MULTIPLY SECTIONED BAYESIAN BELIEF NETWORKS FOR LARGE KNOWLEDGE-BASED SYSTEMS: AN APPLICATION TO NEUROMUSCULAR DIAGNOSIS

By

Yang Xiang

B.A.Sc., Beijing Institute of Aeronautics and Astronautics, 1983

M.A.Sc., Beijing Institute of Aeronautics and Astronautics, 1985

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

THE FACULTY OF GRADUATE STUDIES

ELECTRICAL ENGINEERING

We accept this thesis as conforming

to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

1991

Electrical Engineering
The University of British Columbia
2075 Wesbrook Place
Vancouver, Canada
V6T 1W5

Date: Mar. 11, 1992

Abstract

In medical diagnosis a proper uncertainty calculus is crucial in knowledge representation. Finite calculus is close to human language and should facilitate knowledge acquisition. An investigation into the feasibility of finite totally ordered probability models has been conducted. It shows that a finite model is of limited usage, which highlights the importance of infinite totally ordered models including probability theory.

Representing the qualitative domain structure is another important issue. Bayesian networks, combining graphical representation of domain dependency and probability theory, provide a concise representation and a consistent inference formalism. An expert system QUALICON for quality control in electromyography has been implemented as a pilot study of Bayesian nets. The performance is comparable to that from human professionals.

Extending the research into a large system PAINULIM in neuromuscular diagnosis shows that the computation using homogeneous net representation is unnecessarily complex. At any one time a user's attention is directed to only part of a large net, i.e., there is 'localization' of queries and evidence. The homogeneous net is inefficient since the overall net has to be updated each time. Multiply Sectioned Bayesian Networks (MSBNs) have been developed to exploit localization. Reasonable constraints are derived such that a localization preserving partition of a domain and its representation by a set of subnets are possible. Reasoning takes place at only one of them due to localization. Marginal probabilities obtained are identical to those obtained when the entire net is globally consistent. When the user's attention shifts, a new subnet is swapped in and previously acquired evidence absorbed. Thus, with β subnets, the complexity is reduced approximately to $1/\beta$.

Reducing the complexity with MSBN, the knowledge acquisition of PAINULIM has been conducted using normal hospital computers. This results in efficient cooperation with medical staff. PAINULIM was thus constructed in less than one year. An evaluation shows very good performance.

Coding probability distribution of Bayesian nets in causal direction has several advantages. Initially the distribution is elicited from the expert in terms of probabilities of a symptom given causing diseases. Since disease-to-symptom is not the direction of daily practice, the elicited probabilities may be inaccurate. An algorithm has been derived for sequentially updating probabilities in Bayesian nets, making use of the expert's symptom-to-disease probabilities. Simulation shows better performance than Spiegelhalter's $\{0,1\}$ distribution learning.

Table of Contents

List of Tables

List of Figures

A Guide for the Reader

This thesis has been written for readers from various backgrounds including artificial intelligence, medical informatics, and biomedical engineering.

Chapter 1 contains the background on Bayesian belief networks and related uncertainty management formalisms. Bayesian belief networks combine probability theory and graphical representation of domain models. Appendix A contains an introduction about concepts from graph theory which are relevant to this thesis. Readers unfamiliar with graph theory should read this appendix before reading the main body of the thesis.

Chapter 2 contains results of an investigation into the feasibility of using finite totally ordered probability models for uncertainty management in expert systems. Readers who are mainly interested in positive results and applicable techniques may skip this chapter.

Chapter 3 describes the construction and evaluation of QUALICON, an expert system coupling digital signal processing and Bayesian networks for technical quality control in nerve conduction studies. Researchers in medical informatics and biomedical engineering who are interested in learning about the practical application of Bayesian networks and coupled expert systems will find this chapter useful.

Chapter 4 contains the theory for Multiply Sectioned Bayesian Networks (MSBNs) and junction forests. Its implementation in WEBWEAVR shell and application in PAINULIM are described in Chapter 5.

Chapter 5 describes the construction and evaluation of PAINULIM, an expert neuromuscular diagnostic system for patients presenting a painful or impaired upper limb. Researchers in medical informatics and biomedical engineering who are interested in the practical application of the MSBN technique will find this chapter particularly relevant.

Chapter 6 contains the algorithm of learning for sequential updating conditional probabilities in Bayesian networks using the expert's subjective posterior probabilities.

Acknowledgement

At University of British Columbia, my deepest thanks go to David Poole, Michael Beddoes and Andrew Eisen. David Poole introduced me to the concepts of probabilistic reasoning and Bayesian networks. His critical eye on my work has been extremely beneficial. Michael Beddoes directed me into the field of application of artificial intelligence (AI) in neuromuscular diagnosis, and has been my advisor both in and out of academics. His guidance has been critical to this interdisciplinary research. Andrew Eisen has guided me on making decisions in choosing appropriate subdomains in neuromuscular diagnosis for AI applications, has given me necessary support to conduct the research towards the two expert systems to which he also act as one of the domain experts.

