is a non-trivial, time-consuming task, as it requires gathering often complex domain specific information and expert knowledge into a coherent, easy to understand form. Therefore, a structured mechanism for capturing all relevant knowledge for each of the constituent parts of a BN is required. Elicitation literature is plentiful, however it tends to focus on specific construction tasks [157].

In Chapter 2, we introduced the notion of BN construction; the purpose of this chapter is to describe in detail a range of methods to support the first of these construction approaches — the hand-crafted approach. These methods form a concise framework that can be used as practical guidance for those who wish to apply BN but are unsure of where to start. In addition, we highlight the issues that often arise during construction, including the drawbacks of the methods proposed. References to literature on specific aspects of BN construction, including judgement and elicitation, are provided throughout. Furthermore, this chapter serves to introduce the techniques used to hand-craft BN models for dementia decision support, which is the application domain that this research focuses on.

This chapter is organised in six sections. The overarching process for BN construction is described in Section 4.2, and the main construction components involved are identified. Each of these construction components, namely variables and states, identification of network structure, and numerical probabilities, are described in Section 4.3, 4.4 and 4.5, respectively. Finally, a summary is provided in Section 4.6.

4.2 Construction processes

In general, there are two roles in the hand-crafted BN construction process: firstly, a BN expert, who guides the model building task, often referred to as the *modeler*; and secondly, a domain *expert* whose role it is to supply the modeler with specialist domain knowledge. The number of BN modelers and domain experts required varies depending on the size and complexity of the problem at hand. Financial costs also play a role in dictating the number of people involved in the construction process. Despite resource constraints, the modeler's primary objective is to capture the problem solving techniques of the domain expert and translate these into a BN model. This is achieved by synthesising domain knowledge collected from various sources, and incorporating this information into the BN model in terms of variables, relations and probabilities.

Before the construction task commences, it is recommended that the modeler becomes familiar with the problem domain, and that the domain expert become familiar with the very basic concepts of BNs, if only to aid communication between them — see Section 4.5.2.2. Once this is established, model construction can proceed. A generic process model for constructing BNs is described in Section 4.2.1.

4.2.1 Generic process model

It is widely accepted that there are three important steps in constructing a BN: understanding the problem at hand and selecting the most appropriate domain variables (step 1); determining the graphical structure, that is the relationships among variables (step 2); and specifying of the numerical probabilities (step 3) — each of these steps are treated in Sections 4.3, 4.4 and 4.5 respectively. In order to ensure model accuracy, information required at each step must be acquired in a structured, methodical way. To that end, we recommend the following generic construction process, which is shown graphically in Figure 4.1. The specific construction process is defined as four phases: the initiation phase, in which the problem is investigated and understood; the development phase, in which the model is constructed; model validation, and its ability to solve the problem at hand, is determined during a number of iterated verification/validation phases; and finally, the model enters the maintenance phase, where the BN model is tuned over time.

The primary purpose of the initiation phase (step 1) is to decide what to model. In this step, the purpose of the BN model is specified, and the scope with regards to

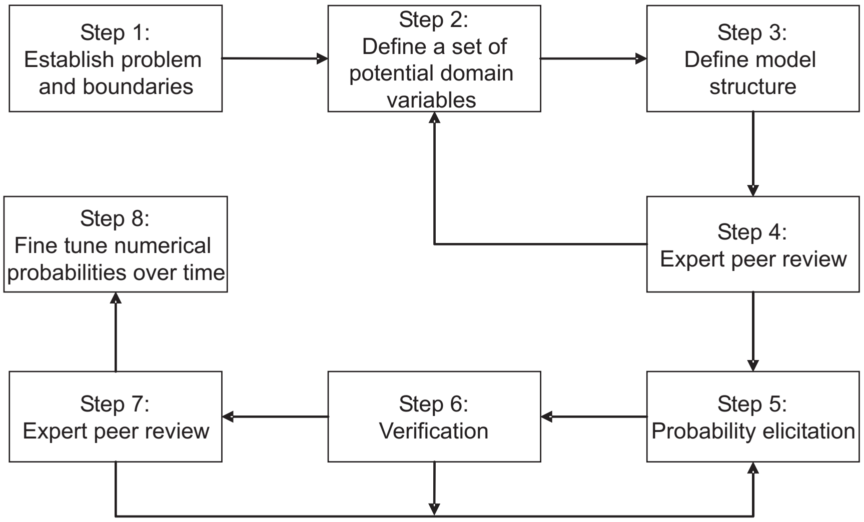


Figure 4.1: A generic Bayesian network construction process diagram.

what variables are to be included in the model is demarcated. Once the problem is defined, the construction/development phase (step 2, step 3 and step 5) can commence. Variables deemed important for the problem domain are identified in step 2 (define a set of potential domain variables). In addition, the range of any continuous variables are defined and the states of discrete variables are specified. Next, the graphical structure is identified in step 3 (define model structure). Steps 1 through 3 lead to the basic model, which, when complete, enters a phase of verification and validation. Verification and validation is carried out by the construction team during development, however a more rigorous and independent review is conducted by independent experts (step 4, expert peer review). Clearly, this is dependent on the availability of multiple domain experts who are willing to peer review the model. If agreement cannot be reached, or if discrepancies in the model are identified, the process loops back to step 2 and the process repeats until a resolution is agreed. Once the first phase of validation and verification is complete, the process enters the second phase of development, which involves numerical probability elicitation (step 5, probability elicitation), followed by a further verification phase. In step 6 (verification), the model goes through a series of user-defined scenarios to satisfy extremity and boundary tests, followed by independent expert vetting (step 7, expert peer review). By the nature of step 6 and 7, a further development phase of refinement/redefinition of the model may be required; a feedback loop allows repeated development and enhancement of the model based on feedback reviews. In the final maintenance phase (step 8), the numerical probabilities are finely tuned over time.

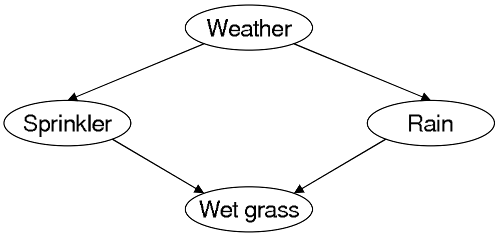
It is clear from Figure 4.1 that BN construction is an iterative process; such an approach is beneficial as it allows the model to evolve and improve. However, in our experience of constructing BNs with a domain expert, it is easy to become embroiled in increasing detail and complexity with each iteration. Hence, it is worth spending time in step 1 to clearly define the purpose, requirements and scope of the model, and actively prevent scope creep at each iteration. Internal validation/verification within the development team can assist this process, as the process ensures that the model conforms to the initial specification as per step 1, and that model accuracy is preserved. Where available, external validation and verification in the form of independent peer-review focus groups is recommended, as it encourages feedback and evaluation of the model independent of the development team. Although independent expert feedback is important, and often valued by the development team, it is ultimately the development team that makes the final decision on what changes are included or rejected.

A comprehensive discussion on alternative development process strategies, including the the spiral development model [19] and the waterfall development model [233], which are based on principles found in Software Engineering, can be found in [149, pp225–230] and [204, 222, 173, 268, 157, 169]

4.3 Defining network variables/nodes and states

The variables of a domain have a direct, one to one mapping onto nodes in a BN model. Therefore, each node can represent either a continuous variable or a discrete variable. A continuous variable is one that take any value (sometimes infinite) between any two points on a scale. For example, height, weight, and age are examples of variables that are, in general, continuous in nature. On the other hand, a discrete variable may take on only a finite number of distinct values. For example, discrete nodes in a BN might represent the outcome of a medical test result. The values taken on by a discrete variable are referred to as *states*. For example, a variable representing a medical test may have states {*pass*, *fail*}. Discrete variables are easy to model, however continuous variables present a greater challenge, particularly during probability elicitation and in inference [96] and [216, pp 345]. It is common, therefore, to transform (or discretise) those variables whose natural form is continuous into a discrete range by creating states to represent sub-ranges of the original continuous range. We use only discrete nodes in this research.

Nodes in the BN model serve different functions, and they can capture different types of information. There are three common types of nodes: first, nodes which represent an outcome, or hypothesis, which provides the user with values of interest — these nodes are the so called *query* or *target* node; second, information nodes that provide "input" to the model — the so called *evidence* node — whose state can be observed; and third, *intermediate* nodes, which summarise the effect of a subset of parent nodes on a child node.



Node	Type	Values
Weather	Nominal	{Sunny, Cloudy, Raining}
Sprinkler	Binary	{On, Off}
Rain	Boolean	{True, False}
Wet_grass	Binary	{Wet, Dry}

Figure 4.2: The simple BN model to predict whether the grass is wet, including a description of nodes and their states.

Nodes can represent a range of different data types; some common data types include: binary nodes, which take on binary values, for example absent, present; multinomial nodes that take on two or more values, for example, {*small, medium, large*}; and integral nodes, for example, a node *no_claims* might represent the a person's no claims discount, which can take on possible values 0 to 60.

An example of different node roles and different types of information that a node can capture is shown in Figure 4.2. This simple weather model attempts to predict whether the grass is wet or dry based on observations of the weather. The root node *Weather* is a multinomial evidence node, as it represent the current observable weather, and it has values {*Sunny*, *Cloudy* and *Raining*}. The intermediate nodes *Sprinkler* and *Rain*, as well as the query node *Wet_grass* are all boolean nodes, and they take on values {*On, Off*}, {*True,False*} and {*Wet,Dry*} respectively.

Practical guidance on defining the nodes and their states is provided in Section 4.3.1.

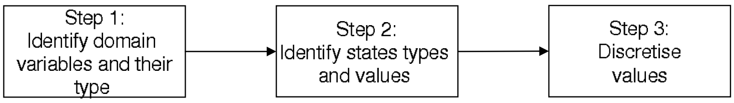

Step 1:
Identify domain variables and their type
Step 2:
Identify states types and values
Step 3:
Discretise values

Figure 4.3: The process of variable/node and state identification.

4.3.1 Guidance on defining BN nodes and their states

Defining the nodes of the BN model and the values that each state can take on is generally one of the easier tasks in the construction process [173]. Despite the apparent simplicity, however, the task should be approached with caution, as modelling errors can be introduced inadvertently. This section provides some practical guidance on variable/node and value identification as introduced in Section 4.3. A simple process flow to guide variable/node and value identification is shown in Figure 4.3; detailed descriptions of each step in the process model are given below.

Step 1: Identification of variables and their purpose

- Bayesian network construction is incremental

  It is rarely possible to elicit every single variable and corresponding values in one step. Therefore, it is recommended that the number of variables of the problem domain is limited in the early stages. Considering only a limited number of variables in the initial stages has the added benefit of focusing the elicitation team to determine only the most important "key" variables. In doing so, the most important variables, which are central to meeting the requirements of the model are identified at the outset.

- A causal interpretation, where appropriate, can be advantageous in identifying key variables. For example, identifying "query" variables first, then work backwards to identify the possible causal "input" or "evidence" variables [149, pp231].

- It should be noted that the type of node is not fixed; as the model evolves, the interpretation or meaning of a node may change depending on how it is to be used.

Step 2: Identify variable states and their values

- The states, or values that a node can take on should adequately represent the level of detail required by the model. As mentioned in Section 4.3, a number of different node data types are available. It is important to balance granularity with efficiency.

- Values of discrete variables must be exhaustive and mutually exclusive [28], which means that each variable must account for all possible cases (states) and each case must fit into one and only one state [131]. For example, consider a possible initial design decision to encode a query node labelled *Pathology* that represents four dementia causing pathologies, with states: AD, VaD, DLB and FTD (see Section 3.2). This initial modelling choice does not satisfy the exhaustive property, as it does not allow for the possibility of another pathological cause of the dementia syndrome, for example, the rare CJD pathology. This can be solved by adding a further state {other}. The addition of this fifth state ("other") does not alone fully solve the problem, though, as the mutually exclusive property is not upheld; taking these as exclusive would imply that the patient can only suffer from one of these diseases. However, evidence from clinical practice shows that co-existing pathologies are common in clinical practice, such as Alzheimer's disease with vascular dementia. From a modelling perspective, it is possible to create a single diagnostic variable with all possible pathologies (singly and in combination); however, it would lead to a complex model, which requires an unmanageable number of probabilities. A potential solution to this problem is to split the pathology node into four distinct boolean variables for each pathology of interest, for example: *AD* {present, absent}, *VaD* {present, absent}, *DLB* {present, absent}, and *FTD* {present, absent}. If the model is to account for a potential other unknown pathology, a node *Other* {present, absent} may be added to the model.

Step 3: Discretisation

- It is theoretically possible, although practically challenging, to incorporate continuous nodes in a discrete BN model. Recall from Section 4.3 that the simplest approach is to discretise the continuous variable. Discretisation can be achieved manually, however, pre-defined user discretisation intervals may result in inaccurate predictions, especially in complex models [84]. Alternatively, software tools, such as Weka [3] and Netica [2], provide automatic discretisation functions. In any case, care must be taken to ensure that information loss is minimised during the discretisation process.

It is important to appreciate the trade-off between model complexity and model accuracy. Clearly, models with many variables and many states leads to highly accurate and detailed models that describe the domain in granular detail. Such complexity, however, can lead to significant challenges during parameter assessment (further details are provided in Section 4.5). Therefore, the modeler and the domain expert should be aware of the complexity required to solve the problem at hand, and strike an appropriate balance between model complexity and model accuracy, ensuring that unnecessary complexity is omitted from the model.

4.4 Defining network structure

Once the variables of a problem domain are decided, the next stage is to define the relationships between variables. The relationships, however, must satisfy the conditions and definition of a BN, given earlier in Section 2.1, namely directed acyclic graph conditions. One approach to defining relationships is based on *causal relationship analysis*. The expert must identify variables that, when the state is changed, causes a related variable to take a particular state or prevents it from taking one of its states. The modeler can assist the expert by asking direct questions relating to causes and effect; for example: questions that begin with "What are the causes of ...?", "What is the effect of X on Y?", "Is there anything that can prevent...?" and "Are X and Y possible explanations for Z?".

An alternative approach to modelling the structure is related to the theory of BNs, introduced in Section 2.3.2.1, namely (in)dependency analysis. This is similar to the causal relationship analysis approach in that the expert is expected to specify whether a variable has an impact on the belief about another related variable. This approach seeks to explore whether variables are directly dependent or whether they are dependent only through other variables (conditionally dependent). The d-separation criterion, which is described in Section 2.3.2.1, can be used to enable this analysis. In addition, expert questioning can assist; for questions that begin with “Does knowing...” are useful

When establishing relationships between variables, the modeler and expert must be cognisant to BNs susceptibility to the *curse of dimensionality*, as mentioned in Section 4.3.1. In the context of BNs, the problem of combinatorial explosion manifests itself in the rapid growth in the number of conditional probabilities required. Assuming all nodes have only two states (binary node), as the number of parents for node X_i increase, the number of conditional probabilities increases exponentially (2^{n+1}). Table 4.1 illustrates the impact on a binary node's conditional probability table as more binary parent nodes are added. Clearly, the severity of the problem is particularly acute when the network consists of multinomial nodes. In some cases multinomial nodes cannot be avoided, as the model may require a large number of states in order to adequately represent the level of granularity required to solve the problem at hand (see step 2, Section 4.3.1).

Number of parents	*Number of parameters*
1	4
2	8
3	16
4	32
5	64
10	2048
15	65536
20	2097152
30	2147483648
40	2.19902×10^{12}
50	2.2518×10^{15}

Table 4.1: Number of conditional probabilities required for a binary node with varying numbers of binary parents

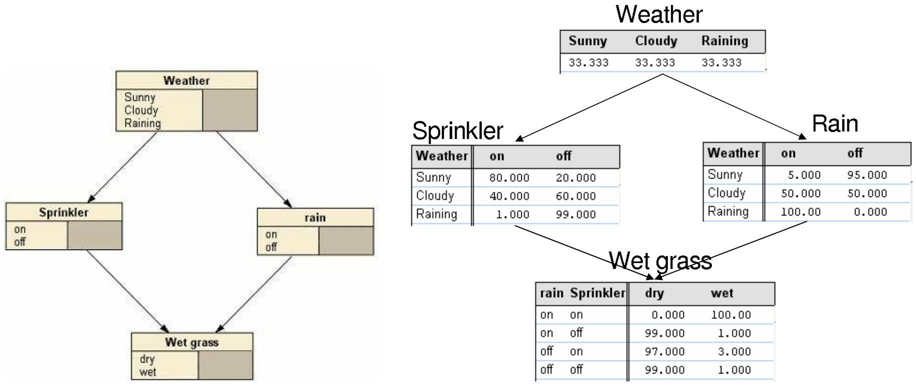
Unnecessarily large and unwieldy conditional probability tables (CPTs) present significant challenges for the domain expert (see Section 4.5.5.1). Therefore, it is desirable to keep the number of parent nodes to a minimum, which can be achieved by applying a maximum upper bound on the number of parents that a node can have. In addition, the CPTs can be reduced by selecting only the most important 'high-impact' parent nodes, at least initially [149, pp230], then expand as necessary in future revisions, is advisable. Sensitivity analysis can assist in determining high impact nodes (see Section 4.5.5.1).

Software support, such as MATILDA [20], is available to assist modelers in defining network structure as well as structure performance analysis. So far, this chapter has focused on the use of expert knowledge to develop the structure. Note, however, that the alternative approach to BN construction involves automatically inducing the structure objectively from data [51, 198, 225]. This is the second approach to BN construction, and it is given a full treatment in Part III.

4.5 Quantifying network probabilities

Finally, once the model structure is complete, the probability distributions, which describe the uncertainty of the domain, are elicited. Each probability distribution encodes the probability that a node will be in one of its states based on observations collected at other nodes. Root nodes — that is, nodes with no edges feeding in — have a marginal probability distribution. Nodes with parents — nodes with edges feeding in, the so called *child* nodes — have probability tables that are conditioned on all possible combinations of the states of the parent set. In the basic wet grass model is shown in Figure 4.4 (a), the *Weather* node is the only root node. The *Sprinkler*, *Rain* and *Wet_grass* nodes are all child nodes. The *Sprinkler* node and *Rain* node are parents of the *Wet_grass* node. Each and every child node has a probability table conditioned on all combinations of parent nodes. The underlying probability distributions for each node in the wet grass model are shown graphically in Figure 4.4 (b). For example, the probability distribution of the intermediate/query node, *Sprinkler*, is dependent on the observed node, *Weather*.

Quantification of the probabilities to specify the uncertainty of the domain is often regarded as a major obstacle in BN construction [131, pp47], and often raises the question, particularly by BN novices, "Where do the numbers come from?" [75].



(a) Basic weather BN model.

Weather

Sunny	Cloudy	Raining
33.333	33.333	33.333

Sprinkler

Weather	on	off
Sunny	80.000	20.000
Cloudy	40.000	60.000
Raining	1.000	99.000

Rain

Weather	on	off
Sunny	5.000	95.000
Cloudy	50.000	50.000
Raining	100.00	0.000

Wet grass

rain	Sprinkler	dry	wet
on	on	0.000	100.00
on	off	99.000	1.000
off	on	97.000	3.000
off	off	99.000	1.000

(b) Probabilities for weather BN model.

Figure 4.4: The wet grass BN model (a) and its probability distributions and from data (b).

On one hand, probability may be interpreted as a statistic of the relative frequency of trials in which a favourable event occurs as the number of trials approaches infinity, the so-called frequentist interpretation, or objective probability. For example, it is possible to determine the probability of a fair coin landing on heads by tossing it a large number of times. The key point here is that frequentist approach to probability elicitation requires repeated trials. On the other hand, probability may be interpreted as representing an individual's "degree of belief" in the occurrence of a particular uncertain event, a so called subjective probability. For example, a domain expert may be asked "what is the chance that it will rain tomorrow?", and reply "I believe that there is a 20% chance that it will rain tomorrow". Where the experiment that generated the data can be repeated several times, or where data is available, then frequency counts can be calculated (see Section 4.5.1). However, furnishing all BN parameters using frequentist methods is untenable in many domains, as the cost associated with completing every experiment many times usually outweighs the benefit of the model, notwithstanding the fact that that it may not be possible to repeat a specific event a sufficient number of times. Therefore, subjective probabilities, discussed in Section 4.5.2, provide a useful alternative in BN parameterisation — this is a major benefit of the BN formalism.

The purpose of this section is to answer the question "where do the numbers come from?". In doing so, three common sources of probabilistic information, which a domain expert can use to furnish the numerical probabilities of the BN model, are introduced. In situations where a data set is available, probabilities can be estimated from frequency counts, see Section 4.5.1. However, in situations where data is not available, or the probabilities are not obtainable algorithmically, probabilistic parameters can be elicited subjectively from domain experts (see Section 4.5.2 or extracted from domain literature (see Section 4.5.3). Combining information sources is described in Section 4.5.4. A generic process for probability elicitation and practical guidance on quantifying BN probabilities is provided in Section 4.5.5.

4.5.1 Data-driven parameter elicitation

Thus far, the discussion has been based on the pure hand-crafted approach, where all information is provided by the domain expert. However, in domains where data is voluminous, it is possible to calculate the parameters of the model directly from the data. Assuming data is available, parameter learning algorithms can be applied to determine the probabilistic parameters of the model. In the simplest case, automatic parameter assessment is characterised by a simple counting exercise where subsets of instances of the data that satisfy the conditions specified by the relationships in the model are enumerated to provide frequency counts[187]. In situations where data is not abundant, and necessary information can not be provided by an expert, a data collection study could be instituted in order to collect samples of data pertinent to the variables and states of the defined model.

4.5.1.1 Guidance on data-driven parameter elicitation

The availability of domain data can prove useful in calculating probabilistic parameters automatically; however, consideration must be given to the following:

1. If an existing data set is to be used, it must contain variables and values that appear in the model. Alternatively, a mapping must exist to transform the variables and values in the data set to those that feature in the model without excessive loss of information.

2. The accuracy of the probabilistic parameters obtained from the data set is dependent on the accuracy of the data collection strategy. In addition, confounding factors that relate to health-care policy may have an impact on the quality of data collected. To that end, biases in the data set will lead to incorrect parameters, which do not accurately reflect the true probability distribution, and may impact on the performance of the model.

3. The data set must be comprehensive to achieve reliable probability assessments, as insufficient samples may result in nonexistent or small probability distributions. In addition, there is a side issue closely related to reliability, which is concerned with incomplete data, and is commonly associated with real-life retrospective data sets. Druzdzel and van der Gaag [75] note that missing values, in many cases, are due to error or omission. Furthermore, the authors provide evidence that suggests that data collectors selectively choose which variables they collect data from. In new data collection studies, data collection participants should collect data from all variables in accordance with the protocol.

Numerous algorithms exist which automatically compute the required probabilities for a given BN model and a given data set. These algorithms can be categorised into algorithms that require the data set to be complete, such as [26, 253], and algorithms that have the ability to estimate probabilistic parameters when the data set is incomplete, such as [227, 56, 8, 160].

If suitable probability distributions cannot be computed using an objective mechanism, for example, if the volume of data is insufficient, or if no mapping exists to transform the variables and values in the data set to those in the model, then subjective probabilities, discussed next in Section 4.5.2, may provide an attractive alternative.

4.5.2 Expert-driven parameter elicitation

In expert-driven parameter elicitation, the required probability distributions are derived from the domain expert in the form of subjective probabilities. Such elicitation is valuable, as practical field knowledge can be obtained, which is especially useful for eliciting rare, complex or poorly understood phenomena. However, probability elicitation from experts is a difficult task to accomplish. In particular, availability and cost, as well as statistical and psychological issues associated with human expert probability elicitation, such as inconsistent [186] and incoherent probabilities probabilities and human bias, make the approach impractical [75] and [207, pp 3]. The issue of human bias is discussed further in Section 4.5.2.1, and protocols that incorporate psychological theory to prevent human bias, are described in Section 4.5.2.2.

4.5.2.1 Issues in expert-driven elicitation

Expert probability elicitation is complex and non-trivial. Experts do not maintain a list of the required probability information in their memory; rather, the expert forms a judgement using knowledge currently available in memory [207, pp 4]. Therefore, in forming their judgement, the expert tends to use information that is most recent, or information that has strongest associations with the required probability . This ad-hoc, or "rule of thumb" approach in forming probabilities is referred to as a heuristic, however, such an approach leads to bias in the expert's judgement [226].

Much of the literature on probability judgement and bias stems from the work of Tversky and Kahneman [265]; the two most common types of bias are cognitive bias and motivational bias. *Cognitive bias* manifests itself in the way in which humans process information, and it occurs when experts' estimates fail to follow normative statistical or logical rules, thus drastically skewing reliability. On the other hand, *motivational biases* reflect an individual's personal motivation, which is often driven by human needs [152]. For example, the desire for a positive self-image or social pressure. Bayesian network modelers and domain experts must be aware of, and deal with, both types of bias.

Many different types of heuristic are used during complex processes such as probability elicitation — see [136] for a comprehensive coverage of heuristics and associated biases. We draw the reader's attention to the common heuristics and associated biases.

1. *Overconfidence.* In this context, the term 'overconfident' relates to the human tendency to provide exaggerated probabilities for extreme events, that is probabilities close to 0 or 1. However, we are less likely to be overconfident about probability assessments that are mid-range [272, 275].

2. *Representativeness/Base-rate neglect.* The representativeness heuristic describes the process where the expert uses the similarity of two events to estimate the degree to which one event is representative of the other. Tversky and Kahneman [136, pp84–98] provide a classic example: A group of subjects are presented with the following description of a person whom they know to come from a population of 70 lawyers and 30 engineers: "Dick is a 30 year old man. He is married with no children. A man of high ability and high motivation, he promises to be quite successful in his field. He is well liked by his colleagues." The description provides no information with respect to Dick's profession. However, when subjects were asked to indicate the probability of Dick being an engineer, the subjects gave a median probability estimate of 0.5. Instead, the answer should have been 0.3. Clearly, the subjects ignored the base-rate and judged the description as equally representative of an engineer or a lawyer, leading to base-rate neglect bias.

3. *Anchoring and adjustment.* In several situations, people assesses probabilities by choosing an initial value (anchor), and then adjusting up or down from the anchor value [24]. For example, consider the following questions: "Is the percentage of European countries in the United Nations greater or less than 70%? What is the exact percent?". Answer: less, 60%. "Is the percentage of European countries in the United Nations greater or less than 20%? What is the exact percent?". Answer: more, 35%. The anchoring and adjustment heuristic can result in bias, as people often misjudge the increase or decrease from the anchor value, resulting in anchoring bias.

4. *Availability.* Assessing probabilities based on the ease with which occurrences can be brought to mind. The rational is that frequent events are more available in memory, and therefore an event that is easily brought to mind will yield a high probability. However, attributing higher than justifiable probabilities can lead to misleading indicators with regards to the frequency with which certain events occur [265].

Elicitation protocols, described in more detail in Section 4.5.2.2, can be of assistance in managing human bias.

4.5.2.2 Elicitation protocols

In general, cognitive biases such as those described in 4.5.2.1 can, to some extent, be suppressed by making the expert aware of their existence [188, pp158]. To that end, elicitation protocols that incorporate cognitive psychology theory have been developed. Although there is no one single protocol, as different elicitation approaches are required in different contexts, a few have gained wide spread acceptance.

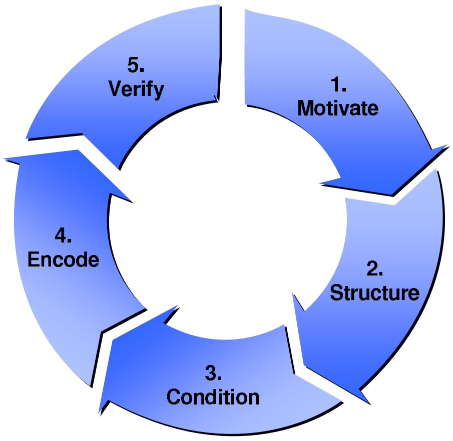


Figure 4.5: Five phase elicitation protocol - adapted from [188]

Three common protocols are the Morgan and Henrion protocol [188], the Stanford / Stanford Research Institute (SRI) [179] protocol and the Wallsten / Environmental Protection Agency (EPA) [275] protocol. The Morgan and Henrion and Stanford/SRI are composed of five phases, as shown in Figure 4.5. The Wallsten/EPA has an extra phase at the beginning, which requires the expert to read a document prior to the elicitation task, which outlines the objectives of the elicitation process and highlights the cognitive biases that can occur.

The five stages of the Morgan and Henrion process are as follows:

1. Selecting and motivating the expert: Establishing Rapport. It can be difficult to find a domain expert as they are a scarce commodity, and their time is often charged at premium rates, as alluded to in Section 4.5.2 above. However, once an expert is found, they must be introduced to the task. The purpose of this step is for the elicitor to establish rapport with the expert, and to provide the expert with the context in which their judgments will be used. A key part of this step requires the elicitor and expert to discuss

methodologies commonly used in expert judgement and probability assessment, and inform the expert of the potential problems and limitations of expert elicitation. In addition, the modeler may include an explanation of the types of judgement heuristics, and explain that the role of an elicitation protocol is to provide a structured guide with a view to obtaining good quality and accurate information.

2. Defining and structuring the uncertain quantities: In this phase, the elicitor seeks to arrive at an unambiguous definition of the quantity to be assessed by identifying the possible range of outcomes and selecting an appropriate measurement scale. In promoting reliable judgements, quantities and scales should be stated in a form with which the expert is most comfortable. For example, the choice of units should be one with which the expert is most comfortable.

3. Training/conditioning the expert: Get the expert to think about all evidence. Part of the purpose of the conditioning phase is to allow the expert to become familiar with the concept of probability assessment. In an attempt to reduce bias during the elicitation process, the expert is briefed on common heuristics and biases (described above). As part of an exercise in becoming accustomed to providing probabilities, the expert may be asked to provide probability assessments for a given domain problem for which known objective frequencies are available. Once the expert has provided all the probabilities, feedback on the true frequencies are presented. This serves as a preparatory training exercise for the expert, and gives them an opportunity to calibrate their responses in the format required.

4. Encode the probability distributions: Quantifying the Experts Judgment. Quantification of the subjective probability distributions, which best reflects the expert's beliefs about the possible range of outcomes and their likelihood for each value of each variable, is carried out in the quantification phase.

5. Verifying: Checking the consistency of the elicited distributions. Once the probabilities are provided, it is important to verify that they are coherent, that is, they all conform to the rules of probability theory (sum to 1); such checks begin during the encoding phase. Also, where possible, the probabilities should calibrate to known observed frequencies, and they should be reliable. However, it is not always possible to verify that the probabilities conform to reality; however, it is possible to test whether the probabilities are reliable. Reliability testing is concerned with determining whether the expert would provide the same estimates when asked for the same probabilities again. However, given the often large number of probabilities required for a BN, it may be impossible to ask the expert to review every probability value a second time. A more tangible reliability test may involve showing the expert probability distributions for a given variable with different conditioning contexts of the parent nodes. The expert would be asked to check whether the relationships for these different contexts are acceptable, and probability adjustments can be made for each discrepancy found.

4.5.2.3 Guidance on expert-driven parameter elicitation

Assessment of the numerical probabilities from a human expert is considered one of the more challenging approaches in quantifying BN probabilities, largely due to the psychological issues associated with human judgement and elicitation. To that end, it is recommended that an elicitation protocol, as described in Section 4.5.2.2, is employed in order to prepare the expert and thus reduce bias. Other, more general guidance on BN quantification is provided in 4.5.5.

The interested reader is directed to Chapter 7 of Morgan and Henrion [188, 141–171] for a detailed discussion on protocols and processes to support expert probability elicitation. Furthermore, an in-depth evaluation and comparison of expert probability assessment techniques applied to real-life problems can be found in Monti and Carenini[186]. Walls and Quigley [274] provide an excellent, sound framework for the elicitation of subjective judgements.

4.5.3 Parameter elicitation using domain literature

A further source of information for network quantification, in addition to data and domain experts, is domain literature. Many domains, especially medicine, have an established body of published literature in the form of peer reviewed journals and textbooks, which includes statistical information on specific domain topics. Amongst other things, statistical information, and information on the causal relationships between the domain variables, may be found in domain literature. The central problem with this approach, however, is concerned with incomplete information. For example, in medical domains, literature often reports conditional probabilities of the presence of a symptom given a disorder, that is $P(Symptom|Disease = T)$; however, the probabilities of the symptoms occurring when the disease is not present, $P(Symptom|Disease = F)$, are not always reported, and are often simply not known because of the difficulties of measuring, thus additional information from other sources may be required.

4.5.4 Combining sources of probabilistic information

When probability information from a single source is insufficient, the modeler and the expert may turn to alternative sources in order to fill the gaps. However, caution must be exercised as the characteristics of the population from which the probability information is collected can vary from source to source. Druzdzel and Díez [74] report the ramifications of combining different sources of probability information. Strict adherence to a single source of probability information, however, is not always possible in practice. For example, medicine is driven by a culture of evidence-based practice, which generates multiple sources of medical information. However, literature does not always provide all the information required — see Section 4.5.3. Therefore, the assumption of a single source of information is restrictive, as valuable information is often available in sources. Fortunately, processes for parameterising BNs, which combines information from different sources have evolved. Pollino et al. [222] and Woodberry et al. [280] provide a procedure for combining probabilistic information obtained from domain experts with probabilistic information found in data. Druzdzel and Díez [73, 74] provide criteria and guidelines for deciding when different sources of data can be combined safely, as well as methods to facilitate mixtures of different information sources.

4.5.5 Guidance on quantifying network probabilities

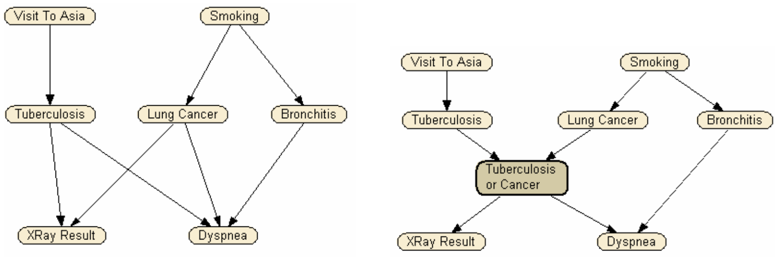
Guidance on specific sources of probability information have been addressed in respective sections above. There are, however, two, wider, practical issues that BN modelers should be aware of. These issues are concerned with the quantity of probabilities required to furnish the model, and the impact that poor quality probabilities have on the model. These two issues are treated in Section 4.5.5.1 and Section 4.5.5.2 respectively.

In addition to elicitation protocol literature, there is an plethora of literature on obtaining the actual numerical probabilities which quantify BNs. The interested reader is referred to some of the most frequently cited literature, which is practical, comprehensive and accessible, such as [75, 159, 123, 186, 226, 24, 207]. O'Hagan et al. [206] provide a comprehensive text on uncertain judgments and eliciting experts' probabilities.

4.5.5.1 Quantity of probabilities

As mentioned in Section 4.4, the number of probabilities that a node requires is exponential in the number of parent variables. For example, a binary child node with n binary parents requires 2^{n+1} probabilities. Although, in cases where a binary child node has entirely binary parents, the expert only needs to specify $\frac{2^{n+1}}{2}$ probabilities, that is one probability per case, as the remainder of the distribution for each case can be computed automatically $(1 - P(X|parents(X)))$. In the case of multinomial nodes, however, that is nodes with > 2 states, the problem becomes severe, and in some cases, leads to an intractable number of probabilities. This problem is generally regarded as a major obstacle of the BN paradigm [131, 47]. Fortunately, there are techniques for modifying the structure of the network, which prevent large and unwieldy conditional probability tables. One such method is known as divorcing, and involves separating parents and introducing intermediate variables [194]. For example, consider the well known Asia/chest clinic BN model, shown in Figure 4.6. All nodes are binary. In Figure 4.6(a), the child node *XRay* (XR) has parents *Tuberculosis* (TB) and *Lung can-*

cer (CA). Similarly, the child node *Dyspnea* (DY) shares the nodes TB and CA, with an additional parent, *Bronchitis* (BR). Therefore, the required probability distributions for XR and DY are $P(XR|TB, CA)$ and $P(DY|TB, CA, BR)$ respectively. In this example, 8 probability values are required for $P(XR|TB, CA)$, and 16 probability values are required for $P(DY|TB, CA, BR)$. In total, the model requires 40 probabilities. This number can be reduced by introducing an intermediate node, *Tuberculosis or cancer* (TBorCA), which *divorces* TB and CA from the other parent BR of DY [149], shown graphically in 4.6(b). This reduces the required number of probabilities: 4 probabilities for $P(XR|TB, CA)$ and 8 probabilities for $P(DY|TB, CA, BR)$, which is a total saving of 4 probabilities. Clearly, 4 probabilities is no overwhelming saving. However, this example serves to demonstrate the concept of divorcing, and it should be noted that larger, more impressive savings are observed in BN models that contain multinomial nodes.



(a) Asia BN before divorcing. (b) Asia BN after divorcing.

Figure 4.6: The Asia BN model before (a) and after divorceing (b).

Another method, which is tailored to the elicitation of a large number of conditional probabilities in the shortest possible time, is that of van der Gaag et al. [268]. In this approach, sensitivity analyses is performed iteratively on the model as it develops. In general, sensitivity analysis is a technique for studying the effects of systematically varying a model's parameters on its predictions [55]. In the context of BNs, sensitivity analysis provides a mechanism to survey the most influential nodes in the BN model by varying the assessment for one of the network's probabilities while keeping all other assessments fixed. The method proposed by van der Gaag et al [268] begins with eliciting approximate initial assessments from domain experts using a probability scale. Thereafter, sensitivity analysis is conducted to survey the most influential as well as redundant nodes in the model, followed by refinement of influential probabilities using conventual elicitation methods. This process is carried out incrementally until there is no improvement in the model's output. Additionally, sensitivity analysis can be used to refine the network structure, as it allows a survey of influential and redundant nodes. After consultation with an expert, redundant nodes can be removed thus reducing the number of entries in the conditional probability table.

4.5.5.2 Quality of probabilities

Discussion thus far has focused on methods to obtain the actual numerical probabilities from human experts — we have not qualified how accurate the judgements need to be. In this section, we comment on the extent to which the quality (or accuracy) of the probabilities obtained from experts influence the model, and the impact that inaccuracies have on model output.

As described in Section 4.5.5.1, sensitivity analysis can be applied to the model in order to determine the extent to which the inaccuracies in the probabilities influence the output of the model, which allows the modeler and the expert to carry out fine-tuning. However, it has been claimed by Pradhan et al. [224] that the behaviour of the network "can be highly insensitive to inaccuracies" in the quality of the majority of probabilities. This implies, therefore, that such fine-tuning is not required, thus networks can be assigned tentative, rough probability assessments and still display reasonable behaviour. Henrion et al. [116] investigated this claim empirically. In their research, random noise was injected into the numerical probabilities of a BN, whose purpose is to diagnose liver and bile diseases. The results of Henrion et al [116] concur with Pradhan et al. [224]. They observed that the BN model was insensitive to imprecision in the numerical probabilities. It is noteworthy, however, that this is an under-researched area, and it appears that the behaviour observed by Henrion et al. is problem dependent, as Coupé et al. [55] report large effects on model output in their congenital heart disease BN model when the probability distributions are varied. Since the available evidence is limited, we recommend that the techniques introduced and the guidance provided throughout Section 4.5 is used to initialise, develop and fine-tune the network probabilities. In addition, the model should be validated at regular intervals, and, if an expert is available, feedback on the model output should be sought.

4.6 Summary

The expert-driven approach to BN construction is a valuable and a useful way to incorporate domain specific knowledge into a model. It may prove difficult, however, to get a domain expert in the first place. Moreover, issues relating to human elicitation and judgement may lead to inaccuracies in the model, such as bias, and discretisation may lead to information loss around the distribution to be captured. When an expert is available, however, the limitations in defining the network structure and quantifying the network can be addressed through careful modelling and appropriate use of methods, procedures and protocols.

This chapter reviewed techniques for constructing BNs using the expert-driven approach, and it has highlighted the challenges and issues in using the handcrafted approach. The chapter reviewed literature on methods for establishing nodes and their states, expressing the structure and relationships between nodes and methods to facilitate probabilistic elicitation. The best nuggets from the literature for each of these components has been cherry-picked to form a framework to assist a non-BN expert with the construction of a BN model.

In the next chapter we demonstrate how the framework can be applied to a real-life problem, specifically construction of BN models for dementia diagnosis.

Chapter 5

Constructing BN for dementia

5.1 Introduction

This part of the thesis (Part II) is concerned with the expert-driven approach to Bayesian network (BN) construction with an application in decision support for dementia diagnosis. The expert-driven approach is described in detail in Chapter 4; dementia, and its diagnosis in clinical practice, is treated in detail in Chapter 3. In this chapter, we describe the implementation of BNs for use in dementia diagnosis decision support, and in doing so demonstrate how the expert-driven BN approach is used in a real-life problem.

The remainder of this section is concerned with providing a brief background to the case study, as well as defining the objectives and scope. The process, which facilitates building the models in this case study, is set out in Section 5.2; the actual construction tasks are described in Section 5.3. Section 5.4 and Section 5.5 deal with model structure construction and Section 5.6 describes network quantification.

5.1.1 Problem background and description

In the first instance of health concern, the affected individual or a carer usually presents to a primary care clinician — the General Practitioner (GP) — thus, the GP and the wider primary care team (PCT) are the initial point of contact. Therefore, the GPs and the wider PCT play an important role in the identification and diagnosis of dementia. However, there are a number of well documented challenges associated with dementia diagnosis at the primary care level — see Section 3.4.1.1 for an overview. To that end, we propose a decision support tool for dementia diagnosis to assist with diagnosis at the primary care level.

In this research, we propose the BN paradigm as a technique to provide decision support in dementia diagnosis at the primary care level. As mentioned in Section 3.2, dementia is a syndrome, which is characterised by an underlying disease process. Therefore, we have developed two BNs models, namely DemNet and PathNet, for dementia syndrome diagnosis and dementia pathology diagnosis, respectively.

5.1.2 Objectives

The primary objective of this case study is to develop BN models for a real-world application — dementia diagnosis decision support. Specifically, models are required to support: 1) dementia syndrome diagnosis; 2) dementia pathology diagnosis, specifically Alzheimer's disease, vascular dementia, dementia with lewy bodies and frontotemporal dementia; and 3) classification of diseases both singly and in combination. As mentioned in Section 4.1, the primary issues that a lay person faces when implementing BNs relate to elements of the BN model and how they fit together (i.e. construction) — that is, how to define the variables and the relationships among them, and in the words of Druzdzel and van der Gaag "Where Do the Numbers Come From?" [75] in order to furnish the probability tables. In this chapter, we hand-craft BN models for dementia diagnosis, demonstrating the construction process as well as the construction methods described in Chapter 4.

5.2 The model building process

The process to develop and quantify the BN models was iterative, and was undertaken at Kildean Hospital, Stirling, which was convenient for the expert. The development process stages are outlined in this section.

5.2.1 Structure building process

The part of the process responsible for building the model structure was developed through a series of workshops. The workshops consisted of a member of the research project team and our clinical lead, Dr. Richard Coles, Community Mental Health Team Elderly (CMHTE), Kildean Hospital, Stirling. Dr. Coles is a consultant in Old Age Psychiatry with many years experience in dementia diagnosis, and is considered an expert in his field for the elicitation process. The workshops were carried out in the style of an interview, where the expert was asked a number of questions relating to how dementia is diagnosed in clinical practice, and the diagnostic variables considered pertinent for use in a primary care setting.

At the outset, it was decided that the frequency of the workshops would be weekly to enable us to plan in detail the steps to achieve construction of the models. The reason that the workshops were not more frequent was due to time constraints associated with having only a single expert providing information, combined with the expert's busy clinical caseload. We set out a plan and agreed that model structure build be carried out across three phases of workshops: 1) variable identification; 2) initial structure; 3) refined structure. In addition, we agreed that the first phase be carried out over four, one hour workshops, and the remaining two phases over six one hour workshops. It was felt that one hour would is long enough to capture chunks of information and prevent expert fatigue.

In the initial planning sessions, we agreed that the three structure build phases be run sequentially, allowing an incremental development. An additional one hour workshop for independent peer review was scheduled in the middle of each phase. A team consisting of four CMHTE nurses specialising in dementia, and an Old Age Psychiatrist independent of the model build, carried out the reviews, and feedback was fed forward into subsequent workshops.

It was explained to the expert in the initial model build workshop that the aim was to develop a graphical model representing the key factors relevant for dementia diagnosis in a primary care environment. The expert was shown the well known Asia Chest Clinic BN model to demonstrate the components of a BN and what was required from him. In addition, the expert was introduced to the Netica [2] BN modelling software, which we used during the sessions to build the models. Control of Netica was the responsibility of one of the research team members, and not the clinician. However, the clinician was provided with scrap paper for rough working.

Throughout the model build, particularly phase one, the expert was asked to justify the selected variables and the links between them. The reason for this justification was to ensure that the models remained realistic and appropriate for use in a primary care environment.

The process concerning variable selection and the way in which the variables are connected for both models is described in detail in Sections 5.4 and 5.5.

5.2.2 Parameter elicitation process

As described in Section 2.1 (page 15), the BN consists of both a qualitative (structure) part and a quantitative (parameters) part. Quantification of BNs requires eliciting information to furnish the conditional probability tables for all combinations of parent nodes feeding into a child node, as well as the probability tables for nodes with no parents. This information was also provided by our clinical collaborator, Dr. Coles.

Similar to the structure specification process outlined above in Section 5.2.3, workshops were planned to elicit the probabilities for the models. More than twenty one hour workshops spread over a three month period were required to quantify the BN models. This included two one hour validation sessions, where the independent panel tested the model on a number of real life cases from clinical practice.

A two phased approach was proposed for quantification. In the first phase, the research project team demonstrated how to furnish the probability for the example Asia Chest Clinic BN model; the second phase was concerned with the expert providing the necessary probabilities. The probabilities specified by the expert were entered into the Netica [2] software tool. The details of the quantification task are given in Section 5.6.

5.2.3 Wider peer review

An initial workshops was set up in June 2004 to socialise the idea of a decision support tool for dementia diagnosis, and gain feedback from a wide clinical audience. The workshop was set up by our clinical lead, and it consisted of a team of multidisciplinary medical professionals specialising in dementia diagnosis and dementia care. The panel consisted of a Neuropsychologist, six Community Psychiatric Nurses (CPNs), an Occupational Therapist (OT), two consultants in Old Age Psychiatry and a General Practitioner (GP). The purpose of this session was to gauge the appetite and usefulness of such a decision support tool, and gain feedback to shape the development of the models. Feedback was positive and encouraging, and comments were taken on board during the model build phases.

A further workshop was set up to include potential end users, namely primary care health professionals (GPs and GP practice nurses). The purpose of this workshop was to drill deeper into the requirements of the model, and the sorts of variables that would be suitable for a primary care setting. Feedback from the primary care workshop concluded that a decision support tool for dementia diagnosis in the primary care setting would be useful, and the diagnostic variables proposed were realistic for the primary care environment. One common point raised by workshop participants was that, due to time restrictions, GPs may be unable to use the tool during consultation, but would perhaps use the system post-consultation as part of the diagnostic checks prior to a follow-up visits. Additionally, it was suggested that the tool may be more appropriate and more beneficial to other primary care clinicians during their consultations, for example, GP practice nurses.

In addition to socialising the concept with medical professionals in — once the initial models were available — the proposition was written up in the form of a paper, and presented at the Health-care Computing Conference in 2005 [213]. In addition, once the final models were more mature in the development cycle, they were presented at an annual dementia conference in 2007 [212] for wider peer review.

5.2.4 Issues with this approach

From the number of workshops required to construct these models, and the resources required for peer review of the models, not to mention hand specification of the probabilities, this approach to BN construction is extremely labour intensive. It relies on a dedicated team to construct the model, with a heavy reliance on a domain expert to provide the knowledge required to accurate model the domain.

5.3 Constructing the models

The BN construction process is composed of three steps (see Section 4.2.1), and, as depicted in Figure 4.1, the construction process is iterative. Firstly, the structure of the model is identified (see Section 5.4), which is achieved by selecting the most appropriate domain variables, defining the relationships between the variables are defined (see Section 5.5), and then the structure is validated. Thereafter, the probability distributions for each of the relationships are quantified (see Section 5.6).

With regard to construction of BN models for dementia diagnosis, the construction process begins by surveying dementia diagnosis in clinical practice, and then identifying the key diagnostic and causal variables. Thereafter, variables suitable for use in the primary care setting are selected, and then the corresponding nodes are connected as per their prescribed dependency relationships. Finally, the model is quantified. Expert knowledge regarding the pathophysiological mechanisms of dementia, including clinical experience and information drawn from medical literature (diagnostic clinical criteria), is used to guide the entire construction process.

As mentioned in Section 5.1.1, two models are proposed for dementia diagnosis: DemNet is the BN model responsible for providing decision support in diagnosing the dementia syndrome; PathNet is the BN model responsible for classifying the underlying dementia causing pathology. The decision to separate the models was motivated by the fact that the models fulfil different purposes, in addition to simplifying the construction process.

5.4 Identification of variables/Nodes and states

The purpose of this step is to identify outcome variables and causal factors in dementia syndrome diagnosis that are used in clinical practice. In particular, the focus is on those variables suitable for use in the primary care setting. With regards to the BN construction process model shown graphically in Figure 4.1, variable identification corresponds to step 2.

Variable identification process is carried out with a domain expert, and clinical guidelines are used to ensure that standard clinical criteria for dementia syndrome diagnosis is included in the model. In addition to identifying potential variables for inclusion in the BN model, the variable identification task allows the modeler to understand better the process and variables involved in dementia diagnosis.

Identification of the diagnostic variables in dementia syndrome and pathology diagnosis is guided by identifying query variables then working backwards to identify the evidence variables (input variables), as described in Section 4.3.1. In doing so, the diagnostic variables are first subdividing into 'query' (or predictor) groupings — the query groupings being the outcome variables that are useful to the user. Thereafter, the evidence variables that lead to each of the query variables are identified. In addition to identification of the domain variables, the possible states that each variable can take on are listed alongside. Nodes that are naturally continuous are discretised.

The variable identification process, carried out over a number of workshops with the domain expert, is responsible for producing a list of candidate variables and their possible states. Variables deemed suitable for use in a dementia decision support tool, particularly in the primary care setting, are selected for use in the diagnostic model.

5.4.1 DemNet variables/Nodes

In the case of DemNet, four query variables of interest are identified: current functioning (CF), global severity (GS), and the class variable, dementia (DEM). Additionally, a total of eight evidence variables are selected. The selected evidence variables and query variables translate to nodes in the BN, and are connected together in accordance with their respective dependencies to form the structure of the model. A list of the nodes selected for use in the model is provided in Table 5.1. Variables that have too many discrete states, such as age,

Node	*States*	*Description*
Current functioning nodes		
Global_PADL (PADL)	Unimpaired, Mild, Severe	Global Personal Activity of Daily Living
Global_DADL (DADL)	Unimpaired, Mild, Severe	Global Domestic Activity of Daily Living
Current_functioning (CF)	Unimpaired, Mild, Severe	Overall assessment assessment of daily functioning
Global severity nodes		
Cognitive impairment (CI)	Level 1 $(21 - 30)$ Level 2 $(11 - 20)$ Level 3 $(0 - 10)$ Level 4: undetermined	Result of cognitive (MMSE) test
Clock_drawing (CDT)	Pass, Fail	Result of clock drawing test
Subtle_functioning (SF)	Unimpaired, Possibly, Definitely	Assessment of subtle functioning
Global_severity (GS)	Unimpaired, Possibly, Definitely	Overall assessment of impairment
Clinical history nodes		
Age (AG)	Level 1 (< 65) Level 2 $(65 - 74)$ Level 3 $(75 - 84)$ Level 4 (> 84)	Age in years
Duration (DR)	Short, Medium, Long	Duration of symptoms
Clear progression (CP)	Yes, No	Steady progression of decline
Class node		
Dementia (DEM)	True, False	Class node.

Table 5.1: Definitions of DemNet's nodes and their states

are regrouped into wider ranges. The actual ranges are denoted in the "states" column by $(\cdot)$.

An individual's ability to function in daily life, which is summarised by the CF node, is characterised by their ability to carry out personal and domestic tasks. Personal activities of daily living include tasks such as dressing, eating, ambulating and hygiene. Examples of daily domestic tasks are shopping, housekeeping, finance management, food preparation and transportation. The overall global degree of impairment in personal and domestic functioning is captured by the PADL node and DADL node in the model. When impairment in CF is detected, ascertaining the reason for impairment can assist in determining the underlying cause. Acute losses, however, often signal cognitive impairment or some other underlying disease process.

The GS node represents the global severity of impairment, and is obtained by aggregating the degree of impairment in current functioning, cognition and subtle functioning. Additionally, the clock drawing test is used as a screening tool for cognitive impairment, particularly in differentiating normal aging from dementia related cognitive impairment. These ‘evidence’ nodes, which are used to predict global severity, are consistent with clinical guidelines for dementia diagnosis. Cognitive impairment is represented by the CI node; it represents a cognitive screening instrument for assessing cognitive function. In this research, the Mini-Mental Status Examination (MMSE) [87] is used as the cognitive screening test of choice, as it provides a superficial assessment of language, visuoperceptual function and memory, and it is widely used in primary care clinical practice. It should be noted that other cognitive impairment screening instruments exist [22], such as the abbreviated mental test score (AMTS) [121], General Practitioner Assessment of Cognition (GPCOG) and Mini-Cog. In general, the MMSE is scored in the range from 0 points to a maximum of 30 points, although the maximum attainable score is adjusted in some specific situations, for example, if the patient is blind. A numerically high MMSE score indicates unimpaired cognition, and a numerically low MMSE score indicates impaired cognition. In order to keep the number of parameters low, the range of the MMSE score has been condensed. The clock drawing test, which attempts to differentiate cognitively normal older adults from those with at least mild dementia, is represented by the CDT node. Note, however, that Nishiwaki et al. [199] and Powlishta et al. [223] have demonstrated empirically that the clock drawing test on its own does not appear to be effective in detecting very mild dementia related cognitive impairment. However, Brodaty [23] demonstrates that the clock drawing test, when combined with the MMSE, is useful in screening for mild dementia. The interested reader is directed to Woodford and George [281] for a comprehensive review of cognitive assessment tools. Evidence of subtle changes in cognition, which impact on global severity, are captured by the subtle functioning (SF) node. For example, progressive difficulty in balancing a cheque book, when previously this was a task carried out quickly and accurately.

Patient history variables are represented by the age (AG), duration of symptoms (DR) and clear progression in symptoms (CP) nodes, which are consistent with clinical guidelines on dementia diagnosis and clinical experience.

Other query and evidence variables were considered, such as neuroimaging and complex blood tests, however, these were not deemed appropriate for the primary care setting.

5.4.2 PathNet variables/Nodes

As noted in Section 3.2, the term dementia is accurately reserved for conditions considered irreversible. A number of different pathologies can lead to the syndrome of dementia, either singly, or in combination. Four dementia causing pathologies considered in this research, which are most commonly associated with later life, are Alzheimer's disease (AD), vascular dementia (VaD), dementia of lewy body type (DLB) and frontotemporal dementia (FTD). These outcomes are captured in the diagnostic model; a 'dementia other' (Other) variable is added to represent other pathologies that are not considered in the model.

A wide variety of risk factors are associated with each of the pathologies considered in this research. However, only the 'high-impact' risk factor variables are selected, as large numbers of evidence nodes (risk factor variables) present significant challenges when quantifying the relationships (see Section 5.5). Additionally, the network must contain only those variables deemed appropriate for use in the primary care setting. Therefore, variables relating to neuroimaging

Node	*States*	*Description*
Clinical evidence nodes (risk factors)		
Age (AG)	Level 1 (< 65) Level 2 ($65 - 74$) Level 3 ($75 - 84$) Level 4 (> 84)	Age in years
Clock_drawing test (CDT)	Pass, Fail	Result of clock drawing test
Hachinski (HI)	Unimpaired, Mild, Severe	Instrument to distinguish between AD and VaD
Memory_impairment (MI)	Impaired, Not impaired	Level of memory impairment
Psychosis (PS)	Yes, Equivocal, No	Loss of contact with reality, including delusions and hallucinations
Tremors (TR)	Present, Absent	Presence of rhythmic involuntary movements
Disease process nodes		
Alzheimer's_disease (AD)	True, False	Class disease
Vascular_dementia (VaD)	True, False	Class disease
Dementia_with_lewy bodies (DLB)	True, False	Class disease
Frontotemporal_dementia (FTD)	True, False	Class disease
Other(Other)	True, False	Class disease

Table 5.2: Definitions of PathNet's nodes and their states

and complex neuropsychiatric batteries are not included, as they are not realistic in the primary care setting. Such diagnostic variables are typically used at the secondary care level during expert-led differential diagnosis.

The query variables (AD, VaD, DLB and FTD), and the selected evidence variables, listed in in Table 5.2, translate to nodes in the BN, and are connected together in accordance with their respective dependencies to form the structure of the model.

Six evidence nodes are selected for inclusion in the PathNet BN model, namely Age (AG), Clock_drawing (CDT), Hachinski (HI), Memory_impairment (MI), Psychosis (PS) and Tremors (TR). The AG and CDT nodes are identical to those in DemNet 5.4.1. If DemNet and PathNet are being used together, it is permissible to carry over the AG and CDT values from DemNet to PathNet. However, if there is a time lapse between the use of DemNet and PathNet, then the CDT may need to be carried out again, as deterioration due to the underlying pathology may lead to changes in the outcome of the CDT score.

The Hachinski (HI) node represents the outcome of a neuropsychiatric test, namely the Hachinski ischaemic scale (HIS) [102]. The HIS test is used in this research as it is a simple clinical tool used to differentiate between dementia causing pathologies on the basis of their differing neuropsychological profiles [214] (see Section 3.4). In clinical practice, the Hachinski test is most commonly used to identify a potential mixed dementia (AD and VaD), or differentiate between VaD and AD. The test is composed of thirteen clinical finding, which attract either one or two points if the finding is present. The sum of all points provides the ischemic score. The minimum score is zero, and the maximum attainable score is 18. Scores are interpreted as follows: > 7, VaD; 4 - 7, borderline mixed; and < 4, primary degenerative dementia (AD). Since eighteen distinct states is too many to model, the range has been reclassified into the three aforementioned intervals.

Impaired connection to reality, auditory or visual hallucinations and delusions are classic ‘psychotic’ symptoms associated with dementia. However, the underlying cause of psychotic episodes should be investigated. For example, not all delusions are psychotic, as they may be explained by impaired memory, or a deluded belief of theft may be explained by impaired memory. Any evidence of psychosis is recorded by the PS node in the network.

The structure of the PathNet BN model is described and shown graphically in Section 5.5.2. below.

5.5 Building the network structure

The nodes selected for inclusion in the respective models, identified in Section 5.4.1 and Section 5.4.2, are connected using arrows to form the BN structure that depicts the relationships between them. Two techniques are available for doing this: firstly, the manual approach, which involves the use of a domain expert; and secondly, the automatic approach, which requires data and a learning algorithm. In this part (Part II), we focus on the former; a full treatment of the latter, the automatic approach, is provided in Part (III).

The BN structure building process corresponds to step 3 in the construction process model, shown graphically in Figure 4.1. In this approach, the BN structure is constructed using expert knowledge as well as domain literature. A number of workshops with the domain expert were held to facilitate this part of the construction process. The Netica [2] BN modelling software tool was used during the workshops, as it made manipulation of the models easy, and assisted in managing progressive versions of the models.

In this case study, causal relationship analysis (see Section 4.4) was used as the primary modelling concept. Information relating to the causal dependencies in dementia diagnosis was provided by the expert, and was used to provide the dependencies (edges) between the selected nodes. For each model, structure building began by constructing a basic BN model consisting of only the query nodes. Future workshops were responsible for iteratively adding/removing evidence nodes and adding/removing relationship edges until a suitable BN model was obtained (see Section 4.2.1). The expert is asked a series of direct questions in deciding the nodes and edges that should be included in the model (see Section 4.4). During each iteration of the structure building task, that is to say adding/removing nodes and edges, great care is taken to ensure a balance between complexity and cost. A complex BN, which encodes granular levels of detail through many multinomial nodes, each with many parents, although may lead to a highly accurate representation of the domain and highly accurate output, will require a large (exponential in the number of nodes) number of probabilities. This presents significant challenges if the probabilities cannot be assigned directly from a data set, as the probabilities would need to be elicited from a domain expert, which may be impossible. Techniques, however, exist to suppress this problem, such as including only high impact nodes, placing an upper bound on the number of parents and divorcing parents, as mentioned in Section 4.4 and Section 4.5.5.1).

This approach to BN construction, namely the expert-driven approach, is highly subjective. It relies on domain experts providing knowledge about the domain in order to construct the BN model. Accordingly, this technique produces a consensus BN model of the domain, given the experts involved in the construction process. Therefore, other experts may produce different BN models of the same problem. To that end, a validation phase is built into the development process to take into account to ensure that the model aligns to common and and popular expert views. In this case study, validation is concerned with determining whether the structure of the model is accurate, and whether it represents to the extent possible both accepted medical literature and necessary elements of clinical practice. Therefore, two validation steps are built into the model. The first layer of validation is carried out by the development team at frequent intervals. This layer of validation allows the development team to ensure that the model is fit for purpose and is clinically accurate. A similar, second layer of validation is carried out by experts not directly involved in the construction task. Modifications to the models are made until suitable agreement is met. This process of validation and modification can be seen in the process model in Figure 4.1, step

4; the arrows feeding into and away from step 4 depict cycles of validation and amendment.

The two BN models developed in this research, DemNet and PathNet, are described and shown graphically in Section 5.5.1 and Section 5.5.2, respectively.

5.5.1 DemNet structure

DemNet is a BN model developed to provide decision support for dementia syndrome diagnosis. By incorporating relevant clinical features, DemNet infers the posterior probability of dementia; the model is shown graphically in Figure 5.1.

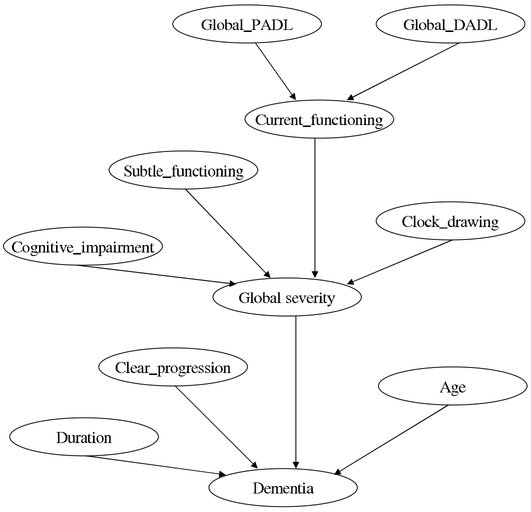


Figure 5.1: DemNet: Dementia syndrome BN

The BN network contains a total of 11 nodes; two of the nodes, namely CF and GS, are query nodes, that is nodes whose output the user is interested in predicting, and one node is the hypothesis node, that is the class node, Dementia. The remainder of the nodes are evidence nodes; they collect observable clinical evidence/information relating to the dementia syndrome.

Current functioning (CF) is a both a query and intermediate node; it cannot be directly observed or measured, and its outcome contributes to assessment of global severity of impairment (GS node in the model). The CF node is influenced by an individual's global ability to function in activities of daily life, in particular, ability to function in personal and domestic activities of daily living [165].

The node GS represents the global severity of dementia; it functions in the same way to the CF node in that it cannot be directly observed or measured, and its output provides useful information to the end user. Global severity is characterised by impairment in cognition (CI and CDT nodes), deterioration in subtle functioning (SF node) and impairment in current functioning (CF). The presence of a CI node facilitates differentiation between normal aging and a cognitive impairment due to a dementia [12].

The hypothesis node (or class node) is the Dementia node, which, in our model, represents the likelihood of the dementia syndrome based on clinical history findings and information relating to global severity of impairment [195, pp 3–4].

5.5.2 PathNet structure

PathNet is a BN model developed to provide decision support in classifying the underlying dementia causing pathology. Currently, there is no laboratory test for the common age-related dementia pathologies pre-death [213]. Therefore the utility of this model is found in the fact that it provides non-specialist medical decision makers with a tool to reason about the underlying cause of the dementia syndrome. By incorporating relevant clinical features, PathNet infers the posterior probability of all four pathologies, based on available evidence; the model is shown graphically in Figure 5.2.

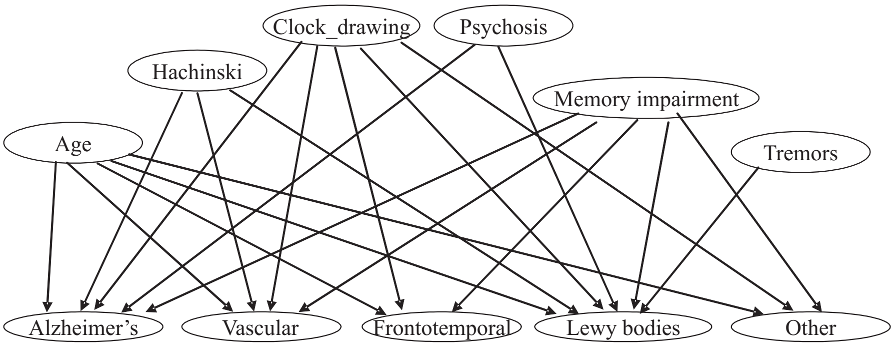


Figure 5.2: PathNet: Dementia pathology BN

The PathNet BN network contains a total of 11 nodes. Each of the disease processes considered in this research are represented by a single outcome/hypothesis, binary node (AD, VaD, DLB, FTD and Other respectively). The remainder of the nodes are clinical evidence nodes, which collect observable clinical evidence/information which helps infer the most likely underlying disease process.

A description of the nodes featured in the PathNet BN model can be found in Table 5.2.

5.6 Quantifying network probabilities

Parameterisation of a BN is concerned with quantifying the uncertainty of the domaim, which is achieved by specifying probability distributions for each of the dependency relations encoded in the BN structure. In Section 4.5 we describe two approaches to facilitate probability assessment: subjective probability assessment, which requires a human, usually a domain expert, to provide subjective 'belief' estimates for the required probability distributions; and objective probability assessment, where the probabilities are derived from repeated observable experiments. The subjective approach is adopted in this research. The primary reason for this lies with the fact that no objective clinical data was available relating to the probability distributions encoded in our BN models. Therefore all the probability distributions in both DemNet and PathNet are subjective. It is the focus of this section to present the subjective probability elicitation process used in this research to quantify both DemNet and PathNet.

5.6.1 The quantification process

The process of quantifying probabilities to furnish the parameters of a BN structure is widely accepted as a challenging aspect of the overall BN construction process, especially when the mode of elicitation is subjective. Three common challenges associated with subjective elicitation, which present significant challenges are: economic factors relating to availability of a suitable domain expert, as well as cost; technical factors, including the psychology of probability elicitation and judgement (see Section 4.5.2.1); and, issues associated with subjective elicitation when the number of probabilities required are large (see Section 4.5.5.1). Regardless of these challenges, the subjective approach is of use, as it facilitates network quantification when the domain is complex and little understood, or when objective data is unobtainable. While the cost associated with hiring a domain expert is a matter concerning the project management team, techniques should be employed to the extent possible to manage issues with regards to the psychology of probabilities elicitation, such as bias, as well as techniques to minimise the number of probabilities required (see Section 4.5.2.3).

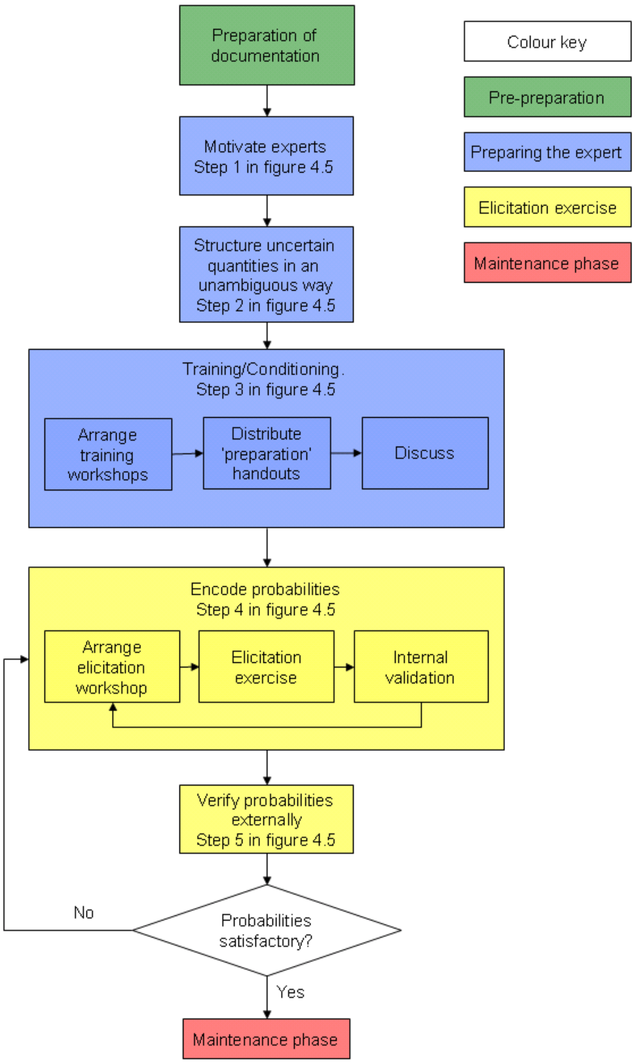
Throughout the development of DemNet and PathNet BN models, the team remained mindful of the aforementioned challenges. Guidance provided throughout Chapter 4 is used in addressing the challenges as they arise. From a structural perspective, an upper bound on the number of parents was imposed to reduce the number of probabilities required — see Section 4.4). To further reduce the number of probabilities, only those 'high-impact' parents were chosen, and node divorcing was used where appropriate — see Section 4.5.5.1. In addition, the number of states defined in multinomial nodes was scrutinised to ensure that as few as possible states were used, whilst retaining a number of states sufficient to provide the level of granularity required. As a result, DemNet requires 395 probabilities, and PathNet requires 526 probabilities; therefore the expert is required to specify a total of 921 probabilities.

From a quantification perspective, with regards to subjectively eliciting the probabilities required, a protocol is used to assist in suppressing biased judgements, as advocated in Section 4.5.2.3. The quantification protocol adopted in this research, and the actual quantification task, is described below in Section 5.6.2 and Section 5.6.3, respectively.

5.6.2 The quantification protocol

In this research, the protocol developed to facilitate probability quantification is based upon the five phase SRI protocol, described in Section 4.5.2.2 and shown graphically in Figure 4.5. However, the protocol has been refined such that it fits better with the needs of our elicitation requirements; it consists of four phases, each phase contains one or more steps. A description of the protocol, as well as a corresponding graphical process flow, is provided below.

Phase 1: Document preparation In this phase, presentations and accompanying documentation is prepared with a view to providing the expert with



Preparation of documentation
Motivate experts Step 1 in figure 4.5
Structure uncertain quantities in an unambiguous way Step 2 in figure 4.5
Training/Conditioning. Step 3 in figure 4.5
Arrange training workshops
Distribute 'preparation' handouts
Discuss
Encode probabilities Step 4 in figure 4.5
Arrange elicitation workshop
Elicitation exercise
Internal validation
Verify probabilities externally Step 5 in figure 4.5
Probabilities satisfactory?
No
Yes
Maintenance phase
Colour key
Pre-preparation
Preparing the expert
Elicitation exercise
Maintenance phase

Figure 5.3: Elicitation protocol used to quantify DemNet and PathNet

background information on elicitation, as well as information relating to the issues that must be considered prior to the elicitation task. The documentation included a description of the motivation for the elicitation exercise, a description of the quantification task, and an explanation of what is actually is required of the expert. Additionally, information relating to common judgement heuristics is prepared — this is used to explicitly make the expert aware of biases and guidance on how to overcome them. Bias management is discussed in more detail in

Section 5.6.3.4. The documents prepared in this phase form the basis of step 2, below.

Phase 2: Preparing the expert Phase 2 is concerned with establishing rapport with the expert/elicitation team and ensuring that the expert is adequately briefed on the elicitation task. The documentation prepared in phase 1 (above) is distributed. At this stage, it is important to motivate the expert to ensure buy-in [274], which is achieved by making clear to the expert the purpose of the elicitation task, why their judgements are required, and how the information that they provide is used. It should be noted, however, that our domain expert was committed to providing the necessary information, and, having participated in previous similar research, was aware of the intrinsic value of the role as expert. Following from the motivation step, the expert is required to provide unambiguous quantities to furnish the probability tables. In this research, the expert is clear on the quantities required — it is the same expert who provided the qualitative structure. The quantities required are prescribed by the conditional probability tables dictated by the structure of BN models developed. The final step in 'preparing the expert' is a training workshop prior to the elicitation exercise. The purpose of this workshop is to introduce the format of the elicitation task, distribute documentation on common heuristics and potential biases, and provide the expert with an opportunity to become familiar with the concept of probability assessment through participation in a sample elicitation tasks. In doing so, the expert is asked to provide probability assessments for a sample problem for which objective measures are available, and feedback is provided to allow the expert to calibrate their responses.

Phase 3: The elicitation exercise It is in this phase, which consists of three steps, that the iterative process of eliciting the actual quantities is carried out. An iteration begins by arranging a time and place to conduct the elicitation workshop. In addition, the purpose and scope of the workshop is defined, that is to say the quantities to be elicited are set out. Once the preparation is complete, the workshop can begin with the aim of eliciting subjective probability distributions that best reflect the expert's beliefs about the possible range of outcomes for each variable. Techniques to encode judgements as probabilities are discussed in Section 5.6.3. A BN development tool, namely Netica [2], is proposed so that the probabilities can be entered directly into the model. Once the workshop has taken place, a process of internal validation is initiated — the expert is expected to test the model and amend simple discrepancies as they arise. More complex discrepancies are added to the "issues log", and are reviewed at the next elicitation workshop. Once a complete version of the model is available, a verification workshop consisting of experts independent of the development team is set up to test the output of the model on a set of well-known scenarios. If agreement cannot be reached, or, if discrepancies in the model are identified, a feedback loop allows repeated development until a resolution is agreed.

Phase 4: Maintenance phase The final phase is concerned with fine-tuning the probabilities over time.

5.6.3 Carrying out the quantification exercise

Thus far discussion has focused on the overall process for quantifying the BNs; the following sections are concerned with methods for encoding judgements as probabilities. Common judgement methods are introduced in Section 5.6.3.2; the approach adopted in this research is described in Section 5.6.3.3, and techniques used to detect and manage bias are described in Section 5.6.3.4.

5.6.3.1 Quantification methods

There are many methods to support expression of probabilities from experts. A plethora of methods is provided in [272, 49, 188, 206]. The most common methods, however, can be classified into one of two categories, namely *direct* methods and *indirect* methods. With direct methods, the expert is simply asked to directly express their degree of belief in the occurrence of an event numerically. This may be a probability, a frequency or an odds ratio. Cooke [49] and van der Gaag et al. [269] note that people find words easier to express than numbers when expressing probabilities, as numerical probabilities can induce discomfort and resistance among experts not used to it [272], and, in addition, such direct methods can lead to bias — see Section 4.5.2.1. To that end, a number of indirect elicitation methods have emerged. With these methods, the expert is asked not for a direct value, rather, the expert is asked for a decision from which their degree of belief is inferred, thus alleviating the requirement to provide probabilities explicitly [188]. Examples of such methods include probability scales, frequency formats, gamble methods and probability wheels. See [188, 97] for further details.

5.6.3.2 Quantification method adopted

During elicitation preparation discussions with our expert, it was decided that direct probability elicitation would be used, as the expert had previous experience in doing so. However, since some time had lapsed since his last elicitation task, it was felt that assistance in specifying the required probabilities was required. One suitably attractive method, which was adopted in our research, is that developed by van der Gaag et al. [269]. In their method, elicitation from experts involves transcribing probabilities as fragments of text and using a scale with both numerical and verbal anchors for marking assessments. Experts are asked to unambiguously mark the scale with their assessment for a particular event. The intuition is found in the fact that the scale allows the experts to specify assessments in terms of visual proportions rather than in terms of precise numbers. Fragments of text are used to explain the the desired event for which a probability is required; these fragments are displayed adjacent to the probability scale. To assist the expert with small, rare events, the text fragments are expressed in terms of likelihood, rather than frequencies. The expert is required to indicate assessments for all conditional probabilities pertaining to a single variable given a single conditioning context on the same scale.

Once the probabilities are available, they are entered directly into the model maintained by the BN development software tool. The software tool used, Netica [2], efficiently displays conditional probability tables that describe compactly the probability distribution over the states of a selected node. This is useful, as it allows the expert to obtain a snap-shot view of the distribution for all possible events. In addition, the expert can use the conditional probability table to view previously supplied values and compare them with different events.

5.6.3.3 The quantification exercise

Complete with an elicitation protocol (Section 5.6.2) and a method for eliciting the required probabilities (Section 5.6.3.2), and, having discussed the orientation documentation with the expert, the elicitation exercise could proceed.

The elicitation workshops were conducted at the expert's site as per prior arrangement, and were conducted as informal, face-to-face interviews.

In the first and second elicitation workshops, the initial work-up to the elicitation exercise took place, that is to say 'preparing the expert', as described in Section 5.6.2. In addition, a plan was set out regarding the order or elicitation for each component. It was decided that DemNet would be quantified first, followed by PathNet, as the structure of DemNet was complete first. With regards to DemNet, it was decided that the order of elicitation be: evidence nodes, current functioning, global severity, followed by the outcome node, dementia. Evidence nodes were chosen first as they are not conditioned, thus they provide the expert with a simple warm-up in providing the required probabilities. The remainder of the order was dictated by the sequential dependencies that exist among the nodes. In the case of PathNet, all evidence node were quantified first, followed by the outcome nodes.

Once the expert was prepared and the implementation plan was complete, the first of the probability capture workshops took place. Each workshop begins with a brief overview of the purpose and scope of the particular workshop, followed by the elicitation exercise. Workshops were scheduled to last no longer than one hour so as to minimise expert fatigue.

5.6.3.4 Managing bias

Various factors in the elicitation process can give rise to bias, and, therefore, result in ill-conceived and inaccurate probabilities. To that end, potential biases in the techniques being used should be identified to the extent possible [226], although some bias occurs only during the elicitation task. We controlled bias in each of our elicitation tasks by using the protocol Meyer and Booker [180].

- Anticipate likely biases in the elicitation process defined
- Modify the elicitation process to mitigate anticipated biases
- Familiarise the expert with common elicitation processes and associated biases
- Pro-actively monitor the elicitation process for biases
- Deal with biases as they emerge

The most common types of bias are described in Section 4.5.2.1; what is relevant to this research, however, is that both cognitive and motivational bias are present.

In our research, the main cognitive bias requiring monitoring and countering is anchoring bias. Anchoring bias can be demonstrated as follows: When someone is asked to provide an estimate for a quantity or assess an uncertain event, they often start with an 'initial estimate', and adjust up or down. Unfortunately, in many cases the expert remains too close to the initial value, therefore not adjusting sufficiently to reflect the uncertainty. This was tackled using evidence provided by Ferrell [85], in which it is demonstrated that making an expert aware of such issues can encourage them to reconsider the judgement. In doing so, the expert was asked occasionally to describe how other experts might disagree with their response, thus forcing the expert to consider if further adjustment up or down from the anchor point was required. Furthermore, this type of bias is more likely to occur when the expert relies too heavily on historical data and so ineffectively summarises expected frequencies of the events" [274]. Recognising that the expert had participated in previous research requiring probability elicitation, the expert was asked to discount recent events in which similar information was required.

In addition to anchoring bias, motivational bias, which is driven by human needs, must also be addressed. At the outset, the expert expressed concern in providing subjective judgements for the BN models. However, it was made clear to the expert that the aim of the elicitation exercise is to obtain subjective degrees of belief about certain events, and was reassured that that initial responses may differ between experts.

5.7 Summary

This chapter demonstrated how the procedures, methods and tools for hand-crafted BN, presented in Chapter 4, can be used to successfully hand-craft BN models for a real-life problem in health decision support. The process and techniques used to define the structure and quantify the probabilities, in addition to the methods used to address the complexities inherent in expert elicitation, are described in detail. However, it is clear that this approach is extremely time consuming and labour intensive, and therefore requires a significant amount of effort, even for simplistic models where the number of nodes is less than ten.

The following chapter evaluates the capability of the dementia BN models developed in this chapter.

Chapter 6

Results and evaluation

The purpose of this chapter is to assess the accuracy of the hand crafted Bayesian network (BN) models developed in Chapter 5 for predicting the posterior probability of the dementia syndrome and underlying pathology. In Section 6.1 we define the experimental design — the data set is described in Section 6.1.1, and the methodology and performance measures are detailed in Section 6.1.2 and 6.1.3 respectively. The experimental results are presented in Section 6.2, with a discussion of the results and implications in Section 6.3. Finally in Section 6.4, we summarise the chapter.

6.1 Experimental design

6.1.1 Description of the dementia data set used

Since there was no single data set in existence that contained patient information relating to all the variables considered in our models (see Section 5.4.1 and 5.4.2), we initiated a data collection study via our medical collaborator, Dr. Richard

Coles. A clinical protocol detailing the data requirements was developed, and the necessary governance process was followed. Local Research Ethics Council (LREC) approval was granted. Community Psychiatric Nurses (CPNs) from CMHTE agreed to collect the data, as it aligned with the diagnostic variables that they recorded during initial assessment of patients where dementia is suspected. Each completed record consisted of the CPNs initial assessment, as well as the actual diagnosis provided by a CMHTE diagnosing physician.

From start to finish, the data collection study lasted one year, and obtained 164 patient records from 'live' clinical practice. We split the data into two sets: one set containing only those features relevant to dementia syndrome diagnosis; and a second set containing features relevant to pathology diagnosis. The rational behind the partition comes from the fact that two models have been developed for simplicity: a model for dementia syndrome diagnosis (DemNet) and a model for dementia pathology diagnosis (PathNet).

The DemNet data set contains a single continuous variable, namely 'AGE'. In addition, a number of discrete variables were deemed to have too many discrete categories, for example 'MMSE' (see Section 5.4.1), which has 30 categories. Our medical collaborator, based on his clinical opinion, provided a discretisation for 'AGE', and a set of refined groupings for those variables with too many categories.

In the case of DemNet, the classification variable is 'DEMENTIA'; its role is simply to provide distribution over the presence and absence of the dementia syndrome. With regard to PathNet, there are five binary variables, one for each of the four possible pathologies, and a single catch all node, 'other', for a case that does not fit into any of the other pathologies.

6.1.2 Experimental methodology

The BN software tool, Netica [2], is used to model the BNs constructed by the domain expert in Chapter 5. Netica case files are generated from the partitioned clinical data, one file for DemNet and one file for PathNet. Together, the Netica and data sets are used to evaluate the classification accuracy of each model.

6.1.3 Performance measures

Classification accuracy is the primary metric used to assess the performance of the diagnostic models. We define classification accuracy as the proportion of test cases “correctly diagnosed”, where “correctly diagnosed” means that the predicted classification of the model matches the actual outcome in the clinical data set. To measure the classification accuracy, we employ four standard test metrics, namely sensitivity, specificity, predictive positive value and predictive negative value. To probe deeper into each model’s performance accuracy, we use a performance metric commonly used in medicine to evaluate the operating characteristics of a particular diagnostic tool [285], namely the Receiver Operator Characteristic Curve (ROC).

6.1.3.1 Performance measures

A description of each measure is listed below.

- Sensitivity: The ability to correctly predict a positive case — also known as the True Positive Rate (TPR). The TPR is the probability that a positive case is correctly classified, and is defined as: $\frac{TP}{TP+FN}$, where TP is the number of true positives and FN is the number of false negatives. Conversely, the False Positive Rate (FPR) is the proportion of cases predicted as positive when the actual outcome is negative, and is defined as $1 - \text{Specificity} = \frac{FN}{TP+FN}$.

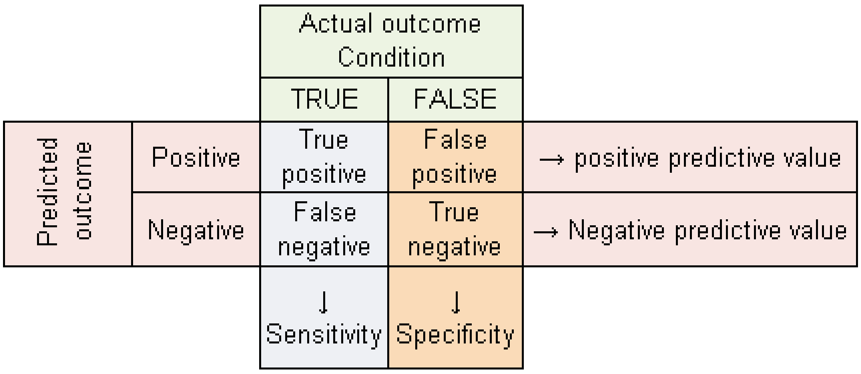
- Specificity: The ability to correctly identify those cases who do not have the disease, otherwise known as the True Negative Rate (TNR). The TNR is defined as: $\frac{TN}{TN+FP}$, where TN represents the number of true negatives and FP the number of false positives.

- Predictive positive value ($PPV+$): Of all the cases predicted as positive, the proportion of cases that were actually positive. It is defined as: $\frac{TP}{TP+FP}$.

- Predictive negative value ($PNV-$): Of all the cases predicted as negative, the proportion of cases that were actually negative. This is defined as: $\frac{TN}{TN+FN}$.

Together, these metrics form a visual representation of cases where the model is confusing two classes, and can be represented as a confusion matrix. An example of a confusion matrix is shown in Figure 6.1.

It is worth noting that a cutoff or threshold needs to be defined so that the performance testing algorithm knows the bounds to use to classify a case as syndrome/disease positive or syndrome/disease negative. For example, a cutoff probability set to 0.0 would result in the model classifying all samples in the data set as being dementia positive. On the other hand, a cutoff of 1.0 would result in classifying all samples as dementia negative. Clearly, these extremes are inappropriate. In our study, each model's class label is assigned based on the maximum likelihood state (the one with highest belief). Using this criterion, if



		Actual outcome Condition		
		TRUE	FALSE	
Predicted outcome	Positive	True positive	False positive	→ positive predictive value
	Negative	False negative	True negative	→ Negative predictive value
		↓ Sensitivity	↓ Specificity	

Figure 6.1: Confusion matrix visualisation. Reproduced from [277].

the model predicts dementia with a probability > 0.5 for a sample case, x, then the model classifies x as dementia present, otherwise negative.

6.1.3.2 Receiver Operating Characteristic (ROC) curve

The Receiver Operating Characteristic (ROC) curve is commonly used in medicine as a mechanism to evaluate the diagnostic performance of classification tools [285]. One of its benefits is that it depicts the accuracy of the predictions made by a model in a visual manner [127, pp 185]. Using values taken from the confusion matrix, the ROC graph plots the sensitivity (TPR) and $1 - specificity$ (FPR) over a range of cutoff/threshold probability values (in the range 0.0 - 1.0). In addition to the curve itself, the Area Under the Curve (AUC) is an important performance metric. Understanding the meaning of the AUC value is important in interpreting the impact of the performance results, therefore we offer an explanation of the AUC and how its value relates to the to the classification accuracy of the model under scrutiny.

The AUC is a measure of the probability that a model will correctly distinguish between two observations, one positive and the other negative [215]. In other words, a randomly selected case from the positive diagnosis group has a predicted value larger than that for a randomly chosen individual from the negative diagnosis group in $x\%$ of the time [285]. For example, an $AUC = 0.5$ indicates that the discrimination ability of a model is equivalent to one where positive and negative cases are classified randomly (a random classifier) [215]. The ROC curve for a model that performs random classification is depicted by the 45° line through the ROC graph. As the discrimination capability increases, the AUC value increases towards a maximum of 1.0 — that is to say perfect discrimination. Visually, as the discrimination accuracy increases, the ROC curve tends towards the upper left hand corner of the graph; conversely, an ROC curve under the 45° line indicates a poor classification performance, although such a model can be inverted so that better performance is achieved.

Pearce and Ferrier [215] offer a generic, qualitative translation of AUC values in the range $0.5 - 1.0$. A model achieving $AUC > 0.90$ may be interpreted as having 'excellent' discrimination characteristics, as the sensitivity values are high relative to the false positive values. AUC values in the range $0.7 - 0.9$ indicate a 'good' level of discrimination. Values in the range $0.5 - 0.7$ indicate 'poor' to 'marginal' discrimination ability, as the sensitivity rate is comparable with the false positive rate. Clearly, this is only a guide; the interpretation of the AUC value may change from application to application.

6.2 Experimental results and model evaluation

Using as input the clinical data set described in Section 6.1.1, we invoke the 'test with cases' function provided by the Netica [2] BN modelling software to test classification performance. The results for the two models — DemNet and PathNet — are presented in Section 6.2.1 and Section 6.2.2 respectively. A statistical analysis tool, SPSS [257], is used to calculate the AUC value for the predicted-actual pairs dictated by the confusion matrix.

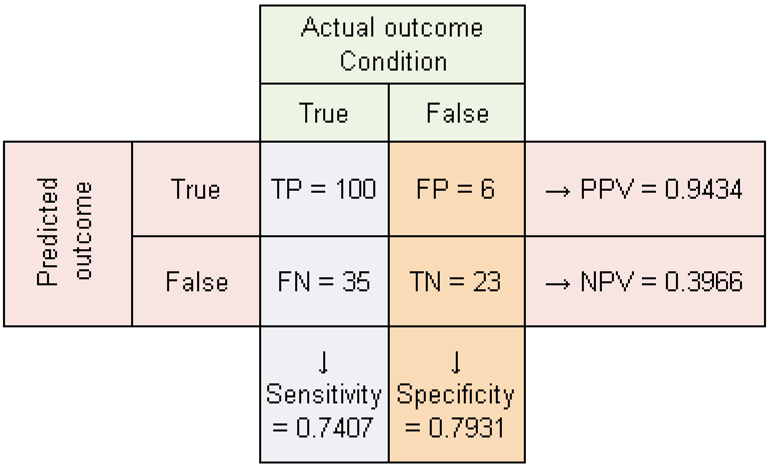
6.2.1 DemNet predictive accuracy

We provide classification accuracy results for the dementia syndrome diagnostic model. In Figure 6.2, we show the full confusion matrix for a cutoff of 0.5 (see Section 6.1.3). Table 6.1 shows the model's ability to distinguish positive and negative cases over a range of cutoffs, specifically those cutoffs in the range $0.0 - 1.0$ that produce unique sensitivity and specificity. The values listed in Table 6.1 are summarised in the ROC curve shown in Figure 6.3, and Area Under the Curve (AUC) statistics are listed in Table 6.1.

As can be seen from the results shown in Table 6.2, DemNet, according to Pearce and Ferrier [215], has a 'good' classification ability as it has an AUC value of 0.764.

Cutoff	Sensitivity	Specificity	$PV+$	$PV-$
0.0	1.0	0.0	0.8232	1.0
0.3	0.9333	0.5517	0.9065	0.6400
0.5	0.7407	0.7931	0.9434	0.3966
0.6	0.5630	0.7931	0.9268	0.2805
0.7	0.3630	0.8966	0.9423	0.2321
0.85	0.2815	0.9310	0.9500	0.2177
0.9	0.0	1.0	1.0	0.1768
1.0	0.0	1.0	1.0	0.1768

Table 6.1: DemNet performance: summary confusion matrix over cutoffs in range $0.0 - 1.0$.