At Vancouver General Hospital, special thanks are due to Bhanu Pant, and Maureen MacNeil, the domain experts for PAINULIM and QUALICON. The technique of multiply sectioned Bayesian networks and junction forests grew out of the discussions with Andrew Eisen and Bhanu Pant. Bhanu Pant also devoted months of painstaking work and much overtime to the construction and evaluation of PAINULIM. Without his enthusiastic cooperation, it would not be possible for PAINULIM to reach its clinical significance in less than a year. Maureen MacNeil devoted months of work for the construction of QUALICON besides her hospital duty.

I thank Romas Aleliunas for helping me to gain the understanding of his theory of probabilistic logic. I also thank Finn Jensen for providing several reprints of his papers; Steen Andreassen for providing publications regarding the progress in MUNIN; and David Heckerman for providing his Ph.D. thesis on similarity networks and his papers on PATHFINDER. Lianwen Zhang suggested the relation between the d-sepset and the

graph separator. Nick Newell provided useful suggestions for programming WEBWEAVR shell.

In addition I thank Sandy Trueman, and Andrew Travlos for participating in the assessment of QUALICON and Tish Mallinson, Karen Burns, and Khadija Amlani for help in nerve conduction potential acquisition.

The National Science and Engineering Research Council, under Grant A3290, 583200, and University of British Columbia, under University Graduate Fellowship, provided the financial support that made this thesis possible.

Finally, I owe a great debt to my mother and father for a lifetime of love, caring and support, and especially to my wife Jingyu Zhu for sheltering me from the travails of the real world and surrounding me with so much love and support.

Chapter 1

INTRODUCTION

In this chapter, the status of computers in medicine is reviewed in Section 1.1: the review covers the status of expert systems in neuromuscular diagnosis. Section 1.2 reviews rule-based expert systems and uncertainty management formalisms other than Bayesian networks. Section 1.3 reviews Bayesian networks as a natural and concise representation for uncertain knowledge.

1.1 Neuromuscular Diagnosis and Expert Systems

The thesis research addresses practical issues in building medical expert systems. The medical area is neuromuscular diagnosis.

The computer is now assisting doctors and technicians in the neuromuscular diagnosis by performing the following functions: data acquisition/analysis; database management; and documentation. Computers have failed to assist doctors directly in their diagnostic inference and clinical decision making [Desmedt 89].

In general medicine, the same failure was true until the mid 70's. The failure was partly due to the limitations of conventional techniques for medical decision making (the clinical algorithm or flow chart, and the matching of cases to large data bases of previous cases [Szolovits 82]). The failure was also partly due to the unawareness of proper ways to apply normative probability and decision theories.

On the other hand, as Szolovits [1982] put it: "Modern medicine has become technically complex, the standards set for it are very high, conceptual and practical advances are rapid, yet the cognitive capabilities of physicians are pretty much fixed. As more and more data become available to the practicing doctor, as more becomes known about the processes of disease and possible interventions available to alter them, practitioners are called on to know more, to reason better, and to achieve better outcomes for their patients." In response to the need for more sophisticated diagnostic aids, medical expert systems appeared in the field of AI in medicine.

Expert systems are computer programs capable of making judgments or giving assistance in a complex area. The tasks they perform are ordinarily performed only by humans. They are commonly separated into 2 major components: *knowledge base* which includes assumptions about the particular domain, and the *inference engine* which controls how the knowledge base is to be used in solving a particular problem. Since the mid 70's, researchers in AI have built expert systems in many medicine areas to assist the doctors in diagnosis or therapy (e.g., MYCIN for the diagnosis and treatment of bacterial infections [Buchanan and Shortliffe 84], and INTERNIST for the diagnosis in internal medicine [Pople 82]). New expert system technologies are still evolving as limitations to existing technologies are recognized. Limited successes in different aspects of performance have been achieved.