		Actual outcome Condition		
		True	False	
Predicted outcome	True	TP = 100	FP = 6	→ PPV = 0.9434
	False	FN = 35	TN = 23	→ NPV = 0.3966
		↓ Sensitivity = 0.7407	↓ Specificity = 0.7931	

Figure 6.2: DemNet performance: confusion matrix for cut off = 0.5.

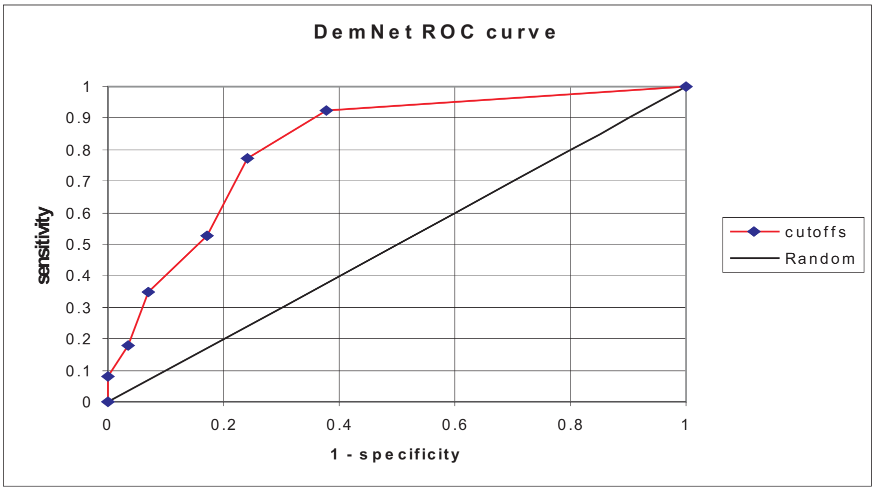


Figure 6.3: DemNet performance: ROC curve.

Measure	Value
Cutoff	0.5
Sensitivity	74.07
1−specificity	0.207
Area	0.764
Significance	< 0.001
95%CI upper	0.665
95%CI lower	0.864

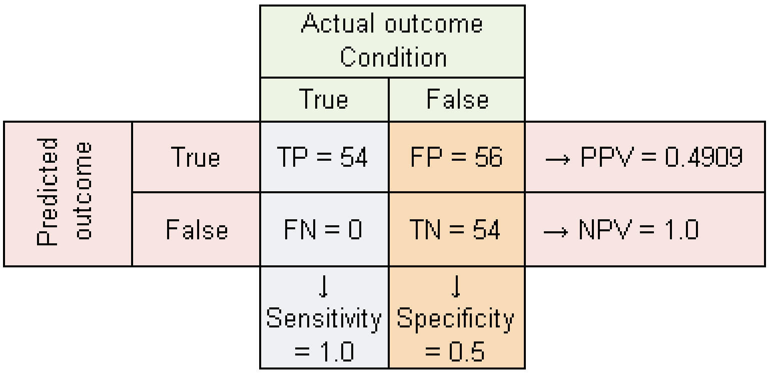
Table 6.2: DemNet performance: Area Under the Curve (AUC) statistics.

6.2.2 PathNet predictive accuracy

Recall from Section 5.1.2 two of the objectives defined for the dementia disease model: 1) diagnostic model must classify Alzheimer's disease, vascular dementia, dementia with lewy bodies, frontotemporal dementia and 'other'; and 2) model must be capable of diagnosing co-existing pathologies. We have implemented a BN model that classifies each of the aforementioned disease processes, and the structure of the model permits classification of co-existing disease processes. Accordingly, we have met both objectives. However, due to a lack of clinical data, we are unable to test the accuracy of the model on all classification tasks, namely frontotemporal dementia, dementia with lewy bodies and co-existing pathologies.

In Sections 6.2.2.1 — 6.2.2.3, we provide the performance results of the model on classifying Alzheimer's disease, vascular dementia and 'other' dementia pathologies. For each pathology, we show the full confusion matrix for a cutoff of 0.5 (see Section 6.1.3), as well as the model's ability to distinguish positive and negative cases over cutoffs in the range $0.0 - 1.0$, specifically those cutoffs that produce unique sensitivity and specificity values . In addition, we show corresponding ROC curves and Area Under the Curve (AUC) statistics.

6.2.2.1 Alzheimer's disease



		Actual outcome Condition True	Actual outcome Condition False	
Predicted outcome	True	TP = 54	FP = 56	→ PPV = 0.4909
	False	FN = 0	TN = 54	→ NPV = 1.0
		↓ Sensitivity = 1.0	↓ Specificity = 0.5	

Figure 6.4: PathNet performance: Alzheimer's disease confusion matrix at cutoff = 0.5.

Cutoff	Sensitivity	Specificity	$PV+$	$PV-$
0.0	1.0	0.0	0.3293	1.0
0.5	1.0	0.5	0.4909	1.0
0.7	0.8519	0.6818	0.5679	0.9036
0.8	0.6667	0.7727	0.5902	0.8252
0.85	0.3519	0.9091	0.6552	0.7407
0.9	0.0	1.0	1.0	0.6707
1.0	0.0	1.0	1.0	0.6707

Table 6.3: PathNet performance: Alzheimer's disease — summary confusion matrix for unique cutoffs in range $0.0 - 1.0$.

Measure	Value
Cutoff	0.5
Sensitivity	1.0
1−specificity	0.509
Area	0.745
Significance	< 0.001
95%CI upper	0.673
95%CI lower	0.818

Table 6.4: PathNet performance: Alzheimer's disease Area Under the Curve (AUC) statistics.

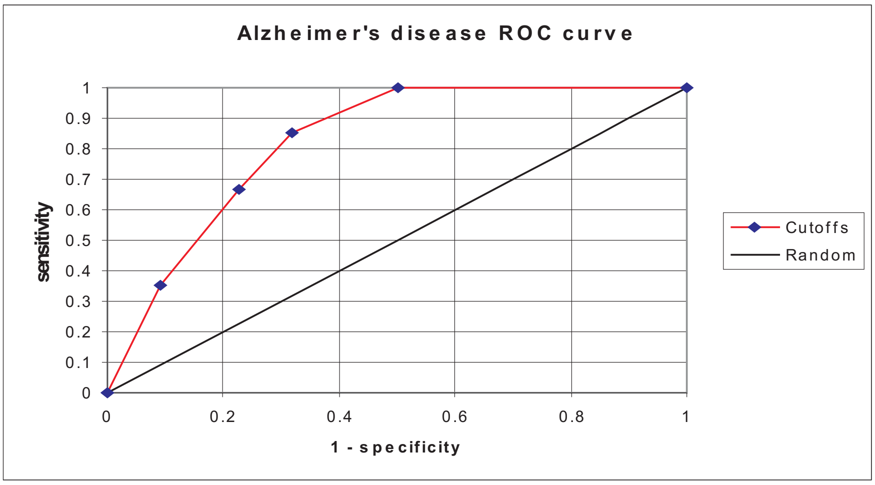
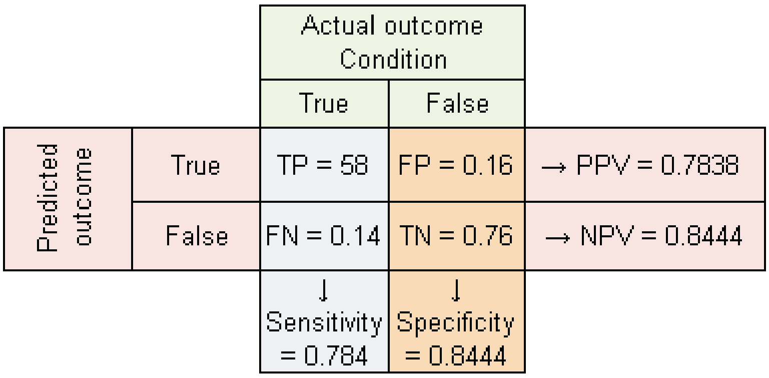


Figure 6.5: PathNet performance: Alzheimer's disease ROC curve.

6.2.2.2 Vascular dementia



		Actual outcome Condition		
		True	False	
Predicted outcome	True	TP = 58	FP = 0.16	→ PPV = 0.7838
	False	FN = 0.14	TN = 0.76	→ NPV = 0.8444
		↓ Sensitivity = 0.784	↓ Specificity = 0.8444	

Figure 6.6: PathNet performance: Vascular dementia confusion matrix at cutoff = 0.5.

Cutoff	Sensitivity	Specificity	$PV+$	$PV-$
0.0	1.0	0.00	0.4390	1.0
0.4	0.8194	0.8152	0.7763	0.8523
0.5	0.7840	0.8440	0.7838	0.8444
0.6	0.6250	0.9022	0.8333	0.7545
0.7	0.3056	0.9674	0.8800	0.6403
0.85	0.0	1.0	1.0	0.5610
1.0	0.0	1.0	1.0	0.5610

Table 6.5: PathNet performance: Vascular dementia — summary confusion matrix of unique cutoffs in range $0.0 - 1.0$.

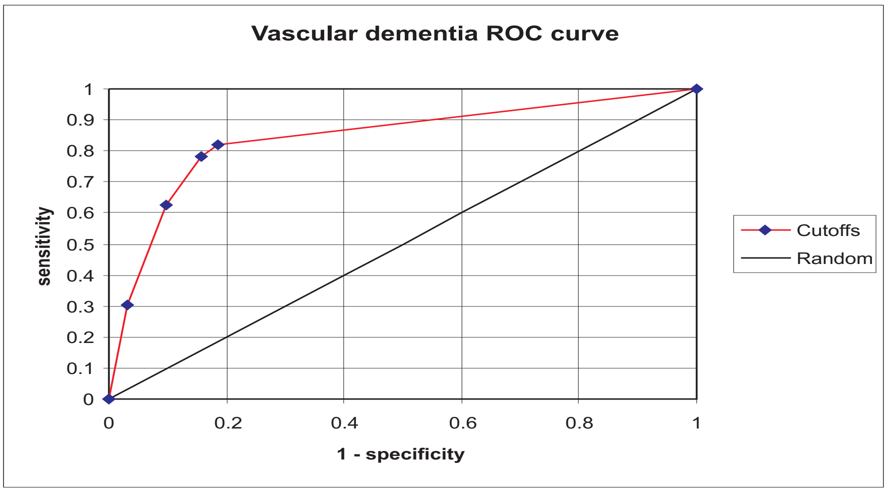
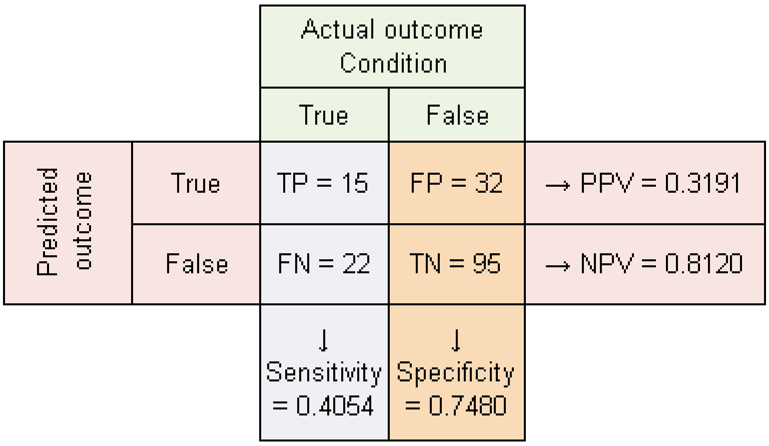


Figure 6.7: PathNet performance: Vascular dementia ROC curve.

Measure	Value
Cutoff	0.5
Sensitivity	0.784
1−specificity	0.156
Area	0.814
Significance	< 0.001
95%CI upper	0.744
95%CI lower	0.884

Table 6.6: PathNet performance: Area Under the Curve (AUC) statistics, Vascular dementia.

6.2.2.3 Other dementia



		Actual outcome Condition		
		True	False	
Predicted outcome	True	TP = 15	FP = 32	→ PPV = 0.3191
	False	FN = 22	TN = 95	→ NPV = 0.8120
		↓ Sensitivity = 0.4054	↓ Specificity = 0.7480	

Figure 6.8: PathNet performance: 'Other pathology' confusion matrix at cutoff = 0.5.

Cutoff	Sensitivity	Specificity	$PV+$	$PV-$
0.0	1.	0.00	0.2256	1.0
0.3	0.6486	0.5118	0.2791	0.8333
0.5	0.4054	0.7480	0.3191	0.8120
0.85	0.1351	1.0	1.0	0.7987
1.0	0.0	1.0	1.0	0.7744

Table 6.7: PathNet performance: 'Other pathology' — summary confusion matrix of unique cutoffs in range 0.0 – 1.0.

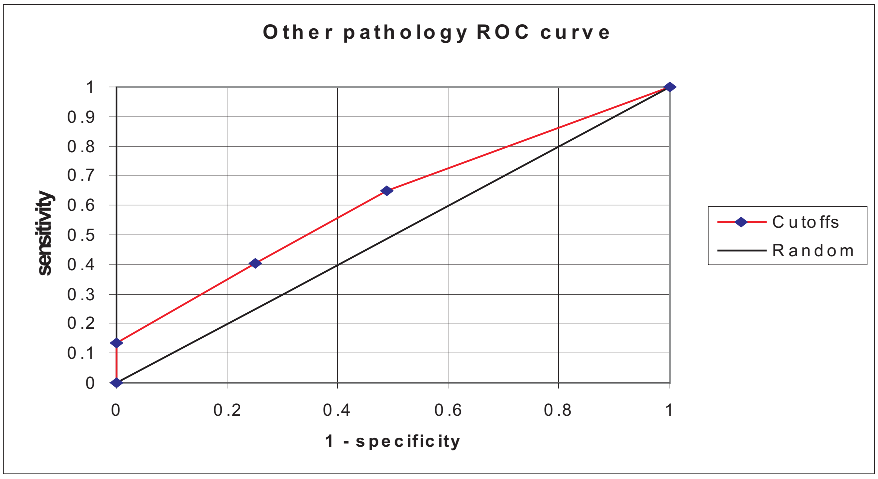


Figure 6.9: PathNet performance: 'Other pathology' ROC curve.

Measure	Value
Cutoff	0.5
Sensitivity	0.4054
1−specificity	0.252
Area	0.577
Significance	< 0.156
95%CI upper	0.469
95%CI lower	0.685

Table 6.8: PathNet performance: Area Under the Curve (AUC) summary statistics, 'Other pathology'.

6.3 Discussion

In our study, we have demonstrated empirically that the BNs developed in Chapter 5 can 1) predict the likelihood of the dementia syndrome in aged populations (65 and over); and 2) predict Alzheimer's disease and vascular dementia. However, we are unable to evaluate the performance of the pathology model on cases where the outcome is dementia with lewy bodies, frontotemporal or co-existing diseases, which is a limitation of this study.

As can be seen from the confusion matrix for the dementia syndrome model, DemNet, shown in Figure 6.2, the sensitivity (TPR) of the model is 0.7407 and FPR (1 − specificity) is 0.2069. In other words — converting these values to percentages — approximately 74.07% of all dementia positive cases in the data set are correctly identified as such by the model, and 20.69% of all dementia absent cases are incorrectly identified as positive. The frequency of negative predictions when the actual outcome was positive, that is to say the false negative ratio (FNR), is $\frac{FN}{TP+FN} = \frac{35}{100+35} \approx 26\%$. Since this is a tool aimed at primary care, it is a concern that some cases may pass unidentified, therefore the model may need tuned in order to reduce the FNR. In contrast, the 20.69% of negative cases that are incorrectly identified as positive are less concerning.

In Figure 6.3, the ROC curve for DemNet lies above the reference (45°) line, which indicates that our model performs better than one that classifies randomly. As can be seen from the curve, a low cutoff value (for example 0.3) results in a high sensitivity, however this comes at a cost as there is a rise in the (FPR) number of cases predicted as positive that should have been negative. The cutoff that we choose, namely 0.5, seems like a reasonable compromise — the sensitivity is not as high as the 0.3 cutoff, but more importantly, the FPR rate is lower. As can be seen from Table 6.1 and Figure 6.3, higher cutoff values seem to have a lower FPR, but the sensitivity is poor.

A key measure of accuracy of the model is the AUC value. From Table 6.2 $AUC_{\text{DemNet}} = 0.764$, with 95% confidence in the range $0.665 - 0.864$. This means that a randomly selected case from the positive diagnosis group has a predicted value larger than that for a randomly chosen individual from the negative diagnosis group in 76.4% of the time. This result indicates that our model has the ability to distinguish between the two diagnostic groups, namely 'dementia present' and 'not dementia', much better than a model that randomly classifies cases. More formally, let $H_{\varnothing} = \text{AUC}_{rand} = 0.5 = \text{AUC}_{DemNet}$; since $\text{AUC}_{DemNet} = 0.764 > \text{AUC}_{rand}$ and $p < 0.001$, we can reject $H_{\varnothing}$. From a qualitative perspective, using the translation offered by Pearce and Ferrier [215] (see Section 6.1.3.2), our results indicate that the model achieves a 'good' level of discrimination on this class variable.

We now turn attention to the model responsible for pathology diagnosis, PathNet. We discuss the results obtained for classification of Alzheimer's disease, vascular dementia and 'other' pathology.

As can be seen from the confusion matrix for Alzheimer's disease shown in Figure 6.4, the sensitivity (TPR) of the model is 1.0 and FPR ($1-$ specificity) is 0.5. In other words — converting these values to percentages — 100% of Alzheimer's disease cases in the model were correctly predicted as positive by the model. However, approximately 50% of all Alzheimer's disease absent cases are incorrectly identified as positive. We acknowledge that the FPR is high. A possible explanation may be that the model is trying to predict a co-existing pathology, where Alzheimer's disease forms one of the possible pathologies. This claim can only be validated with data containing co-existing cases, and a modification to the evaluation set up. Alternatively, it may be that the model requires further tuning.

In Figure 6.5, the ROC curve for Alzheimer's disease lies above the reference (45°) indicating that our model performs better than one that classifies using a random guess strategy. A cutoff value of 0.5 leads to an excellent level of sensitivity, however, as discussed above, the FPR in this case is high. It seems that a more restrictive cutoff of 0.7 has a more balanced outcome.

From Table 6.4 $AUC_{\text{Alzheimer}} = 0.745$, with 95% confidence in the range $0.673 - 0.818$. This means that a randomly selected case from the positive Alzheimer's disease group has a predicted value larger than that for a randomly chosen individual from the negative diagnosis group in 74.5% of the time. This result indicates that our model has the ability to distinguish between the two diagnostic groups, namely Alzheimer's disease present and Alzheimer's disease absent, much better than a model that randomly classifies cases. More formally, let $H_{\varnothing} = \text{AUC}_{rand} = 0.5 = \text{AUC}_{\text{alzheimer's}}$; since $\text{AUC}_{\text{alzheimer's}} = 0.745 > \text{AUC}_{rand}$ and $p < 0.001$, we can reject $H_{\varnothing}$. From a qualitative perspective, using the translation offered by Pearce and Ferrier [215] (see Section 6.1.3.2), our results indicate that the model achieves a 'good' level of discrimination on this class variable.

From the vascular dementia confusion matrix shown in Figure 6.6, the sensitivity (TPR) of the model is 0.784 and FPR (1 − specificity) is 0.156. In other words — converting these values to percentages — approximately 78.4% of all vascular dementia positive cases in the data set are correctly identified as such by the model, and 15.6% of all vascular dementia absent cases are incorrectly identified as positive. The frequency of negative predictions when the actual outcome was positive, that is to say the false negative ratio (FNR), is $\frac{FN}{TP+FN} = \frac{14}{58+14} \approx 19.4\%$. Since this is a tool aimed at primary care, it may be of concern that some cases may pass unidentified.

In Figure 6.7, the ROC curve for vascular dementia lies above the reference (45°) indicating that our model performs better than one that classifies using a random guess strategy.

From Table 6.6 $AUC_{\text{vascular}} = 0.745$, with 95% confidence in the range $0.673 - 0.818$. This means that a randomly selected case from the positive vascular dementia disease group has a predicted value larger than that for a randomly chosen individual from the negative diagnosis group in 74.5% of the time. This result indicates that our model has the ability to distinguish between the two diagnostic groups, namely vascular dementia present and vascular dementia absent, much better than a model that randomly classifies cases. More formally, let $H_{\varnothing} = AUC_{rand} = 0.5 = AUC_{vascular}$; since $AUC_{vascular} = 0.745 > AUC_{rand}$ and $p < 0.001$, we can reject $H_{\varnothing}$. From a qualitative perspective, using the translation offered by Pearce and Ferrier [215] (see Section 6.1.3.2), our results indicate that the model achieves 'good' level of discrimination on this class variable.

The performance of PathNet on 'other' is not as good. As can be seen from the confusion matrix shown in Figure 6.6, the sensitivity (TPR) of the model is very low 0.4054 and the FPR ($1 -$ specificity) is on the high side at 0.252. In other words — converting these values to percentages — approximately 40.54% of all 'other' positive cases in the data set are correctly identified as such by the model, and 25.2% of all 'other' absent cases are incorrectly identified as positive. The frequency of negative predictions when the actual outcome was positive, that is to say the false negative ratio (FNR), is $\frac{FN}{TP+FN} = \frac{22}{15+22} \approx 59.5\%$. Clearly, one of the reasons that the TPR is so low is because the positive cases are being incorrectly classified as negative.

In Figure 6.9, the ROC curve for 'other' lies (barely) above the reference (45°), which would indicate that our model performs better than one that classifies using a random guess strategy. From Table 6.8 $\mathrm{AUC}_{\mathrm{other}} = 0.577$, with 95% confidence in the range $0.673 - 0.818$. This means that a randomly selected case from the positive 'other' group has a predicted value larger than that for a randomly chosen individual from the negative 'other' group in 57.7% of the time. Since $\mathrm{AUC}_{\mathrm{other}} = 0.577 > AUC_{H_\varnothing}$, it would appear that our model just manages to distinguish between the two diagnostic groups, namely 'other' true and 'other' false, better than a model that randomly classifies cases. However, after probing the results further, it would appear that PathNet classifies the 'other' class variable only as well as a random classifier. More formally, let $H_\varnothing = \mathrm{AUC}_{\mathrm{rand}} = 0.5 = \mathrm{AUC}_{\mathrm{other}}$. Despite the fact that $\mathrm{AUC}_{\mathrm{other}} = 0.577 > AUC_{rand}$, the significance is $p < 0.156$, therefore we can not reject $H_\varnothing$. From a qualitative perspective, using the translation offered by Pearce and Ferrier [215] (see Section 6.1.3.2), our results indicate that the model achieves a 'poor' level of discrimination for this class variable.

Despite the poor performance of PathNet on the 'other' class variable, our other results are encouraging and indicate that our BN models have the potential to be used as a decision support tool that could assist primary care clinicians during dementia diagnosis.

6.4 Summary

This chapter described how the two dementia models developed by hand in Chapter 5 are evaluated using a real-life clinical data set.

We define a number of metrics that are used to measure the diagnostic accuracy of the models, then we demonstrate empirically the classification performance of each model against a number of cutoffs. The classification accuracy of the models against the metrics is generally good. We recognise, however, that the model responsible for dementia pathology diagnosis, PathNet, is stronger at identifying Alzheimer's disease and vascular dementia than it is at identifying the "Other" diagnostic class. It is thought that this is due to the lack of test data representing the "Other" class variable. In addition, also due to the lack of data, it has not been possible to empirically evaluate the model on the other diagnostic tasks, such as dementia with lewy bodies, frontotemporal dementia or co-existing pathologies. Nonetheless, when the models are tested by domain experts using a small sample of cases the output appears to be inline with experts' expectations. Further data is required in order to provide a complete assessment of the other diagnostic outputs, and, in addition, the parameters of the model may require further tuning as a result.

This part of the thesis has focused on the hand-crafted approach to BN construction, and we have demonstrated how the approach is applied to a real-life problem. The next part is concerned with the second approach which relies on data for BN construction.

Part III

Data-driven Bayesian network construction

Chapter 7

Data driven Bayesian network construction

7.1 Introduction

In Chapter 4 we present a range of methods to support Bayesian network (BN) construction using human expert knowledge — the so called "hand crafted" approach. As mentioned in Chapter 4, such an approach is time consuming, and heavy emphasis is placed on human experts to provide both the dependency relationships and the local probability parameters. An alternative method for BN construction seeks to induce the structure and parameters from data. It is the purpose of this chapter to describe the mechanics of this approach. Note that this chapter describes only BN structure learning from data and not parameter learning. Where necessary, we use the Netica [2] BN modelling software tool to perform parameter learning for a given (learned) model. Note that when we refer to BN learning in this chapter, we mean only structure learning, unless otherwise stated.

While the data driven approach is a popular alternative, which greatly reduces the dependence on human experts, and in some cases increases the accuracy of the model, it is no “silver bullet”. The primary drawback with BN learning from data is that the number of possible structures for a given problem grows super-exponentially with the number of variables in the problem domain, n. For a problem consisting of n variables, Robinson [229] calculates the complexity of the search space as $O(n!2^{\binom{n}{2}})$. The number of possible BN structures for various values of n are computed using Robinson's formula, shown in 7.1, and listed in Table 7.1.

$$f(n) = \sum_{k=1}^{n} (-1)^{k+1} \binom{n}{k} 2^{k(n-k)} f(n-k); f(0) = 1, f(1) = 1 \qquad (7.1)$$

n	Number of BNs
1	1.0×10^0
2	3.0×10^0
3	2.5×10^1
4	5.4×10^2
5	2.9×10^4
10	4.2×10^{18}
20	2.3×10^{72}
50	7.2×10^{424}
100	1.1×10^{1631}

Table 7.1: The number of possible BNs for a given number of variables as per Robinson's formula.

A number of algorithms have attempted to reduce the complexity of the problem by imposing restrictions and assumptions, however the problem remains complex and hard [45].

Given that the number of possible structures for a given problem domain grows super-exponentially, exact and exhaustive methods for BN learning become unfeasible. Therefore, approximate methods based on heuristics become advantageous. Many such algorithms can be found in BN learning literature, however the majority of these techniques can be separated into two broad categories: 1) methods that carry out *dependency analysis* tests; and 2) methods that search in a solution space and utilise a fitness metric to drive exploration of the search space — the so called “*search and score*” approach.

The organisation of this chapter is as follows. In Section 7.2 we provide an introduction to the dependency analysis approach. The class of algorithms that operate using a search and score strategy, which is the basis of the algorithms developed in this research, is described in detail in Section 7.3. Existing techniques and new techniques, which are used as a basis for comparing the algorithms developed in this research, are described in Section 7.4. Finally, we summarise this chapter in Section 7.5.

7.2 Dependency analysis approach

Learning structure by dependency analysis is based on performing conditional independence (CI) test on tuples of variables. Using *statistical tests* such as χ^2 or *information theoretic* measures such as mutual information, these constraint based algorithms attempt to determine whether pairs of variables are independent or dependent given a set of conditioned variables.

The most common of the statistical approaches is the PC algorithm, developed by Spirtes and Glymour [255]. It begins with the complete undirected graph, then ‘thins’ the graph by removing edges with zero order conditional independence relations, then thins again with first order conditional independence relations, and so on until an optimal BN is generated. Algorithms based on information theoretic measures include that of Cheng et al. [40, 41] and Thomas [263].

Singh and Valtorta [250] report three drawbacks in using CI approaches: 1) extensive testing of independence relations is required to derive the final network structure; 2) when condition sets are large, CI tests may become unreliable unless an enormous volume of data is available; 3) the set of all independence statements which hold for a given domain grows exponentially as the number of variables grow, making CI approaches unrealistic. Nevertheless, Spirtes and Glymour [255] have developed a CI based algorithm that yields good computational efficiency with sparse networks and limited data.

7.3 Search and score approach

The search and score approach is an alternative offering to the dependency analysis approaches described in Section 7.2. The search and score paradigm for BN discovery seeks to explore a search space of candidate BN structures for the one that best represents the probabilistic dependency relationships that exist in the data. In other words, the approach seeks to discover the probabilistic dependency network that most likely generated the data set [51]. Under this approach, the BN discovery problem can be viewed as an optimisation based search problem that consists of three components:

1. *Search engine*: Drives exploration of the search space.
2. *Search space*: Defines the space of feasible solutions.
3. *Scoring function*: A consistent mechanism for measuring the quality of a solution. The score or "fitness" metric allows the search algorithm to differentiate between "good" and "bad" solutions/structures; additionally,

the value produced by the scoring function is used to guiding the direction of the search engine's exploration.

In contrast to the CI based algorithms described in Section 7.2, the search and score approaches employ a search heuristic to search the space of candidate structures (solutions) for one that maximises the score by making perturbations to the solution. For example, edges are added or removed, then the effect is measured using the scoring function. The search continues until an optimal solution is found or some predefined stopping criterion is met.

We describe each of the three search and score components in more detail in the following sections.

7.3.1 Search engines

In Section 7.1 we stated that the complexity of the search space was one of the major challenges faced in BN discovery from data. Accordingly, many search heuristics have been proposed. The search engines proposed in the literature can be broadly classified into two categories: those that iteratively build upon a single candidate solution (network structure), the so called *sequential algorithms*; and, in contrast, *population algorithms* that develop a collection of candidate solutions in parallel. Clearly, the advantage of the population based approach is that a wider area of the solution space is explored; in addition, since the population based approaches explore groups of solutions, the chance of getting stuck in local optima is reduced.

A selection of popular sequential and population based search algorithms are described in more detail in Section 7.4.

7.3.2 Search spaces

The search space defines all the possible candidate solutions. It is these candidate solutions that the search engine explores. A common approach is to perform a search in the space of all possible BN structures — the *space of all directed acyclic graphs* (DAGs). However, recall from Section 7.1 that the cardinality of this search space is super-exponential in the number of variables in the domain. The primary drawback with this approach is computational inefficiency related to redundancy in repeated analysis of equivalent directed acyclic graphs Anderson et al. [5]. Alternative sub-spaces exist which seek to: 1) make the search more tractable by reducing the cardinality of the search space; and 2) reduce redundancy.

One such alternative search space is the *space of equivalence classes* in which a single solution represents those BN structures that represent the same set of conditional independencies [256, 43]. Gillispie and Perlman [100] have experimented with the space of equivalence classes, however, they did not find convincing reduction in the cardinality of the search space despite the high computational cost generated by searching in this space. However, Chickering [44] has developed a representation for the equivalence classes and a greedy search heuristic which yields "slightly better results in less time than the same search applied to the traditional space of DAGs".

Another approach, which seeks to reduce the complexity of the search space, is a search in the *space of orders*. In this approach an order is imposed between the n nodes of the problem domain. In other words, a node X_j can not be a parent of node X_i if X_i precedes X_j in the order. The maximum number of parents that a node can take on is $\binom{n-1}{k}$, where n is the number of nodes and k is an upper bound on the number of parents that a node can take.

Many authors have proposed a search in the space of orders, as it is significantly smaller than the space of entire network structures [262]. However, the mechanism used to evaluate candidate orders may lead a high computational overhead. For example, Larrañaga et al. [154] uses a Genetic Algorithm (see Section 7.4.2.1) to evolve candidate orders (solutions), then they apply the the K2 algorithm (see Section 7.4.1.1) to evaluate each candidate order.

Although the search of BN structures is typically conducted in a single search space, Kočka and Castelo [150] have proposed a hybrid which attempts to combine characteristics of the entire space of DAGs with the space of equivalence classes.

7.3.3 Scoring functions

In Sections 7.3.1 and 7.3.2 we described the search engine and search space components of the search and score methodology, which are responsible for generating candidate BN solutions. It is the purpose of this section to introduce the mechanism by which candidate BN structures are gauged for quality — it is this quality value that search engines use to differentiate between good and poor BN solutions with respect to the database of cases. Several scoring metrics have been proposed in the literature. Note, however, that the scope of this section is to provide necessary background with regards to the fitness metric used within this research, and not an in-depth analysis of all the scoring metrics available.

In this research we use a decomposable scoring function — the 'score' of a BN structure given a data set is the sum of scores associated with individual families, where a family is defined as a node and its parents:

$$score(bs) = \sum_{i=1}^{n} score(X_i, \pi(X_i)) \tag{7.2}$$

The basic function of the metric is to calculate a quantitative measure that expresses the probability that the candidate structure, bs, generated the probabilistic relationships that exist in the data set.

To that end, a number of implicit assumptions are made regarding the data set of cases used:

- Variables are discrete: Continuous variables should be discretised at the risk of losing information.
- No missing values in the data set: Estimate a distribution over the variable's values to estimate its value [254].
- Cases in the data set are independent of each other.
- The process that generated the data is not time dependent.

In Section 7.3.3.1 we define the fundamentals of the fitness metric used in our research — Bayesian scoring metrics — and in Section 7.3.3.2 we present the actual metric used.

7.3.3.1 Bayesian-based scoring metrics

The Bayesian approach is a practical method for selecting statistical models — for example Bayesian network models — given a database of cases [51]. In this approach, Bayes theorem is used to calculate the posterior probability, $P(B_s|D)$, of a network structure given the database of cases. This is achieved by: 1) defining a prior distribution over all network structures; and 2) for each structure, use Bayes' theorem to compute the posterior probability given the data set. Using this approach, it is possible to compute ratios for pairs of networks structure and thus rank order a set of structure by their posterior probabilities. To calculate the ratio of posterior probabilities, say $P(Bs_1|D)$ and $P(Bs_2|D)$, we use the following equivalence:

$$\frac{P(B_{s1}|D)}{P(B_{s2}|D)} = \frac{\frac{P(B_{s1}|D)}{P(D)}}{\frac{P(B_{s2}|D)}{P(D)}} = \frac{P(B_{s1}, D)}{P(B_{s2}, D)} \tag{7.3}$$

Using this principle, it is possible to compute $P(Bs|D)$ for any candidate network structure. Cooper and Herskovits have proposed a scoring metric for assessing the quality of Bayesian network structures based on the Bayesian approach — see Section 7.3.3.2.

7.3.3.2 Cooper Herskovits (K2) metric

Cooper and Herskovits [51] have derived a scoring metric based on the Bayesian approach to compute $P(Bs|D)$, where $P(Bs|D)$ is described in their theorem:

Theorem 1. *Let Z be a set of n discrete variables, where variable x_i in Z has r_i possible value assignments: $(v_{i1}, \ldots, v_{ir_i})$. Let D be a data set containing m cases, where each case contains a value assignment for each variable in Z. Let*

Bs_1 denote a Bayesian network structure containing only the variables in Z. Each variable X_i in Bs_1 has a set of parents π_i, where π_i may be empty. Let ϕ_{ij} denote the j^{th} unique instantiation of π_i relative to D. Suppose there are q_i such unique instantiations of π_i. Define N_{ijk} to be the number of cases in D in which variable x_i is instantiated as v_{ik} and π_i is instantiated as ϕ_{ij}. Let $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$. If the cases in the database occur independently, there are no missing values, and the prior probability density function of the parameters given the structure is uniform, then it follows that:

$$P(Bs|D) = P(Bs) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}! \tag{7.4}$$

This score is known as the K2 metric. The $P(Bs)$ term in the right hand side of Formula 7.4 captures information about the "real" network structure prior to observation of the data set. For example, if specific edges are known in advance, then the network structures that admit this information are given a higher prior probability. If no information is available, all structures are equiprobable, hence the $P(Bs)$ term can be dropped from Formula 7.4. Note that the logarithm of Formula 7.4 is used for numerical convenience.

7.4 Existing search and score algorithms

This section presents some examples of algorithms that can be used to search through any space of Bayesian network (BN) structures. As introduced in Section 7.3.1, there are two categories of search algorithm, namely sequential algorithms (Section 7.4.1) and population based algorithms (Section 7.4.2).

7.4.1 Sequential algorithms

We begin by describing the "classic" K2 algorithm proposed by Cooper and Herskovits [51], which relies on an order among the variables as input. Thereafter, we present Buntine's algorithm [25], which does not rely on an order among the variables. Finally, we present an extension to the K2 algorithm, the CB algorithm, which combines conditional independence tests and Bayesian learning to discover BNs. Note that other sequential algorithms are used for BN learning, for example Simulated Annealing [33].

7.4.1.1 Classic K2

Cooper and Herskovits [51] proposed a greedy, deterministic search heuristic which searches for the most probable belief network structure given a database of cases, and it uses the CH metric, described in Section 7.3.3.2, to guide the search. The algorithm assumes that an order among the domain variables is available, an upper bound on the number of parents is imposed, and that a priori, all structures are equally likely. The order assumption states that a node x_i can only have node x_j as a parent node if in the ordering node x_j comes before node x_i. The K2 algorithm starts with an empty network and iterates through each of the nodes according to their position in the order. For each node, K2 assumes that a node has no parents, and then adds incrementally that parent whose addition maximises the probability of the resulting structure. When the addition of no single parent can increase the probability, the K2 algorithm stops adding parents to the node.

The K2 algorithm pseudocode is shown in Figure 1 — the notation is defined in Section 2.2.

Algorithm 1 Pseudocode of the K2 algorithm [51].

```
Input: A set of $n$ nodes, an ordering on the nodes, an upper bound $u$ on the number of parents a node may have, and a database $D$ containing $m$ cases.
Output: For each node, a printout of the parents of the node.
for i:= 1 to $n$ do
    $\pi_i = \emptyset$
    $P_{old}$:= $f(i, \pi_i)$; computed using Equation 9.1
    okToProceed := true
    while okToProceed AND $|\pi_i| < u$ do
        let $z$ be the node in $Pred(x_i)$; i.e. $\pi_i$ that maximises $f(i, \pi_i \cup \{z\})$;
        $P_{new} := f(i, \pi_i \cup \{z\})$;
        if $P_{new} > P_{old}$ then
            $P_{old} := P_{new}$
            $\pi_i := \pi_i \cup \{z\}$
        else
            okToProceed := false;
        end if
    end while
    print 'Node:', $x_i$, 'Parents of this node:', $\pi_i$
end for
```

Crucially, Cooper and Herskovits [52] show that the metric on which K2 is based (CH, see Section 7.3.3.2) is minimized as the number of cases increases without limit on "those [BN] structures that, for a given node order most parsimoniously capture all the independencies manifested in the data". In other words, the CH metric favours the optimal (in)dependency model consistent with the given order, as the number of cases in the data set increase without limit.

The main drawback with this algorithm is the fact that it requires as input an order among the variables. The order imposed among the variables has an impact on the resulting quality of the network structure [250], therefore to guarantee a good performance, a good order needs to be specified as input for K2. In some cases suitable "domain knowledge" may not be available, and where it is expert time is charged at premium rates. As an alternative, automated order learning is proposed. In addition, the K2 algorithm requires a decomposable scoring function (see Section 7.3.3.2), which may be viewed as a drawback.

7.4.1.2 Buntine's algorithm

Buntine's [25] 'B' algorithm is a popular greedy sequential search algorithm for BN structure discovery. In contrast to Cooper and Herskovits K2 algorithm, the B algorithm does not require a node ordering. It starts with an empty parent set. At each iteration an edge that does not lead to a cycle and maximises the scoring function is added, and the process continues until the fitness quality no longer increases or all nodes in the order have been visited. Although the original B algorithm considers a single operator — the add link operator — variations could be proposed that include additional operators such as edge deletion and edge reversal.

Like the majority of sequential greedy search algorithms, the primary drawback of the B algorithm is premature convergence on local optima. For example, when the B algorithm decides to add an edge, it can not be removed at a later iteration. To that end, Blanco et al. [18, 17] have proposed new algorithms which allow the search engine to reconsider previous decisions at a later time in the search.

7.4.1.3 CB algorithm

Singh and Valtorta [249, 250] proposed an algorithm which cherry-picks the best properties of the algorithms described above, namely the Conditional independence + Bayesian learning (CB) algorithm. Recall from Section 7.2 that dependency analysis algorithms use conditional independence (CI) test to construct the network; although effective, the main drawback in this approach concerned the number of tests required to search through all possible CI statements. In Section 7.3, we describe the search and score approach to BN discovery from data, which typically consists of a greedy search heuristic and a scoring function. In Section

7.4.1.1 we introduce the K2 algorithm and note the metric used by the algorithm favours the most optimal structure given an order among the variables. However, the requirement of an order as input is the algorithm's primary drawback, since it may not be possible to provide an order for domains where there is very little expertise or the number of variables is large.

The CB algorithm seeks to harness the qualities of each of the aforementioned algorithms by combining them with a view to provision a "computationally tractable algorithm which is not over dependent on CI tests nor does it require [as input] a node ordering" [249]. It achieves this through an orchestration involving CI tests to generate a "good" node order from the data, which is then used by the K2 algorithm to discover the underlying BN. The CB algorithm executes in two phases: in phase one, the CB algorithm starts with a complete, undirected graph over all nodes, then CI tests are executed to remove the edges between adjacent nodes that are unconditionally independent (CI tests of order zero). The remaining edges in the undirected graph are orientated to form an order among the nodes. In Phase II, the order derived in phase I is fed into K2 as input to construct the network. The process is repeated as follows: Using the resulting undirected graph from the previous iteration, the algorithm removes those conditional independent edges given one node (CI test of order 1), then two nodes and so on. The algorithm iteratively constructs the network increasing the order of CI tests until the termination criteria is met.

7.4.2 Population based algorithms

A class of algorithms referred to as "nature inspired" search heuristics are loosely based on systems found in nature; they are typically population-based and stochastic. Given the dimensionality and complexity of the Bayesian network (BN) search space, those nature inspired algorithms which operate on groups of candidate solutions in parallel are of particular interest in BN learning. Many of these population-based nature-inspired algorithms have been proposed as search algorithms in the BN discovery from data problem.

Wong et al. [279], Novobilski and Kamangar [202] and Lee et al. [162] proposed Evolutionary/Genetic Programming (EP/GP).

Larrañaga et al. [154, 155, 156] and Novobilski and Fesmire [201, 203] have proposed a range of Genetic Algorithm (GA) based approaches. Estimation of Distribution Algorithms (EDA) have been applied by Romero et al. [232].

More recently, de Campos et al. [62] and Daly et al. [59] have employed Ant Colony Optimisation (ACO), and Castro and von Zuben [35] apply the Artificial Immune System (AIS) algorithm.

In Chapter 9, we compare our approach to BN discovery from data with algorithms from the literature. In particular, we compare our algorithms with two Genetic Algorithm (GA) based approaches. The following two sections provide an introduction to these two approaches.

7.4.2.1 Genetic Algorithm approach for Bayesian network discovery

> " ...the metaphor underlying genetic algorithms is that of natural evolution. In evolution, the problem each species faces is one of searching for beneficial adaptations to a complicated and changing environment. The 'knowledge' that each species has gained is embodied in the makeup of the chromosomes of its members [60]."

Originally developed by John Holland [122], Genetic Algorithms (GAs) are inspired by, and based loosely on, principles of Darwinian evolution, and are used extensively to solve search and optimisation problems. The algorithm's strength lies in its ability to evolve near optimal (or, in some cases, optimal) solutions to complex problems, sometimes involving multiple objectives with minimal problem information and without searching the entire search space [64, pp85] — a strength favourable for problems where little is known about the underlying search space. The ability of GAs to explore a search space is achieved by mechanisms employed from evolution theory, which simulate natural selection and genetic inheritance. Accordingly, individuals yielding a high fitness in the population are likely to survive and generate off-spring, thus transmitting their biological heredity to new generations. As the name suggests, genetic algorithms and their mechanics allude to natural genetics. However, note that computational GAs are simply inspired by principles of natural genetics, and are not, in general, biologically plausible.

A GA has a number of components. Each potential solution to a problem is encoded as a *chromosome*; one or more chromosomes make a *population* of solutions. The algorithm embeds a fitness metric (*objective function*, for example, a BN scoring metric), which facilitates solution evaluation. At each *iteration* of the algorithm, chromosomes are selected according to some predefined *selection* criteria and entered into a *mating pool*. The next population (generation) of chromosomes are formed by performing *genetic recombination* on chromosomes in the mating pool. In order to maintain diversity in the population and prevent premature convergence, a *mutation* operator is applied at various positions in the chromosome with probability P. A diagram of the GA process is shown in Figure 7.1.

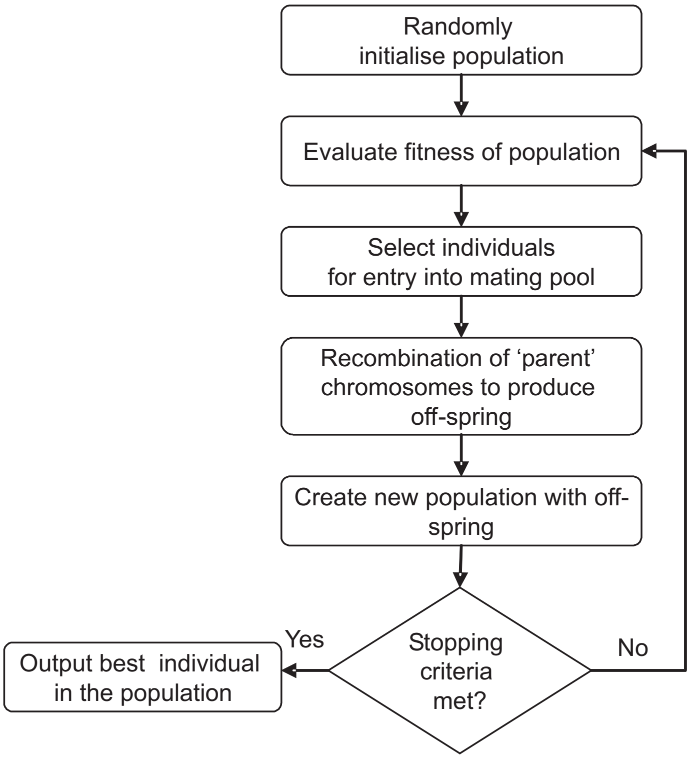

Randomly initialise population
Evaluate fitness of population
Select individuals for entry into mating pool
Recombination of 'parent' chromosomes to produce off-spring
Create new population with off-spring
Stopping criteria met?
Yes
No
Output best individual in the population

Figure 7.1: Generic GA process diagram.

The interested reader is directed to Mitchell [183] for a useful, introductory text on GAs.

From a BN discovery perspective, GA chromosomes encode solutions according to the search space. For example, if the search space is defined as the space of all BN structures, then a GA chromosome may encode an n by n connectivity matrix, where n is the number of variables in the problem — see Section 8.5.1. Alternatively, a chromosome may encode an order among the variables — see Section 8.6.1. Note, that, in general, the GA operators are not closed with respect to Directed Acyclic Graph (DAG) conditions — see Section 2.1. In other words, the GA is stochastic, therefore the algorithm may evolve a solution that includes cycles. However, Novobilski [201] has proposed genetic operators that guarantee acyclicity. An example of a GA process for BN discovery from data is shown in Figure 7.2.

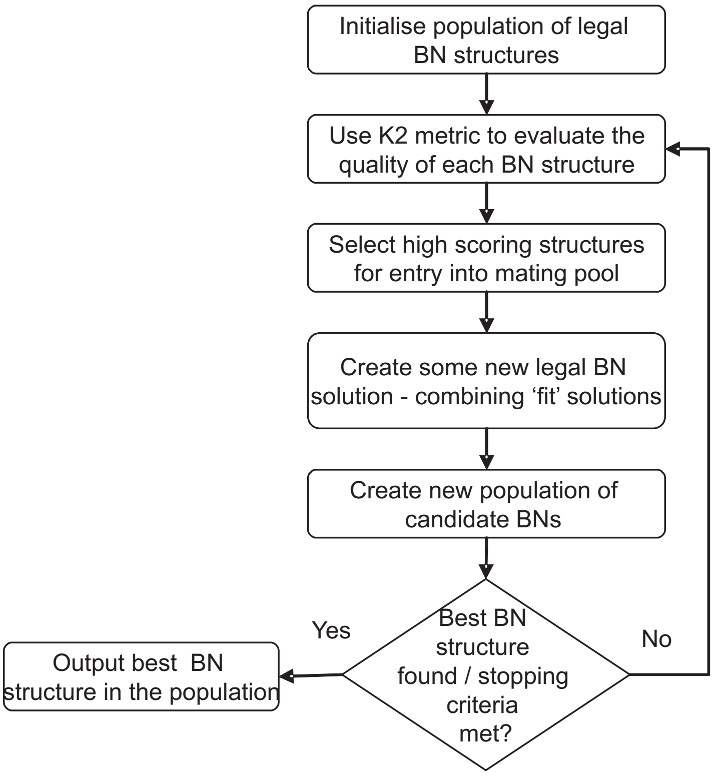

Initialise population of legal BN structures
Use K2 metric to evaluate the quality of each BN structure
Select high scoring structures for entry into mating pool
Create some new legal BN solution - combining 'fit' solutions
Create new population of candidate BNs
Best BN structure found / stopping criteria met?
Yes
No
Output best BN structure in the population

Figure 7.2: GA process diagram for BN discovery.

Larranaga et al. [154] et al. proposed the GA as a search heuristic for exploring the entire space of BN structures and the space of orders. When applied to the entire space of structures, each candidate solution is evaluated using the K2 fitness metric — see Section 7.3.3.2. However, the GA variant, which searches in the space of orders, executes the K2 algorithm on each candidate solution with the purpose of achieving a score — see Section 7.4.1.1. Clearly, application of the K2 algorithm to score candidate structures is computationally expensive. However, Kabli et al. [135] have proposed a GA based algorithm that seeks to reduce the computation expense. This algorithm is introduced in Section 7.4.2.2.

7.4.2.2 ChainGA algorithm

In Section 7.3.2, we acknowledge that good orders lead to good Bayesian network (BN) structures, and, given a good order, the K2 algorithm finds good BN structures (see Section 7.4.1.1). In Section 7.4.2.1 we present a technique for BN discovery from data which uses a Genetic Algorithm (GA) to search for 'good' orders that are evaluated using the K2 algorithm. However, we note that use of the K2 algorithm leads to high computational overheads 7.4.2.1.

Kabli et al. [135] have synthesised all the good properties of order-based search into an algorithm called Chain-Model GA, and they demonstrated that their approach is superior and computationally more efficient to the order-based GA approach proposed by Larranaga et al. [154]. In the Chain-Model GA, Kabli et al. hypothesis that " a chain (order) is a sufficiently good model to local node orderings of which good Bayesian network structures can be built [135]". The reduction in computational expense is gained by evaluating the *chain structure* rather than using the full K2 algorithm to evaluate all orders. For example, given a domain of variables X_1, X_2, X_3, X_4 and a candidate solution (order) X_2, X_1, X_4, X_3, then it admits a chain structure shown in Figure 7.3.

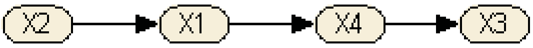


Figure 7.3: An example of a chain-structure.

Instead of using the K2 algorithm to evaluate the quality of the order, the chain structure is evaluated using only the K2 metric with respect to the data. In other words, the chain structure is the candidate BN model. Kabli et al. [135] make the assumption that good orders lead to good structures, and that the 'chain-structure' may hold important information about the ordering and its relation to the network structure. This is a similar assumption to that made by Cooper and Herskovits [51] in that there is a reliance on a good order if the K2 algorithm has a chance of finding a good BN structure. At the end of each iteration, a group of high scoring chain structures are fed into the K2 algorithm as orderings; the resulting structures and scores are stored. Experiments conducted by Kabli et al [135] demonstrate a reduction in computational expense compared to a similar order-based approach proposed by Larranaga [156]

7.5 Summary

This chapter presented the second approach to BN construction, namely the data-driven approach. Two data-driven approaches are described in detail, namely algorithms based on dependency analysis and algorithms based on the search and score paradigm, and existing algorithms found in the literature are described. Challenges and issues regarding BN construction from data, as well as development opportunities within existing algorithms, are highlighted.

In the next chapter we present new algorithms for BN discovery from data, which are based on binary Particle Swarm Optimisation.

Chapter 8

Constructing BN using PSO

In this chapter we present a novel application of Particle Swarm Optimisation (PSO), demonstrating its use as a search heuristic in the Bayesian network (BN) structure discovery from data problem.

The necessary PSO background is presented in Section 8.1 and the motivation for using PSO in the BN structure discovery problem is presented in Section 8.2. Some existing PSO-based approaches to the structure learning problem are presented in Section 8.3. We present our approach in Section 8.4, and the two resulting PSO-based algorithms for BN structure discovery are described in detail in Section 8.5 and Section 8.6. It should be noted that this chapter does not present any evaluation of the PSO-based techniques for BN discovery; such evaluation is given a full treatment in chapter 9.

8.1 Particle Swarm Optimisation

Particle Swarm Optimisation (PSO) is a nature-inspired, population-based stochastic search and optimisation heuristic which was first proposed by James Kennedy and Russell Eberhart [77]. The origins of PSO are sociologically inspired, as it evolved from the notion of "producing computational intelligence by exploring simple analogues of social interaction, rather than purely individual cognitive abilities [221]" . In the early part of the 1990's, Heppner and Grenander [117] published work on the sociology of bird flocking behaviour when searching for corn. Eberhart and Kennedy syndicated the sociocognitive behaviours observed by Heppner and Grenander, and harnessed these characteristics to develop three basic, abstract principles (evaluate, compare and imitate — see Section 8.1.1) to produce a powerful optimisation method — Particle Swarm Optimisation.

As mentioned above, the mechanistic functioning of PSO is derived from social interaction theory, and its underlying mechanics are based on social principles. These principles — their underlying rudiments and their relation to PSO — are discussed in Section 8.1.1. Thereafter, the anatomy of the fundamental component of PSO, namely the particle, is presented in Section 8.1.2, and corresponding notation is provided in Section 8.1.2.1. A description of the "original", canonical PSO algorithm is presented in Section 8.1.3.

8.1.1 Rudiments of the classical PSO

The PSO algorithm maintains a population of simple entities — the particles — where each particle represents a single potential solution in the search space of the optimisation problem at hand. Each particle in the population "flies" through the search space, interacting with the rest of the population to seek new "optimal" areas of the search space. In order to explore the search space effectively, particles use three sociocognitive behaviours, introduced in Section 8.1. These innate social behaviours provide each particle with the ability to *evaluate* their current position in the search space and *compare* their current and previous flying experience with others, and *imitate* the behaviours of those who have succeeded finding features in the search space that are important [142, pp. 288]. Before describing these fundamental, sociocognitive behaviours (evaluate, compare and imitate), it is important to note that particles are connected together according to a topological neighbourhood. Each particle is a member of a neighbourhood, and within these neighbourhoods particles communicate with each other to share information. The concept of neighbourhood is described in more detail in Section 8.1.1.1.

Description of the three sociocognitive behaviours:

1. **Evaluate** In order to distinguish features of the search space that attract and features that repel, particles have the ability to evaluate themselves, usually by means of an objective function (or fitness function). The ability of a particle to evaluate itself is an important asset, as it provides the particle with a score that is a quantitative measure of goodness.

2. **Compare** Since particles have a consistent ability to evaluate themselves, they therefore have a mechanism to compare themselves to others in the population. The notion that particles with a high[1]score occupy "good" areas of the search space is central. In order to explore new and better areas of the search space, particles compare themselves to others in the population and imitate the behaviour of those particles who are superior to themselves.

3. **Imitate** Individual particles imitate their neighbours on the basis of their performance. For example, if a particle's neighbour in the population has a better solution to the problem at hand, the particle will endeavour to

be more like its neighbour, flying through the search space towards the "better" neighbouring particle.

The mechanics of particle flight in PSO is simple [77]; a particle's trajectory along each dimension of the solution space is determined according to a set of rules. At a high level of abstraction, these rules represent a combination of the particle's own flying experience and the successes of the particle's neighbours. The social behaviour principles that governs particle flight is described in more detail in Section 8.1.1.1.

8.1.1.1 Solution exploration through social interaction

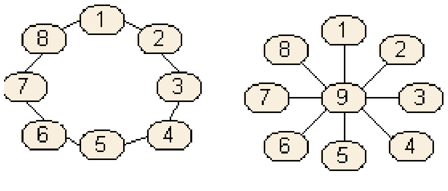
In a PSO, the particles interact in social groups to solve the problem at hand, as they do not have the ability to solve problems as individuals [221]. The population of particles is split and organised into sub-groups of particles according to a specified topological communication/interaction infrastructure, sometimes referred to as a social network, or neighbourhood.

Particles "fly" through the problem search space looking for good solutions. A particle's flight is controlled by iteratively adjusting its trajectory through calculated changes in velocity. The velocity change is calculated by applying a number of rules, which include combining the particles current and previous best flying experience with the best point found by any particle in the surrounding neighbourhood. To prevent premature convergence and encourage exploration, stochastic perturbations are injected into the velocity update rules [77]. See Section 8.1.3 for further details on the PSO algorithm and particle update rules.

[1] Qualification of the "high" is dependent on the problem. For example, in a minimisation problem, a high scoring particle may be the one with a score closest to zero.

Eventually, after successive iterations of the algorithm, the swarm is likely to gravitate to an optimum of the fitness function. This behaviour aligns with Heppner's [117] observations of bird flocks foraging for food.

The neighbourhood strategies that govern how individual particle are connected to one another (a particles "social network") are numerous, however each grouping strategy is engineered using only one of two social interaction principles, namely the global best(*gbest*) principle and the local best (*lbest*) principle [185]. On one hand, the gbest social network topology connects all members of the population to one another, thus individual particles can compare their performance to every other member of the population, imitating the one single "best" particle in the entire population. The gbest principle is known to converge quickly on solutions, however its weakness is found in its propensity to become trapped in local optima [142, pp.344]. On the other hand, lbest neighbourhood strategies allow each individual to be influenced by some smaller number of topological members of the population, allowing particles to "flow around" local optima, as sub-populations explore different regions, striking a careful balance between *exploration* of the search space and *exploitation* of local optima.



(a) Ring ($k = 2$) (b) Wheel

Figure 8.1: Common neighbourhood topologies — ring (b) and wheel (a)

The two most common lbest topologies are Ring and Wheel [140], shown in Figure 8.1.1.1. In the Ring topology (Figure 8.1(a)), each particle communicates with, and is directly influence by, its k adjacent neighbours; it is common for

$k = 2$ in lbest strategies. Particles are assigned to a neighbourhood based on their index. For example, assuming $k = 2$, the social neighbourhood for particle 1 includes particles 8 and 2. Similarly, the social neighbourhood of particle 2 includes particles 1 and 3.In this neighbourhood strategy, influence spreads from neighbourhood to neighbourhood. If one segment of the population converges on a local optimum while another segment converges on a different optimum, if an optimum really is the best found by any part of the population, it will eventually attract all other neighbourhoods towards it. Note that the lbest ring topology is a special case of the gbest ring topology. On the other hand, the Wheel topology (Figure 8.1(b)) prohibits particles from communicating with each other, permitting communication through only a single, focal individual. The central particle compares its performance against the rest of the population and modifies its trajectory towards the best members, thus creating a 'follow the leader' effect. Such a topology slows the speed of transmission of good solutions through the population, thus only the very highest quality areas in the search space will be communicated to the rest of the population. Kennedy [140] tested these topologies using a number of benchmark problems, and results demonstrated differences in performance. However, it should be noted that the difference found was problem dependent, and there was no conclusive evidence to suggest that any one topology was better than another. The interested reader is directed to [144, 178] for detailed discussions on the effects of topology on performance in a PSO.

8.1.2 Anatomy of a particle

Before presenting the PSO algorithm we describe the elements that constitute a particle, as well as the PSO notation used throughout this thesis.

Each particle is composed of five attributes, summarised in Table 8.1. It is worth noting that the "evaluate" behaviour described in Section 8.1.1 is represented by the particle's fitness score — row 4 in Table 8.1.

Attribute	*Definition*
$\overrightarrow{X_i}$	Position vector, which stores the current location of the i^{th} particle in D-dimensional space.
$\overrightarrow{P_i}$	Previous best position found in D-dimensional space.
$\overrightarrow{V_i}$	Particle's current velocity along each of the D-dimensions.
$f(\overrightarrow{x_i})$	Fitness (score) of a particle at position $\overrightarrow{x_i}$.
PB_i^k	Fitness of the best position found so far (i.e. score of $\overrightarrow{p_i}$).

Table 8.1: The five components of a particle.

A particle's current position is denoted by the vector $\overrightarrow{X_i}$, which represents a single point in D-dimensional space, which, in turn, represents a single solution to the problem at hand. Therefore, each of the D dimensions represents a variable in the problem at hand.

At each iteration of the PSO algorithm, particles determine their own fitness, $f(\overrightarrow{X_i})$. If the current point $\overrightarrow{X_i}$ is the fittest solution that the particle has found so far, then its coordinates, $\overrightarrow{X_i}$, are stored in $\overrightarrow{P_i}$ and the corresponding fitness, $f(\overrightarrow{X_i})$, is stored in *pBest*. Thereafter, each particle updates its trajectory by computing a new velocity, $\overrightarrow{v_i^{k+1}}$, which is applied to its current position, $\overrightarrow{X_i}$; this enables the particle to fly to a new position in D-dimensional space.

A more detailed description of the PSO algorithm and the rules used to compute new trajectories is provided in Section 8.1.3.

8.1.2.1 Notation

In this section we present standard notation to describe PSO. The notation presented in this section will be used where appropriate through the remainder of this thesis.

Swarm, or population A swarm (or population) of n particles is denoted as the vector S^k; the superscript, ^{k}, denotes the swarm at the k^{th} iteration. Accordingly, a swarm of n particles at iteration k is denoted as: $S^k = [X_1^k, X_2^k, \ldots, X_n^k]$.

A particle The i^{th} particle, X_i^k, in the swarm at iteration k represents a single point in d-dimensional space. Therefore, X_i^k is defined as: $X_i^k = [x_{i1}^k, x_{i2}^k, \ldots, x_{ij}^k]$, where x's are the parameters (variables) and x_{ij}^k is the position of the i^{th} particle along dimension j at iteration k, where $1 \leq j \leq d$.

Particle velocity Each particle has a velocity at iteration k. Velocity dictates the rate of position change, and therefore enables the particle to fly. The velocity of particle X_i^k along each dimension at iteration k is given by the vector $V_i^k = [v_{i1}^k, v_{i2}^k, \ldots, v_{ij}^k]$. Accordingly, each element, v_{ij}^k, represents the velocity of particle i along dimension j at iteration k. Changes in particle velocity are sensitive: too little velocity and the search will never converge; too much velocity and the particle will pass good solutions and may never converge. To ensure a balance, the velocities are restricted to a user defined range, $V_i^k = [V_{min}, V_{max}]$ — see Equation 8.4 in Section 8.1.3.1.

Personal best, pBest The best position in the search space found by particle i up to iteration k is denoted by the vector PB_i^k. At each iteration, pBest is updated using Equation 8.1.

$$PB_i^{k+1} = \begin{cases} PB_i^k & \text{if } f(X_i^{k+1}) \leq f(BP_i^k) \\ X_i^k & \text{if } f(X^{k+1_i} > f(BP_i^k) \end{cases} \tag{8.1}$$

In other words, if the fitness of the particle at its current position is fitter than the fitness of the current personal best, then set the current personal best to the current position. Otherwise, leave the personal best position unchanged.

Neighbourhood best In section 8.1.1.1 we noted two neighbourhood strategies, namely gBest and lBest. In PSO algorithms that assume a gBest strategy, the neighbourhood best particle is determined by, at each iteration, iterating through every particle in the swarm and selecting the particle with the best *pBest* score. The vector GB^k retains the position vector of the highest achieving particle in the swarm until a better position is found.

In the lBest strategy, the entire swarm is divided into sub-populations; each particle is influenced by other members in its social network. In order to achieve this, each particle has a separate memory store known as *lBest* to stores the best location found by any member of the social network. At each iteration, the personal best position found by each particle is compared against the known best position in the neighbourhood. If there is a higher scoring position found, then $LBEST$ takes a copy the position of PB_i^k. However, if no better score is found, $LBEST$ retains its current vector.

Now that the components of a particle and notation have been defined, we are now positioned to present the original PSO algorithm.

8.1.3 The original Particle Swarm Optimiser algorithm

The original PSO algorithm is simple [141]; it operates in continuous, real number space, requires primitive mathematical operators, makes minimal use of computational resources such as memory and processing power, and it can be implemented in a few lines of computer code [77].

The equations required to update a particle's position in the search space are described in Section 8.1.3.1, and the algorithm that implements the original PSO is presented in Section 8.1.3.2.

Modifications made to the original PSO, including a technique to improve convergence as well as a modification to support binary encoded problems, are introduced in Section 8.1.4.

8.1.3.1 Updating the particles' trajectory

Two equations are required to enable particles to fly to new areas in the search space. The particle must first compute a new velocity, and using the new velocity, fly from its current position to a new position in the search space. The two equations for describing a particle's flying trajectory are:

$$v_{ij}^{k+1} = v_{ij}^{k} + c_1\varphi_1(pb_i^{k+1} - x_{ij}^{k}) + c_2\varphi_2(gb_j^{k} - x_{ij}^{k}) \tag{8.2}$$

$$x_{ij}^{k+1} = x_{ij}^{k} + v_{ij}^{k+1} \tag{8.3}$$

Equation 8.2 defines how particle velocities are updated, and Equation 8.3 defines how the particle flies to a new area of the search space, given the new velocities.

Note that the term gb_j^k in Equation 8.2 assumes denotes a gBest neighbourhood strategy; this can easily be changed to a lBest strategy.

As can be seen from Equation 8.2, the velocity update equation is composed of three parts:

1. Momentum, v_{ij}^k: The particle's velocity is based on its current value to prevent abrupt changes.

2. Cognitive influence, $c_1\varphi_1(pb_{ij}^k - x_{ij}^k)$: This represents private thinking and conservative tendencies. In other words, flight is based on previous flying experience (pBest).

3. Social influence, $c_2\varphi_2(gb_j^k - x_{ij}^k)$: This represents social collaboration between particles i.e. sheep-like (or follower) tendencies. In other words, flying is influenced by experience of the rest of the swarm — in particular, the flying experience of best particle in the neighbourhood.

The values of c_1 and c_2 are used to modulate the cognitive influence and social influence of the particle as it accelerates through the search space, and are initially set to the value 2 [139]. Acceleration in the direction of pBest (c_1) and gBest (c_2) is weighted by two positive, independent random terms, φ_1 and φ_2 respectively, drawn from a uniform distribution in the range $U(0, 1)$. It is these terms that provide the stochastic element in a PSO; the terms weight the particle's individual-learning and social-influence such that sometimes the effect of one, and sometimes the effect of the other, will be stronger. This encourages diversity and exploration. Therefore, when a particle moves towards its best previous position or towards its neighbourhood best position, it moves towards a point which is around the best position. However, it is easy to see that the PSO can become unstable quickly, as particles have the potential to increase speeds uncontrollably, ultimately enabling the particle to leave the search space. To that end, a clamping function, shown in Equation 8.4, is applied to each velocity once it has been calculated to ensure that it remains within a specified range $[V_{min}, V_{max}]$. Usually, $V_{min} = -V_{max}$.

$$restrict(v_{ij}^k) = \begin{cases} V_{max} & \text{if } v_{ij}^k > V_{max} \\ v_{ij}^k & \text{if } v_{min} \leq v_{ij}^k \leq V_{max} \\ V_{min} & \text{if } v_{ij}^k \leq V_{min} \end{cases} \tag{8.4}$$

Applying hard bounds on velocities should be done carefully for two reasons: 1) there is no optimal value for $\pm V_{max}$ — the value of $\pm V_{max}$ is very much problem dependent; however, there is no rule of thumb for deriving $\pm V_{max}$ [221]; and 2) incorrect selection of $\pm V_{max}$ has the potential to allow particles to fly far past good solution areas, and small $\pm V_{max}$ has the potential to trap particles into local minima [77]. This issue is discussed further in Section 8.1.4.1.

Once the velocity along each dimension is calculated, the particle flies to the next point in the search space using Equation 8.3.

For a deeper discussion on parameter selection in PSO, the interested reader is directed to [46, 163, 32, 246, 79, 210].

8.1.3.2 The Particle Swarm Optimiser algorithm

The PSO algorithm consists of repeated applications of the trajectory update equations presented in Section 8.1.3.1. The original PSO algorithm is shown in

Figure 8.2 as a process flow diagram, and the corresponding pseudocode is given in Algorithm 2.

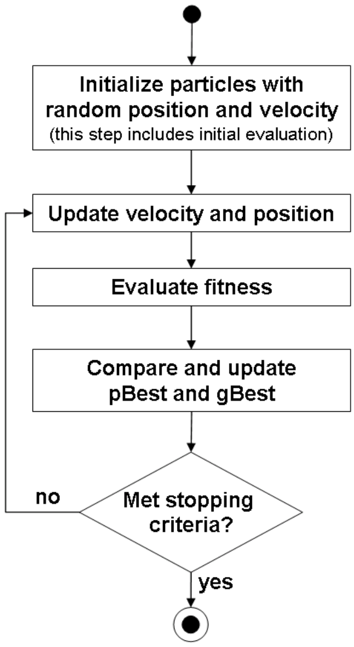

Initialize particles with random position and velocity
(this step includes initial evaluation)
Update velocity and position
Evaluate fitness
Compare and update pBest and gBest
Met stopping criteria?
no
yes

Figure 8.2: Particle Swarm Optimisation flow diagram.

Algorithm 2 Original particle swarm algorithm pseudocode

```
Create and initialise a swarm of $d$-dimensional particles: S
repeat
    for each particle $X_i^k \in [1, \dots, |S|]$: do
        Perform update using equations 8.2, 8.4 and 8.3
        if $f(S.X_i^k) < f(pBest_{S.X_i^k})$ then
            $pBest_{S.X_i^k} = S.X_i^k$
        end if
        if $f(pBest.X_i^k) < f(gBest_{S.X_i^k})$ then
            $gBest_{S.X_i^k} = X_i^k$
        end if
    end for
until termination criterion is met
```

Creation and initialisation of each particle in the swarm consists of 3 steps:

1. Initialise the particle's n-dimensional position vector, X_i, using a uniformly distributed random number generator in the range X_{min}, X_{max} for each

dimension, where $\pm X_{max}$ is the bounds of the dimension (variable) at hand. This encourages the particle to scatter its initial positions throughout the search space.

2. Initialise the particle's n-dimensional velocity vector, V_i, to zero. Alternatively, a uniformly distributed random number generator can be used in the interval $\pm V_{max}$.
3. Copy the initial position vector, X_i, to the particle's pBest vector: $pBest \leftarrow X_i$.

Optimisation algorithms have many different termination criteria options. In many cases, however, the termination criteria are problem dependent. Common termination strategies include running the algorithm for a fixed number of fitness function evaluations or running the algorithm until a solution is found with a fitness within a pre-specified error threshold.

8.1.4 Modifications to the original PSO

A number of improvements have been made to the original PSO algorithm, the most pertinent being enhancements to the convergence capability, and the introduction of a binary version of the PSO algorithm. These two topics are discussed in Sections 8.1.4.1 and 8.1.4.2 respectively.

8.1.4.1 Improving search scope and convergence

In Section 8.1.3.1, one of the two caveats concerning the use of V_{max} was the rate of convergence. In particular, premature convergence on local optima when V_{max}

is low. A number of modifications have since been made to the PSO algorithm in order to gain better control of the scope of the search and enhance its convergence capability.

One of the most common enhancements is the *inertia weight* term, ω, proposed by Shi and Eberhart [245]. The inertia weight is a scaling factor that governs the amount of the previous velocity that should be retained when calculating the new velocity. The change required in the algorithm to support inertia is not structural; rather, the velocity update equation is modified to incorporate the inertia weight. Therefore, Equation 8.2 becomes

$$v_{ij}^{k+1} = \omega \cdot v_{ij}^{k} + c_1\varphi_1(pb_i^{k+1} - x_{ij}^{k}) + c_2\varphi_2(gb_j^{k} - x_{ij}^{k}) \tag{8.5}$$

It is possible to restore the original PSO velocity update equation by setting $\omega = 1$.

In most PSO implementations that use inertia weight, the inertia is decreased linearly over time, and it starts with an initial value close to 1 [246, 79]. The inertia weight has the desired effect of modulating the particle's velocity in such a way that it balances exploration and exploitation. At the start of the search the algorithm favours global exploration, as the inertia weight is high, therefore permitting the particle to accelerate up to its maximum velocity. As the inertia decreases, the particle's velocity decreases towards zero enabling it to refine the search and become more exploitative.

Various empirical experiments have been conducted to investigate the effect of different inertia weight values. Shi and Eberhart [78] investigated the effect of ω in the range $[0, 1.4]$. They found that the PSO converged quicker when ω took on values in the range $[0.8, 1.2]$. However, when $\omega > 1.2$, Shi and Eberhart found that the PSO reached convergence less often. In later experiments, it was found that a linear decrease in ω from 0.9 to 0.4 was most effective on four different problems [79]. Note that these ranges may vary between problems.

Relative to other nature-inspired search heuristics, for example Genetic Algorithms, PSO has fewer parameters. However, it remains for the required parameters to be set, and like other algorithms, the choice of parameters is very much problem dependent. A number of theoretical and empirical studies appear in the PSO literature to assist users in parameter selection. The interested reader is directed to [79, 77, 266, 246, 247, 82, 78]

8.1.4.2 The binary PSO algorithm

The original PSO algorithm proposed by Kennedy and Eberhart was designed for solving real-value problems. However, with slight modifications, the original PSO algorithm is capable of solving problems that are binary in nature [143].

In Kennedy and Eberhart's version of binary PSO, a particle's position vector is restricted to values in the set $\{0, 1\}^d$, though no restriction is applied to particles' velocity vector.There is no change to the original velocity update equation (Equation 8.2), however the notion of velocity in the binary PSO is different to that of the real-value version. In the real-value version, velocity is interpreted as a rate of change; it is clear, though, that this makes no sense in a binary space. Therefore, in Kennedy and Eberhart's binary PSO, velocity is interpreted as a probability threshold which is used to determine whether x_{ij}^k — the j^{th} component of x_i^k, a bit — should be in one state or another (0 or 1). Once the velocity update equation is applied,

$$v_{ij}^{k+1} = v_{ij}^{k} + c_1\varphi_1(pb_i^{k+1} - x_{ij}^{k}) + c_2\varphi_2(gb_j^{k} - x_{ij}^{k}) \ , \tag{8.6}$$

the new velocity, v_{ij}^{k+1}, is converted to a probability using a logistic transform function (shown graphically in Figure 8.3), defined as

$$sig(v_{ij}^{k+1}) = \frac{1}{1 + exp(-v_{ij}^{k+1})} \tag{8.7}$$

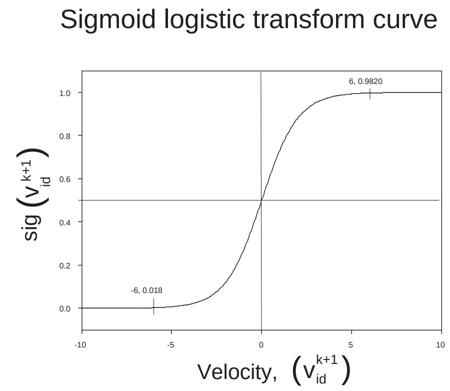


Figure 8.3: Sigmoidal logistic transform graph.

Although the original velocity update equation does not change, the position update equation differs from the original (Equation 8.3), and is defined as

$$x_{ij}^{k+1} = \begin{cases} 0 & \text{if } \varphi_3 \geq sig(v_{ij}^{k+1}\ ) \\ 1 & \text{if } \varphi_3 < sig(v_{ij}^{k+1}\ ) \end{cases} \quad , \tag{8.8}$$

where φ_3 is a random number drawn from a uniform distribution in the range $U[0, 1]$. In other words, generate a random number φ_3 for each bit-string site and compare it to the squashed velocity, $sig(v_{ij}^{k+1})$, for that site. For example, for each bit-string site, if φ_3 is less than $sig(v_{ij}^{k+1})$, then x_{ij}^{k+1} is 1, otherwise x_{ij}^{k+1} is 0.

Analysis of the logistic transform function used to squash the velocity into a probability shows that the position of the particle fixes on $x_{ij}^{k+1} = 0$, with less chance of change, as $sig(v_{ij}^{k+1})$ approaches 0. In other words, when $v_{ij}^{v+1} \lesssim -10$ [143]. For example, if $v_{ij}^{v+1} = -11$, $sig(v_{ij}^{k+1}) \approx 0.00002$, rounded $sig(v_{ij}^{k+1}) = 0$; since φ_3 generates positive random real numbers, application of Equation 8.8 for any value of φ_3 results in $x_{ij}^{k+1} = 0$. On the other hand, the sigmoid function becomes saturated when $v_{ij}^{v+1} > 10$, resulting in a greater probability of $x_{ij}^{k+1} = 1$.

In order to prevent $sig(v_{ij}^{k+1})$ becoming too close to 0.0 or 1.0, the velocities are clamped using a constant parameter V_{max} — the same as the real-value PSO in Section 8.1.3.1. Kennedy and Eberhart [142, pp. 296] recommend clamping the velocities in the range $[-4, 4]$. As can be seen from the logistic transform graph shown in Figure 8.3, the probability of state change is reduced to the range between ≈ 0.018 and ≈ 0.9820. This means that there is always a chance (≈ 0.018) that a bit will change state, which Kennedy and Eberhart [142, pp. 296] equate to a mutation rate in Genetic Algorithms.

8.2 Motivation for using Particle Swarm Optimisation

Recall from Section 7.1 that the primary problem with Bayesian network (BN) discovery from data is the size of the search space, as it grows with the number of variables in the problem domain — see Section 7.1. There is a plethora of literature which documents the ability of nature inspired algorithms to search effectively and produce good results in problems characterised by massive, high-dimensional, complex search spaces — problems such as BN discovery from data. Researchers who have looked at this subject (in the context of BN structure discovery) include Larrañaga et al. [155], who proposed GAs for BN structure learning. Other researchers include Romero et al. [232], de Campos et al. [62] and Castro et al. [35] who applied Estimation of Distribution Algorithm (EDA), Ant Colony Optimisation (ACO) and Artificial Immune System (AIS) to BN structure learning, respectively. They have all demonstrated that nature-inspired algorithms yield successful results for BN structure discovery from data.

More recently, Particle Swarm Optimisation (PSO) has emerged as a nature-inspired search and optimisation heuristic. Evidence from the literature suggests that, for certain problems, PSO is superior to its GA counterpart. For example, Petrovski et al. [219] compare GAs and PSO techniques in the evolution of optimal chemotherapy schedules for patients suffering from cancer. Their results concluded that PSO was able to find feasible regions for possible solutions faster than GAs. In addition, the PSO algorithm was successful in finding better solutions to the problem than the GA approach. Another example is the work of Mouser and Dunn [189]. In their research, they show that PSO outperforms GAs in optimal design of aircraft. Hassan et al. [109] examined the claim that PSO has the same effectiveness (finding the true global optimal solution) as the GA, but with significantly better computational efficiency (fewer function evaluations). Kennedy and Spears [145] compared binary PSO and the GA on a number of different randomly generated multimodal problems. In that study, the binary PSO was superior in performance to the GA — the binary PSO found the global optimum on every trial, regardless of the problem features; the same could not be said for the GA.

A possible explanation for the improved performance of the PSO over the GA may be found in the underpinning mechanics of the respective algorithms. The PSO has strong sociological underpinnings that promotes information sharing and, in addition, it has the capacity to store a limited amount of information about good, previously explored areas of the search space. Using its social communication principles, it has the ability to communicate this information between particles at each iteration. GAs, on the other hand, do not have the ability to conserve information from "ancestors". Learning from peers is limited to application of genetic operators on the individuals in the population at the current iteration (i.e. there is no interaction across all candidate solutions).

Since PSO has shown evidence of enhanced performance over GAs on a selection of problems, we seek to determine PSO's performance on the task of BN structure discovery, and compare the results to work previously done using GAs for BN structure discovery. Before presenting our approach, though, we present some of the current approaches PSO approaches to BN structure learning.

8.3 Existing PSO-based structure learning approaches

As mentioned in Section 8.2, PSO has been shown to be an effective algorithm for searching through complex, high-dimensional search spaces. In addition, research has demonstrated that, on some problems, PSO is more effective in terms of computational cost and convergence to an optimal solution when compared with other nature-inspired search heuristics (for example, the GA). These characteristics have encouraged active research into the application of PSO for BN structure learning, and as a result, a number of papers have been published. It would appear that this element of our research has been actively pursued by others in the field at the same time. This is encouraging, as it would indicate that others in the field view this approach worthy of investigation. We will review some of the recent literature in this section.

Du et al. [76] use PSO to explore Directed Acyclic Graphs (DAG) space, where each particle represents a single "point" in the space (i.e. a candidate BN structure), and is encoded using an n x n connectivity matrix, where each element in the matrix represents a dimension in the particle, and n is the number of nodes in the problem. The presence of a "$0''$ in any element indicates no edge, and a "1" indicates a connecting edge. In this implementation, velocities are discrete $\{-1, 0, 1\}$. Therefore, when calculating the velocities for each position, the resulting value indicates whether the edge represented by the position in question is to be removed, added or left unchanged. An example is shown in Table 8.2. In the example, dimension 1 for an arbitrary particle is examined. The personal best value for dimension 1 is shown in the first column, and the current value of the particle along dimension 1 is shown in the second column. The new cognitive portion of the velocity for dimension 1, as per Equation 8.2, is shown in the third column; a translation for the new velocity is given in column four.

PB_{i1}^k	x_{i1}^k	$(pb_{i1}^k - x_{i1}^k)$	Translation
0	0	0	unchanged
0	1	-1	remove
1	0	1	add
1	1	0	unchanged

Table 8.2: Illustration of Du et al. velocity update.

The velocity is applied to the particle's current position in order to move it to a new point in the search space. If the position currently represents no edge ($x_{i1}^{k} = 0$), when $v_{i1}^{k} = 1$, then a new edge would be added at the position ($x_{i1}^{k+1} = 0$); otherwise, the edge stays present or is removed, i.e. $x_{i1}^{k+1} = 0$.

As can be seen from Section 8.4, our approaches are based on the binary PSO, which bases velocity update on the notion of a probability of change for each position, and so our position update rules are different to those used by Du et al.

Another PSO approach is that of Heng et al. [115, 114]. The encoding scheme is different to that mentioned above. In this approach, each position in the particle encodes a variable, and the contents of the position is the list of those variables that are direct parents. A set of discrete operators are defined, which, when applied to a position instruct the particle whether to add a variable, remove a variable or take no action. These operators are known as Switch Operators, and a list of such Switch Operators can be applied to a single position. The interested reader is directed to [115] for a detailed description.

Sahin et al. [236, 235] exploit the advantages of the parallel nature of PSO. In their implementation, they use a distributed binary PSO algorithm across 48 processors. The approach is very similar to ours in that it is implemented as a binary PSO; however, the benefit here is that the computational expensive aspect of the search, namely computing the fitness score, is distributed across many processors in parallel. They apply their algorithm to airplane engine fault diagnosis.

Other PSO-based approaches to BN structure learning include [39, 53].

There are a number of differences and similarities between these approaches and the approaches that we have developed. Firstly, we represent and encode our solutions using a connectivity matrix. In one case as an nxn connectivity matrix, and in the other case the upper triangulated matrix complete with a concatenated permutation. Secondly, we explore the space of DAGs; in the first case we repair solutions that do not respect DAG constraints, and in the other case we do not need to repair at all. Common to our approaches and the approaches discussed here is the use of a Bayesian fitness metric, typically the Cooper-Herskovits [51] metric, which provides the particles with an objective scoring function to evaluate their positions as they fly through the search space.

8.4 Proposed approach PSO

In this section we present an overview of our approach to Bayesian network (BN) structure induction driven by a Particle Swarm Optimiser. Detailed descriptions of the two binary PSO-based algorithms developed are provided in Section 8.5 and Section 8.6.

Our approach to BN structure discovery is based on the search and score paradigm, described in Section 7.3, and uses binary PSO. It is worth noting that our approach is closest to Du et al. [76] in terms of solution representation and in terms of how the algorithm functions. In the first of our approaches, we use the binary PSO algorithm as the search engine to explore the space of BN structures. In our implementation, we represent solutions using the full $n \times n$ connectivity matrix. This method for solution representation, however, requires a mechanism to repair solutions that are invalid due to cycles, caused by the PSO's stochastic update algorithm. Since this algorithm first generates solutions and then repairs those that are invalid, we have named the algorithm CONAR (CONstruct And Repair), and it is described in detail in Section 8.5.

In our second approach, we remove the the need for a repair operator. In this algorithm, we propose the use of only the upper triangular portion of the $n \times n$ binary connectivity matrix to represent solutions. In doing so, we can exploit a solution representation strategy that restricts generation of illegal BN structures, thus produces only legal solutions. We have named this algorithm REST (REstricted STructure), and it is described in detail in Section 8.6.

We propose the Cooper and Herskovits [51] BN scoring metric (K2), described in Section 7.3.3.2, expressed in terms of the natural logarithm, to score candidate BNs. At each iteration of the PSO algorithm, the K2 metric is used to evaluate the quality of the generated solutions. Therefore, our aim is to find the structure with highest probability of representing the probabilistic relations embedded in the data set — the one with the K2 score closest to zero. The K2 scoring metric was adopted to enable us to compare our research with similar work on BN structure discovery using nature-inspired methods, such as [135, 16, 62, 155, 156].

Details of the two algorithms, CONAR and REST, have been published in [57].

Before the two algorithms are presented, we first give an overview of the core representation strategy that we propose to encode BN structures in both CONAR and REST.

8.4.1 Bayesian network representation

Each individual particle in the swarm represents a single BN structure; each BN structure is encoded in a $n \times n$ binary connectivity matrix. In a n-dimensional domain, where n is the number of domain variables, a particle encodes the flattened binary connectivity matrix, denoted as C, such that,

$$C[i,j] = \begin{cases} 1 & \text{if } x_i \rightarrow x_j \\ 0 & \text{otherwise} \end{cases}, \tag{8.9}$$

where i infers the row, and j the column.

As alluded to in Section 8.4, the use of a $n \times n$ binary connectivity matrix gives rise to two different representation strategies:

1. The full, $n \times n$ connectivity representation in which all nodes of the network can be parents of all other nodes in the domain. This is the representation scheme used by CONAR.

2. The upper triangular connectivity matrix representation which requires an order among the variables such that a node has the potential to be a parent of only nodes preceding it in the proposed ordering. This representation is used by REST.

8.5 CONstruct And Repair (CONAR)

The CONAR algorithm is based on the search and score paradigm for BN discovery, as described in Section 7.3. The algorithm performs a search in the space of BN structures looking for the BN that best models the probabilistic relations embedded in the data set. As noted earlier, in this research we use the $K2$ metric proposed by Cooper and Herskovits [51] as the scoring function to evaluate the quality of individual particles (solutions).

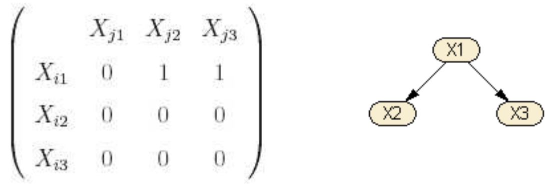
8.5.1 Solution representation

In CONAR, particles encode the flattened, full $n \times n$ binary matrix, where each dimension of the particle maps to one element in the matrix. For example, the flattened matrix for an arbitrary BN with n variables, with respect to Equation 8.9, is defined as

$$C_{11}C_{12}\ldots C_{1n}C_{21}C_{22}\ldots C_{2n}\ldots C_{n1}C_{n2}\ldots C_{nn} \tag{8.10}$$

Once a particle's velocity is updated and it moves to a new point in the search space, the particle may well represent an invalid BN solution. This is because the encoding scheme permits all nodes to be parents of other nodes in the domain, and therefore does not permits cycles — such solutions do not respect Directed Acyclic Graph (DAG) conditions — see Section 2.1.

For example, consider a domain of $n = 3$ variables, and an arbitrary binary particle, x_i, with $n \times n = 9$ dimensions set as follows: [011000000]. The populated matrix and corresponding BN structure is shown in Figure 8.4.

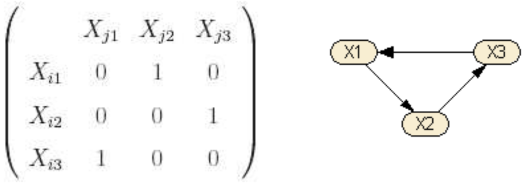


$$\left(\begin{array}{cccc} & X_{j1} & X_{j2} & X_{j3} \\ X_{i1} & 0 & 1 & 1 \\ X_{i2} & 0 & 0 & 0 \\ X_{i3} & 0 & 0 & 0 \end{array}\right)$$

(a) Matrix before update (b) BN before update

Figure 8.4: A BN solution in CONAR, before update.

After a single iteration of the PSO, x_i occupies a new point in the search space: [01001100]. The updated matrix and corresponding BN structure is shown in Figure 8.5.



$$\left(\begin{array}{cccc} & X_{j1} & X_{j2} & X_{j3} \\ X_{i1} & 0 & 1 & 0 \\ X_{i2} & 0 & 0 & 1 \\ X_{i3} & 1 & 0 & 0 \end{array}\right)$$

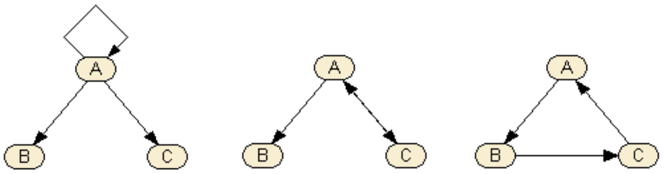
(a) Matrix after update (b) BN model after update

Figure 8.5: A BN solution post CONAR update.

It is clear, however, that the new solution is illegal as it corresponds to a cyclic graph: $X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow X_1$. We introduce a simple *repair operator* to transform DAGs that violate DAG conditions back into legal DAGs. This is achieved by repeatedly eliminating (at random) edges that break DAG conditions until a legal DAG is recovered. Section 8.5.1.1 describes in detail the cycle detection and elimination strategies employed.

8.5.1.1 Validation and repair in CONAR

In the CONAR algorithm, once particles fly to a new position in the search space, their position is verified to ensure that it is legal. In other words, the solution that the particle represents is validated in that it respects Directed Acyclic Graph (DAG) conditions — see Section 2.1. If DAG conditions are violated, then the particle is repaired by randomly eliminating cycle-causing edges. CONAR identifies cycles in a 3-stage process: firstly, self-referencing edges (or self-cycles) are detected, followed by bi-cycles, and lastly regular-cycles. The method used to determined the presence of cycles and the strategies employed to repair them are discussed in this section.



(a) Self cycle (b) Bi cycle (c) Regular cycle

Figure 8.6: Cycles caused using the full $n \times n$ representation.

$$\left(\begin{array}{cccc} & A_j & B_j & C_j \\ A_i & 1 & 1 & 1 \\ B_i & 0 & 0 & 0 \\ C_i & 0 & 0 & 0 \end{array}\right) \quad \left(\begin{array}{cccc} & A_j & B_j & C_j \\ A_i & 0 & 1 & 1 \\ B_i & 0 & 0 & 0 \\ C_i & 1 & 0 & 0 \end{array}\right) \quad \left(\begin{array}{cccc} & A_j & B_j & C_j \\ A_i & 0 & 1 & 0 \\ B_i & 0 & 0 & 1 \\ C_i & 1 & 0 & 0 \end{array}\right)$$

(a) Self cycle (b) Bi cycle (c) Regular cycle

Figure 8.7: Matrices for BNs shown in Figure 8.6.