Back to the area of neuromuscular diagnosis, several (prototype) expert systems have appeared since the mid 80's: LOCALIZE [First et al. 82] for localization of peripheral nerve lesions; MYOSYS [Vila et al. 85] for diagnosing mono- and polyneuropathies; MYOLOG [Gallardo et al. 87] for diagnosing plexus and root lesions; Blinowska and Verroust's system [1987] for diagnosing carpal tunnel syndrome; ELECTRODIAGNOSTIC ASSISTANT [Jamieson 90] for diagnosing entrapment neuropathies, plexopathies, and radiculopathies; KANDID [Fuglsang-Frederiksen and Jeppesen 89] and MUNIN [Andreassen et al. 89] aiming at diagnosing the complete range of neuromuscular disorders. Satisfaction in system testing with constructed cases have been reported, while only one of them (ELECTRODIAGNOSTIC ASSISTANT) reported clinical evaluation using 15 cases of 78% agreement rate with electromyographers. Most of the systems are rule-based. The limitation of rule-based systems is reviewed in Section 1.2. MUNIN uses Bayesian networks for representation of uncertain knowledge; and is still under development. Just as the expert system techniques are evolving, their application to neuromuscular diagnosis is still one for ongoing research.

1.2 Representation Formalisms Other Than Bayesian Networks

Construction of an expert system faces an immediate question: What is a proper knowledge representation and inference formalism? The answer depends largely on the characteristics of the task to be performed by the target system.

Medical diagnosis is characterized with probable reasoning, or more formally, reasoning under uncertainty. The uncertainty comes from the incomplete knowledge about biological process within the human body, from the incomplete observation and monitoring of patient conditions, and from dynamically changing relations between many factors entering diagnostic process. A proper knowledge representation for reasoning under uncertainty is required which includes the representation of qualitative domain structure and an uncertainty calculus.

Perhaps the most widely used structure in expert systems is a set of rule in rule-based systems. This section reviews rule-based systems, their advantages and limitations. There has been a set of uncertainty calculuses used in expert systems. This section also reviews those calculuses notable in AI which associate single real numbers with uncertainty. An investigation of the feasibility of using a finite number of symbols to denote degrees of uncertainty will be presented in Chapter 2. The reason for reviewing those calculuses together with rule-based systems is because some of them (i.e., MYCIN certainty factor and odds likelihood ratios) can only be applied in conjunction with rule-based systems.

Bayesian networks has been chosen as the knowledge representation in this thesis research. The networks combine graphic representation of domain dependence and probability theory. This combination overcomes many limitations of rule-based systems. Bayesian network techniques will be reviewed in section 1.3.

There has been controversy in the artificial intelligence field about the adequacy of using probability for representation of beliefs [Cheeseman 88a, Cheeseman 88b]. I will not attempt to enter into this controversy. Rather, my review discusses why Bayesian networks seem appropriate for building medical expert systems and thus are adopted in this thesis research.

1.2.1 Rule-Based Systems

An expert system which deals with probable reasoning will have a knowledge base which consists of a *qualitative* structure of the domain model and a *quantitative* uncertainty calculus. The same qualitative structure can usually accommodate different uncertainty calculuses. Perhaps the most widely used structure in expert systems is a set of rules in rule-based systems. All the methods for representing uncertainty reviewed in this section can be incorporated with rule-based systems. Rules in such systems consist of antecedent-conclusion pairs, "if X, then Y" with the reading "if antecedent X is true, then conclusion Y is true". A rule encodes a piece of knowledge.

An *inference network* is a directed acyclic graph (DAG) in which each node is labeled with a *proposition.* The set of rules in a rule-based system, which does not contain cyclic rules, can be represented by an inference network in which each arc corresponds to a rule with direction from antecedent to conclusion. The 'roots' of an inference network are labeled with variables about which the user is expected to supply information. The 'leaves' are variables of interest. Figure 1.1 is a inference network representing 4 rules: "if B, then A", "if C, then A", "if D, then A" and "if E, then D". B, C, E are roots and A is a leaf. Below rule-based systems are considered in terms of inference networks.

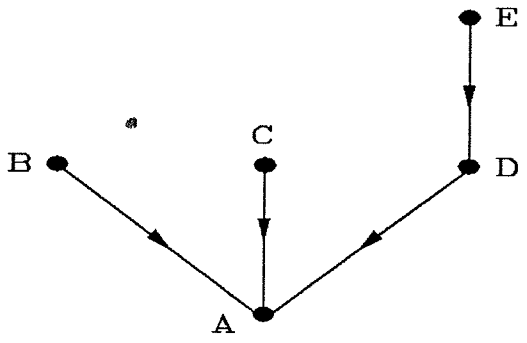


Figure 1.1: An inference network

1.2.2 The Method of Odds Likelihood Ratios

This subsection reviews the representation of uncertainty in rule-based expert systems by odds and likelihood ratios, and the propagation of evidence under this representation. Such an approach is used in the expert system PROSPECTOR [Duda et al. 76] which helps geologists evaluate the mineral potential of exploration sites. The formulation of Neapolitan [1990] is followed.