Self-cycles A self-referencing cycle (or self-cycle) occurs when a node has an edge that points directly to itself. An example of a self-cycle is shown in Figure 8.6(a), and the corresponding $n \times n$ binary matrix is shown in Figure 8.7(a). Detecting self-cycles is trivial. The presence of a '1' in any element along the diagonal of the $n \times n$ matrix identifies a self-cycle. The repair strategy simply replaces any '1' in the diagonal with a '0'.

Bi-cycles A bi-directional cycle (or bi-cycle) occurs when two nodes in a BN structure appear to influence each other. In Figure 8.6(b), we see the scenario where Node A influences node C, and in turn, Node C influences Node A. The corresponding matrix is shown in Figure 8.7(b). Bi-cycles are detected by, for each edge in the upper triangular portion of the $n \times n$ matrix, checking the corresponding reverse edge. For example, if element $C_{21} = 1$ and $C_{12} = 1$, then clearly a bi-cycle exists, therefore one of the edges is selected at random and removed. It should be noted that CONAR does not make an informed decision about which edge to remove, therefore there is a 0.5 probability that an optimal edge is lost.

Regular cycles Cycles spanning ≥ 3 nodes are referred to as regular cycles, as shown in Figure 8.6(c). The $n \times n$ binary matrix for this example is shown in Figure 8.7(c), where n is the number of variables in the domain. Node A influences node B, Node B influences Node C, and Node C influences Node A. Such cycles are identified using Warshalls algorithm [276], which calculates the shortest path between pairs of variables. In doing so, it is possible to calculate whether a path exists from Node $X_i \rightarrow X_i$, which therefore indicate the existence of a cycle. Therefore, to detect a regular cycle from a, Warshall's algorithm is run to check, for each node, if there's a path from X_i to X_i, with computational complexity $O(n^3)$. Note that this computational complexity is to detect a regular cycle from a single node, and not all n nodes. Sub-graphs containing regular cycles are repaired in the same fashion as the bi-cyclic sub-graphs, where offending edges are removed at random until the solution respects DAG conditions.

8.5.2 Algorithm

The CONAR algorithm makes use of the canonical binary PSO algorithm, with an extra step concerned with validation and repair; this is shown in Algorithm 3. The following two points are worth noting:

1. The create and initialise swarm, step 1 in Figure 3, is as per Section 8.1.3.2
2. The initial dimensions are set such that they correspond to a legal DAG

Algorithm 3 Pseudocode for CONAR

```
Create and initialise a swarm of (valid) $n^2$-dimensional particles: S
Select arbitrary gBest
repeat
    for each particle $X_i \in [1, \dots, |S|]$: do
        Perform update using equations 8.6, 8.7 and 8.8
        if isIllegal $(S.X_i)$ then
            $S.X_i \leftarrow repair(S.X_i)$
        end if
        if $f(S.X_i) > f(pBest_{S.X_i})$ then
            $pBest_{S.X_i} = S.X_i$
        end if
        if $f(pBest.X_i) > f(gBest_{S.X_i})$ then
            $gBest_{S.X_i} = pBest.X_i$
        end if
    end for
until termination criterion is met
```

Although the CONAR algorithm uses the basic canonical binary PSO algorithm with simple repair strategies for repairing illegal structures, we feel that this proposition makes for a good starting point to evaluating the potential of PSO for BN structure learning.

8.6 REstricted STructure (REST)

The REST algorithm is almost identical to CONAR; the primary difference is the solution representation, which is modified to guarantee generation of only legal solutions, therefore eliminating the need for validation and repair operators. REST uses binary Particle Swarm Optimisation to perform a search in the space of legal BN structures, and it uses the same scoring function ($K2$) as CONAR to evaluate the quality of candidate solutions.

8.6.1 Solution representation

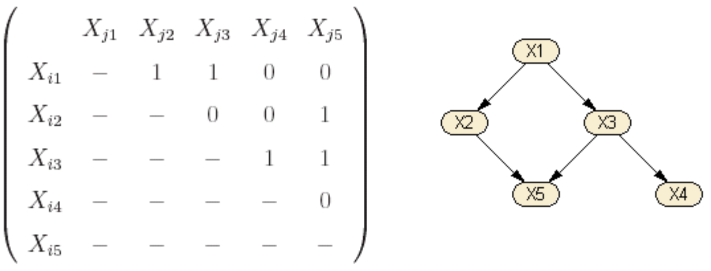
It is possible to guarantee a legal BN structure by encoding the solution using the upper triangular portion of an $n \times n$ (where n is the number of variables) connectivity matrix [54, 155, 16]. In this representation, particles encode the flattened upper triangular portion of the $n \times n$ binary matrix, where each dimension of the particle maps to one element in the matrix. The flattened matrix for an arbitrary BN with n variables, with respect to Equation 8.9, is defined as

$$C_{12}C_{13}C_{14}\ldots C_{1n},\ldots C_{23}C_{24}\ldots C_{n2},\ldots C_{n-2n-1},C_{n-2n},C_{n-1n} \tag{8.11}$$

Therefore, the number of elements required to store an arbitrary BN structure with n variables is defined as

$$\frac{n(n-1)}{2} \tag{8.12}$$

An example of this representation scheme is as follows. Consider a domain of $n = 5$ variables, and a binary particle, X_i, with $\frac{n(n-1)}{2} = 10$ dimensions. Assume that the dimensions of the particle are set as follows: [1100001110]. The corresponding matrix and the resulting BN structure is shown in Figure 8.8.



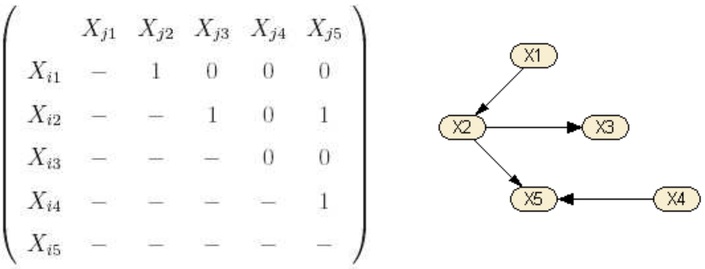
$$\left(\begin{array}{cccccc} & X_{j1} & X_{j2} & X_{j3} & X_{j4} & X_{j5} \\ X_{i1} & - & 1 & 1 & 0 & 0 \\ X_{i2} & - & - & 0 & 0 & 1 \\ X_{i3} & - & - & - & 1 & 1 \\ X_{i4} & - & - & - & - & 0 \\ X_{i5} & - & - & - & - & - \end{array}\right)$$

(a) Triangulated matrix

(b) BN model

Figure 8.8: Triangulated $n \times n$ representation.

Imagine that particle X_i has values [1000101001] after a single (stochastic) iteration of the PSO algorithm; the updated matrix and corresponding BN structure is shown in Figure 8.9. Notice that the DAG remains valid.



$$\left(\begin{array}{cccccc} & X_{j1} & X_{j2} & X_{j3} & X_{j4} & X_{j5} \\ X_{i1} & - & 1 & 0 & 0 & 0 \\ X_{i2} & - & - & 1 & 0 & 1 \\ X_{i3} & - & - & - & 0 & 0 \\ X_{i4} & - & - & - & - & 1 \\ X_{i5} & - & - & - & - & - \end{array}\right)$$

(a) Matrix after update

(b) BN model after update

Figure 8.9: Triangulated $n \times n$ after update.

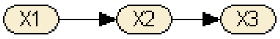
The issue with this approach is that the representation scheme is restrictive in that the search becomes a search in the space of BNs that admit a specific order. In other words, it is not possible to encode any arbitrary BN structure. For example, in Figure 8.8 and 8.9, the order of the variables in the matrix dictates the possible parent set for each variable. An order, by definition, states that a node X_j can only have node X_i as a parent node if, in the order, node X_i comes before node X_j [232]. For example, the edges $X_1 \rightarrow X_2, X_1 \rightarrow X_3$ and $X_3 \rightarrow X_4$ are permitted; however, the edges $X_3 \rightarrow X_2$ and $X_4 \rightarrow X_1$ are not permitted.

In order to make the representations scheme flexible, and provide the ability to represent any legal directed acyclic graph, we append one of the possible $n!$ order permutations to create a complete representation of the BN structure. Romero et al. [232] use a similar encoding approach; however, they seek to find the optimal ordering for the K2 algorithm, rather than the optimal BN structure. Our encoding scheme is best understood by example.

Consider a domain consisting of $n = 3$ variables, and an order X_1, X_2, X_3. An example is shown in Figure 8.10.

$$\left(\begin{array}{cccc} & X_{j1} & X_{j2} & X_{j3} \\ X_{i1} & - & 1 & 0 \\ X_{i2} & - & - & 1 \\ X_{i3} & - & - & - \end{array}\right)$$

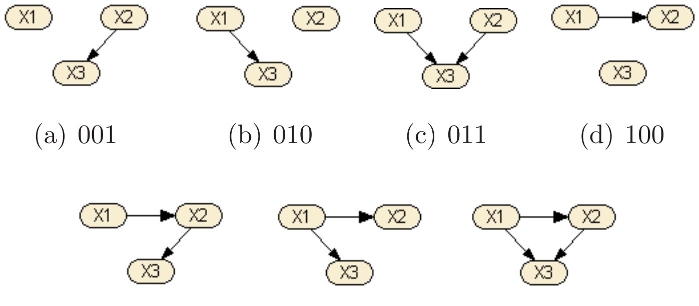
(a) Connectivity matrix



(b) Corresponding BN model

Figure 8.10: Triangulated $n \times n$ after update.

Given that the encoding has $\frac{n(n-1)}{2}$ bits, and each bit represents a different structure, the total number of possible structures for 3 variables, ordered as X_1, X_2, X_3, is $2^{n(n-1)/2} = 8$. The 7 of the 8 structures are shown in Figure 8.11 — the 1^{st} candidate structure, that is the one with no edges [000], has been omitted. The matrix and structure shown in Figure 8.10 corresponds to the structure shown in Figure 8.11(e).



(a) 001 (b) 010 (c) 011 (d) 100

(e) 101 (f) 110 (g) 111

Figure 8.11: All BN structures admitting order X_1, X_2, X_3.

To make this representation flexible, we attach one of $n!$ systematically generated permutations (shown in Table 8.3) to the triangulated matrix encoding. Together, the binary connectivity encoding and permutation provide a specification that allows representation of any legal BN structure, given n number of variables. Since the encoding mechanism is composed of both a flattened triangulated matrix and a permutation, the full encoding bit string length becomes $\frac{n(n-1)}{2} + permbits(n!)$, where $n!$ is an integer stored in a binary representation that corresponds to the number of permutations for n variables, and $permbits(n!) = \lceil \log_2(n!) \rceil$ is the maximum number of bits required to store $n!$ in a binary representation.

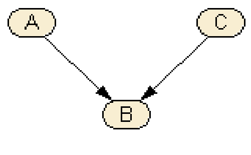
000 = 0 = ABC	001 = 1 = ACB	010 = 2 = BAC
011 = 3 = BCA	100 = 4 = CAB	101 = 5 = CBA

Table 8.3: Table of permutations for $n = 3$ variables .

In this example, with $n = 3$ variables, the connectivity portion requires $\frac{3^2-3}{2} = 3$ bits, and the permutation portion requires $\lceil \log_2(n!) \rceil = \lceil \frac{\log(n!)}{\log(2)} \rceil = 3$. Accordingly, 6 bits are required to represent solutions. Consider a particle, X_i with 6 dimensions set to [011101]. The first 3 bits ([011]) define the connectivity, and the remaining 3 bits ([101]) define order. Converting the order bit-string ([101]) to an integer, we get 5; the corresponding k^{th} permutation, where $k = 5$ and the index based at zero, see Table 8.3, is CAB. The connectivity matrix (which admits the order CAB) and the resulting BN structure is shown in Figure 8.12.

$$\left(\begin{array}{cccc} & C_j & A_j & B_j \\ C_i & - & 0 & 1 \\ A_i & - & - & 1 \\ B_i & - & - & - \end{array}\right)$$

(a) Connectivity matrix



(b) Decoded BN model

Figure 8.12: An example of the flexible binary encoded BN.

Combining the connectivity matrix and the permutation bit-string provide a complete binary specification for a BN structure. As such, the PSO algorithm can optimise the encoded bit-string without the need for validation or repair. The role of the PSO algorithm, therefore, is to find the area of the search space that corresponds to a good order permutation and connectivity.

8.6.2 Algorithm

Since REST does not require any validation or repair steps, the algorithm is identical to CONAR without steps $6 - 8$, shown in Figure 3

8.7 Summary

In the previous chapter we introduced the notion of BN construction from data, identified existing data-driven algorithms, and highlighted the issues and challenges with existing algorithms. In this chapter we presented a new algorithm for BN construction from data and illustrated in detail its operation. The new algorithm, binary PSO, seeks to address some of the issues found in other algorithms.

Two algorithms are proposed in this chapter, namely CONAR and REST, and are both based on binary PSO. CONAR serves to demonstrate that binary PSO can be used as a search heuristic for BN construction, and that there is no need to specify an order among the nodes or search in a two-tiered search space. However, CONAR requires expensive validation and repair operators to ensure the integrity of candidate solutions. With a view to alleviating the validation and repair requirements, the representation mechanism employed by CONAR was modified, which resulted in REST.

In the next chapter, the algorithms are evaluated empirically against a number of performance metrics, and are compared against the genetic algorithm approaches for BN discovery from data.

Chapter 9

Experimental evaluation

9.1 Introduction

In this chapter we analyse, evaluate and compare the performance of the Bayesian network (BN) learning algorithms described in Chapter 8 with a view to addressing three questions:

1. What is the ability of PSO to search the entire space of Bayesian network structures without the need to learn an order, and, in addition, can the variable order requirement be suppressed completely?
2. What is the performance capability of the algorithms developed compared against each other, and how do they compare to other algorithms in the literature (both order-based algorithms and algorithms that explore the whole space)?

For the purpose of experimentation and evaluation, we use a number of synthetic data sets from the literature, which are commonly used for the purpose of evaluation and comparison. In addition, we examine the performance of our algorithms on real-life clinical data pertaining to dementia diagnosis.

We begin this chapter in Section 9.2 by defining and describing the data sets used, performance metrics and experimental parameters used in experimentation. An empirical evaluation of CONAR and REST on the synthetic and dementia data is provided in Section 9.3 and 9.4 respectively. We end with a summary of the results in Section 9.5. Note that this chapter does not evaluate the BN construction approach; rather, it presents the experimental results. A review of the approach, along with an review of the difference in the dementia models across construction approaches, is treated in Chapter 10.

9.2 Experimental design

For the purpose of assessing the performance of the BN discovery algorithms using known models, and to provide a mechanism to compare BN construction approaches described in Chapter 4 and Chapter 7, we have used two groups of test data. One group consists of synthetically generated data using a known solution, which enables testing the performance of the algorithm. The other group of data consists of real-life clinical data, which allows us to test the algorithms on a "real-life" problem as well as obtain models for the purpose of comparing construction approaches.

9.2.1 Synthetic test problems and databases

A common approach to evaluating BN learning algorithms involves generating a synthetic database from a pre-specified network, applying the learning algorithm to the synthetic data set, then comparing the learned network with the original one. A description of the benchmark BN problems and generated databases used in our research is given below (in order of complexity).

1. *Asia* The Asia database represents an artificial problem relating to medical diagnosis. It was proposed proposed by Lauritzen and Spiegelhalter [161], and it is commonly used to demonstrate the workings of BNs. The model is basic, consisting of 8 binary nodes and 8 edges, as shown in Figure 9.6(a). A 'learning' data set consisting of $5,000$ cases is generated using Netica [2] (a BN software tool).

2. *Car* This is a demonstrative BN, which represents the working relationships between parts of a car thus facilitating fault diagnosis. As can be seen from figure 9.9, the Car BN consists of 18 nodes and 17 edges. Many versions of the Car problem exist — we have chosen the version supplied by BayesiaLab [1], from which $10,000$ cases were generated.

3. *ALARM* The ALARM network, proposed by Beinlich et al. [11], is a medical decision support tool for monitoring patients in intensive care. The network consists of 37 nodes and 46 edges, and is shown in Figure 9.12(a). The Alarm BN is considered a challenging problem for BN learning algorithms. For the purpose of experimentation, a database consisting of $3,000$ cases is generated using Netica [2].

9.2.2 Real life clinical data set - dementia diagnosis

We apply CONAR and REST to the dementia data set with the purpose of evaluating the algorithms on a "real-life" problem. Furthermore, the models obtained from these experiments feed into the comparison in Chapter 10, which addresses structural differences between the models created by an expert and the models derived from data.

Since there was no single data set pertaining to the hand-crafted models in Chapter 5, we invoked a data collection study to provide samples for evaluating the diagnostic accuracy of the hand-crafted models and data-driven discovery. The dementia data set used in this chapter is that described in Section 6.1.1, which is used to test the diagnostic accuracy of the hand-crafted models. Note, however, that the variables dementia with lewy bodies and frontotemporal dementia do not appear, as there was a serious under-representation of these pathologies.

9.2.3 Performance measures

Since the original networks are known, it is possible to evaluate the performance of the algorithms by comparing the learned solutions to the reference models. In order to enable evaluation, we collected performance measures relating to the quality of the learned structures, as well as computational complexity. In evaluating the models derived using the dementia data set, the hand-crafted models described in Chapter 5, Section 5.5.1 and Section 5.5.2, are taken to be the reference models — the models discovered from the data set will be compared against these.

- Metrics to evaluate solution quality:
    - Quantitative measure - the value of the CH (K2) metric (see Section 7.3.3.2). It is important to note that the learning problem is a maximisation of the log of a probability, therefore lies in the range $(-\infty, 0)$.

In other words, a score closer to zero is “better” than a score that is more negative.

  - Qualitative measures - structural differences between the learned network and the original network. The average and variance of the total number of edges (**TE**) is shown. **TE** is defined as the number of correctly orientated edges in a solution with respect to the reference model (**TC**), plus the number of inverted edges in a solution with respect to the reference model (**IE**). In addition, the number of extra edges (**EE**) — that is edges that appear in a solution and are not correctly orientated or inverted in the reference solution — and the number of missing edges (**ME**) are recorded.

- Algorithm complexity metric:

  - Note from Section 7.3.3 that the scoring function used to evaluate candidate solutions is decomposable. Therefore, the computational expense associated with scoring a single candidate solution is measured in terms of the number of function evaluations (**FE**) required to score all families encoded by the solution. The **FE** counter is incremented for each parent, therefore:

$$f_{ch}(x_i, \pi(x_i)) = \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}! \tag{9.1}$$

9.2.4 Experimental parameters

A summary of the parameters used for experimentation are shown in Table 9.1. Systematic tuning of the parameters has not been carried out; however, the values have been selected to be similar to those used in other optimisation problems: population sizes for each benchmark problem are the same as [135]; we use [142, pp. 344] as a guide for neighbourhood size; and ϕ_1 and ϕ_2 are randomly generated in range $[0, 1]$ [244]. The inertia weight, which is used to favour exploration in the first stages of the search and exploitation in the second part of the search, is decreased over time from [0.9 to 0.4] as [79]. The velocity clamp term, $\pm V_{max}$, is set to ±4.0, as described in [142, pp. 296].

Problem	Population size	Neighbourhood size
Asia	100	4
Car	20	3
Alarm	10	2
DemNet	100	4
PathNet	100	4

Table 9.1: Experimental parameters.

9.3 Experimental results and analysis: Synthetic data

We begin by presenting empirical results regarding the performance of CONAR and REST in Section 9.3.1. In Section 9.3.2, we compare performance results of CONAR and REST to order-based algorithms.

9.3.1 Comparison between CONAR and REST

Note that number of independent executions (or runs) of each algorithm for each problem varies — the corresponding values used in our experiments are shown in Table 9.2.

Problem	Runs	Iterations
Asia	30	500
Car	50	$30,000$
Alarm	60	$8,000$

Table 9.2: Number of independent runs for each problem.

The experimental results for each problem are displayed in Tables 9.4 - 9.6. Each table shows: the average score found and the standard deviation[1]($\mu \pm \sigma$), the best result found in all the runs, as well as qualitative and complexity measures. We should note that the best score row in each table relates to the best score found in all runs across both algorithms. The metric values for the best solution found are denoted (.).

The quality of the "known" graphical structures, measured using the CH metric, are provided in Table 9.3. We include these values as a comparative reference of the goodness of the results obtained by our algorithms.

	Asia	Car	Alarm
Known score	-11,264.83	-23,151.45	-29,796.59
Known edges	8	17	37

Table 9.3: Scores and edges count for the "known" models.

	CONAR	REST
Score(CH)	-11,241.04 ± 0.0	-11,261.64 ± 5.15
Best (in all runs)	-11,241.04	-11,251.93
TE (9)	9 ± 0.0	20.63 ± 2.58
TC (6)	6 ± 0.0	1.67 ± 0.96
IE (1)	1 ± 0.0	2.47 ± 0.97
EE (2)	2 ± 0.0	16.5 ± 2.57
ME (1)	1 ± 0.0	3.87 ± 1.00
FE (395,560)	418,615.53 ± 14,096.17	327,835.67 ± 179,497.48

Table 9.4: Asia results — 30 executions.

[1] For consistency, we use $\mu \pm \sigma$ as it aligns with the statistics used in the research that we use for comparison (in Section 9.3.2).

	CONAR	REST
Scores (CH)	-23,163.76 ± 4.48	-23,673.78 ± 95.15
Best (in all runs)	-23,158.45	-23,464.81
TE (25)	27.22 ± 2.0	43.78 ± 1.64
TC (15)	13.68 ± 2.13	5.70 ± 1.84
IE (2)	2.02 ± 1.53	4.62 ± 1.89
EE (8)	11.52 ± 2.51	33.46 ± 2.26
ME (0)	1.3 ± 1.02	6.68 ± 1.68
FE (13,037,127)	15,034,616.4 ± 1,491,111.95	12,691,603.96 ± 6,622,990.97

Table 9.5: Car results — 50 executions.

	CONAR	REST
Scores (CH)	-32667.71 ± 404.36	-41772.75 ± 423.97
Best (in all runs)	-31683.42	-40451.7390
TE (107)	133.12 ± 5.24	129.88 ± 3.11
TC (21)	16.08 ± 2.55	7.65 ± 2.22
IE (14)	14.92 ± 2.50	7.55 ± 1.95
EE (72)	82.12 ± 5.34	114.68 ± 4.00
ME (11)	15.00 ± 2.91	30.80 ± 2.23
FE (7,673,475)	8,228,848.42 ± 360,817.67	5,999,871.133 ± 2,974,604.498

Table 9.6: Alarm results — 60 executions.

Graphics depicting the best score achieved on a run by run basis for each problem are shown Figures 9.1 – 9.3.

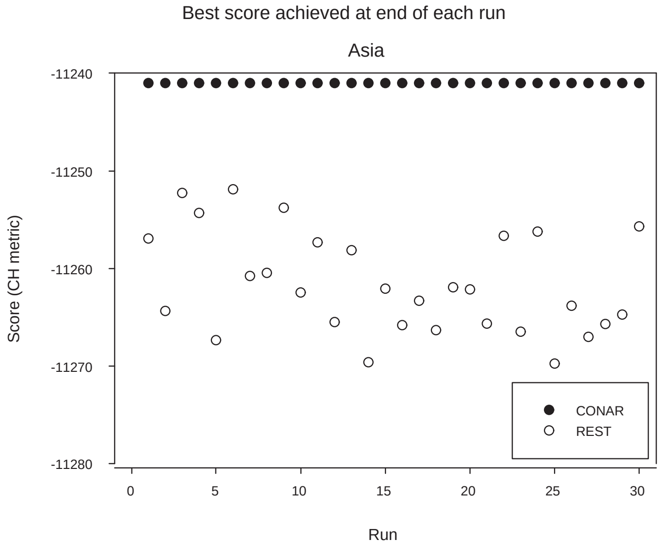


Figure 9.1: Asia problem — best score achieved at the end of each run.

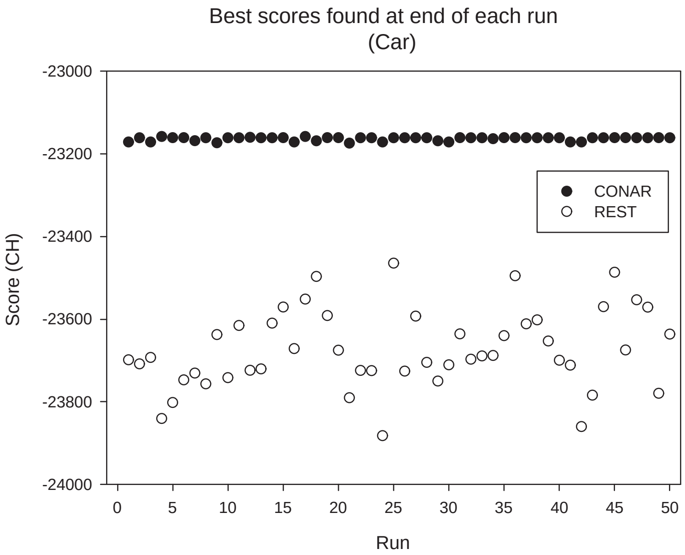


Figure 9.2: Car problem — best score achieved at the end of each run.

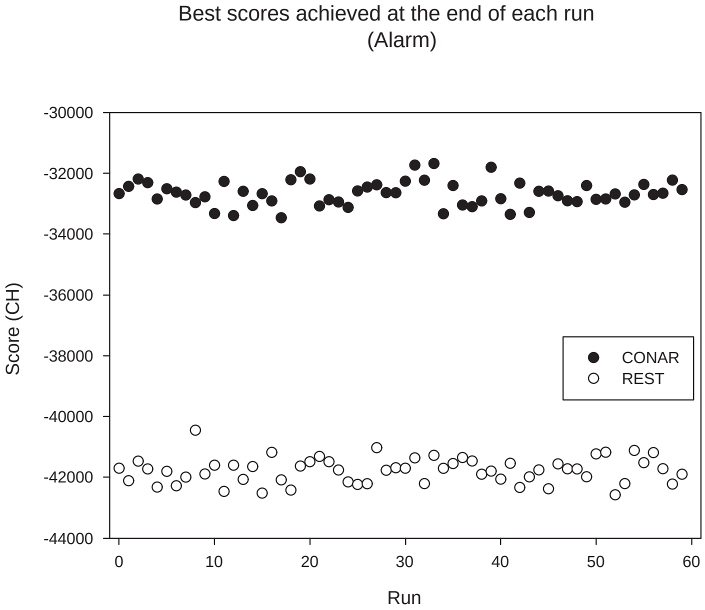


Figure 9.3: Alarm problem — best score achieved at the end of each run.

A deeper, quantitative and qualitative analysis of the results is provided in Sections 9.3.1.1 and 9.3.1.2 respectively.

9.3.1.1 Quantitative analysis

A statistical analysis has been carried out between CONAR and REST in order to determine the significance of the differences in the scores and in the number of function evaluations. We have used the Mann-Whitney [172] statistical significance test at 95% significance level, unless otherwise stated.

As can be seen from the scores achieved by both algorithms across all runs (shown in tables 9.4 – 9.6), CONAR is undoubtedly the better performing algorithm, as it finds better scoring solutions than REST (on average). The difference between the two algorithms is significant: the Mann-Whitney test shows a significance level of $p < 0.001$ for all problems.

Probing the results further, we make a number of observations and conclusions.

- The best result found for Asia has a CH value of $-11,241.04$ across all runs, which was found by CONAR. We should note that the BN learning literature regards the Asia domain as a trivial problem; however, it is considered a useful test-bed for new BN learning algorithms. Between the two algorithms, CONAR and REST, CONAR, in addition to finding the best scoring network in the Asia domain, finds the best CH value for the Car and Alarm domains, where the best networks found have CH values of $-23,158.45$ and $-31,683.42$ respectively. As can be seen from Table 9.4 and 9.5, the best scores achieved by CONAR on Asia and Car exceed the score values of the respective known reference models (shown in table 9.3).
- Regarding the accuracy of the solutions produced by each algorithm, it is clear that CONAR outperforms REST on each problem. The CH value attained by CONAR, on average, is closest to the CH value of the reference

structure. Compared to the reference CH score, both CONAR and REST improve on the CH value for the Asia reference model, however CONAR achieves the best overall value. Similarly, the average CH value achieved by CONAR on the Car domain is consistently closest to the reference model. An interesting observation is found in the variation in the results produced by CONAR: the standard deviation associated with the average scores produced by CONAR is lower than REST. Therefore, we can conclude that the solutions found by CONAR are consistently good over all runs. This can be seen graphically in Figures 9.1 – 9.3.

- In addition to achieving CH values close to the reference model, CONAR produces solutions that are qualitatively closer to the reference models. That is to say CONAR's solutions exhibit less structural differences (TE, EE, ME, I) than REST (see Table 9.4 – 9.6). Qualitative accuracy is discussed in more detail in Section 9.3.1.2.

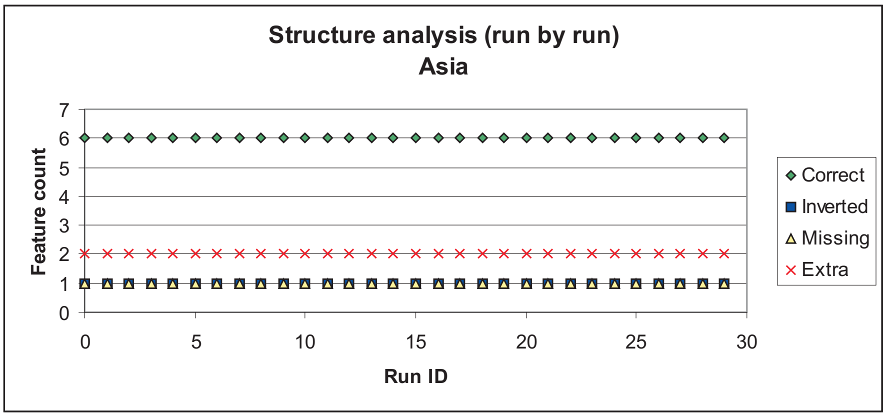
- With regard to efficiency, CONAR requires on average more FEs to converge to a solution than REST. The difference in FEs between CONAR and REST is statistically significant in the Asia problem ($p < 0.030$), however the difference in FEs on the car problem is not significant ($p < 0.059$). Although the average number of FEs required by REST is lower than CONAR, the variance in the number of FEs is lower for CONAR than REST, thus the computational effort required by CONAR is more consistent. It is worth noting that REST's efficiency gain in the number of FEs is compromised by the quality of solutions obtained. As can be seen in Tables 9.4 – 9.6, CONAR requires more FEs, however the quality of CONAR's solutions is significantly superior to those found by REST in all the problems examined.

9.3.1.2 Qualitative analysis

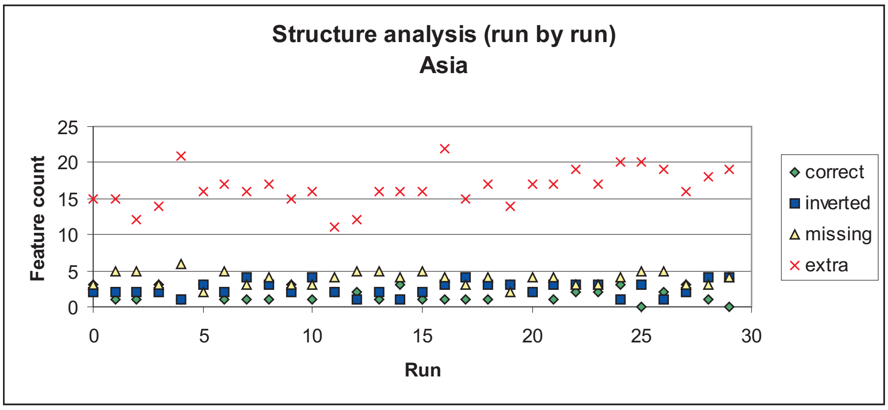
In this section, we provide an analysis of the solutions found by CONAR and REST from a structural perspective. That is to say that we: 1) compare the morphological differences in network topology across the solutions found during experimentation; and 2) compare morphologically the best scoring network structure with the corresponding reference model.

In order to evaluate and compare the learned BNs with the known reference models, we use the five structure feature performance metrics (**TE**, **CE**, **IE**, **ME**, **EE**) introduced in Section 9.2.3. The average and standard deviation for each of the five features is shown in Tables 9.4 - 9.6. Figures 9.4 – 9.11 drill into these results in more detail. Specifically, the feature count for the best solution found at the end of each run is shown graphically, as well as a summary of the proportion of different feature types found in the best solutions across all runs.

By comparing visually the graphs shown in Figures 9.4 – 9.11, it is clear that CONAR enjoys superior performance to REST in that the morphology of the structures found by CONAR is consistently closest to the reference model on each problem. The Mann-Whitney [172] test is performed to determine the significance of differences between the two algorithms on each of the five feature metrics. At the 99% level, the test shows a significance of $p < 0.001$ for all problems on all metrics.



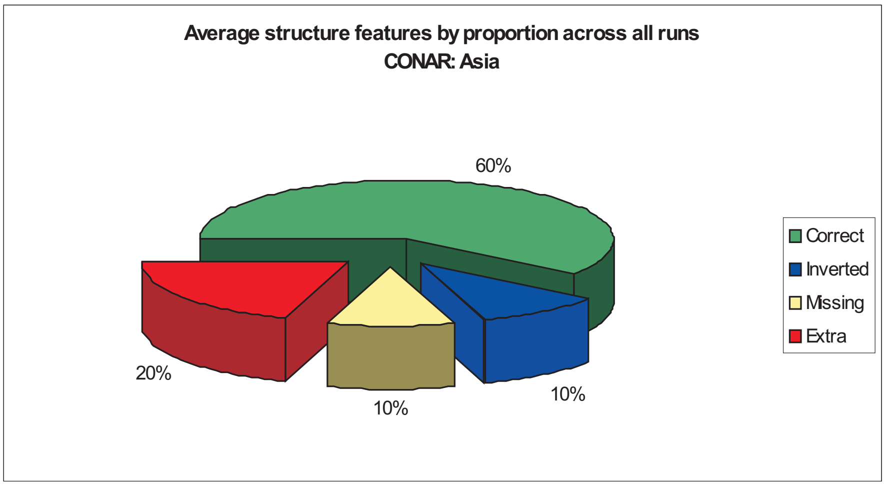
(a) CONAR: Solution features by count on each run.



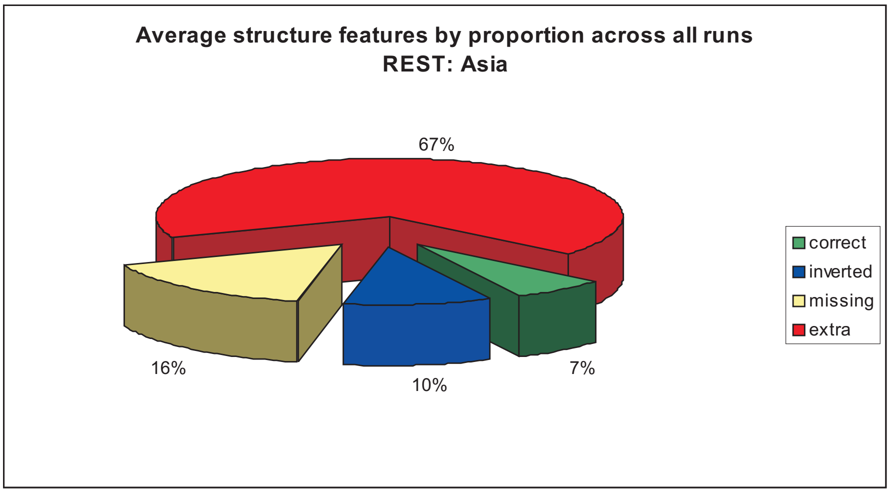
(b) REST: Solution features by count on each run.

Figure 9.4: Count of structural features of the best solutions in each run (Asia).

The Asia reference model is shown in Figure 9.6(a), and the converged solution is shown in Figure 9.6(b). To assist visual comparison, the edges in the learned model are colour coded to show correct, inverted, missing and extra edges with respect to the reference model.

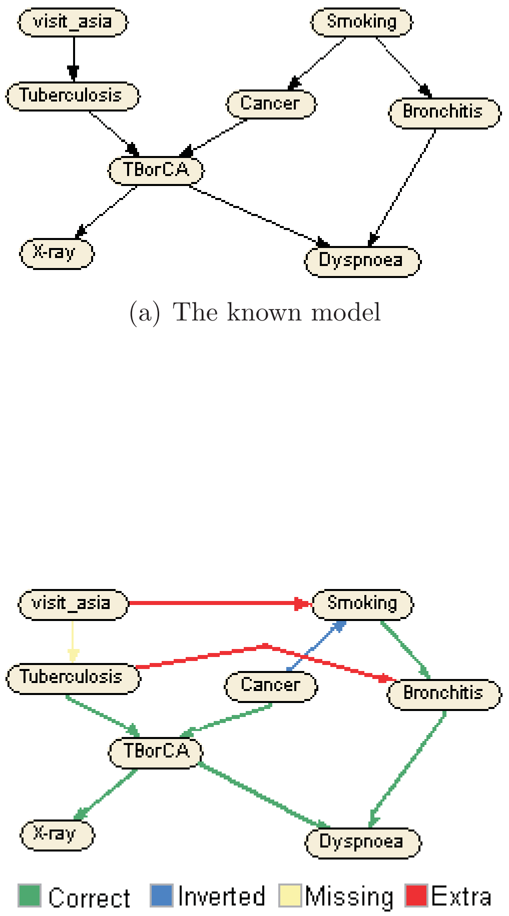


(a) CONAR: Solution features by proportion across all runs.



(b) REST: Solution features by proportion across all runs.

Figure 9.5: Proportion of structural features in the best solutions across all runs (Asia).



(a) The known model

(b) The best learned model (discovered by CONAR)

Figure 9.6: The Asia BN problem: Reference and learned models.

Asia - key observations

- Solutions found by REST consist of more edges than the solutions found by CONAR, which may be explained by REST's extremely limited learning capability, which is reflected in its poor performance profile across all problems.

- The structure of the models learned by REST deviate significantly from the reference models. On average, REST finds 21 (rounded) edges in total, 4 of which are correct edges (2 correctly orientated and 2 inverted), and 17 extra edges. In addition, REST fails to find on average 4 edges that appear in the reference model.

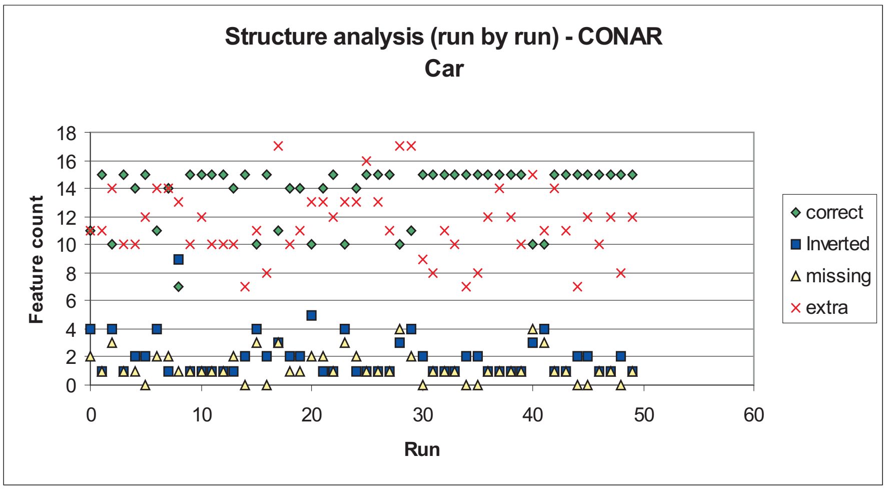
- As can be seen from Figure 9.7, it is clear that the number of extra and missing edges attained by REST is consistently higher than CONAR, and the number of inverted and correct edges discovered by REST is significantly lower than those found by CONAR.

- Analysing deeper the structural differences in the solutions found by each algorithm, the proportion of correctly orientated edges found by CONAR is significantly greater than those learned by REST. In particular, CONAR finds 6 correctly orientated edges on every run, meaning that 60% of the edges in solutions generated by CONAR are correctly orientated. In contrast, REST finds on average 2 (rounded) correct edges, which accounts for only 7% of the edges found in solutions across all runs.

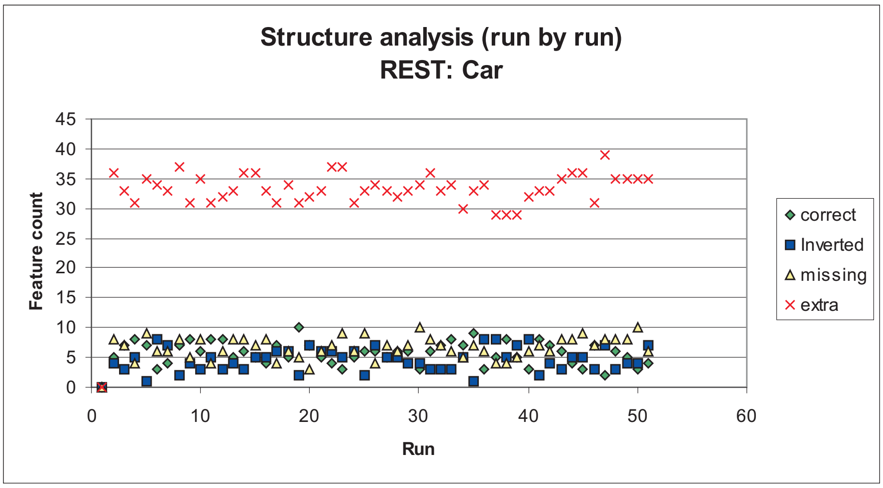
- Regarding extra edges, CONAR finds 2 (on average) compared to REST where the average is 17. Therefore, across the population of runs executed, 67% of edges found by REST are extra. In contrast, only 20% of the edges found in CONAR's solutions are extra.

- The extra edges found by REST do not increase the overall quality of solutions found, since the score of the reference solution is better than the average score of REST's solutions.

- Considering the reference model and best learned model, shown in Figure 9.6, 7 out of 8 edges are found. Only few errors were obtained. The extra edges were 'visit_asia' → 'smoking' and 'tuberculosis' → 'bronchitis'. The inverted edge is ‘cancer’ → ‘smoking’, and the missing edge is ‘visit_asia’ → ‘tuberculosis’.

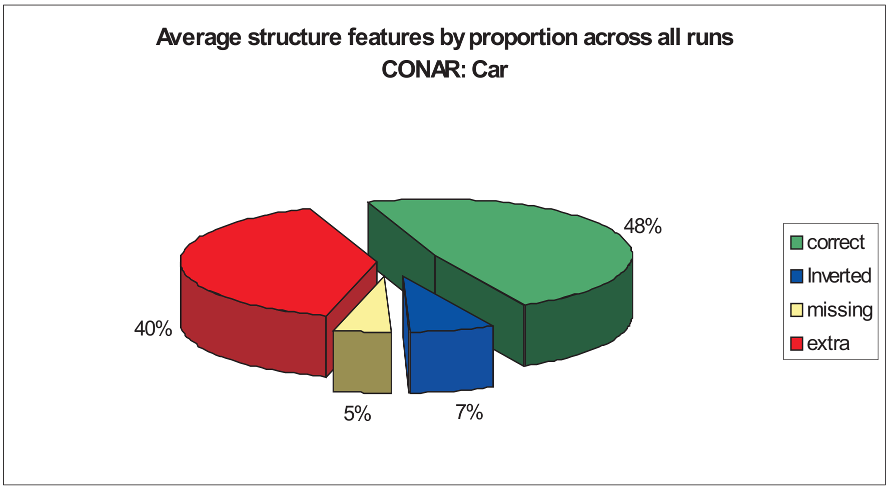


(a) CONAR: Solution features by count on each run.

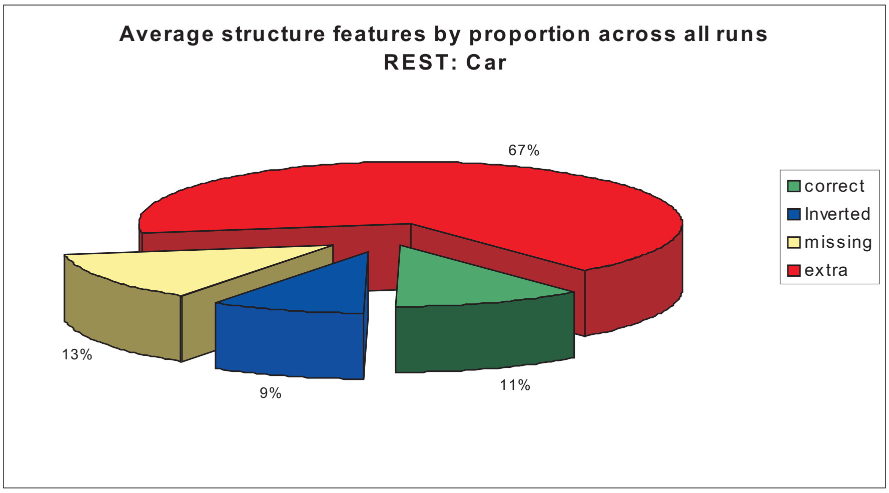


(b) REST: Solution features by count on each run.

Figure 9.7: Count of structural features of the solutions found in each run (Car).