With the method of odds likelihood ratios, uncertainty to the rules are represented by conditional probabilities in the form $p(X|Y)$ where X is the antecedent and Y is the conclusion[1]. Suppose for each rule "if X, then Y", the conditional probabilities $p(X|Y)$ and $p(X|\overline{Y})$ are determined; and for each node Y in the corresponding inference network, except for the roots, the prior probability $p(Y)$ is determined. Then the *prior odds* on

[1] In Bayesian networks the order of X and Y will be reversed.

Y is defined as $O(Y) = p(Y)/p(\overline{Y})$. The *posterior odds* on Y upon learning that X is true is defined as $O(Y|X) = p(Y|X)/p(\overline{Y}|X)$. The *likelihood ratio* of X is defined as $L(X|Y) = p(X|Y)/p(X|\overline{Y})$. Notice that $p(Y) = O(Y)/(1 + O(Y))$.

With the above definition, the evidence propagation can be illustrated with the example in Figure 1.2. As in the figure, likelihood ratios are stored at each arc, and prior probabilities are stored at each node except for roots. If one has evidence that E is true, then $O(D|E) = L(E|D)O(D)$. To propagate the evidence to A in a simple way, it is assumed that A and E are conditionally independent given D. Then $p(A|E) = p(A|D)p(D|E) + p(A|\overline{D})p(\overline{D}|E)$. Note the assumption is not valid if multiple paths exist from E to A. If in another case, one has evidence that B and C are true, one would like to know $O(A|BC)$ where the concatenation implies 'AND'. Assume that B and C are conditionally independent given A, then $O(A|BC) = L(B|A)L(C|A)O(A)$. Again, the assumption is not valid'in a multiply connected network.

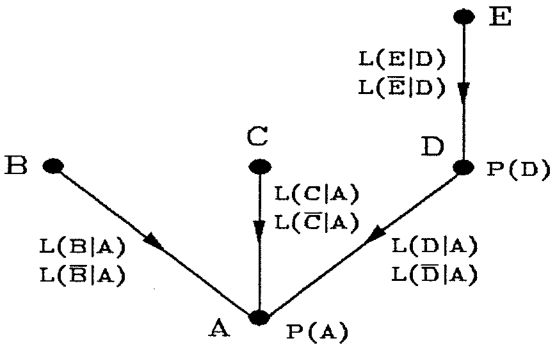


Figure 1.2: An inference net using the method of odds likelihood ratios

Several limitations of the method of odds likelihood ratios have been reviewed by Neapolitan [1990]:

1. The method is restricted to singly connected networks. Many application domains

can not be represented.

2. An inference network with odds and likelihood ratios associated can only be used for reasoning in the designed direction but not in the opposite direction.

3. For all the nodes except roots, prior probabilities are to be specified. Since priors are population specific, they are difficult to ascertain. A medical expert system constructed using data from one clinic might not be deployed in another clinic because the population there was different.

1.2.3 MYCIN certainty factor

The most widely used uncertainty calculus in rule-based systems is called *certainty factor* and it was originally used in MYCIN [Buchanan and Shortliffe 84]. General probabilistic reasoning based on probability theory required the specification of exponentially large number of data which lead to intractable computation associated with belief updating [Szolovits and Pauker 78, Buchanan and Shortliffe 84]. For example, a full joint probability distribution over a domain with α binary variables requires specification of $2^{\alpha} - 1$ parameters. The parameters had to be updated when the new evidence became available. The primary goal in creating the MYCIN certainty factor was to provide a method to avoid this difficulty.

In a rule-based system using certainty factors for reasoning under uncertainty, a rule "if X, then Y" is associated with a *certain factor* $CF(Y, X) \in [-1, 1]$ to represent the *change* in belief about Y given the verity of X. $CF(Y, X) > 0$ corresponds to *increase* in belief, $CF(Y, X) < 0$ corresponds to *decrease* in belief, and $CF(Y, X) = 0$ corresponds to *no change* in belief. Representing a rule-based system by an inference network, the certainty factors are stored at arcs as in Figure 1.3.

The MYCIN certainty factor model provides a set of rules for propagating uncertainty

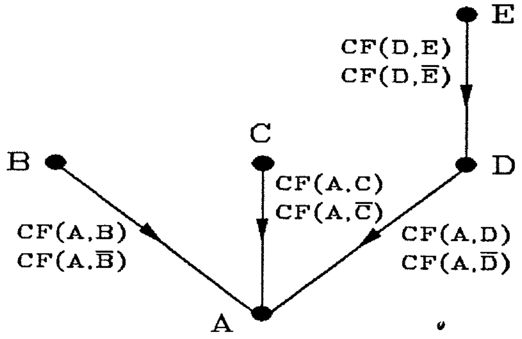

E
CF(D,E)
CF(D,$\overline{\text{E}}$)
C
B
D
CF(A,C)
CF(A,$\overline{\text{C}}$)
CF(A,B)
CF(A,$\overline{\text{B}}$)
CF(A,D)
CF(A,$\overline{\text{D}}$)
A

Figure 1.3: An inference net using the MYCIN certainty factors

through an inference network. Figure 1.3 illustrates sequential and parallel combination in an inference network. If one has evidence that E is true, then the certainty factors $CF(D,E)$ and $CF(A,D)$ can be combined to give the certainty factor $CF(A,E)$ by *sequential combination*

$$CF(A,E) = \begin{cases} CF(D,E)CF(A,D) & CF(D,E) \geq 0 \\ -CF(D,E)CF(A,\overline{D}) & CF(D,E) < 0 \end{cases}$$

If instead one has evidence that B and C are true, then the certainty factor $CF(A,BC)$ is given by *parallel combination*

$$CF(A,BC) = \begin{cases} CF(A,B) + CF(A,C)(1 - |CF(A,B)|) & CF(A,B)CF(A,C) \geq 0 \\ \frac{CF(A,B)+CF(A,C)}{1-\min(|CF(A,B)|,|CF(A,C)|)} & CF(A,B)CF(A,C) < 0 \end{cases}$$

The calculus of the MYCIN certainty factor has many desirable properties which would be expected from an uncertain reasoning technique. However, in the original work of the MYCIN certainty factor, there was no operational definition of a certainty factor [Heckerman 86, Neapolitan 90]. That is, the definition of a certainty factor does not prescribe a method for determining a certainty factor. Without an operational definition there is no way of knowing whether 2 experts mean different things when they assign different certainty factors to the same rule. Heckerman [1986] gives probabilistic interpretations of the MYCIN certainty factor. He shows that these interpretations are monotonic transformations of the likelihood ratio reviewed in section 1.2.2. Therefore the MYCIN certainty factor makes the same assumptions as the method of odds likelihood ratios and shares the same limitations [Neapolitan 90].

1.2.4 Dempster-Shafer Theory

Unlike probability theory, which assigns probabilities to every member of a set Ψ of mutually exclusive and exhaustive alternatives and requires that the probabilities sum to unity, the Dempster-Shafer (D-S) theory [Shafer 76] assigns *basic probability assignments* (bpa) to every subset of Ψ (member of 2^{Ψ}) and requires that the bpas sum to unity. For a proposition C, the bpa assigned to $\{C\}$ and the bpa assigned to $\{\overline{C}\}$ do not have to sum to unity. Thus D-S theory allows some of the probability to be unassigned, that is, it accepts an incomplete probabilistic model when some parameters (either prior probabilities or conditional probabilities) are missing [Pearl 88]. In this sense, D-S theory is an extension of probability theory [Neapolitan 90]. Medical diagnosis involves repetition and it is possible to obtain a complete probability model. The model may not be accurate and further refinement may be required (Chapter 6).

Unlike probability theory, which calculates the conditional probability that Y is true given evidence X, D-S approach calculates the probability that the proposition Y is *provable* given the evidence X and given that X is consistent. Pearl [1988] argues that D-S theory offers a better representation when the task is one of synthesis (e.g., the class scheduling problem) where external constraints are imposed and the concern centers on issues of possibility and necessity; he also argues that probability theory is more suitable for diagnosis.

A pragmatic advantage of probability theory over D-S theory is that it is well founded and well known. An expert system based on a well known theory will certainly benefit in the knowledge acquisition and in communicating with users.

1.2.5 Fuzzy Sets

Fuzzy set theory [Zadeh 65] deals with propositions which have *vague* meaning such as “the symptom is severe” or “the median nerve conduction velocity is normal”. It associates a real number from $[0, 1]$ with the membership of a particular element in a set. For example, if a patient has a median nerve conduction velocity of 58 m/sec, the result has its membership 1 in the set of ‘normal’ results. If the velocity is 44 m/sec, the result has membership 0 in the ‘normal’ set. When the velocity value is 50 m/sec, the result has a partial membership, say, 0.7 in the ‘normal’ set.

Some researchers [Pearl 88, Neapolitan 90] view fuzzy set theory as addressing a fundamentally different class of problems than those addressed by probability theory, certainty factor and the Dempster-Shafer theory. All but fuzzy set theory deal with *well defined* propositions which are definitely either true or false. One is simply uncertain as to the outcome. Cheeseman [1986,1988a,1988b] makes a strong claim that fuzzy sets are unnecessary (for representing and reasoning about uncertainty, including vagueness), probability theory is all that is required. He shows how probability theory can solve the problems that the fuzzy approaches claim probability cannot solve.

1.2.6 Limitations of Rule-Based Systems

Section 1.2.2 and 1.2.3 reviewed 2 methods for reasoning under uncertainty in rule-based systems. This subsection concentrates on the questions: Are rule-based systems suitable for probable reasoning? In which situations are they appropriate knowledge representation?