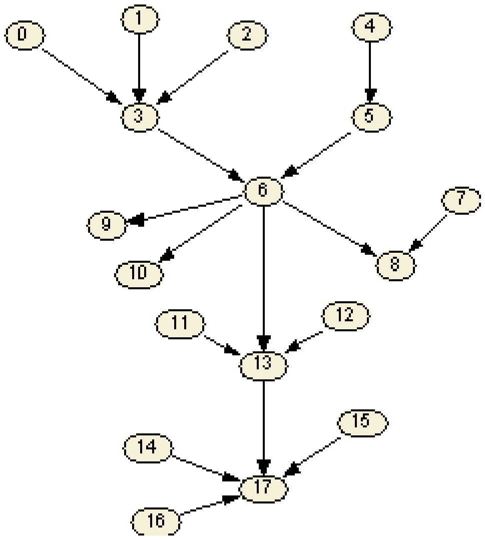
(a) CONAR: Solution features by proportion across all runs.



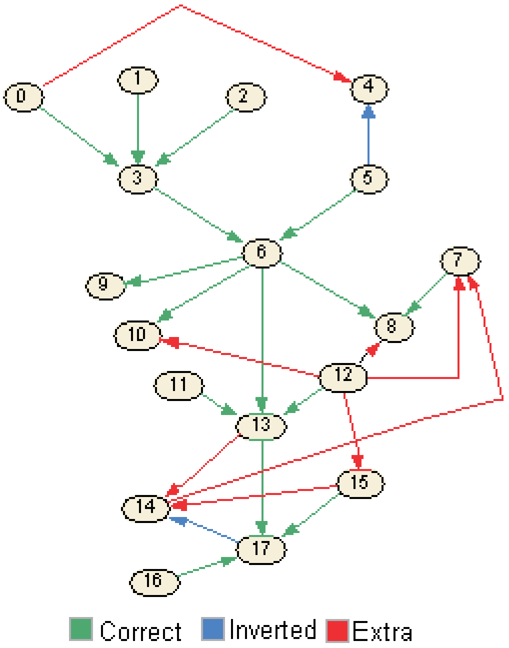
(b) REST: Solution features by proportion across all runs.

Figure 9.8: Proportion of structural features of the solutions across all runs (Car).

The known Car model is shown in Figure 9.9(a), and the best scoring model achieved across all runs is shown in Figure 9.9(b). To assist visual comparison,



(a) The known model



(b) The learned model (discovered by CONAR)

Figure 9.9: The Car BN problem: Reference model and best learned model.

the edges in the learned model are colour coded to show correct, inverted and extra edges with respect to the reference model.

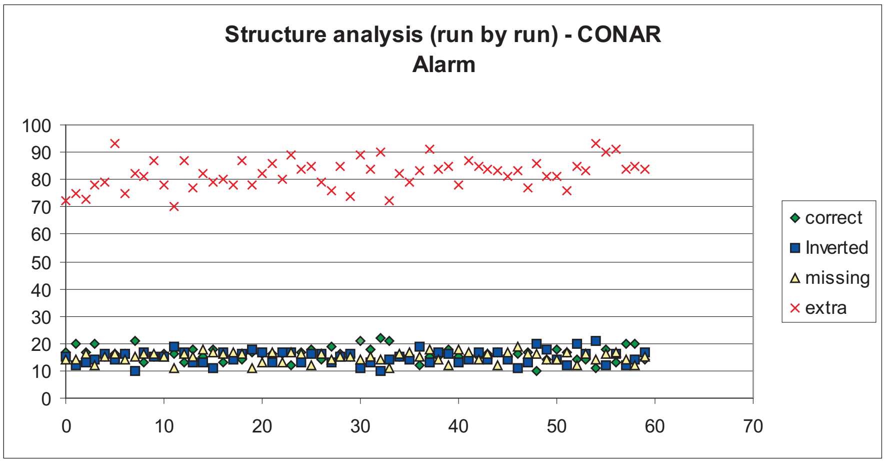
Car - key observations

- From a score perspective, the quality of solutions found by CONAR are significantly more superior to REST (see section 9.3.1.1).
- As can be seen from Table 9.5, the structure of the models learned by REST deviates significantly from the reference model, which has 17 edges. On average, REST finds 44 (rounded) edges in total, 11 of which are edges in the reference model(6 correctly orientated and 5 inverted), and 33 extra edges. In addition, REST fails to find on average 7 (rounded) edges that appear in the reference model. It is clear that the solutions found by REST are further from the known model compared to those solution found by CONAR (irrespective of orientation). The main difference is in the number of extra and missing edges found by REST.
- It is noteworthy that CONAR, on average, finds all but one of the edges in the reference model, albeit that 2 of the 16 edges are inverted. In contrast, REST finds only 11 of the possible 17 edges, 6 of which are correctly orientated and 5 are inverted edges.

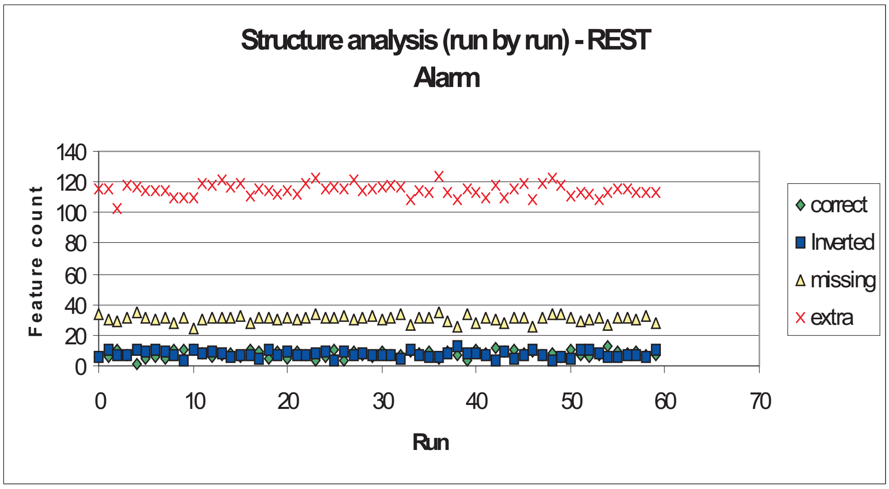
  Regarding extra edges, REST finds significantly more extra edges than CONAR. However, similar to the Asia problem, the extra edges found by REST do not increase the overall quality of the solutions, since the score of the reference solution is better than the average score of REST's solutions with extra edges.

- It is interesting to note that best learned model, shown in Figure 9.9, contains all the edges that feature in the reference model. There are, however, a small number of edge errors, as well as a number of extra edges. The inverted edges are 5 → 4 and 14 → 17; the extra edges are shown in red on Figure 9.9(b).

- Compared with the Asia problem, the proportion of edges (correctly orientated and inverted edges) is reduced: the number of edges found by CONAR was 70% on the Asia problem; however in the Car problem, the proportion of edges found by CONAR is 55%. In addition, there is an increase in the proportion of extra edges found by CONAR in the Car problem (40%), compared to the Asia problem (20% extra). However, the number of edges missed by CONAR on the Car problem is reduced to 5%, where 10% of the edges were missing in the Asia problem. The proportion of edges discovered by REST on the Asia and Car remain approximately equivalent.

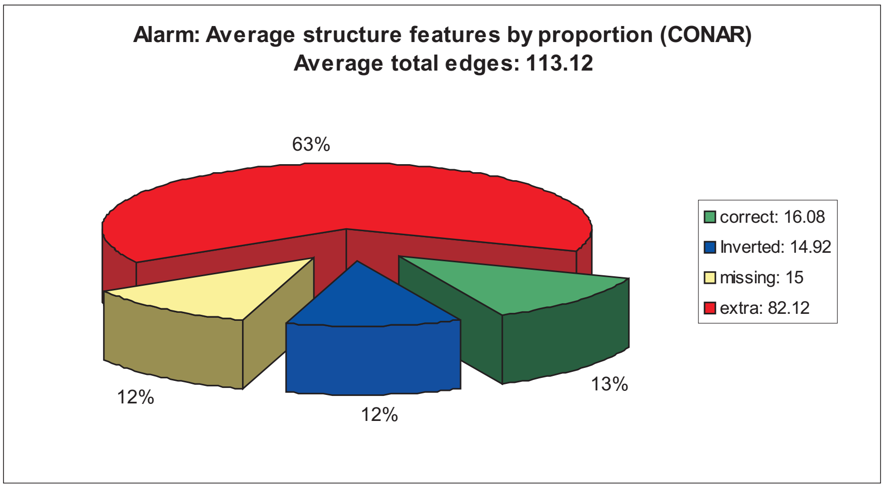


(a) CONAR: Solution features by count on each run.

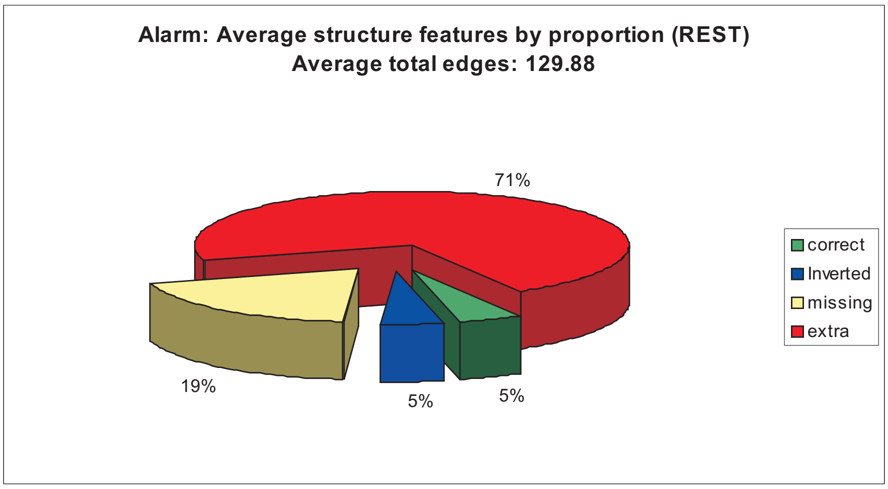


(b) REST: Solution features by count on each run.

Figure 9.10: Count of structural features of the solutions found in each run (Alarm).

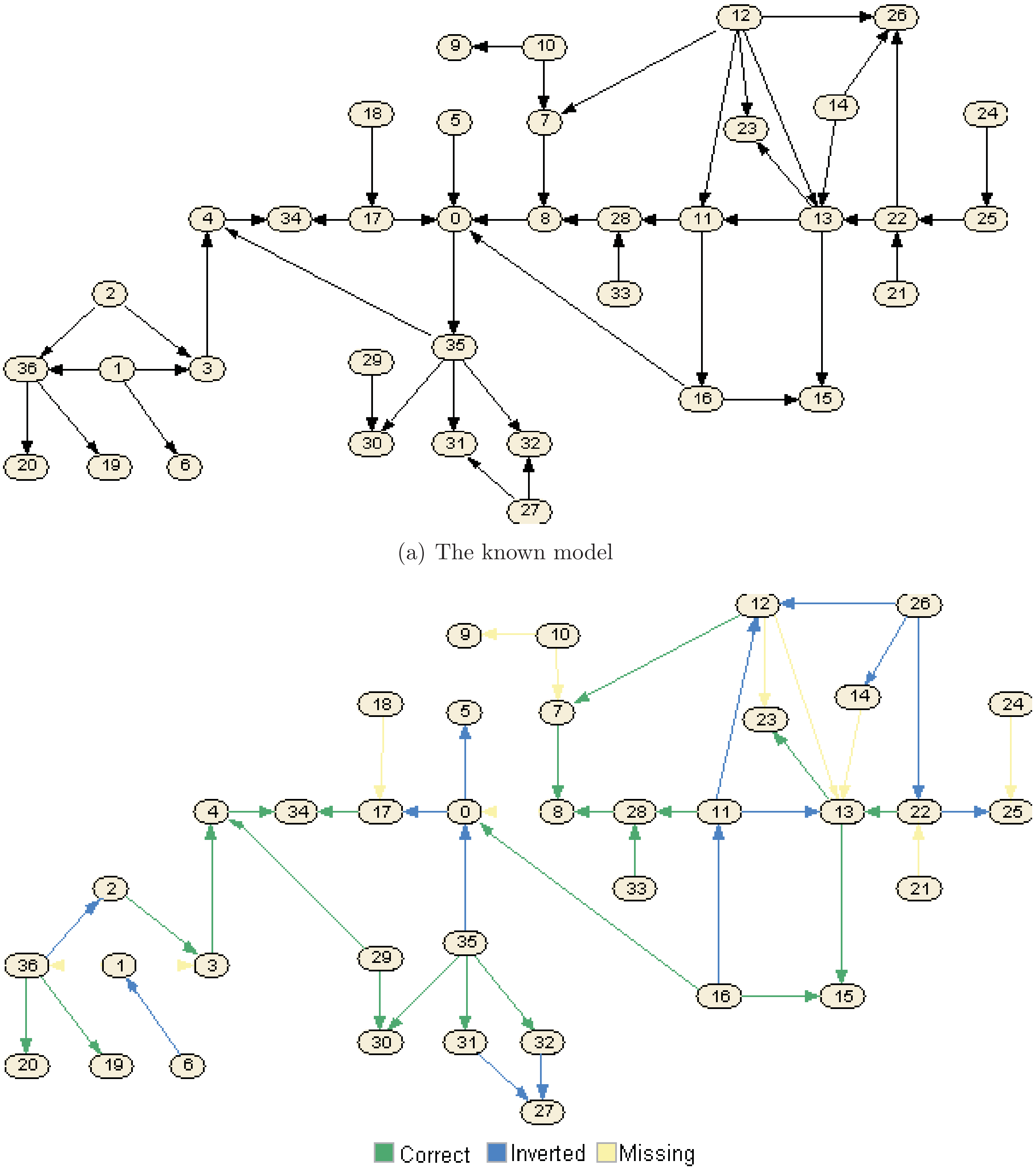


(a) CONAR: Solution features by proportion across all runs.



(b) REST: Solution features by proportion across all runs.

Figure 9.11: Proportion of structural features of the solutions across all runs (Alarm).



(a) The known model

(b) The learned model (discovered by CONAR)

Figure 9.12: The Alarm BN problem: Reference model and best learned model. Note that extra edges are not shown in order to simplify the diagram.

The Alarm reference model is shown in Figure 9.12(a) and the best scoring model achieved across all runs is shown in Figure 9.12(b). Extra edges are not shown, as the model is larger in size, and the number of extra edges (72) would over complicate the diagram.

Alarm - key observations

- Despite reduced performance compared to the other problems considered in this research, CONAR continues to discover, on average, more than 70% of the edges in the reference model. However, it should be noted that the number of correctly orientated edges is lower than those found in the Asia and Car domains.
- Irrespective of lower score performance compared to Asia and Car, the best learned model, shown in Figure 9.12(b), contains 75% of the edges found in the reference model. Of that, 60% are correctly orientated, and 40% are inverted.

9.3.2 Comparison with order-based techniques

In Section 7.4.1.1, we acknowledge that order-based techniques, such as the K2 algorithm proposed by Cooper and Herskovits [51] (see Section 7.4.1.1), produce good Bayesian network structures when an optimal order is provided. The primary issue associated with this approach, however, relates to specification of the order, as this impacts on the quality of the resultant network [126]. It is possible for a human expert to specify the order, although this increases the risk of error. Alternatively, algorithms exist that seek to learn the order. However, two-tiered algorithms that attempt to learn the order then the structure, such as Larrañaga et al. [154, 155], are expensive.

The purpose of this section is to compare our algorithms, which do not make an order assumption, with algorithms that that require, or make use of a variable order during learning. The comparisons are defined in terms of three objectives:

1. Compare CONAR and REST algorithms to those that require an order as input to learn the BN structure. Specifically:
   (a) How do CONAR and REST perform against an order-based algorithm when random orders are specified?
   (b) How do CONAR and REST perform against an order-based algorithm when a good order is specified?
2. Investigate the performance of CONAR and REST compared to algorithms that evolve optimal orders.
3. Determine whether the performance of the algorithms developed in this research have sufficient capability to relax the need to learn order.

9.3.2.1 Comparison experimentation

In order to answer the question stated in Section 9.3.2, four comparison experiments are conducted:

1. K2no - K2 with random orders. The K2 algorithm is considered a benchmark in BN learning literature. It requires as input an order among the variables. In demonstrating K2's reliance on correct orders and enabling

a comparison between K2, CONAR and REST, K2no is executed $10 \times n$ times with unique, randomly generated orders, where n is the number of variables.

2. K2 - standard K2, with good order. When K2 is given a good order, good BN structures are produced [51]. Therefore, for each problem, we execute K2 with a good order. This enables a comparison between our algorithms (that do not rely on an order) and the benchmark algorithm that requires an order to perform well. In addition, this experiment contributes towards answering the question concerning relaxation of the need to learn order,introduced in Section 9.3.2.

3. K2GA - Order learning using a Genetic Algorithm (GA). Clearly, the risk of error is increased if humans are responsible for defining the order of variables. Acknowledging, however, that a good order leads to good BN structures, we seek to compare our algorithms with a technique that learns an optimal order using a GA, such as that developed by Larrañaga et. al [154]. While this approach results in good solutions, it should be noted that this particular technique is computationally expensive, as it uses the K2 algorithm as a mechanism to evaluate each candidate solution (order) evolved by the GA. We use this algorithm as a comparison to investigate how our algorithms (which do not require order information) perform when compared against an existing approach that learns orders.

4. ChainGA - As mentioned above, the K2GA algorithm requires execution of the K2 algorithm for each and every solution in the population, which expensive. Kabli et al. [135] have developed a variant of the K2GA algorithm, which greatly reduces K2GA's dependency on the K2 algorithm for

evaluating every solution in the population. This experiment will enable a comparison with a state of the art algorithm which learns orders efficiently.

Together, the K2GA and ChainGA experiments seek to answer the question concerning the performance of CONAR and REST compared to algorithms that automatically learn a suitable order. The K2 algorithm experiments combined with the K2GA and ChainGA experiments, will address the question relating to the value of algorithms that learn the order.

9.3.2.2 Comparative results

The experimental results generated by CONAR and REST are listed alongside the K2, K2GA and ChainGA algorithms in tables 9.7 – 9.9. Note that K2O refers to the K2 algorithm with an ordering, and K2no refers to the K2 algorithm with a random order.

It is worth noting that we have obtained the data set used by Kabli et al. [135] in order to compare our algorithms with theirs. However, we are unable to make comparisons of statistical significance between our algorithms and the K2GA and ChainGA algorithms, we do not have access to the raw results generated by their algorithms. Although not ideal, we use the summary statistics published in their paper as a basis for comparison.

	Rank	Evaluation	FE
CONAR	1	-11,241.04 ± 0.0	418,615.53 ± 14,096.17
REST	5	-11,261.64 ± 5.15	327,835.67 ± 179,497.48
K2no	6	-11,266.12 ± 19.81	31.93 ± 1.65
K2O	4	-11,248.23 ± n/a	-
K2GA	2	-11,244.3 ± 2.0	3,645 ± 968
ChainGA	3	-11,248.1 ± 11.0	1,924 ± 152

Table 9.7: Algorithm comparison - Asia.

	Rank	Evaluation	FE
CONAR	2	-23,163.76 ± 4.48	1,491,111.95
REST	6	-23,673.78 ± 95.15	6,622,990.97
K2no	5	-23,262.71 ± 79.94	80.65 ± 4.45
K2O	1	-23,149.65 ± n/a	-
K2GA	3	-23,168.5 ± 21.0	2,227.96 ± 395
ChainGA	4	-23,213.3 ± 152	1,018.7 ± 177

Table 9.8: Algorithm comparison - Car.

	Rank	Evaluation	FE
CONAR	5	-32,667.71 ± 404.36	8,228,848.42 ± 360,817.67
REST	6	-41,772.75 ± 423.97	5,999,871.13 ± 2,974,604.49
K2no	4	-30,658.86 ± 313.08	185.51 ± 4.66
K2O	3	-30,129.82 ± n/a	-
K2GA	1	-30,068.4 ± 164	2,458.4 ± 474
ChainGA	2	-30,097.3 ± 141	1,844.0 ± 116

Table 9.9: Algorithm comparison - Alarm.

9.3.2.3 Discussion of experimental comparisons

With the purpose of answering the questions outlined in Section 9.3.2, the results obtained in Section 9.3.2.2 are discussed here in detail.

Comparison with algorithms requiring an order. We should note that the K2O algorithm is a deterministic, thus if it is run n times the result is constant with zero standard deviation. Therefore in comparisons involving the K2O algorithm, the 1-sample Wilcoxon [278] test at the 95% confidence level appropriately replaces the Mann-Whitney test.

As can be seen in Tables 9.7 through 9.9, it is clear that the capability of K2O is superior to that of K2no across all problems. On the Car and Alarm problems, the difference is vast: $P < 0.001$; on the Asia problem, however, the magnitude is slightly less severe: $p = 0.007$. There is no surprise in the results generated by this experiment; the results provide demonstrable evidence to support the fact that good BN structures are obtained if an optimal order is supplied.

With respect to the scores of the known models, K2O attains consistently good scores on all problems compared to K2no. In addition, the solutions found by K2O are structurally more similar to the known models than those found by K2no.

In comparing the algorithms developed in this research with K2no, CONAR outperforms K2no on the Asia and Car domains; the difference in the scores obtained by each algorithm is statistically significant: ($P < 0.001$). However, REST absolutely struggles to compete with K2no on the Car and Alarm problems, where the difference between the algorithms is extreme ($P < 0.001$). It is interesting to observe that REST appears to produce solutions that are comparable to those generated by K2no, as indicated by the statistical significance test ($P = 0.65$). On the basis of an analysis of the results generated during experimentation, we conclude that REST is inadequate for data-driven BN construction, since it does not have the ability to at least compete with an algorithm that executes with random orders. On the other hand, CONAR shows more promising results, as it has the ability to produce good BN structures which are close in score and morphology to the known reference model, and is at very least competitive with other relevant algorithms.

Comparing analysed results generated by CONAR and K2, it is clear that K2 outperforms CONAR on two problems, namely Car and Alarm with a significance $P < 0.001$. In addition to the difference in scores, K2 uses a significantly lower number of FEs than both CONAR and REST, again the difference is hugely significant $P < 0.001$. The explanation for this may be found in the different underlying mechanics of the respective algorithms. K2 differs from that of nature inspired algorithms such as ours: the K2 algorithm develops a single solution over time, where as our algorithms (nature-inspired) are population based and iterative, therefore an increase in FEs is expected.

Problem	TE	TC	IE	EE	ME
Asia	7	7	0	0	1
Car	17	16	0	1	1
Alarm	77	22	19	37	5

Table 9.10: Features found by K2 on Asia, Car and Alarm problems

By comparing the summary of the feature count produced by K2 (Table 9.10) with that of the best solution found by CONAR for each problem, denoted by (.) in Tables 9.4 – 9.6, it would appear that CONAR is comparable with K2 on Asia and CAR. There are larger difference in the feature count achieved by CONAR and K2 on the Alarm problem.

Comparison with algorithms that learn the order. As stated in Section 9.3.2.2, we reemphasise that we do not have access to the raw result data generated by Kabli et al. [135], and therefore are unable to make comparisons of statistical significance between our algorithms, K2GA and ChainGA.

On the Asia problem, CONAR achieves the overall best average score with the smallest standard deviation, as can be seen from Table 9.7. However, there does not appear to be an obvious difference in scores achieved by CONAR and K2GA, although there does appear to be a small difference between CONAR and ChainGA. Without a statistical test, it is impossible to determine whether CONAR is comparable or significantly better than ChainGA. To that end, we conclude that the scores attained by CONAR are comparable to K2GA, and at the very least comparable to ChainGA.

We observe a similar patter to Asia in the Car problem; in Table 9.8, CONAR appears comparable to K2GA on the Car problem. With regard to ChainGA, CONAR has a better average score, and its standard deviation is much lower. We conclude that the average score achieved by CONAR is at the very least comparable to K2GA, and possibly significantly better than ChainGA.

It appears that CONAR experiences a degradation in search capability on the Alarm problem. Given the number of variables in this problem, the solution space is extremely large and complex. However, given comparable achievements against the Asia and Car problem, the results on the Alarm problem are disappointing. Future work is required to conduct further experiments, as the parameters used may not be sufficient for the complexity of this problem and may need tuned.

The performance of REST compared to K2GA and ChainGA is poor; the same pattern is observed on all other algorithms considered during experimentation. Despite the potential benefits realised by removing the need for validation and repair operators, REST is not capable of achieving results that are at least comparable to any of the other algorithms, as it does not deliver the requisite level of performance.

With regard to the computational effort required by CONAR and REST compared to the effort required by K2GA and ChainGA, it is clear that CONAR and REST require a significantly larger number of FEs. A possible explanation for this lies in the fact that CONAR and REST search in the larger, more complex space of all possible BN structures, while K2GA and ChainGA search in the smaller, more restricted space of orders. It is for this reason that the reduction in the number of fitness function evaluations is observed in ChainGA, as it is only the chain structure that is evaluated on a regular basis.

On the value of algorithms that use an order to learn. It is well documented that high-scoring network structures can be found when an optimal ordering among the nodes is given [154], and a variety of algorithms have been developed that exploit this, such as K2. This behaviour is demonstrated in our experiments: when K2 is executed with random orders, K2no, it performs poorly; however, when a good order is provided, K2 performs very well — see Tables 9.7 – 9.9. Based on the discussion of the results which compares CONAR and REST to algorithms that learn the order (K2GA and ChainGA), we explore the value-add gained by using an order to learn Bayesian network structures.

The value in using the K2 algorithm is highlighted above; however, its major drawback is its requirement of an order as input. This may not be available, or, if specified by a human, may be incorrect, which may result in poor scoring structures. We turn, therefore, to algorithms that learn optimal orders, such as K2GA and ChainGA.

The primary issue with K2GA is the computational expense required by constant dependency on the K2 algorithm as a mechanism for evaluating GA candidate solutions. One of the original questions that this thesis seeks to address is: Are the algorithms developed in this research sufficiently powerful to produce solutions that are competitive with the solutions produced by an expensive algorithm (K2GA) that learns in the space of orders, without having to employ an expensive two-tiered algorithm. The results presented above go some way to answering this question; they demonstrate that our CONAR algorithm is capable of structure discovery from data without the use of a two-tiered algorithm. However, the ChainGA algorithm has very recently emerged as a promising algorithm that seeks to improve the performance of K2GA by placing much less emphasis on the K2 algorithm. The algorithm is competitive with ours in terms of solution quality, however it appears to have better performance in terms of computational effort.

In two of the three problems in the experiments that we have conducted, there appears to be no benefit to the score obtained by searching in the the space of orders over searching in the entire space of Bayesian network structures. In fact, there appears to be a small benefit in learning in the entire space, as the average scores and associated standard deviations achieved by CONAR appear to be better than K2GA and ChainGA, although the statistical significance is unfounded. The slight increase in the quality of the scores obtained may be explained by a more thorough exploration in the whole space of structures, as opposed to the structures limited by the learned orders. However, there is clearly a difference in computational expense between our algorithms at the K2GA and ChainGA, as shown in the FE column in Tables 9.7 – 9.9. Clearly, this impacts on our ability to achieve learning with less computational expense.

9.4 Experimental results and analysis: Clinical data

In this section we present empirical results on the performance of CONAR and REST on the task of constructing BN models for dementia diagnosis using a real-life clinical data set. Note that this section reports only the results of CONAR and REST on the construction task; a comparison of the “learned” models with the original reference models is treated in Chapter 10.

9.4.1 Comparison between CONAR and REST

Each algorithm is executed 30 independent times on each data set (dementia syndrome data set and the pathology data set). The experimental results for each are displayed in Tables 9.12 - 9.13. Each table shows: the average score found and the standard deviation ($\mu \pm \sigma$), the best result found in all the runs, as well as qualitative and complexity measures. We should note that the best score row in each table relates to the best score found in all runs across both CONAR and REST. The scoring metric values for the best solution found are denoted by (.).

The dementia models developed in Chapter 5 are taken to be the reference models, and the quality of these structures, measured using the CH metric, are provided in Table 9.11.

	DemNet	PathNet
Known score	-1,312.15	-794.90
Known edges	10	21

Table 9.11: Scores and edges count for reference DemNet and PathNet models.

	CONAR	REST
Score(CH)	-1,113.13 ± 0.12	-1,162.32 ± 5.87
Best (in all runs)	-1,113.01	-1,149.32
TE (16)	16.87 ± 1.00	14.6 ± 2.04
TC (2)	1.93 ± 0.37	1.53 ± 1.31
IE (4)	4.40 ± 0.56	3.30 ± 1.40
EE (10)	10.53 ± 0.57	10.03 ± 1.54
ME (4)	3.67 ± 0.61	7.43 ± 1.57
FE (938,086)	909,623.3 ± 5,5623.28	443,199.6 ± 355,662.98

Table 9.12: DemNet results for CONAR and REST — 30 executions.

	CONAR	REST
Score(CH)	-676.82 ± 0.0	-697.21 ± 2.94
Best (in all runs)	-676.82	-690.17
TE (9)	9 ± 0.0	14 ± 2.04
TC (1)	1 ± 0.0	1.53 ± 1.30
IE (1)	1 ± 0.0	3.03 ± 1.40
EE (7)	7 ± 0.0	10.03 ± 1.54
ME (10)	10 ± 0.0	7.43 ± 1.57
FE (522,639)	540,636.4 ± 12,124.54	420,108.2 ± 250,274.54

Table 9.13: PathNet results for CONAR and REST — 30 executions.

Graphics depicting the best score achieved on a run by run basis for each problem are shown in Figures 9.13 and 9.14.

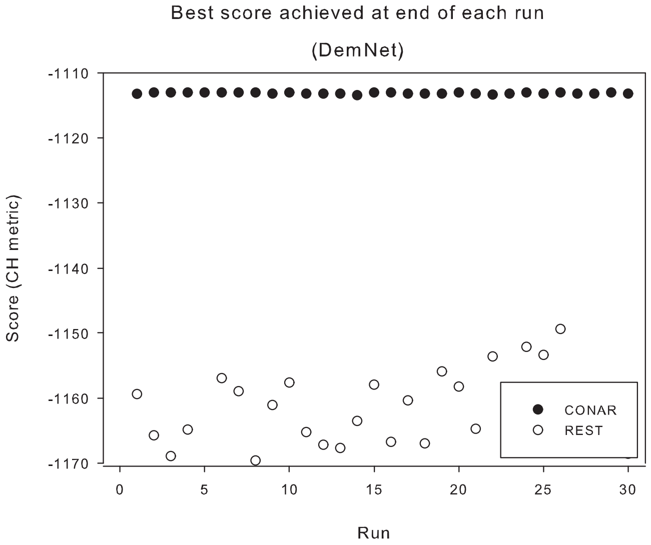


Figure 9.13: DemNet — best score achieved at the end of each run.

A deeper, quantitative and qualitative analysis of the results is provided in Sections 9.4.1.1 and 9.4.1.2 respectively.

9.4.1.1 Quantitative analysis

A statistical analysis has been carried out between CONAR and REST in order to determine the significance of the differences in the scores and in the num-

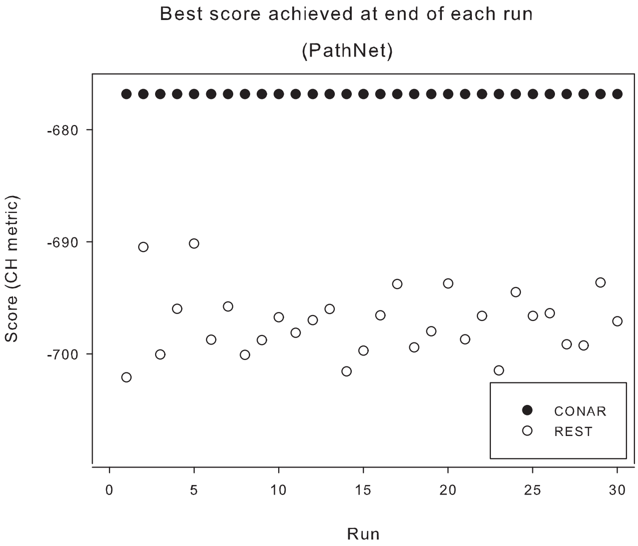


Figure 9.14: PathNet — best score achieved at the end of each run.

ber of function evaluations. We have used the Mann-Whitney [172] statistical significance test; in all cases a 95% significance level is used.

As can be seen from the scores achieved by both algorithms across all runs (shown in Tables 9.12 and Table 9.13), CONAR is undoubtedly the better performing algorithm, as it finds better scoring solutions than REST (on average). The difference between the two algorithms on both DemNet and PathNet is significant: the Mann-Whitney test shows a significance level of $p < 0.001$.

Probing the results further, we make the following observations and conclusions:

- The best result found for DemNet has a CH value of $-1,113.01$ across all runs, which was found by CONAR. In addition, CONAR found the best scoring PathNet model, which had attained a CH value of -676.82. As can be seen from Table 9.12 and Table 9.13, the best scores achieved by CONAR exceed the score values of the respective known reference models (shown in Table 9.11).

- From Figure 9.13 and Figure 9.14, it is clear that CONAR is more consistent (and reliable) in terms of the solutions that it finds across multiple runs.

- Despite CONAR finding consistently better scoring solutions than REST, the average score for both DemNet is not close to the known score, and in the case of PathNet the average score exceeds the reference score. This is exemplified by the qualitative results (TE, EE, ME, I) shown in Table 9.12 and 9.13. With regards to DemNet, 6 out of 10 edges were recovered. However, many extra edges were produced (10), which may offer an explanation for the gap between the average score attained by CONAR and the reference score for each problem. In the case of PathNet, only 2 of the 12 edges were recovered. Qualitative accuracy is discussed in more detail in Section 9.4.1.2.

- As seen in the results of the benchmark problems, Section 9.3, CONAR requires on average more FEs than REST on both DemNet and PathNet. The difference is statistically significant for DemNet ($p < 0.001$); however, the difference is only just significantly different for PathNet ($p < 0.049$).

9.4.1.2 Qualitative analysis

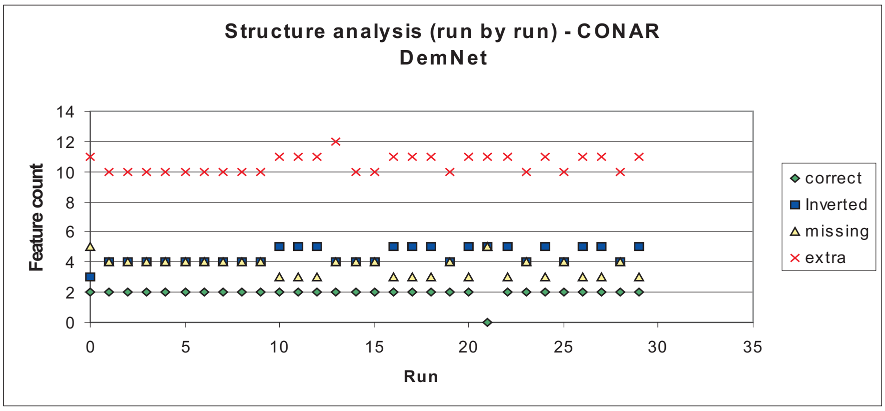
In this section we provide an analysis of the experimental results on the clinical data set. The analysis method is identical to that performed on the synthetic benchmark data sets described in Section 9.3.1.2.

The average and standard deviation for each of the five feature metrics (**TE**, **CE**, **IE**, **ME**, **EE**) introduced in Section 9.2.3 are shown in Tables 9.12 and 9.13. Figures 9.15 through 9.18 drill into these results in more detail. Specifically, the feature count for the best solution found at the end of each run is shown graphically, as well as a summary of the proportion of different feature types found in the best solutions across all runs.

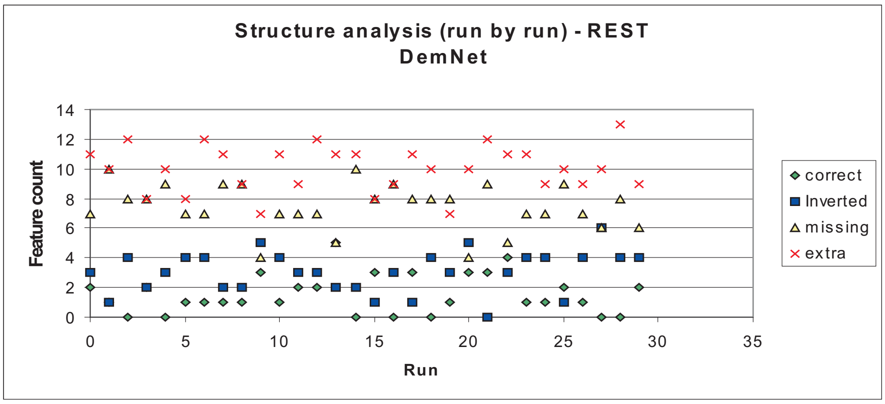
By comparing visually the graphs shown in Figure 9.15 and Figure 9.17, it is clear that CONAR enjoys superior performance to REST in that the morphology of the structures found by CONAR is consistently closest to the reference model on each problem. The Mann-Whitney [172] test is performed to determine the significance of differences between the two algorithms on each of the five feature metrics. The significance values at the 95% are shown in Table 9.14; non-significant differences between CONAR and REST are denoted by †.

Feature	DemNet	PathNet
TE	< 0.001	< 0.001
CE	0.027	0.112†
IE	< 0.001	< 0.001
ME	< 0.001	< 0.001
EE	0.208†	< 0.001

Table 9.14: Significance of differences in structural features between CONAR and REST on DemNet and PathNet.



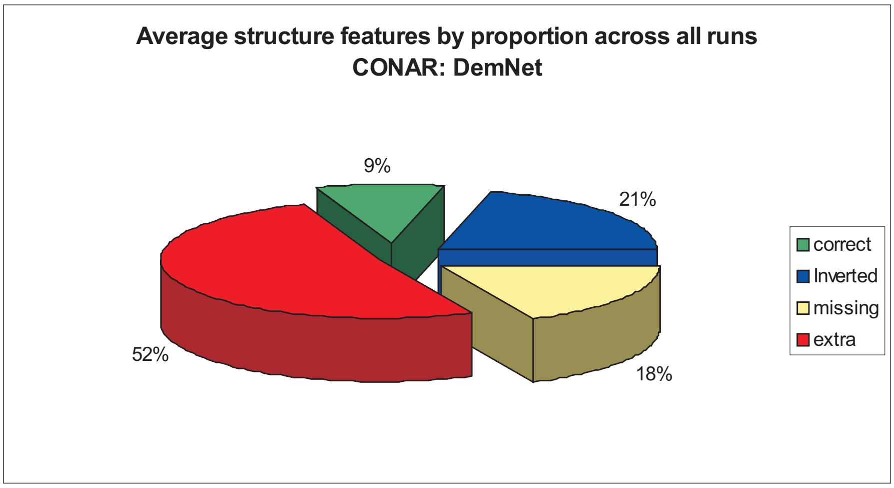
(a) CONAR: Solution features by count on each run.



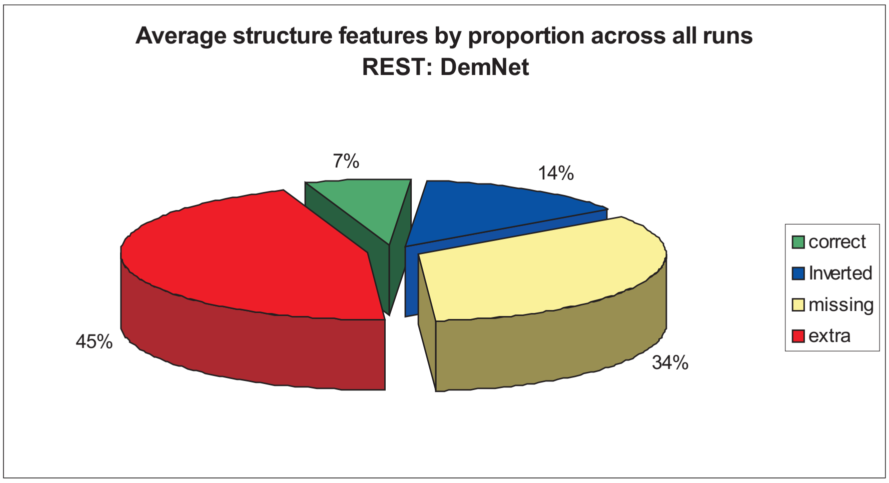
(b) REST: Solution features by count on each run.

Figure 9.15: Count of structural features of the best solutions in each run (DemNet).

Structural difference between the the "learned" models and the reference models are discussed in detail in Chapter 10.

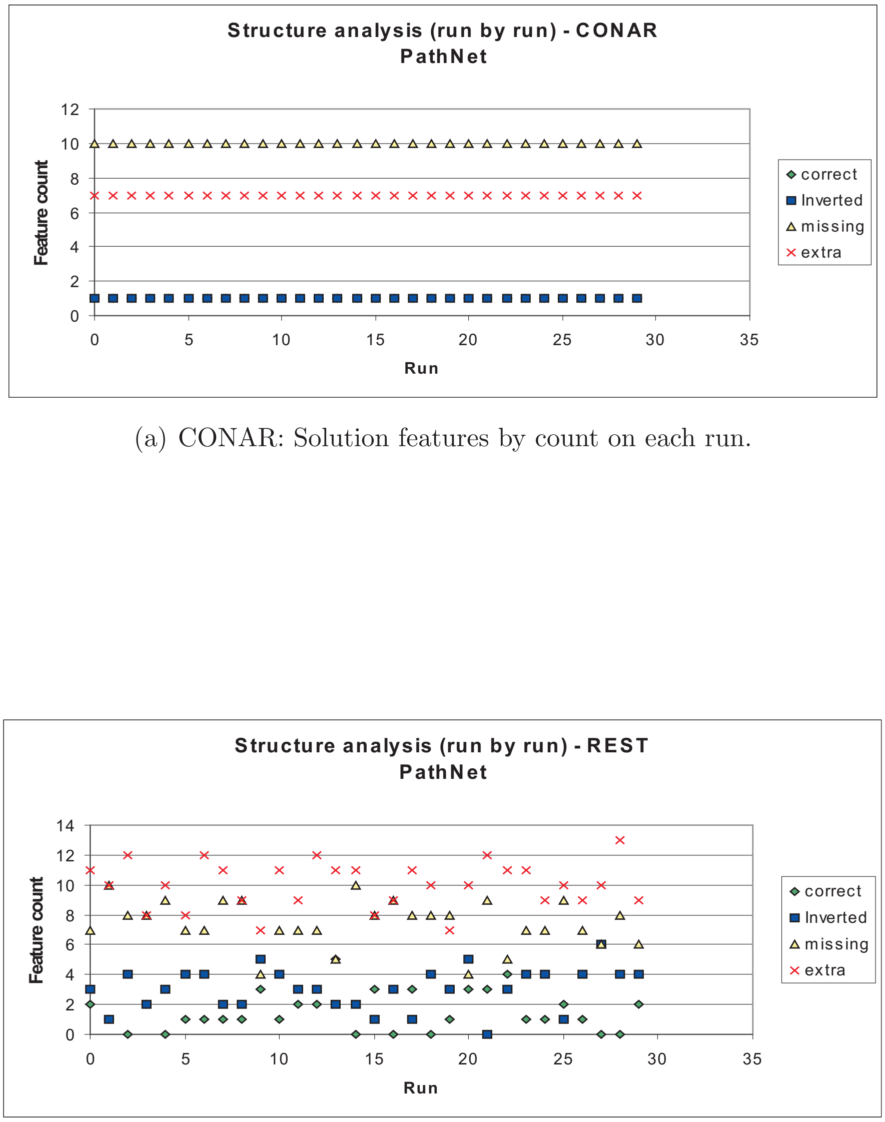


(a) CONAR: Solution features by count on each run.



(b) REST: Solution features by count on each run.

Figure 9.16: Proportion of structural features in the best solutions across all runs (DemNet)



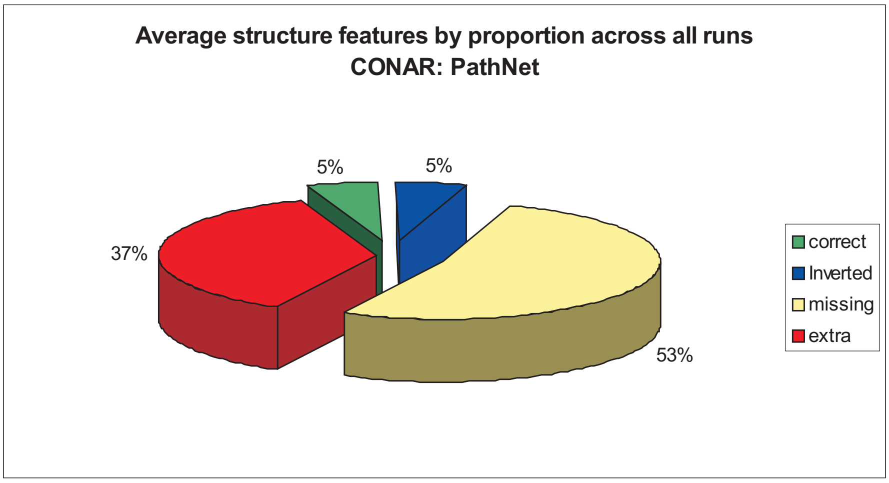
(a) CONAR: Solution features by count on each run.

(b) REST: Solution features by count on each run.