The basic principle of rule-based systems is *modularity.* A rule “if X, then Y” has the procedural interpretation: "If X is in the knowledge base, then regardless of what other things the knowledge base contains and regardless of how X was derived, add Y to the knowledge base" [Pearl 88]. The attractiveness of rule-based systems is high efficiency in inference. This stems from modularity. However, the principle is valid only if the domain is *certain*. That is, rule-based systems are appropriate for applications involving only *categorical* (as defined by Szolovits and Pauker [1978]) reasoning [Neapolitan 90].

When reasoning under uncertainty, the rule "if X, then Y with uncertainty w" reads as: "If the certainty of X undergoes a change δ_X, then regardless of what other things the knowledge base contains and regardless of how δ_X was triggered, modify the current certainty of Y by some amount δ_Y, which may depend on w, on δ_X, and on the current certainty of Y" [Pearl 88]. Pearl discusses three major problems resulted from the principle of modularity in probable reasoning.

- Improper handling of bidirectional inferences. This is a restatement of the third limitation in section 1.2.2. If X implies Y, then verification of Y makes X more credible. Thus the rule "X implies Y" should be used in two different directions: *deductive* - from antecedent to conclusion, and *abductive* - from conclusion to antecedent. However, rule-based systems require that the abductive inference to be stated explicitly by another rule and, even worse, that the original rule be removed. Otherwise, a cycle would be created.

- Difficulties in retracting conclusions. Two rules "If the ground is wet then it rained" and "If the sprinkler was on then the ground is wet" may be contained in a system. Suppose "the ground is wet" is found. The system will conclude "it rained" by first rule. But if later "the sprinkler was on" is found, the certainty "it rained" should be decreased significantly. However, this can not be implemented naturally in a rule-based system.

- Improper treatment of correlated sources of evidence. Recall that both the odds likelihood ratio method and the certainty factor method assume singly connected inference networks. Rule-based systems respond only to the degrees of certainty to antecedents and not to the origins of these degrees. As a result the same conclusions will be made whether the degree of certainty originates from identical or independent sources of information.

Heckerman and Horvitz [1987] analyze the inexpressiveness of rule-based systems.

- Only binary variables (proposition variables) allowed. When multi-valued random variables are needed, they have to be broken down into several binary ones and the naturalness of representation is lost.

- Multiple causes. When multiple causes exist, they are represented by an exponential number of binary variables, and the representation can not accommodate different evidence patterns.

With these problems in mind, one must conclude that generally rule-based systems are not appropriate for applications that require probable reasoning.

1.3 Representing Probable Reasoning In Bayesian Networks

This section briefly introduces Bayesian networks as a natural, concise knowledge representation method and a consistent inference formalism for building expert systems. The substantial advances of Bayesian network technologies in recent years are reviewed.

1.3.1 Basic Concepts of Probability Theory

Two major interpretations of probability exists: *objective* or *frequentist* interpretation, and *subjective* or *Bayesian* interpretation [Savage 61, Shafer 90, Neapolitan 90]. The former defines probability of an event E as the *limiting frequency* of E in repeated experiments. The latter interprets probability as *degree of belief* held by a person. In building a medical expert system based on probability theory, one usually has to take a medical expert as the major source of the probabilistic information. Thus Bayesian interpretation is naturally adopted.

Although philosophically the 2 interpretations represent 2 different camps, practically, they are not significantly different as far as physicians' belief in uncertain relations in medicine is concerned. The medical literature substantiates the fact that many physicians believe that probabilities, according to the frequentist's definition, exist, and that the likelihoods which they assign, are estimates of these probabilities based on their experiences with frequencies [Neapolitan 90]. My personal experience with neurologists in building PAINULIM expert system is also in agreement with this viewpoint. The Bayesian formulation of probability will be adopted in this thesis. In Chapter 6, an interaction between the 2 interpretations is utilized to improve the accuracy of knowledge representation.

The framework of Bayesian probability theory consists of a set of axioms which describes constraints among a collection of probabilities provided by a given person [Pearl 88, Heckerman 90b].

$$0 \leq p(A|C) \leq 1$$

$$p(C|C) = 1$$

If A and B are mutually exclusive,

$$\text{then } p(A \text{ or } B|C) = p(A|C) + p(B|C) \quad \text{(sum rule)}$$

$$p(AB|C) = p(A|BC)p(B|C) \qquad \text{(product rule)}$$

where A, B are arbitrary events and C represents the background knowledge of the person who provides the probability. The probability $p(A|C)$ represents a person's belief in A given his background knowledge C. The following rules, to be used in the thesis, can be proved from the above axioms.

$p(\overline{A}|C) = 1 - p(A|C)$ (negation rule)

If $B_1, \ldots, B_i$ are mutually exclusive and exhaustive then

$p(AB_1|C) + \ldots + p(AB_i|C) = p(A|C)$ (marginalization)

$p(A|BC) = p(B|AC)p(A|C)/p(B|C)$ (Bayes theorem)

1.3.2 Inference Patterns Embedded In Probability Theory

As a well founded theory, probability theory embeds many intuitive inference patterns, which renders it a suitable uncertainty calculus for probable reasoning. The following briefly discusses some patterns used in medical diagnosis.

Bidirectional reasoning Probability allows both *predictive* and *abductive (diagnostic)* reasoning. Carpal tunnel syndrome (cts) is a common cause of pain in forearm. If one has a probability p(painful forearm|cts) = 0.75 and also knows John has cts, then one would *predict* with high confidence John would suffer from pain in forearm. On the other hand, if one also has p(cts), p(painful forearm), and knows Mary does suffer from pain in forearm, then one can compute p(cts|painful forearm) by applying Bayes theorem. The *diagnosis* will be based on p(cts|painful forearm).

Context sensitivity Both cts and thoracic outlet syndrome (tos) cause pain in forearm. The pain in the forearm is equally likely from either cause. Cts has much higher incidence than tos, say p(cts) = 0.25 and p(tos) = 0.02 in a clinic. A patient with painful forearm is 12 times more likely to have cts than tos. But if later through other means the patient is found to have tos, then the painful forearm can be explained by tos. Cts becomes much less likely. The probability for this case will have p(cts|painful forearm) = HIGH, and p(cts|painful forearm & tos) = LOW. That is, the original belief in cts is retracted in the new context.

Explaining away In the above example, the confirmation of tos makes the alternative explanation tos for painful forearm less credible. That is, tos explains painful forearm away from cts.

Dynamic dependence Cts and tos are independent diseases, i.e., knowing only a patient having or not having cts tells one nothing about whether he has tos or not. However, knowing a patient has a painful forearm will render the 2 diseases related in the diagnostic process. Further evidence supporting one of them will decrease the likelihood of another.

Readers are referred to Pearl [1988] for an elaborate discussion of the relationship between probability theory and probable reasoning.

1.3.3 Bayesian Networks

Bayesian networks are known in the literature as *Bayesian belief networks, belief networks, causal networks, influence diagrams*, etc. They have a history in decision analysis [Miller et al. 76, Howard and Matheson 84]. They have been actively studied for probable reasoning in AI for about a decade. Formally a Bayesian network [Pearl 88] is a triplet (N, E, P).

- The *domain* N is a set of nodes each of which is labeled with a random variable characterized by a set of *mutually exclusive* and *exhaustive* outcomes. 'Node' and 'variable' are used interchangeably in the context of Bayesian nets.
- E is a set of arcs such that (N, E) is a DAG. The arcs signify the existence of direct causal influences between the linked variables. The basic dependence assumption

embedded in Bayesian nets is that a variable is independent of its non-descendants given its parents.

Uppercase letters (possibly subscripted) in the beginning of the alphabet are used to denote variables, corresponding script letters are used to denote their sample spaces, and corresponding lowercase letters with subscripts are used to denote their outcomes. For example, in binary case, a variable A has its sample space $\mathcal{A} = \{a_1, a_2\}$, and H_i has its sample space $\mathcal{H}_i = \{h_{i1}, h_{i2}\}$. Uppercase letters towards the end of the alphabet are used to denote a set of variables. If $X \subseteq N$ is a set of variables, the *space* $\Psi(X)$ of X is the cross product of sample spaces of the variables $\Psi(X) = \times_{A \in X} \mathcal{A}$. π_i is used to denote the set of parent variables of $A_i \in N$.

- P is a joint probability distribution quantifying the strengths of the causal influences signified by the arcs. P is specified by, for each $A_i \in N$, the distribution of the random variable labeled at A_i conditioned by the values of A_i's parents π_i in the form of a conditional probability table $p(A_i|\pi_i)$. $p(A_i|\pi_i)$ is a normalized function mapping $\Psi(\{A_i\} \cup \pi_i)$ to $[0, 1]$. The joint probability distribution P is

$$P = p(A_1 \ldots A_\alpha) = \prod_{i=1}^{\alpha} p(A_i|\pi_i)$$

For medical application, nodes in a Bayesian net represent disease hypotheses, symptoms, laboratory results, etc. Arcs signify the causal relations between them. The probability distribution quantifies the strengths of these relations. For example, "cts (C) often causes pain in forearm (F)" can be represented by an arc from C to F and a probability $p(f_1|c_1) = 0.75$.