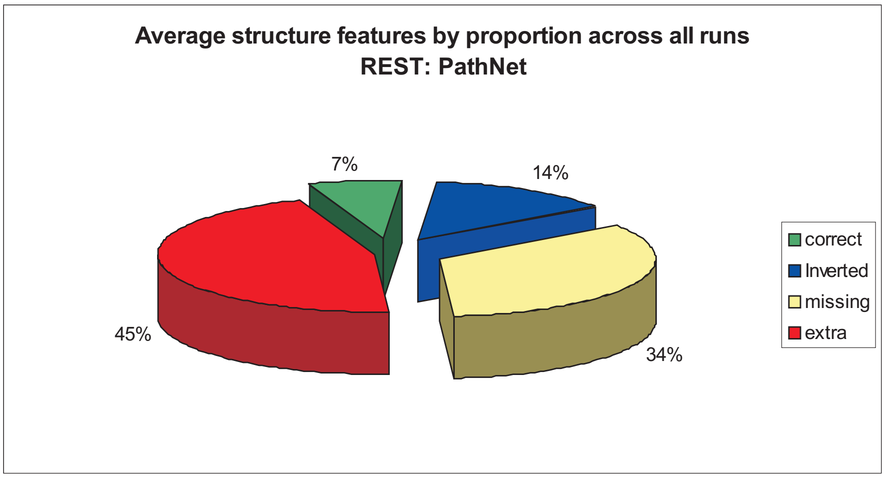
Figure 9.17: Count of structural features of the best solutions in each run (PathNet)

9.5 Summary

This chapter presents an empirical evaluation of the performance of the techniques developed in the previous chapter. Experiments are conducted with a



(a) CONAR: Solution features by proportion across all runs.



(b) REST: Solution features by proportion across all runs.

Figure 9.18: Proportion of structural features in the best solutions across all runs (PathNet)

range of standard benchmark problems, as well as a real-life problem — dementia diagnosis. The purpose of the experiments was to demonstrate:

1. The application of PSO to BN construction from data.

2. That the techniques developed do not require as input an order among the variables, and produce results that are comparable and in some cases superior to existing approaches.

3. The performance and efficiency of PSO-based approaches compared to other order-based algorithms.

From the results presented in this chapter, it is clear that the benefits of the CONAR algorithm are strong on some of the synthetic problems, however the same cannot be said for the REST algorithm. Structures close to the original were achieved on each of the synthetic problems. With regard to the performance metrics, CONAR significantly outperforms REST on all synthetic problems. Similarly, REST is outperformed by all the comparator algorithms, most notably K2no, which executes the K2 algorithm with random orders. However, comparing CONAR with other algorithms in the literature, it exhibited at least the same performance as the algorithms used for comparison, except on one problem (Alarm). It is anticipated, however, that further parameter tuning would result in an improvement in the quality score value for the Alarm problem.

In Section 8.2, we outline the motivation for PSO, and in doing so suggest that PSO has greater efficiency in some problems in exploring the search space than its GA counterpart. It is evident that in our implementation, however, efficiency (number of Fitness Evaluations, FEs) is poor in comparison to the standard GA approach (K2GA) and advanced order-based approach (ChainGA) proposed by Kabli et al. [135]. One possible explanation regarding the comparison with the standard K2GA algorithm may be associated with the use of neighbourhood strategies in the PSO algorithm (see Section 8.1.1.1). A neighbourhood strategy allows a wider area of the search space and reduces premature convergence, however the added expense is that the algorithm takes longer to propagate. With regard to the ChainGA approach, a search is conducted in the space of orders, however reliance on the K2 algorithm is greatly reduced due to the evaluation of chain structures, which ultimately reduces the number of fitness evaluations.

Nevertheless, the CONAR PSO algorithm finds equally comparable solutions to those found by the standard GA (K2GA) and the advanced ChainGA approach. Moreover, in some problems, our CONAR approach finds structures that are marginally superior to the standard K2GA and the advanced two-tiered ChainGA approach.

A comparison of the data-driven models and hand-crafted expert models (Part II), as well as a comparison of the two construction approaches is provided in the next part of this thesis.

Part IV

Comparison of approaches

Chapter 10

Comparison and evaluation of construction approaches

10.1 Introduction

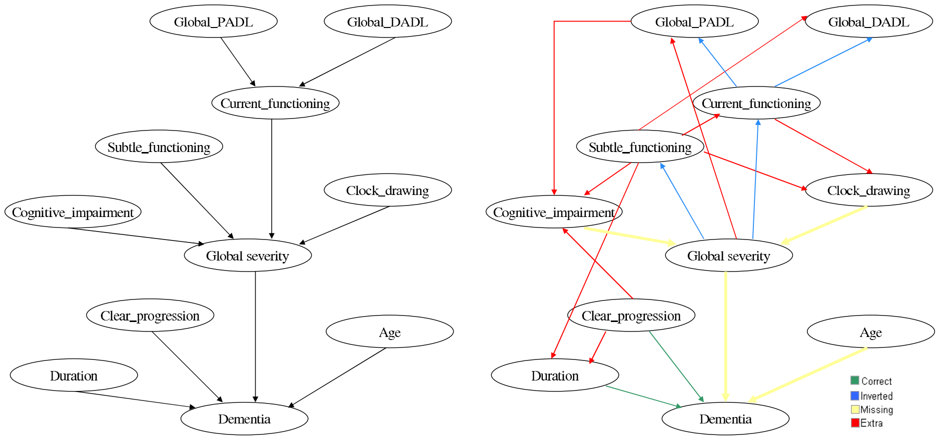
The purpose of this chapter is to assess the concordance of the models derived using the two construction approaches presented Part II and Part III. In addition, we assess the main benefits and challenges of each construction approach.

10.2 Comparison of models

In this section we show the BN structures created using each of the construction approaches and highlight and explain the differences.

10.2.1 DemNet

As can be seen from Figure 10.1(b), there is a lot of disparity between the hand-crafted and data-driven model. The hand-crafted model shown in Figure (a)



(a) Hand-crafted (reference model).

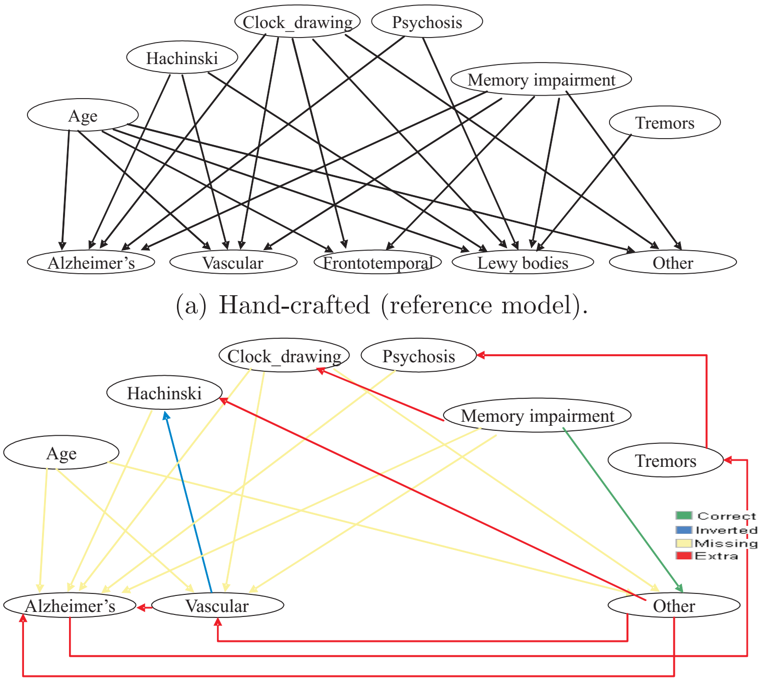
(b) Data-driven (clinical data).

Figure 10.1: DemNet constructed by hand (a) and from data (b)

is taken to be the reference model. With respect to the reference model, the “learned” model constructed from data in Figure (b) has only two correctly orientated edges and four inverted edges; irrespective of orientation, this is a total of six edges out of ten possible edges.

10.2.2 PathNet

The hand-crafted and data-driven models for dementia pathology diagnosis are shown in Figure 10.2. The hand-crafted model shown in Figure (a) is taken to be the reference model, and the model constructed from the clinical data is shown in Figure (b). In comparison to the DemNet models shown above, there is not much concordance between the hand-crafted model and the model derived from data. In fact, the model derived from clinical data has recovered only 2 edges from a possible 12, and one of those is inverted. Note that the variables frontotemporal and lewy bodies have been removed due to the lack of data.



(a) Hand-crafted (reference model).

(b) Data-driven (clinical data).

Figure 10.2: PathNet constructed by hand (a) and from data (b)

10.2.3 Discussion

From the models presented in Section 10.2.1 and Section 10.2.2, it is clear that there are marked differences in structure between the hand-crafted, reference model and the models constructed from a clinical data set.

Two reasons may explain the structural differences in the models. Firstly, the desired volume of real-life clinical data was not achieved during the data collection study. Secondly — closely related to the low data volume issue — the spread and diversity of data did not meet expectations. For example, the the reference model supports diagnosis of dementia with lewy bodies and frontotemporal dementia, notwithstanding co-existing pathologies. However, the clinical data set contained only a hand-full of these cases, therefore the construction algorithm could not recover relationships to satisfy this requirement. Accordingly, the two data-related factors are highly influential contributors, if not wholly responsible, for the structural differences that appear in the data-driven models (particularly the red and yellow edges). Clearly, when constructing BN models from data, the volume and spread of the data are two important factors that need to be considered in order to ensure the best possible chance of constructing the most accurate model.

Nevertheless, it is encouraging that 60% of the possible correct edges appear in the "learned" DemNet model. In this case, the volume of data available was enough to recover the core relationships of the model. However, the downside is that the eight extra edges (shown in red in Figure 10.1(b)), which illuminate the difference in the models, are probably due to the lack of data. For example, in a bigger data set, the distributions would be more pronounced, thus the construction engine may find a better model that has a distribution that does not support these edges. However, it is also worth considering the notion that extra edges are due to error, which may have manifested themselves during data collection.

To demonstrate that the algorithm is capable of network construction in the first place, we show the construction engine's capability and performance on data set synthetically generated from the reference model. The Netica [2] tool was used to generate $5,000$ cases. As can be seen from the resulting model, shown in Figure 10.3, the algorithm is capable of constructing the network using data generated from the reference model; it contains 90% of the edges that appear in the known model, therefore the algorithm is capable of building a model that represents the probability distributions in the data.

The comparison between the reference model and the model constructed from real-life clinical data for dementia syndrome diagnosis, shown in Figure 10.1,

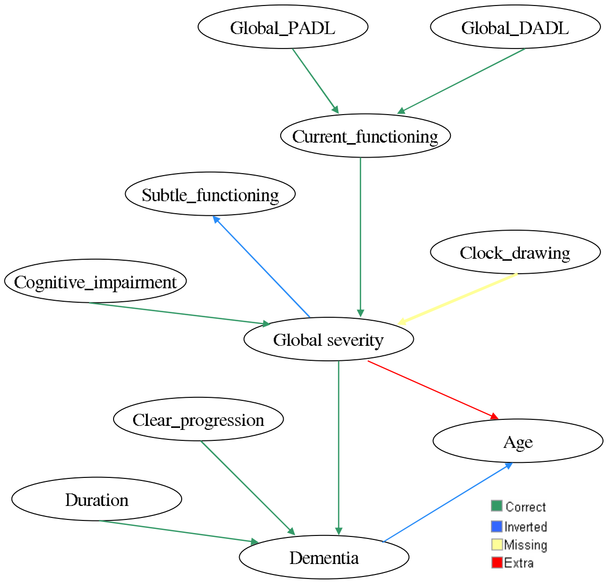


Figure 10.3: DemNet constructed from data generated from the reference model.

serves as a good indicator that there is a reasonable level of concordance with the hand-crafted reference model and the true distributions found in clinical practice. Clearly, more clinical data is required in order to test more rigorously the levels of concordance between the models. However, the 60% edge recovery rate from the clinical data set may be deemed as an achievement against a backdrop of low data volumes and limited data spread.

As can be seen from Figure 10.2, the disparity between the reference and data-driven models for dementia pathology diagnosis is vast. Clearly, this is because of the small quantity of data. Even when $5,000$ data samples are synthetically generated from the reference model, only 12 (60%) of the edges are recovered, and 8 (38%) edges are missed, as shown in Figure 10.4.

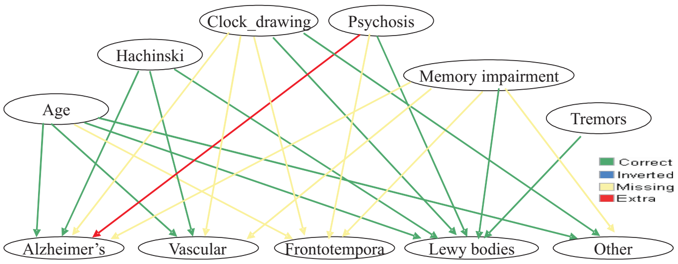


Figure 10.4: PathNet constructed from data generated from the reference model.

10.3 Evaluation of approaches

In this section we discuss the two approaches for BN construction, described in detail in Chapter 4 and Chapter 7 respectively.

10.3.1 Evaluation of hand-crafted approach

Hand-crated BN construction in its pure form is completely dependent on access to human expertise. This approach has many benefits, however there are also a number of barriers that present significant challenges during construction. Such benefits and barriers are discussed in this section.

10.3.1.1 Benefits of hand-crafted approach

One of the key benefits to be realised from the hand-crafting approach is that the model is built interactively by a domain expert, therefore vital domain knowledge is embedded in the model. In addition, the nodes, states and relationships are fully understood by the human expert, therefore the rationale can be articulated easily to others. Another benefit of the approach is that the model is built primarily using subjective consensus information, therefore uncommon scenarios that may be rare in data can be captured. The value is found in the fact that the

BN formalism does not care whether the information is objective or subjective; in fact, intrinsically, the BN formalism assumes that the information is subjective, hence the alternative name, Bayesian ‘belief’ network. The ‘belief’ word reflects the expert’s own uncertainty about the probability distribution in question. This could be a known value, however it may not be known and therefore is based on their expert judgement. Clearly, different experts will have different opinions on the subjective probability distributions, and indeed the distributions may change over time.

Another benefit of the hand-crafted approach is that one or more experts can be involved in the construction process, and therefore the technicalities of the domain and the factors that the model represents can be verified or discussed in detail at each stage of the development cycle. However, Walls and Quigley [274] note a potential drawback with group elicitation: one or more experts may be more domineering over others in the group. In order to work around this, they suggest that in “situations where the persuasive argument of dominant or over confident experts might prevail or less-confident experts might be isolated” that judgement elicitation should be carried out independently and combined afterwards.

10.3.1.2 Barriers and challenges of hand-crafted approach

Despite the benefits associated with hand-crafted approach, there are a number of barriers and challenges.

The primary barrier with this approach is concerned with probability elicitation and judgement: humans find it difficult to provide assessments for many conditional probabilities at once [124]. To that end, people have a tendency to use a heuristics to assist them in determining conditional probability distributions for several conditioning factors [188, pp 163]. However, such heuristics tend to give rise to bias [207, pp 35]. Another approach to making the assessment task easier is to restructure the model and prune less important parent nodes, especially where parents have more than two states. This has the effect of reducing the complexity of the conditional distributions, thus easing elicitation. However, restructuring or pruning the model may have an impact on the level of granularity.

With regards to assisting the expert in probabilistic elicitation and reducing the bias that is inherent in elicitation, controls in the form of protocols (formal elicitation processes) can be employed. In general, these protocols introduce the expert to the elicitation task, and make them familiar with the issues that may arise so biases can be detected as they arises during elicitation exercises.

10.3.2 Evaluation of data-driven approach

In Section 10.3.1 we reflected on the hand-crafted approach to BN construction, and highlighted the benefits and challenges. In this section, we reflect on the alternative approach: the data-driven approach.

10.3.2.1 Benefits of data-driven approach

Constructing BNs from data involves discovering the graphical structure of the network, that is the relationships between the variables, and estimation of the conditional probability distribution given the structure. It is the fact that both components of the BN are derived from data that make this an attractive approach, especially when no expert is available.

Many medical (and other) domains create, collect and maintain data sets over periods of time. Within these data sets are valuable pieces of information about the relationships that exist between the variables. Using a data-driven BN construction algorithm, it is possible to realise the relationships that exist in the data graphically in the form of a BN. Accordingly, the issues associated with the hand-crafted approach are eradicated; however, the data-driven approach comes with its own set of challenges.

10.3.2.2 Barriers and challenges of data-driven approach

The two most common challenges with this approach are: 1) data quality and quantity; and 2) automatic discovery of a BN from data is computationally complex.

Data While data may be readily available in many domains, it must be comprehensive and it must satisfy a number of criteria if it is to be suitable for BN learning. In the first instance, the variables and values in the data set should match the variables and values of the desired BN; otherwise, there must exist a consistent and reliable one-to-one mapping between the variables in the data set and the requisite variables. Also, the data collection process must be executed carefully to ensure that it is free from bias, as this could have a negative effect on the resulting model. In addition, to ensure that the construction algorithm has a chance of reliably recovering the probabilistic relationships, the volume and spread of data must be proportional to the number of variables. In general, there is no defined rule to express the volume of data samples required for BN learning. As can be seen in Section 10.2, small, sparse sample sizes can lead to models that are not representative of the true relations of the problem at hand. Another assumption that is made by many learning algorithms is that the samples in the dat set are complete — there should be no missing values. This may be viewed as an unrealistic, and to some extent utopian requirement, as most real-life data sets are not complete, especially medical data sets. However, algorithms exist that discover the network structure and parameters when data is incomplete [90, 112, 225]. Finally, algorithms that construct BNs from data, in general, make the assumption that the cases in the data set are independent of each other.

Complexity The primary drawback of the data-driven approach is that it is computationally hard. The problem arises because the number of possible structures for a given problem grows super-exponentially with the number of variables in the problem domain (see Section 7.1).

10.3.3 Other broad issues

A number of other issues, listed below, arise from the construction approaches discussed above.

Scalability Despite the tools and methods available to assist construction, the hand-crafted approach is very labour intensive. As a result, the approach can be tiring for experts. This begs the question regarding the scalability of this technique for large BN model. Two potential solutions to this scaling issue include: 1) a hybrid approach that combines construction approaches: first, a base-line model is constructed from data, then the expert tunes and moulds; and 2) a technique inspired by Object-Orientated programming, which treats pieces of knowledge as artifacts (or fragments), which are combined into the bigger BN model [158].

Network complexity The question of how complex the network should be is an interesting one. It is reasonable to think that a complex topology with many variables and relations is more beneficial than a simpler structure. However, for classification problems, Thomas [263] and Domingos and Pazzani [70] have demonstrated that a simple Bayesian classifier, such as a Naive Bayes model, perform just as well as (and in some situations outperforms) complex, sophisticated BN. In addition, Cheng and Greiner [42] propose a less restrictive option, which relies on tree-augmented networks. Such models are more flexible, as they permit extra dependency relations between the variables. Nonetheless, there remains situations when more complex models are required.

The best approach? Like many algorithms in computing science, no single construction approach is a "silver bullet" for BN construction. It may be reasonable, however, to assume that the hand-crafted approach may provide more valuable knowledge than the data-driven approach when data is sparse — this was our experience (see Section 10.2). However, there is very little evidence to support this point of view. In addition, there may equally be a scenario where the expert is overwhelmed by the complexities of elicitation, leading to a poorly defined model. In which case, a data-driven model may well derive a more accurate model, even with sparse data [191].

10.4 Summary

In this chapter we have compared and evaluated the two approaches for BN construction investigated in this thesis. We have applied both approaches to a real-life problem, and we have compared the resulting models.

There is no one single "best" approach to BN construction — each approach has its own merits and challenges. On one hand, the hand-crafted approach captures expert knowledge of a domain, however there are significant barriers in terms of elicitation. On the other hand, the data-driven approach does not rely on a domain expert, however it does rely on a comprehensive, complete data-set. In addition, the data-driven BN problem is computationally complex and therefore requires powerful algorithms to construct the model from data.

The approaches are not necessarily independent of each other. There may be situations where a hybrid that combines both approaches would be more suitable. For example, where there are hundreds of variables and there is only one expert to construct the model. In this scenario, the data-driven approach combined with the hand-crafted approach would reduce the burden on the expert and aid the construction task.

Part V

Conclusion and future work

Chapter 11

Summary, reflection, achievements and conclusions

This thesis has presented BN construction approaches, it has identified opportunities to improve automatic construction from data, and it has implemented new algorithms to implement some of the opportunities. In addition, a novel application of BN has been presented — decision support for dementia diagnosis. In this chapter the body of work contained in this thesis is summarised. The overall achievements are detailed along with a discussion on how the aims defined in Chapter 1 have been met. Afterwards, limitations of the research are addressed, and suggestions about how they could be resolved are discussed in the future work section.

11.1 Summary

This thesis started by identifying the potential of Bayesian networks (BN) for expressing and reasoning about problems characterised by uncertainty (Chapter 2). Thereafter, in Chapter 3 we identified challenges and barriers regarding the diagnosis of dementia in primary care practice, particularly at the General Practitioner (GP) level. To address this issue, we proposed a novel application of Bayesian networks (BN) to provide medical decision support for the diagnosis of dementia in primary care practice. However, the primary challenge with any application of BNs is that they first need to be constructed before they can be used.

We then focused on the two approaches for BN construction, namely the hand-crafted approach and the data-driven approach (Part II and Part III). We investigated and reviewed a range of techniques for each of these approaches. In Part II we provide a detailed description of a number of tools to support the hand-crafted approach, and we demonstrated how these techniques are applied to a real-life problem. In Part III, we investigated issues with the the data-driven construction approach and proposed a new algorithm to address some of these issues. We evaluated the new algorithms against a range of problems, including the real-life dementia diagnosis problem, and compared their performance to existing and new data-driven algorithms that appear in the literature.

Finally, in Part IV, we compare and discuss the two approaches, as well as the BN models derived from each approach.

11.2 Contributions of research

The initial aims of this research are detailed in Chapter 1; however, they are summarised below.

1. To conduct research in the area of BN construction

2. To investigate the hand-crafted approach to BN construction and create a practical framework to assist non-BN experts with model construction for real-life applications

3. To investigate the potential for BN models for dementia diagnosis, and develop a hand-craft for dementia diagnosis using expert knowledge

4. To determine the performance of BN models for dementia diagnosis

5. To investigate existing data-driven approaches and identify and implement development opportunities

6. To investigate the performance of the new algorithms compared to new and existing data-driven algorithms that appear in the literature

Through this programme of research, it is felt that the research objectives outlined have been met. A breakdown of the contributions and how they have been met are listed below.

This thesis started by identifying the potential of Bayesian networks (BN) for expressing and reasoning about problems characterised by uncertainty (Chapter 2). Thereafter, in Chapter 3, we identified challenges and barriers regarding the diagnosis of dementia in primary care practice, particularly at the General Practitioner (GP) level. To address this issue, we proposed a novel application of Bayesian networks (BN) to provide medical decision support for the diagnosis of dementia in primary care practice. However, the primary challenge with any application of BNs is that they first need to be constructed before they can be used.

Compiled a framework using common methods selected from the literature to support hand-crafted BN construction, and showed how they are applied in a novel way to a real-life problem The potential of (BN) for expressing and reasoning about problems characterised by uncertainty is well known. However, before a BN can be used it must first be constructed — this is a significant challenge, and perhaps one of the perceived drawbacks of the BN formalism. In Part II of this thesis the hand-crafted construction approach is treated.

- An in-depth investigation into the the hand-crafted BN construction approach is provided in Chapter 4. The challenges and barriers of the hand-crafted approach, such as cognitive bias, are reviewed. At the beginning of this research project, it was felt that one of the barriers preventing the adoption of BNs is due to their construction. Therefore, in order to equip non-BN experts for the challenge of hand-crafting BNs, a framework containing a selection of methods from the literature that address issues such as bias are presented.
- Chapter 3 describes in details a real-life domain — dementia diagnosis — and highlights the challenges with dementia diagnosis in clinical practice. Bayesian networks are proposed as a decision support engine.
- The research contribution is extended in Chapter 5 through a demonstration of how the framework presented in Chapter 4 can be applied to the construction of BN models for dementia diagnosis decision support. The models are evaluated empirically in Chapter 6.

Investigated the data-driven approach to BN construction, identified issues, and developed a new construction algorithm to address these issues The alternative to the hand-crafted construction approach is the data-driven approach, which is treated in Part III. However, this approach is no silver-bullet to the construction problem.

- An introduction and review of the data-driven approach is provided in Chapter 7. The primary contribution of this chapter is to bring to the forefront the computational complexity associated with automatically constructing BN models from data, as well as to review existing algorithms in the literature. At the outset of this research, it was felt that some of the existing data-driven algorithms that require human input at the outset may be restrictive, as the information may not be available. In this chapter, we identify development opportunities to improve BN construction from data, specifically targeting algorithms with the following characteristics: 1) algorithms that search the entire space of structures, such as Larrañaga et al. genetic algorithm [155] – is it possible to improve the quality of solutions found as well as efficiency?; and 2) order-based algorithms reduce the search space, and when a good order is provided, good BN structures are found. However, we argue that the prerequisite to input an order among the nodes is not always practical as an expert is not always available to provide the order – can we find a suitable algorithm to search the entire space efficiently such that an order is not required?.

- Since the primary issue of data-driven BN construction lies in the computational complexity of searching through a complex, high-dimensional search space, we turned to the field of Evolutionary Computation (EC), as these algorithms are generally well suited to exploring such problem spaces. In order to investigate the above, a set of new algorithms from the EC field are proposed. Chapter 8 expands this research by detailing the two algorithms developed, which are based on binary Particle Swarm Optimisation (PSO); these algorithms search through the complex, high-dimensional space of BN structures for the one that best represents the relations in the data set.

- Chapter 9 provide details of the experiments conducted, which demonstrate the performance of the algorithms on synthetic and real-life problems. Experimentation in Chapter 6 shows that the models, in general, have good ability to distinguish between the diagnostic groups (according to Pearce and Ferrier's [215] qualitative translation of classification performance). Using synthetic benchmark data sets and a real-life clinical data set, the performance of the new algorithms is assessed against classic and new algorithms in the literature. One of the algorithms performed well on most of the test problems.In addition, the algorithms are compared to other algorithms taken from the literature that explore the entire space of candidate structures as well as those order-based algorithms.

Compared the construction approaches and the models derived by each approach Two approaches for BN construction are discussed in this thesis. At the end of Part II, the performance of the hand-crafted dementia models is measured; similarly, the performance of the new data-driven algorithms are measured at the end of Part III. The contribution of this part is to allow comparison in Part IV of the two BN construction approaches, as well as the models derived by each approach.

- Like many approaches in Computing Science (and other domains), there is no single “best” approach to BN construction. The pros and cons of each construction approach are set out and discussed in Chapter 10.

11.3 Key strengths

The key strengths of this research are covered by the aims defined in Chapter 1.3.

1. **Review construction approaches** The review identified two approaches for BN construction, and it highlighted the challenges and barriers with each of the approaches. As a result, a number of development opportunities were proposed.

2. **Implement development opportunities** The hand-crafted approach by definition relies on human experts. In addition, there are a number of challenges for non-BN experts, such as cognitive bias. To manage end to end construction and the common issues faced, a construction process framework is defined to assist non-BN experts. Moreover, a framework is provided to assist during elicitation tasks.

   The data-driven construction approach also faces challenges. Algorithms considered in this research require an order among the variables as input, or use a two-tiered algorithm to achieve construction. New algorithms are proposed. The approach presented is able to find comparable solutions (worst case) by searching the entire space of BN structures without an order as input, and the algorithms do not require a two-tiered algorithm for searching the order space.

The space in which we perform our search (entire space) requires additional operators to verify integrity of the solutions. In one of our approaches we offer a mechanism that relaxed the need for such operators (but it didn't work well).

3. **BN for dementia diagnosis** Bayesian networks have successfully been applied to dementia diagnosis. The output is probabilistic therefore aligns closely with clinical practice — other tools for dementia diagnosis in the literature do not give probabilistic output. Reference models have been hand-crafted, serving to demonstrate the construction process and tools presented in Chapter 4. The models developed are capable of assisting dementia syndrome and pathology diagnosis, both singly and in combination.

4. **Systematic evaluation of new approaches** The algorithms presented seek to improve on the performance of existing algorithms in the literature. This was demonstrated on a number of benchmark problems as well as a real-life problem.

5. **Evaluate construction approaches** Two different approaches are adopted for BN construction; they are both analysed and compared. In addition, the structural differences between the derived models is assessed.

11.4 Limitations of research

Although the research in this thesis has many advantages and contributes to applications in health-decision support and data-driven BN construction, it does have some limitations. We divide the limitations into two parts, and address them in this section.

Bayesian network construction

1. **REST construction algorithm** The REstricted STructure algorithm is an extension to the CONstruct And Repair (CONAR) algorithm. REST was designed to circumvent the need to employ validation and repair operators when searching in the entire space of BN structures. This solution modifies the representation scheme such that a particle encodes two components: 1) triangulated connectivity matrix, which guarantees legal solutions and 2) a permutation, which provides flexibility in the possible relationships between nodes (see Section 8.6). Using this encoding scheme, no repair or validation operators are required.

   However, as can be seen from the results in Chapter 9, REST does not appear capable of recovering structures from data.

   Novobilski [201] has proposed a mechanism to ensure legally encoded BN models during stochastic update, however this technique has been implemented for a genetic algorithm, although it is possible to adopt Novobilski's approach in Particle Swarm Optimisation. However, analysing further the mechanics of Novobilski's solution, it appears that there is still an element of validation and repair.

   A suitable binary representation is required that permits only legal BN structures during stochastic update.

2. **Parameter setting** Section 9.3 shows the results of the approaches when executed on a number of benchmark data sets from the literature. Out of all the test problems, CONAR performs worst on the Alarm problem. It is thought that the parameters need tuned further in order to improve

the results. The problem of parameter tuning, however, is faced by many nature-inspired algorithms.

3. **Comparative performance - computational effort** The computational effort required by the algorithm is measured in terms of the number of fitness evaluations (see Section 9.2.3) required on average to converge to a solution. In Section 8.2, we note that part of the motivation for PSO is its ability to produce high quality solutions with lower effort when compared to the genetic algorithm, for example. Although our approach produces, in general, solutions of comparable quality to other algorithms, as can be seen from Table 9.7 through Table 9.9, the number of fitness evaluations required by our approach is far more than those the number of fitness evaluations required by the approaches that we use for comparison.

4. **Comparative performance - significance** In Section 9.3.2.2 we provide an empirical comparison of our approach with three other algorithms in the literature, namely K2, K2GA and ChainGA. We were able to obtain our own results from the K2 (K2O and K2NO) algorithm, however the results of the K2GA and ChainGA were taken from Kabli et al. [135]] paper. We have been unable to obtain the raw result data. Although PSO appear to marginally outperform the K2GA and ChainGA, we are unable to validate the statistical significance without the raw results data.

Application to dementia diagnosis

1. **Elicitation** One of the limitations in the approach to hand-crafting the dementia models is that only one expert was used during elicitation. However,

the models received peer review at a number of points during development, and the models were socialised at health-care conferences. Feedback was incorporated.

2. **Clinical data** No pertinent clinical data existed at the start of this research, and in order to test the accuracy of the hand-crafted models, as well as the data-driven construction approaches, data from clinical practice was required. A data collection study was initiated with a target of 350 samples. However, only 154 samples were attained. Furthermore, the spread of the data was limited, particularly the pathology data. To that end, we were able to measure the classification accuracy of only three of the diagnostic outputs, namely Alzheimer's disease, vascular dementia and other. We were not able to measure the classification accuracy of the hand-crafted models on dementia with lewy bodies, frontotemporal dementia and co-existing pathologies. Extremity testing was carried out on the diagnostic outcomes omitted from the test strategy, and the model appeared to output values inline with the expert's expectations.

   In addition, due to the lack of data, it was decided not to test the classification accuracy of the models derived from the clinical data set.

11.5 Future work

The BN models for dementia diagnosis have been peer-reviewed by experts during their development; however, if they are to be used in live clinical practice, they will require full evaluation in the form of a clinical trial.

In our experimentation, we have evaluated the the prediction accuracy of the hand-crafted models against clinical data. However, further evaluation is required by measuring the prediction accuracy of the learned, although more data would be required.

Analysis of CONAR and REST results on benchmark problems showed that they under-performed on the complex Alarm problem. An extension to the PSO algorithm, namely Clerc's restriction criteria, may assist convergence, and hence improve the quality of solutions [47].

It may be possible to reduce the number of fitness function evaluations by caching scoring computations. However, this would depend on how much the solutions in the swarm change over time, which would require prior analysis.

Both CONAR and REST use PSO and search in a different space from the nature-inspired algorithms used for comparison, namely K2GA and ChainGA. Since CONAR and REST differ in two fundamental ways from these algorithms (search space and algorithm), it is difficult to explain the reason for the difference in performance. To enable more straight forward comparisons, we propose that K2GA and ChainGA are compared with K2PSO and ChainPSO equivalents.

With regards to the consistently poor results achieved by REST, further investigation is required to determine ways in which the encoding mechanism can be finely tuned to increase performance.

The CONAR and REST algorithms have been tested on standard benchmark problems, and they have been applied to dementia diagnosis data from clinical practice. However, a range of other real-life applications from various domains ought to be tested.

A number of nature-inspired algorithms have been applied to the BN construction from data. However, there is no overarching study that systematically measures performance across a number of problems. Such a study would be useful, as there may be trends that emerge indicating that certain algorithms perform better/worse on specific types of problems.

Bibliography

[1] Bayesialab. http://www.bayesia.com.

[2] Netica, norsys corporation. http://www.norsys.com.

[3] Weka: Data mining software in java. www.cs.waikato.ac.nz/ml/weka/.

[4] *Diagnostic and Statistical Manual of Mental Disorders.* American Psychiatric Publishing, Inc, fourth edition, 2004.

[5] S. Anderson, D. Madigan, and M. Perlman. A characterization of markov equivalence classes for acyclic digraphs. *Annals of Statistics*, 25:505–541, 1997.

[6] S. Andreassen, M. Woldbye, B. Falck, and S.K. Andersen. Munin–a causal probabilistic network for interpretation of electromyographic findings. In *Proceedings of the 10th International Joint Conference on Artificial Intelligence*, pages 366–372, 1987.

[7] D.N. Barton, T. Saloranta, S.J. Moe, H.O. Eggestad, and S. Kuikka. Bayesian belief networks as a meta-modelling tool in integrated river basin management – pros and cons in evaluating nutrient abatement decisions under uncertainty in a norwegian river basin. *Ecological Economics*, 66(1):91–104, 2008.

[8] E. Bauer, B. Koller, and Y. Singer. Update rules for parameter estimation in bayesian networks. In D. Geiger and P. Shanoy (Eds), editors, *Proceedings Thirteenth Conference on Uncertainty in Artificial Intelligence (UAI '97)*, pages 3–13. Morgan Kaufmann, San Francisco, Calif., 1997.

[9] K.C. Baumbgartner, S. Ferrari, and G.C. Salfati. Bayesian network modeling of criminal behavior for criminal profiling. In *44th IEEE Conference on decision and control, 2005 and 2005 European Control Conference (CDC-ECC '05)*, pages 6480–6485, Seville, Spain, December 2005.

[10] M.A. Beaumont and B. Rannala. The bayesian revolution in genetics. *Nature*, 5(4):251–261, April 2004.

[11] I. Beinlich, H.J. Suermondt, R.M. Chavez, and G.F. Cooper. The alarm monitoring system: A case study with two probabilistic inference techniques for bayes nets. In *Proceedings of the Second European Conference on Artificial Intelligence in Medicine*, pages 247–256, 1989.

[12] B. Bergener. Normal aging and cognitive impairment revisited. *International Psychogeriatrics*, 7:475–477, 1995.

[13] BMJ BestTreatments. Dementia (alzheimer's disease, lewy body dementia, vascular dementia): What treatment works? rivastigmine. World Wide Web, June Accessed 6 August 2007.

[14] A. Bianchetti and M. Trabucchi. Behavioural and psychological symptoms of dementia: clinical aspects. *Neuroscience Research Communications*, 35(3):173–183, March 2005.

[15] A.D. Blackwell, B.J. Sahakian, R. Vesey, J.M. Semple, T.W. Robbins, and J.R. Hodges. Detecting dementia - novel neuropsychological markers of preclinical alzheimers disease. *Dementia and Geriatric Cognitive Disorders*, 17:42–48, October 2004.

[16] R. Blanco. *Learning Bayesian Networks from data with Factorisation and Classification Purposes. Applications in Biomedicine.* PhD thesis, Department of Computer Science and Artificial Intelligence of the University of the Basque Country, Spain, 2005.

[17] R. Blanco, I. Inza, and P. Larrañaga. *[91]*, chapter Learning Bayesian networks by floating search methods, pages 181–200. [91], 2004.

[18] R. Blanco, I. Inza, and P. (2002) Larrañaga. Floating search methods in learning bayesian networks. In *Proceedings of the First European Workshop on Probabilistic Graphical Models, PGM'02*, pages 9–16, 2002.

[19] B.W. Boehm. A spiral model of software development and enhancement. *IEEE Computer*, 21(5):61–72, 1988.

[20] T. Boneh, A. E. Nicholson, and E. A. Sonenberg. Matilda: A visual tool for modeling with bayesian networks: Research articles. *International Journal of Intelligent Systems*, 21(11):1127–1150, 2006.

[21] H. Brodaty, G.C. Howarth, A. Mant, and S.E Kurrle. General practice and dementia. *Medical Journal of Australia*, 160:10–14, 1994.

[22] H. Brodaty, L-F. Low, L. Gibson, and K. Burns. What is the best dementia screening instrument for general practitioners to use? *American Journal of Geriatric Psychiatry*, 14:391–400, 2006.

[23] H. Brodaty and C.M. Moore. The clock drawing test for dementia of the alzheimer's type: A comparison of three scoring methods in a memory disorders clinic. *International Journal of Geriatric Psychiatry*, 12:619–627, 1997.

[24] D.W. Bunn. Anchoring bias in the assessment of subjective probability. *Operational Research Quarterly*, 26(2):449–454, July 1975.

[25] W. Buntine. Theory refinement in bayesian networks. In *In Proceedings of the 7th Conference on Uncertainty in Artificial Intelligence.*, pages 52–60, 1991.

[26] W. Buntine. Operations for learning with graphical models. *Artificial Intelligence Research*, 2:159–225, 1994.

[27] W. Buntine. A guide to the literature on learning probabilistic networks from data. *Ieee Trans. On Knowledge And Data Engineering*, 8:195–210, 1996.

[28] E. Burnside. Bayesian networks: Computer-assisted diagnosis support in radiology. *Academic Radiology*, 12(4):422–430, 2005.

[29] E.S. Burnside, D.L. Rubin, J.P. Fine, R.D. Shachter, G.A. Sisney, and W.K. Leung. Bayesian network to predict breast cancer risk of mammographic microcalcifications and reduce number of benign biopsy results: Initial experience. *Radiology*, 240:666–673, September 2006.

[30] S. Cahill, M. Clark, C. Walsh, H. O'Connell, and B. Lawlor. Dementia in primary care: the first survey of Irish general practitioners. *International Journal of Geriatric Psychiatry*, 21:319–324, 2006.

[31] H. Campbell, R. Hotchkiss, N. Bradshaw, and M. Porteous. Integrated care pathways. *British Medical Journal*, 316:133–137, January 1998.

[32] A. Carlisle and G. Dozier. An off-the-shelf pso. In *Proceedings Workshop on Particle Swarm Optimization.*, Indianapolis, 2001.

[33] M.A. Carrillo, F.J.C. Ortiz, R. Morales-Menndez, and L.E.G. Castaon. Learning bayesian network structures from small datasets using simulated annealing and bayesian score. In *[104]*, pages 375–380, 2005.

[34] E. Castillo, J.M. Menendez, and Sanchez-Cambronero S. Predicting traffic flow using bayesian networks. *Transportation Research Part B: Methodological*, 42(5):482–509, 2008.

[35] P.A.D. Castro and F.J. von Zuben. An immune-inspired approach to bayesian networks. In *HIS '05: Proceedings of the Fifth International Conference on Hybrid Intelligent Systems*, pages 23–28, Washington, DC, USA, 2005. IEEE Computer Society.

[36] Jr C.E. Kahn, L.M. Roberts, K.A. Shaffera, and P. Haddawya. Construction of a bayesian network for mammographic diagnosis of breast cancer. *Computers in Biology and Medicine.*, 27(1):19–29, 1997.

[37] Alzheimer's Disease Education & Referral (ADEAR) Center. Alzheimer's disease genetics - fact sheet. `http://www.nia.nih.gov/NR/rdonlyres/3C4B634E-A2D8-4415-927F-4B79BEC47EA6/2377/Alzheimers\_Disease\_Genetics\_Fact\_Sheet.pdf`.

[38] F.T.S. Chan and M. KumarTiwari, editors. *Swarm Intelligence, Focus on Ant and Particle Swarm Optimization*. I-Tech Education and Publishing, Vienna, Austria, 2007.

[39] M. Chávez, G. Casas, R. Falcón, J.E. Moreira, and R.G. Ábalo. Building fine bayesian networks aided by pso-based feature selection. In *[94]*, pages 441–451, 2007.

[40] J. Cheng, D.A. Bell, and W. Liu. An algorithm for bayesian belief network construction from data. In *Proceedings of the Sixth International Workshop on Artificial Intelligence and Statistics, AI&STAT'97*, pages 83–90, 1997.

[41] J. Cheng, D.A. Bell, and W. Liu. Learning belief networks from data: An information theory based approach. In *Proceedings of the Sixth ACM International Conference on Information and Knowledge Management, CIKM'97*, pages 325–331, 1997.

[42] J. Cheng and R. Greiner. Comparing bayesian network classifiers. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI'99)*, pages 101–107. Morgan Kaufmann Publishers, August 1999.

[43] D.M. Chickering. A transformational characterization of equivalent bayesian networks. In P. Besnard and S. Hanks, editors, *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 87–98. Morgan Kaufmann, 1995.

[44] D.M. Chickering. Learning equivalence classes of bayesian-network structures. *Machine Learning Research*, 2:445–498, 2002.

[45] D.M. Chickering, D. Geiger, and D. Heckerman. Learning bayesian networks is np-hard. Technical Report MSR-TR-94-17, Microsoft, November 1994.

[46] M. Clerc and J. Kennedy. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evolutionary Computation*, 6(1):58–73, 2002.

[47] M Clerk and J. Kennedy. The particle swarm - explosion, stability and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation.*, 6(1):58–73, 2002.

[48] Audit Commission. Forget me not: Mental health services for older people. Report, Audi Commission, London, 2000.

[49] R.M. Cooke. *Experts in Uncertainty.* Oxford University Press, USA, 1991.

[50] G.F. Cooper. *Computation, Causation & Discovery*, chapter An overview of the representation and discovery of causal relationships using Bayesian networks, pages 3–62. AAAI Press/MIT Press, Cambridge, MA, 1999.

[51] G.F. Cooper and E. Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–347, 1992.

[52] G.F. Cooper and E. Herskovits. A bayesian method for the induction of probabilistic networks from data. Technical Report KSL-91-02, Stanford University, November 1992.

[53] E.S. Correa, A.A. Freitas, and C.G. Johnson. Particle swarm and bayesian networks applied to attribute selection for protein functional classification.

In *GECCO '07: Proceedings of the 2007 GECCO conference companion on Genetic and evolutionary computation*, pages 2651–2658. ACM, 2007.

[54] C. Cotta and J. Muruzábal. Towards a more efficient evolutionary induction of bayesian networks. In *Parallel Problem Solving from Nature VII*, pages 730–739, 2002.

[55] V.M.H CoupéE, L.C van der Gaag, and J.D.F. Habbema. Sensitivity analysis: an aid for belief-network quantification. *The Knowledge Engineering Review*, 15(3):215–232, 2000.

[56] R. G. Cowell. Parameter estimation from incomplete data for bayesian networks. In D. Heckerman and J. Whittaker (Eds.), editors, *In Artificial Intelligence and Statistics: Proceedings of the 7th International Workshop on Artificial Intelligence and Statistics*, pages 193–196. Morgan Kaufmann, San Francisco, 1999.

[57] J. Cowie, L. Oteniya, and R. Coles. Particle swarm optimisation for learning bayesian networks. In Sio Iong Ao, Leonid Gelman, David W. L. Hukins, Andrew Hunter, and A. M. Korsunsky, editors, *Proceedings of World Congress on Engineering*, Lecture Notes in Engineering and Computer Science, pages 71–76. Newswood Limited, 2007.

[58] N. Cruz-Ramirez, H.G Acosta-Mesa, H. Carrillo-Calvet, L.A Nava-Fernandez, and R.E. Barrientos-Martinez. Diagnosis of breast cancer using bayesian networks: A case study. *Computers in Biology and Medicine*, 37(11):1553–1564, 2007.

[59] R. Daly, Q. Shen, and S. Aitken. Using ant colony optimisation in learning bayesian network equivalence classes. In *Proceedings of the 2006 UK Workshop on Computational Intelligence*, pages 111–118, 2006.

[60] L. Davis and M. Steenstrup. *Genetic Algorithms and Simulated Annealing: An Overview.* Morgan Kaufman, CA, 1987.

[61] A.P. Dawid. Conditional independence in statistical theory. *J. R. Statist. Soc. Series B*, 41(1):1–31, October 1978.

[62] L.M. de Campos, J.M. Fernández-Luna, J.A. Gamez, and J.M. Puerta. Ant colony optimization for learning bayesian networks. *International Journal of Approximate Reasoning - Elsevier*, 31(3):291–311, 2002.

[63] J. De Lepeleire and J. Heyrman. Diagnosis and management of dementia in primary care at an early stage: the need for a new concept and an adapted procedure. *Journal of Theoretical Medicine Bioethics*, 20(3):215–228, 1999.

[64] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms.* John Wiley & Sons, Inc., New York, NY, USA, 2001.

[65] R.J.P. deFigueiredo, W.R. Shankle, A. Maccato, M.B. Dick, P. Mundkur, I. Mena, and C.W. Cotman. Neural-network-based classification of cognitively normal, demented, alzheimer disease and vascular dementia from single photon emissionwith computed tomography image data from brain. *Proceedings of the National Academy of Sciences*, 92:5530–5534, 1995.

[66] Alzheimer's disease society. Clinical features of dementia, August 2007. `http://www.alzheimers.org.uk/Working\_with\_people\_with\`

_dementia/Primary_care/Dementia_diagnosis_and_management_in_primary_care/dementia.html.

[67] Alzheimer's disease society. Facts about dementia: Genetics and dementia, July 2007.

[68] Alzheimer's disease society. How is dementia diagnosed?, August 2007. http://www.alzheimers.org.uk/How_is_dementia_diagnosed/index.htm.

[69] Alzheimer's disease society. Integrated care pathway for young onset dementia in the west midlands, August 2007. http://www.alzheimers.org.uk/Younger_People_with_Dementia/PDF/YODRegionalPathway.pdf.

[70] P. Domingos and M. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29(2-3):103–130, 1997.

[71] C. Donaldson, N. Tarrier, and A. Burns. The impact of the symptoms of dementia on caregivers. *British Journal of Psychiatry*, 170:62–68, 1997.

[72] M.G. Downs. The role of general practice and the primary care team in dementia diagnosis and management. *International Journal of Geriatric Psychiatry*, 11:937–942, 1996.

[73] Marek J. Druzdzel and F. Javier Díez. Criteria for combining knowledge from different sources in probabilistic models. In *Proceedings of Sixteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-2000)*, Working Notes of the Workshop on Fusion of Domain Knowledge with Data for Decision Support, pages 23–29, Stanford, CA, June 2000.

[74] Marek J. Druzdzel and Francisco J. Díez. Combining knowledge from different sources in causal probabilistic models. *Journal of Machine Learning Research*, 4:295–316, July 2003.

[75] M.J. Druzdzel and L.C. van-der Gaag. Building probabilistic networks: 'where do the numbers come from?'. *IEEE Transactions on Knowledge and Data Engineering*, 12(4):481–486, July/August 2000.

[76] T. Du, S.S. Zhang, and Z. Wang. Efficient learning bayesian networks using pso. In *International conference on Computational intelligence and security (CIS2005)*, volume 3801/2005 of *Lecture notes in computer science*, pages 151–156. Springer Berlin / Heidelberg, December 2005.

[77] R. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In T. Fukuda, editor, *Proceedings of the Sizth International Symposium on Micro Machine and Human Science*, pages 39–43, Nagoya Municipal Industrial Research Institute, Nagoya, Japan, October 1995. Nagoya Municipal Industrial Research Institute, IEEE.

[78] R. Eberhart and Y. Shi. Modified particle swarm optimizer. In *IEEE International Conference on Evolutionary Computation (ICEC'98)*, pages 69–73. IEEE, 1998.

[79] R. Eberhart and Y. Shi. Empirical study of particle swarm optimization. In *In Proceedings of the IEEE Congress on Evolutionary Computation(CEC'99)*, pages 1945–1950, 1999.

[80] M. Eccles, J. Clarke, M. Livingston, N. Freemantle, and J. Mason. North of England evidence based guidelines development project: guideline for

the primary care management of dementia. *BMJ*, 317:802–808, September 1998.

[81] S.B. English, S-C Shih, M.F. Ramoni, L.E. Smith, and A.J. Butte. Use of bayesian networks to probabilistically model and improve the likelihood of validation of microarray findings by rt-pcr. *Journal of Biomedical Informatics*, 2008. In Press.

[82] H.-Y. Fan and Y. Shi. Study on vmax of particle swarm optimization. In *Proceedings of the Workshop on Particle Swarm Optimization 2001*, Indianapolis, IN: Purdue School of Engineering and Technology, IUPUI, 2001.

[83] G. Feder, M. Eccles, R. Grol, C. Griffiths, and J. Grimshaw. Clinical guidelines: Using clinical guidelines. *British Medical Journal*, 318:728–730, March 1999.

[84] N. Fenton, M. Neil, and D. Marquez. Agena: Bayesian network and simulation software for risk and decision support. `http://www.agena.co.uk/resources/white_papers/fentonMMR_Full_v1_0.pdf`, August 2007.

[85] W. R. Ferrel. *Subjective Probability*, chapter Discrete Subjective Probabilities and Decision Ananlysis: Elicitation, Calibration and Combination. John Wiley and Sons Ltd., 1994.

[86] S.I. Finkel, A. Burns, and G. Cohen. Behavioural and psychological symptoms of dementia: a clinical and research update. *International Journal of Psychogeriatrics*, 12(s1):13–18, 2000.

[87] M.F. Folstein, Folstein. S.E., and P.R. McHugh. "mini-mental state". a practical method for grading the cognitive state of patients for the clinician". *Journal of Psychiatric Research*, 12(3):189–98, 1975.

[88] R.H. Fortinsky, A. Leighton, and J.H. Wasson. Primary care physicians' diagnostic, management, and referral practices for older persons and families affected by dementia. *Research on Aging*, 17(2):124–148, 1995.

[89] B.M. French, M.R. Dawson, and A.R. Dobbs. Classification and staging of dementia of the alzheimer type: a comparison between neural networks and linear discriminant analysis. *Archives of Neurology*, 54(8):1001–1009, August 1997.

[90] F. Friedman. Learning belief networks in the presence of missing values and hidden variables. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 125–133. Morgan Kaufmann, 1997.

[91] J.A. Gámez, S. Moral, and A. Salmerón, editors. *Advances in Bayesian Networks*, volume 146 of *Studies in Fuzziness and Soft Computing*. Springer Verlag, 2004.

[92] P. Garcia, A. Amandi, S. Schiaffino, and M. Campo. Evaluating bayesian networks' precision for detecting students' learning styles. *Computers & Education*, 49(3):794–808, 2007.

[93] Serge Gauthier. *Alzheimers Disease in Primary Care: Pocketbook*. Martin Dunitz Medical Pocket Books. Taylor & Francis, 1999.

[94] A.F. Gelbukh and A.F.K Morales, editors. *MICAI 2007: Advances in Artificial Intelligence, 6th Mexican International Conference on Artificial*

Intelligence, Aguascalientes, Mexico, November 4-10, 2007, Proceedings, volume 4827 of *Lecture Notes in Computer Science*. Springer, 2007.

[95] D.S. Geldmacher, G. Provenzano, T. McRae, V. Mastey, and J.R. Ieni. Donepezil is associated with delayed nursing home placement in patients with alzeimer's disease. *Journal of the American Geriatrics Society*, 51:937–944, 2003.

[96] O. Gevaert, F. De Smet, E. Kirk, B. van Calster, T. Bourne, S. van Huffel, T. Moreau, D. Timmerman, B. De Moor, and G. Condous. Predicting the outcome of pregnancies of unknown location: Bayesian networks with expert prior information compared to logistic regression. *Human Reproduction*, 21(7):1824–1831, 2006.

[97] G. Gigerenzer and U. Hoffrage. How to improve bayesian reasoning without instruction: frequency formats. *Psychol Rev*, 102:684–704, 1995.

[98] H. Giles. Using bayesian networks to examine consistent trends in fish farm benthic impact studies. *Aquaculture*, 274(2-4):181–195, 2008.

[99] C.J Gill, L. Sabin, and C.H. Schmid. Why clinicians are natural bayesians. *British Medical Journal*, 330:1080–1083, 2005.

[100] S. Gillispie and M. Perlman. Enumerating markov equivalence classes of acyclic digraph models. In *In Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence*, pages 171–177, 2001.

[101] R. Grol, J. Dalhuijsen, S. Thomas, C. Veld, G. Rutten, and H. Mokkink. Attributes of clinical guidelines that influence use of guidelines in general prac-

tice: observational study. *British Medical Journal*, 317:858–861, September 1998.

[102] V.C. Hachinski, L.D. Iliff, E. Zilhka, G.H. du Boulay, V.L. McAllister, J. Marshall, R.W. Russell, and L. Symon. Cerebral blood flow in dementia. *Archives of Neurology*, 32(9):932–637, Sep 1975.

[103] P.W. Hamilton, N. Anderson, P.H. Bartels, and D. Thompson. Expert system support using bayesian belief networks in the diagnosis of fine needle aspiration biopsy specimens of the breast. *Clinical Pathology*, 47:329–336, April 1994.

[104] M. H. Hamza, editor. *Artificial Intelligence and Applications (AIA2005)*. IASTED/ACTA Press, December 2005.

[105] Richard Harvey, Nick C. Fox, and Martin N. Rossor. *Dementia Handbook*. Medical Pocketbooks. Informa Healthcare, 1999.

[106] D.G. Harwood, W.W. Barker, R.L. Ownby, M. Mullan, and J. Duara. No association between subjective memory complaints and apolipoprotein e genotype in cognitively intact elderly. *International Journal of Geriatric Psychiatry*, 19:1131–1139., 2004.

[107] Masato Hasegawa. ß-amyloid and tau protein. *Psychogeriatrics*, 4(s2):S62–S69, December 2004.

[108] M. Hashimoto, E. Rockenstein, L. Crews, and E. Masliah. Role of protein aggregation in mitochondrial dysfunction and neurodegeneration in alzheimer's and parkinson's diseases. *Neuromolecular Medicine*, 4(1-2):21–36, 2003.

[109] R. Hassan, B. Cohanim, O.L. de Weck, and G. Venter. A comparison of particle swarm optimization and the genetic algorithm. In *1 st AIAA Multidisciplinary Design Optimization Specialist Conference*, Austin , Texas, April 2005.

[110] Z. Hawi, K. Sheehan, A. Lynch, I. Evans, N. Lowe, B. Lawlor, and M. Gill. Late onset alzheimer's disease and apolipoprotein association in the irish population: relative risk and attributable fraction. *Irish Journal of Medical Science*, 172(2):74–76, Apr-Jun 2003.

[111] D. Heckerman, A. Mamdani, and P.M. Wellman. Real-world applications of bayesian networks. *Commun. ACM*, 38(3):24–26, 1995.

[112] D. Heckerman, C. Meek, and G. Cooper. A bayesian approach to causal discovery. Technical report, Microsoft Research, Redmond, Washington, 1997.

[113] P. Hedera. Ethical principles and pitfalls of genetic testing for dementia. *Journal of Geriatric Psychiatry and Neurology*, 14(4):213–221, 2001.

[114] X-C. Heng, Q. Zheng, T. Lei, and S. Li-Ping. Learning bayesian network structures with discrete particle swarm optimization algorithm. In *Foundations of Computational Intelligence Foundations of Computational Intelligence (FOCI 2007)*, pages 47–52, April 2007.

[115] X-C. Heng, Q. Zheng, X-H. Wang, and S. Li-Ping. Research on learning bayesian networks by particle swarm optimization. *Information Technology Journal*, 5(3):540–545, 2006.

[116] M. Henrion, M. Pradhan, B. del Favero, K. Huang, G. Provan, and O. O'Rorke. Why is diagnosis using belief networks insensitive to imprecision in probabilities? In *Uncertainty in Artificial Intelligence*, 1996.

[117] F. Heppner and U. Grenander. *A stochastic nonlinear model for coordinated bird flocks.* AAAS Publications., Washington, DC., 1990.

[118] D.G. Hermans, U.H.H. Htay, and R. McShane. Non-pharmacological interventions for wandering of people with dementia in the domestic setting. *Cochrane Database of Systematic Reviews*, (1), 2007.

[119] B. Herrero, L.M. Laita, E. Roanes-Lozano, V. Maojo, L. de Ledesma6, J. Crespo, and L. Laita. *Artificial Intelligence, Automated Reasoning, and Symbolic Computation*, volume 2385/2002 of *Lecture Notes in Computer Science*, chapter A Symbolic Computation-Based Expert System for Alzheimers Disease Diagnosis, pages 149–163. Springer Berlin / Heidelberg, 2002.

[120] C.D. Herzog, K.A. Nowak, M. Sarter, and J.P. Bruno. Microdialysis without acetylcholinesterase inhibition reveals an age-related attenuation in stimulated cortical acetylcholine release. *Neurobiology of Aging*, 24:861–863, 2003.

[121] H.M. Hodkinson. Evaluation of a mental test score for assessment of mental impairment in the elderly. *Age Ageing*, 1:233–238, 1972.

[122] J.H. Holland. *Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence.* Cambridge, Mass., 1992.

[123] L.R. Hope, A.E. Nicholson, and K.B. Korb. Knowledge engineering tools for probability elicitation. Technical Report tr-2002-111, Monash University, June 2002.

[124] S. Hora and M. Jensen. Expert judgement elicitation. Tech Report 19, The Swedish Radiation Protection Authority (SSI), The Swedish Radiation Protection Authority, S-171 16 Stockholm, Sweden, 2002.

[125] Eric Horvitz and Finn Verner Jensen, editors. *UAI '96: Proceedings of the Twelfth Annual Conference on Uncertainty in Artificial Intelligence, August 1-4, 1996, Reed College, Portland, Oregon, USA*. Morgan Kaufmann, 1996.

[126] William H. Hsu, Haipeng Guo, Benjamin B. Perry, and Julie A. Stilson. A permutation genetic algorithm for variable ordering in learning Bayesian networks from data. In W. B. Langdon, E. Cantú-Paz, K. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M. A. Potter, A. C. Schultz, J. F. Miller, E. Burke, and N. Jonoska, editors, *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference*, pages 383–390, New York, 9-13 July 2002. Morgan Kaufmann Publishers.

[127] M. Huninkm, P.P. Glasziou, J.E. Siegel, J.C. Weeks, J.S. Pliskin, A.S. Elstein, and M.C. Weinstein. *Decision Making in Health and Medicine: Integrating Evidence and Values.* Cambridge University Press, 2001.

[128] S. Iliffe, T. Austin, J. Wilcock, M. Bryans, S. Turner, and M. Downs. Design and implementation of a computer decision support system for the diagnosis and management of dementia syndromes in primary care. *Methods of information in medicine*, 41(2):98–104, 2002.

[129] S. Iliffe, J. Manthorpe, and A. Eden. Sooner or later? Issues in the early diagnosis of dementia in general practice: a qualitative study. *Family Practice*, 20:376–381, 2003.

[130] S. Iliffe, J. Wilcock, T. Austin, K. Walters, G. Rait, S. Turner, M. Bryans, and M. DOWNS. Dementia diagnosis and management in primary care. *Dementia*, 1:11–23, 2002.

[131] Finn V. Jensen. *Introduction to Bayesian Networks*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1996.

[132] F.V. Jensen. *Bayesian Networks and Decision Graphs*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.

[133] K.A. Jobst, L.P.D. Barnetson, and B.J. Shepstone. Accurate prediction of histologically confirmed alzheimer's disease and the differential diagnosis of dementia: The use of nincds-adrda and dsm-iii-r criteria spect x-ray ct and apo e4 in medial temporal lobe dementias. *International Psychogeriatrics*, 10(03):271–302, September 1998.

[134] R. W. Jones. Dementia, postcode prescribing and NICE. *Age and Ageing*, 33(4):331–332, 2004.

[135] R. Kabli, F. Herrmann, and J. McCall. A chain-model genetic algorithm for bayesian network structure learning. In *Proceedings of Genetic and Evolutionary Computation Conference, London, UK.*, pages 1264–1271. ACM, 2007.

[136] D. Kahneman, P. Slovic, and A. Tversky. *Judgment under Uncertainty: Heuristics and Biases*. Cambridge, Univ. Press, 1982.

[137] E. Kapaki, G.P. Paraskevas, I. Zalonis, and C. Zournas. Csf tau protein and ß-amyloid (1–42) in alzheimers disease diagnosis: discrimination from normal ageing and other dementias in the Greek population. *European Journal of Neurology*, 10:119–128, 2003.

[138] J.I. Kazi, P.N. Furness, and M. Nicholson. Diagnosis of early acute renal allograft rejection by evaluation of multiple histological features using a bayesian belief network. *J. Clin. Pathol.*, 50:108, 1998.

[139] J. Kennedy. The behavior of particles. In *EP '98: Proceedings of the 7th International Conference on Evolutionary Programming VII*, pages 581–589, London, UK, 1998. Springer-Verlag.

[140] J. Kennedy. Small worlds and mega-minds: Effects of neighbourhood topology on particle swarm performance. In *In Proceedings of the IEEE Congress on Evolutionary Computation.*, pages 1931–1938. IEEE, 1999.

[141] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of IEEE International Conf. on Neural Networks*, pages 1942–1948, Perth, Australia, Nov 1995. IEEE.

[142] J. Kennedy and R. Eberhart. *Swarm intelligence.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001.

[143] J. Kennedy and R.C Eberhart. A discrete binary version of the particle swarm algorithm. In *"In: Proceedings of the World Multiconference on Systemics, Cybernetics, and Informatics"*, pages 4104–4109. Piscataway, New Jersey., 1997.

[144] J. Kennedy and R. Mendes. Population structure and particle swarm performance. In David B. Fogel, Mohamed A.El-Sharkawi, Xin Yao, Garry Greenwood, Hitoshi Iba, Paul Marrow, and Mark Shackleton, editors, *Proceedings of the IEEE Congress on Evolutionary Computation (CEC2002)*, pages 1671–1676. IEEE, 2002.

[145] J. Kennedy and W.M. Spears. Matching algorithms to problems: an experimental test of the particle swarm and some genetic algorithms on the multimodal problemgenerator. In *IEEE World Congress on Computational Intelligence. Proceedings of the 1998 IEEE International Conference on Evolutionary Computation*, pages 78–83, Bureau of Labor Statistics, Washington, DC, May 1998. IEEE.

[146] C.E. Khan, L. Roberts, K. Wang, D. Jenks, and P. Haddawy. Preliminary investigation of a bayesian network for mamographic diagnosis of breast cancer. In R.M. Gardner, editor, *Proceedings of the American Medical Informatics Association*, pages 208–212. Hanley and Belfus, 1995.

[147] U.B. Kjrulff and A.L. Madsen. Probabilistic networks - an introduction to bayesian networks and influence diagrams, August 2006.

[148] Martin Knapp and Martin Prince and. Dementia UK. Technical Report 820, Alzheimers Society, U.K., 2007.

[149] Kevin B. Korb and Ann E. Nicholson. *Bayesian Artificial Intelligence*. Chapman & Hall/CRC Press UK, 2003.

[150] T. Kočka and R. Castelo. Improved learning of bayesian networks. In D. Koller and J. Breese, editors, *In Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence*, pages 269–276, 2001.

[151] L. Kristiansen, O. Hellzén, and K. Asplund. Swedish assistant nurses' experiences of job satisfaction when caring for persons suffering from dementia and behavioural disturbances. An interview study. *International Journal of Qualitative Studies on Health and Well-Being*, 1(4):245–256, 2006.

[152] Z. Kunda. The case for motivated reasoning. *Psychological Bulletin*, 108(3):480–498, 1990.

[153] J. Kuusisto, K. Koivisto, K. Kervinen, L. Mykkanen, E-L. Helkala, M. Vanhanen, T. Hanninen, L. Pyorala, Y.A. Kesaniemi, P. Riekkinen, and M. Laasko. Association of apolipoprotein e phenotypes with late onset alzheimer's disease: population based study. *BMJ*, 309:636–638, September 1994.

[154] P. Larrañaga, C.M. H. Kuijpers, R.H. Murga, and Y. Yurramendi. Learning bayesian network structures by searching for the best ordering with genetic algorithms. *IEEE Transactions Systems, Man and Cybernetics, Part A.*, 26(4):487–493, 1996.

[155] P. Larrañaga, M. Poza, Y. Yurramendi, R.H. Murga, and C.M.H. Kuijpers. Structure learning of bayesian networks by genetic algorithms: A performance analysis of control parameters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9):912–926, 1996.

[156] Pedro Larrañaga, Basilio Sierra, Miren J. Gallego, Maria J. Michelena, and Juan M. Picaza. Learning bayesisan networks by genetic algorithms: A case study in the prediction of survival in malignant skin melanoma. In *AIME*, pages 261–272, 1997.

[157] Kathryn Blackmond Laskey and Suzanne M. Mahoney. Network engineering for agile belief network models. *IEEE Transactions on Knowledge and Data Engineering*, 12(4):487–498, 2000.

[158] K.B. Laskey and S.M. Mahoney. Network fragments: Representing knowledge for constructing probabilistic models. In *Uncertainty in Artificial Intelligence: Proceedings of the Thirteenth Conference*. Morgan Kaufmann, San Mateo, CA, 1997.

[159] A.H. Lau and T.Y. Leong. Probes: a framework for probability elicitation from experts. In *Proceedings of Artificial Intelligence and Mathematics Annual Symposium.*, 1999.

[160] S. L. Lauritzen. The em algorithm for graphical association models with missing data. *Computational Statistics and Data Analysis.*, 19:191–201, 1995.

[161] S.L. Lauritzen and D.J. Spiegelhalter. *Readings in uncertain reasoning*, chapter Local computations with probabilities on graphical structures and their application to expert systems, pages 415–448. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1990.

[162] S.Y. Lee, K.S. Leung, and M.L. Wong. Improving the efficiency of using evolutionary programming for bayesian network learning. In E.D. Goodman, editor, *2001 Genetic and Evolutionary Computation Conference Late Breaking Papers*, pages 252–259, 2001.

[163] Z. Li-ping, Y. Huan-jun, and H. Shang-xu. Optimal choice of parameters for particle swarm optimisation. *Journal of Zhejiang University, Science*, 6A(6):528–534, 2005.

[164] H. Lindgren. *Decision support in dementia care: developing systems for interactive reasoning.* Phd, Umeå University, Faculty of Science and Technology, Computing Science, Umeå University, Faculty of Science and Technology, Computing Science, 90187 Umeå, May 2007.

[165] K.Y.P. Liu, C.C.H Chan, M.M.L. Chu, T.Y.L. Ng, L.W. Chu, F.S.L. Hui, H.K. Yuen, and A.G. Fisher. Activities of daily living performance in dementia. *Acta Neurologica Scandinavica*, 116:91–95, 2007.

[166] D. Luciani, S. Cavuto, L. Antiga, M. Miniati, S. Monti, M. Pistolesi, and G. Bertolini. Bayes pulmonary embolism assisted diagnosis: a new expert system for clinical use. *Emergency Medicine Journal*, 24:157–164, 2007.

[167] The Lund and Manchester Groups. Clinical and neuropathological criteria for frontotemporal dementia. *Neurology, Neurosurgy and Psychiatry*, 57:416–418, 1994.

[168] R.W. Mahley and Y. Huang. Apolipoprotein (apo) e4 and alzheimer's disease: unique conformational and biophysical properties of apoe4 can modulate neuropathology. *Acta Neurologica Scandinavica*, 114(185):8–14, August 2006.

[169] S.M. Mahoney and K.B. Laskey. Network engineering for complex belief networks. In *[125]*, pages 389–396, 1996.

[170] S. Mani, M.B. Dick, M.J. Pazzani, E.L. Teng, D. Kempler, and I.M. Taussig. Refinement of neuro-psychological tests for dementia screening in a cross cultural population using machine learning. In *Artificial Intelligence in Medicine, AIMDM'99*, volume 1620 of *Lecture Notes in Artificial Intelligence*, pages 326–335. Springer, 1999.

[171] S. Mani, M. Valtorta, and S. McDermott. Building bayesian network models in medicine: The mentor experience. *Applied Intelligence*, 22:93–108, 2005.

[172] H. Mann and D. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *Annals of Mathematical Statistics*, 18(1):50, 60 1947.

[173] B.G. Marcot, J.D. Steventon, G.D. Sutherland, and R.K. McCann. Guidelines for developing and updating bayesian belief networks applied to ecological modeling and conservation. *Canadian Journal of Forest Research*, 36(12):3063–3074, 2006.

[174] I.G. McKeith. Dementia with lewy bodies. *The British Journal of Psychiatry*, 180:114–147, 2002.

[175] G. McKhann, D. Drachman, M. Folstein, R. Katzman, D. Price, and E.M. Stadlan. Clinical diagnosis of alzheimer's disease: report of the nincds-adrda work group under the auspices of department of health and human services task force on alzheimer's disease. *Neurology*, 34(7):939–944, Jul 1984.

[176] S. McLean. Assessing dementia: Part 1: Difficulties, definitions and differential diagnosis. *Australian and New Zealand Journal of Psychiatry*, 18(11):13–16, 1987.

[177] R. Mead, J Paxton, and R. Sojda. Applications of bayesian networks in ecological modeling. In H.Q. Tian, editor, *Environmental Modelling and Simulation*, St. Thomas, US Virgin Islands, November 2006. ACTApress.

[178] R. Mendes. *Population topologies and their influence in particle swarm performance.* Phd, Departamento de Informatica, Escola de Engenharia, Universidade do Minho, 2004.

[179] M. Merkhofer. Quantifying judgemental uncertainty: methodology, experiences, and insights. *OEEE Transactions on Systems, Management Cybernetics*, 17:741–752, 1987.

[180] M.A. Meyer and J.M. Booker. *Eliciting and Analyzing Expert Judgment: A Practical Guide.* ASA-SIAM Series on Statistics and Applied Probability. Society for Industrial Mathematics, 2001.

[181] S. Middleton, J. Barnett, and D. Reeves. What is an integrated care pathway? *Bandolier Journal*, 3(3), 2001. Published by Hayward Medical Communications, a division of Hayward Group plc. Copyright (c) 2001 Hayward Group plc.

[182] A.J. Milne, K. Hamilton-West, and E. Hatzidimitriadou. Gp attitudes to early diagnosis of dementia: Evidence of improvement. *Aging & Mental Health*, 9(5):449–455, 2005.

[183] Melanie Mitchell. *An Introduction to Genetic Algorithms.* MIT Press, Cambridge, MA, USA, 1998.

[184] M.S. Mittelman, S.H. Ferris, G. Steinberg, E. Shulman, J.A. MacKeil, A. Ambinder, and J. Cohen. An intervention that delays institutionalization of alzheimer's disease patients: treatment of spouce-caregivers. *The Gerontologist*, 33:730–740, 1993.

[185] Arvind Mohais, Rui Mendes, Christopher Ward, and Christian Postoff. Neighborhood re-structuring in particle swarm optimization. In Shichao Zhang and Ray Jarvis, editors, *18th Australian Joint Conference on Artificial Intelligence (AI 2005)*, pages 776–785, Sydney, Australia, Dec 2005. Springer-Verlag GmbH.

[186] S. Monti and G Carenini. Dealing with the expert inconsistency in probability elicitation. *IEEE Transactions on Knowledge and Data Engineering*, 12(4):499–508, July/August 2000.

[187] R. Montironi, W.F. Whimster, Y. Collan, P.W. Hamilton, D. Thompson, and P.H. Bartels. How to develop and use a bayesian belief network. *Journal of Clinical Pathology*, 49(3):194–201, March 1996.

[188] M. Morgan and M. Henrion. *Uncertainty : a guide to dealing with uncertainty in quantitative risk and policy analysis.* Cambridge University Press, London, 1990.

[189] C. R. Mouser and S. A. Dunn". Comparing genetic algorithms and particle swarm optimisation for an inverse problem exercise. In Rob May and A. J. Roberts, editors, *Proc. of 12th Computational Techniques and Applications Conference CTAC-2004*, volume 46, pages 89–101, mar 2005.

[190] K. Murphy. An introduction to graphical models. Technical report, University of British Columbia, 2001.

[191] K. Murphy. Learning bayes net structure from sparse data sets. 2001, MIT, 2001.

[192] M. Naidoo and R. Bullock. *An Integrated Care Pathway for Dementia. Best practice for dementia care.* Harcourt, London, 2001.

[193] R.E. Neapolitan. *Learning Bayesian Networks.* Prentice Hall, April 2003.

[194] M. Neil, N. Fenton, and L Nielson. Building large-scale bayesian networks. *The Knowledge Engineering Review*, 15(3):257–284, 2000.

[195] Scottish Intercollegiate Guidelines Network. *Management of patients with dementia: A national clinical guideline.* Number 86 in j. Scottish Intercollegiate Guidelines Network., 2006.

[196] NICE. A guide to NICE. Technical Report N0869, National Institute for Clinical Excellence., MidCity Place, 71 High Holborn, London, WC1V 6NA., March 2005.

[197] nice. Drugs for alzheimer's disease - full guidance. nice nice, National Institute for Clinical Excellence., nice, nice 2007.

[198] Daniel Nikovski. Constructing bayesian networks for medical diagnosis from incomplete and partially correct statistics. *Transactions on Knowledge and Data Engineering*, 12(4):509 – 516, 2000.

[199] Y. Nishiwaki, E. Breeze, L. Smeeth, C.J. Bulpitt, R. Peters, and A.E. Fletcher. Validity of the clock-drawing test as a screening tool for cognitive impairment in the elderly. *American Journal of Epidemiology*, 160(8):794–807, 2004.

[200] J-P. Nordmann and G. Berdeaux. Use of a bayesian network to predict the nighttime intraocular pressure peak from daytime measurements. *Clinical Therapeutics*, 29(8):1751–1760, 2007.

[201] A. Novobilski. The random selection and manipulation of legally encoded bayesian networks in genetic algorithms. In *Proceedings of The 2003 International Conference on Artificial Intelligence, (ICAI) 2003*, pages 438–443, 2003.

[202] A. Novobilski and F. Kamangar. Bayesian learning with selective subsets of populations in genetic programming. In *Proceedings of The Conference on Smart Engineering System Design: Neural Networks, Fuzzy Logic, Evolutionary Programming, Complex Systems and Data Mining (ANNIE) 2002*, 2003.

[203] A. Novobilski, J. Kline, and F. Fesmire. Using a genetic algorithm to identify predictive bayesian models in medical informatics. 2004.

[204] Andre Oboler. The kebn process: A new approach to knowledge engineering with bayesian nets. Technical report, Monash University, 2002.

[205] Royal College of Psychiatrists. Forgetful but not forgotten: assessment and aspects of treatment of people with dementia by a specialist old age psychiatry service. Council Report CR119, Royal College of Psychiatrists, London, April 2005.

[206] A. O'Hagan, C.E. Buck, A. Daneshkhan, J.R. Eiser, P.H. Garthwaite, D.J. Jenkison, J.E. Oakley, and T. Rakow. *Uncertain Judgements: Eliciting Experts' Probabilities.* John Wiley & Sons Ltd, 2006.

[207] Anthony O'Hagan. *Bayesian statistics and its applications*, chapter Research in Elicitation. Anshan Ltd, 2007.

[208] M. Olafsdottir and J. Marcusson. Diagnosis of dementia at the primary care level. *Acta Neurologica Scandinavica Supplementum*, 165:58–62, 1996.

[209] M. Olafsdttir, M. Foldevi, and J. Marcusson. Dementia in primary care: why the low detection rate. *Scandinavian Journal of Primary Health Care*, 19(3):194–198, 2001.

[210] M.G.H. Omran. *Particle Swarm Optimization Methods for Pattern Recognition and Image Processing.* Ph.D, Built Environment and Information Technology, Unversity of Pretoria, November 2004.

[211] World Health Organization. International statistical classification of diseases and related health problems. Technical report, World Health Organization, Geneva, 1992.

[212] L. Oteniya, J. Cowie, and R. Coles. Demnet and pathnet: clinical decision support for dementia diagnosis. Presentation at 1st International Conference at Stirling. Citizenship: Responding to the Challenges of Dementia., April 2007.

[213] Lloyd Oteniya, Julie Cowie, and Richard Coles. A clinical decision support system to aid in the diagnosis of dementia. In *Proceedings of the 22nd HealthCare Computing Conference*, pages 289 – 297, March 2005.

[214] L. Pantoni and D. Inzitari. Hachinski's ischemic score and the diagnosis of vascular dementia: A review. *The Italian Journal of Neurological Sciences*, 14(7):539–546, Oct 1993.

[215] J. Pearce and S. Ferrier. Evaluating the predictive performance of habitat models developed using logistic regression. *Ecological Modelling*, 133(3):225–245, September 2000.

[216] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[217] E. Perry. Acetylcholine and alzheimer's disease. *The British Journal of Psychiatry*, 152:737–740, 1988.

[218] Linda C. van der Gaag Peter J. F. Lucas and Ameen Abu-Hanna. Bayesian networks in biomedicine and health-care. *Artificial Intelligence in Medicine*, 30(3):201–214, March 2004.

[219] A. Petrovski, B. Sudha, and J. Mccall. Optimising cancer chemotherapy using particle swarm optimisation and genetic algorithms. In *In Proc. 8th International Conference on Parallel Problem Solving from Nature*, 2004.

[220] G. Pinner. Truth-telling and the diagnosis of dementia. *The British Journal of Psychiatry*, 176:514–515, 2000.

[221] R. Poli, J. Kennedy, and T. Blackwell. Particle swarm optimization: An overview. *Swarm Intelligence*, 1:33–57, 2007.

[222] C.A. Pollino, O. Woodberry, A. Nicholson, K. Korb, and B.T. Hart. Parameterisation and evaluation of a bayesian network for use in an ecological risk assessment. *Environmental Modelling & Software*, 22(8):1140–1152, August 2007.

[223] K.K. Powlishta, D.D. Von Dras, A. Stanford, B.D Carr, C. Tsering, J.P. Miller, and J.C Morris. The clock drawing test is a poor screen for very mild dementia. *Neurology*, 59:898–903, 2002.

[224] M. Pradhan, M. Henrion, G. Provan, B. del Favero, and K. Huang. The sensitivity of belief networks to imprecise probabilities: an experimental investigation. *Artif. Intell.*, 85(1-2):363–397, 1996.

[225] M. Ramoni and P. Sebastiani. Learning bayesian networks from incomplete databases. In P. Shenoy (Eds.) D. Geiger, editor, *Proc. of the Conf. on Uncertainty in AI*, pages 401–408, 1997.

[226] Silja Renooij. Probability elicitation for belief networks: issues to consider. *The Knowledge Engineering Review*, 16(3):255–269, 2001.

[227] Carsten Riggelsen. Learning parameters of bayesian networks from incomplete data via importance sampling. *International Journal of Approximate Reasoning*, 42(1-2):69–83, 2006.

[228] T. Rivas, J.M. Matias, J. Taboada, and A. Arguelles. Application of bayesian networks to the evaluation of roofing slate quality. *Engineering Geology*, 94(1-2):27–37, 2007.

[229] R.W. Robinson. Counting labelled acyclic digraphs. In F. Harary, editor, *In New Directions in Graph Theory.* New York: Academic Press, 1973.

[230] S.L. Rogers, M.R. Farlow, R.S. Doody, R. Mohs, and L.T. Friedhoff. A 24-week, double-blind, placebo-controlled trial of donepezil in patients with alzheimer's disease. donepezil study group. *Neurology*, 50:136–145, 1998.

[231] G.C. Román, T.K. Tatemichi, T. Erkinjuntti, J.L. Cummings, J.C. Masdeu, J.H. Garcia, L. Amaducci, J.-M. Orgogozo, A. Brun, D.M. Hofman, A. Moody, M.D. O'Brien, T. Yamaguchi, J. Grafman, B.P. Drayer, D.A. Bennett, M. Fisher, J. Ogata, E. Kokmen, F. Bermejo, P.A. Wolf, P.B. Gorelick, K.L. Bick, A.K. Pajeau, M.A. Bell, C. DeCarli, A. Culebras, A.D. Korczyn, J. Bogousslavsky, A. Hartmann, and P. Scheinberg. Vascular dementia: Diagnostic criteria for research studies: Report of the ninds-airen international workshop. *Neurology*, 43:250–260, Feb 1993.

[232] R. Romero, P. Larrañaga, and B. Sierra. Learning bayesian networks in the space of orderings with estimation of distribution algorithms. *IJPRAI*, 18(4):607–625, 2004.

[233] W. W. Royce. Managing the development of large software systems: concepts and techniques. In *Proc. IEEE WESTCON*, 1970.

[234] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, second edition, 2003.

[235] F. Sahin and A. Devasia. *[38]*, chapter Distributed Particle Swarm Optimization for Structural Bayesian Network Learning, pages 505–532.

[236] F. Sahin, M. Yavuz, Z. Arnavut, and O. Uluyol. Fault diagnosis for airplane engines using bayesian networks and distributed particle swarm optimization. *Parallel Computing*, 33(2):124–143, 2007.

[237] D.P. Salmon, R.G. Thomas, M.M. Pay, A. Booth, C.R. Hofstetter, L.J. Thal, and R. Katzman. Alzheimers disease can be accurately diagnosed in very mildly impaired individuals. *Neurology*, 59(7):1022–1028, October 2000.

[238] Alzheimer Scotland. Branches and services of alzheimer scotland, April 2007. `http://www.alzscot.org/downloads/branches\&services.pdf`.

[239] Alzheimer Scotland. Services we provide, July 2007. `http://www.alzscot.org/pages/services.htm`.

[240] National Health Service Quality Improvement Scotland. Integrated care pathways for mental health: bipolar disorder, borderline personality disorder, dementia, depression and schizophrenia. Technical report, National Health Service Quality Improvement Scotland, April 2007.

[241] W.R. Shankle, S. Mani, M.B. Dick, and M.J. Pazzani. Simple models for estimating dementia severity using machine learning. In *In proceedings of MedInfo'98: 9th World Congress on Medical Informatics*, Seoul, Korea, 1998.

[242] W.R. Shankle, S. Mani, M.J. Pazzani, and P. Smyth. *Detecting very early stages of Dementia from normal aging with Machine Learning methods*, volume 1211 of *Lecture Notes in Artificial Intelligence*, pages 73–85. Springer, 1997.

[243] W.R. Shankle, S. Mani, M.J. Pazzani, and P. Smyth. *Intelligent Data Analysis in Medicine and Pharmacology*, chapter Dementia Screening with Machine Learning methods, pages 149–166. Kluwer Academic Publishers, 1997.

[244] Y. Shi. Particle swarm optimization. Feature article, IEEE Neural Networks Society, February 2004.

[245] Y. Shi and R. Eberhart. A modified particle swarm optimizer. In *IEEE World Congress on Computational Intelligence: Proceedings of The 1998 IEEE International Conference on Evolutionary Computation*, pages 69–73, May 1998. doi:10.1109/ICEC.1998.699146 http://ieeexplore.ieee.org/Xplore/login.jsp?url=/iel4/5621/15048/00699146.pdf.

[246] Y. Shi and R Eberhart. Parameter selection in particle swarm optimization. In *In Proceedings of the Seventh Annual Conference on Evolutionary Programming*, pages 591–600, 1998.

[247] Y. Shi and R.C. Eberhart. Fuzzy adaptive particle swarm optimization. In Xin Yao, editor, *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'2001)*, volume 1, pages 101–106, 2001.

[248] S. Simoncic. A bayesian network model of two-car accidents. *Journal of Transportation and Statistics*, 7(2/3):13–26, 2007.

[249] M. Singh and M. Valtorta. An algorithm for the construction of bayesian network structures from data. In *Proceedings of the 9th Annual Conference on Uncertainty in Artificial Intelligence (UAI-93)*, San Francisco,CA, 1993. Morgan Kaufmann.

[250] M. Singh and M. Valtorta. Construction of bayesian network structures from data: A brief survey and an efficient algorithm. *Int. J. Approx. Reasoning*, 12(2):111–131, 1995.

[251] M. Sj¨gren, L. Gustafson, C. Wikkels¨, and A. Wallin. Frontotemporal dementia can be distinguished from alzheimer's disease and subcortical white matter dementia by an anterior-to-posterior rcbf-spet ratio. *Dementia and Geriatric Cognitive Disorders*, 11(5):275–285, September 2000.

[252] J.S Snowden. Neuropsychological evaluation and the diagnosis and differential diagnosis of dementia. *Reviews in Clinical Gerontology*, 9:65–72, 1999.

[253] D. J. Spiegelhalter and S. L. Lauritzen. Sequential updating of conditional probabilities on directed graphical structures. *Networks*, 20:579–605, 1990.

[254] D.J. Spiegelhalter, A.P. Dawid, S.L. Lauritzen, and R.G. Cowell. Bayesian analysis in expery systems. *Statistical Science*, 8:219–283, 1993.

[255] P. Spirtes, C. Glymour, and R. Scheines. An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review*, 9:62–72, 1991.

[256] R. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search.* Springer Verlag, New York, 1993.

[257] SPSS Inc., Chicago IL. *SPSS Base 13.0 for Windows User's Guide.*, 2004.

[258] G. Stoppe, S. Haak, A. Knoblauch, and L. Maeck. Diagnosis of dementia in primary care: A representative survey of family physicians and neuropsychiatrists in germany. *Dementia and Geriatric Cognitive Disorders*, 23(7):207–214, 2007.

[259] K. Sullivan and F. Oconor. Should a diagnosis of alzheimers disease be disclosed? *Aging & Mental Health*, 5(4):340–348, 2001.

[260] P.N. Tariot and H.J. Federoff. Current treatment for alzheimer disease and future prospects. *International Journal of Alzheimer Disease and Associated Disorders*, 17(4):s105–s113, July/September 2003.

[261] P.N. Tariot, P.R. Solomon, J.C. Morris, P. Kershaw, S. Lilienfeld, and C. Ding. A 5-month, randomized, placebo-controlled trial of galantamine in AD. *Neurology*, 54(2269–2276), 2000.

[262] Marc Teyssier and Daphne Koller. Ordering-based search: A simple and effective algorithm for learning bayesian networks. In *UAI*, pages 548–549, 2005.

[263] C.S. Thomas. *From 'Tree' Based Bayesian Networks To Mutual Information Classifiers: Deriving a Singly Connected Network Classifier Using an Information Theory Based Technique.* Phd, Department of Computing Science and Mathematics, University of Stirling., University of Stirling, Stirling, U.K., May 2005.

[264] S. Turner, S. Iliffe, M. Downs, J. Wilcock, M. Bryans, E. Levin, J. Keady, and R. O'Carroll. General practitioners' knowledge, confidence and attitudes in the diagnosis and management of dementia. *Age and Ageing*, 33(5):461–467, 2004.

[265] A. Tversky and D. Kahneman. Judgement under uncertainty: Heuristics and biases. *Science*, 185:1124–1131, 1974.

[266] F. van den Bergh. *An Analysis of Particle Swarm Optimizers.* Ph.d, Faculty of Natural an dAgricultural Science, University of Pretoria., University of Pretoria, November 2001.

[267] W.M. van der Flier and P. Scheltens. Use of laboratory and imaging investigations in dementia. *Journal of Neurology, Neurosurgery and Psychiatry*, 76:45–52, 2005.

[268] L.C. van der Gaag, S. Renooij, C.L.M. Witteman, B.M.P. Aleman, and B.G. Taal. How to elicit many probabilities. In Laskey and Eds. Prade, editors, *In UAI99 — Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 647–654, 1999.

[269] L.C. van der Gaag, S. Renooij, C.L.M. Witteman, B.M.P. Aleman, and B.G. Taal. Probabilities for a probabilistic network: a case study in oesophageal cancer. *Artificial Intelligence in Medicine*, 25:123–148, 2002.

[270] M. Verduijn, N. Peek, P.M.J. Rosseel, E. de Jonge, and B.A.J.M. de Mol. Prognostic bayesian networks: I: Rationale, learning procedure, and clinical use. *Journal of Biomedical Informatics*, 40(6):609–618, 2007.

[271] A.C. Vink, J.S. Birks, M.S. Bruinsma, and R.J.P.M. Scholten. Music therapy for people with dementia (review). *Cochrane Database of Systematic Reviews*, (4), 2003.

[272] D. von Winterfeldt and W. Edwards. *Decision analysis and Behavioural Research.* Cambridge University Press, 1986.

[273] G. Waldemar, M. Dubois, B. Emre, J. Georges, I. G. McKeith, M. Rossor, Scheltens P., P. Tariska, and B. Winblad. Recommendations for the diagnosis and management of alzheimers disease and other disorders associated with dementia: Efns guideline. *European Journal of Neurology*, 14(1):e1–e26, January 2007.

[274] L. Walls and J. Quigley. Building prior distributions to support bayesian reliability growth modelling using expert judgement. *Reliability Engineering and System Safety*, 74:117–128, November 2001.

[275] T.S. Wallsten and D.V. Budescu. Encoding subjective probabilities: A psychological and psychometric review. *Management Science*, 29(2):151–173, 1983.

[276] S. Warshall. A theorem on boolean matrices. *J. ACM*, 9(1):11–12, 1962.

[277] Wikipedia. Positive predictive value. World Wide Web, Accessed 19 April 2008.

[278] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, Dec 1945.

[279] M.L. Wong, W. Lam, and K.S. Leung. Using evolutionary programming and minimum description length principle for data mining of bayesian networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 21(2):174–178, February 1999.

[280] O. Woodberry, A. Nicholson, K.B. Korb, and C. Pollino. Parameterizing bayesian networks. In *AI 2004: Advances in Artificial Intelligence. 17th Australian Joint Conference on Artificial Intelligence. Proc. of Australian Joint Conference on Artificial Intelligence*, volume 3339/2004 of *Lecture Notes in Computer Science*, pages 1101–1107, Cairns, Australia., December 2004. Springer Berlin / Heidelberg.

[281] C.H.J. Woodford and J. George. Cognitive assessment in the elderly: a review of clinical methods. *QJM*, 100(8):469–484, 2007.

[282] B. Woods, A. Spector, C. Jones, M. Orrell, and S. Davies. Reminiscence therapy for dementia (review). *Cochrane Database of Systematic Reviews*, (2), 2005.

[283] S. Wright. Correlation and causation. *Journal of Agricultural Research*, 20(7):557–585, 1921.

[284] R. Yale. Developing support groups for individuals with early-stage alzheimer's disease. Baltimore, USA: Health Professions Press., 1995.

[285] M.H. Zweig and G. Campbell. Receiver operator characteristic (roc) plots: A fundamental evaluation tool in clinical medicine. *Clinical Chemistry*, 39:561–577, 1993.