The term 'causal' above is to be interpreted in a broad sense. Correspondingly, arcs in a Bayesian net could go either direction. But in medical applications, it is usual to consider diseases as causes and symptoms as effects. Shachter and Heckerman [1987] show that if arcs in Bayesian nets are directed from disease to symptoms, the DAG construction is usually easier and the resultant net topologies are usually simpler. Furthermore, conditional probabilities of symptoms given diseases are related to the symptom causing mechanisms of diseases, and are usually irrelevant to the patient population. Thus only priors of diseases need to be changed when an expert system is built at one clinic and used at another location with a different patient population. Whereas both the priors for symptoms and the probabilities of diseases given symptoms are patient population dependent. Directing arcs from symptoms to diseases will require total revision of probability values in a Bayesian net when the system is to be used with a different population.

As stated above, Bayesian networks combine probability theory with graphic representation of domain models. The necessity of this combination is two-edged. For one thing, encoding a domain model with a DAG conveys directly the dependence and independence assumptions made of the domain. The DAG facilitates knowledge acquisition and makes the representation transparent. For another, graphical models allow quick identification of dependencies by examining DAGs locally. Therefore efficient belief propagations are possible [Pearl 88] and the difficulty associated with general probabilistic reasoning (section 1.2.3) can be avoided. The study of inference in Bayesian networks heavily depends on the study of the DAG topologies. In the thesis, when the topology of a Bayesian network is mentioned, it always means the topology of the corresponding DAG.

Probabilities associated with Bayesian networks have different meanings depending on their location of storage in the networks and the stage of inference. Some convention is appropriate to avoid confusion. Before any inference takes place, the probabilities associated with root nodes (with zero in-degree) are called *prior* probabilities or priors. The probabilities associated with non-root nodes are called *conditional* probabilities. After the evidence is available, the updated probabilities conditioned on evidence are called *posterior* probabilities. When it is clear from the context, they are just called probabilities.

1.3.4 Propagating Belief in Bayesian Networks

Inference in a Bayesian network is essentially a problem of propagating changes in belief and computing posterior probabilities as new evidence is obtained. Since the appeal of Bayesian networks for representing uncertain knowledge has become increasingly apparent over the last few years [Howard and Matheson 84, Pearl 86], there has been a proliferation of research seeking to develop new and more efficient inference algorithms. Two classes of approaches can be identified. One class of approaches explores *approximation* using stochastic simulation and Monte Carlo schemes. A good review on this class is given by Henrion [1990]. Another class of approaches explores specificity in computing *exact* probabilities. Since this thesis research adopts the exact method, the major advances in this class are reviewed below.

The first breakthrough in efficient probabilistic reasoning in Bayesian networks is made by Kim and Pearl [1983]. They develop an algorithm, applicable to singly connected Bayesian networks (Figure 1.4), for propagating the effect of new observations. Each node in the network obtains messages from its parents (π messages) and its children (λ messages). These messages represent all the evidence from the portion of the network lying beyond these parents and children. The single-connectedness guarantees that the information in each message to a node is independent and so local updating can be employed. The algorithm's complexity is linear in the number of variables.

Heckerman [1990] provides QUICKSCORE algorithm which addresses networks of diameter 1 as the one in Figure 1.5. The upper level consists of disease variables and the lower level consists of 'finding' variables. Note the net is multiply connected (Appendix A). The algorithm makes three assumptions. All variables are binary. Diseases

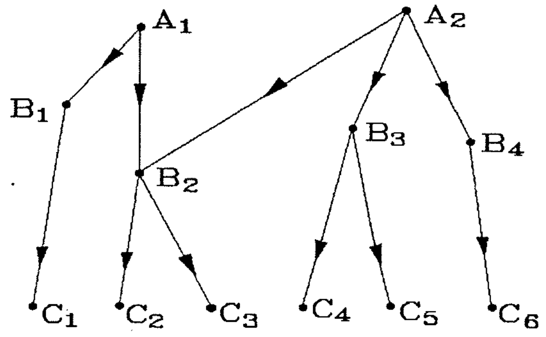


Figure 1.4: The DAG of a singly connected Bayesian network

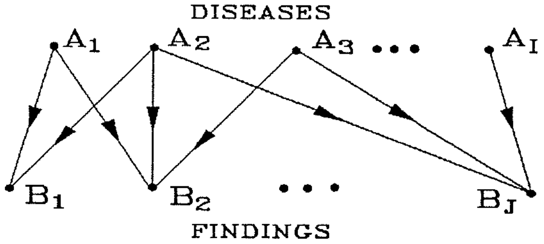


Figure 1.5: The DAG of a Bayesian network with diameter 1

are marginally independent and findings are conditionally independent given diseases[2]. Diseases interact to produce findings via a noisy-OR-gate. The time complexity of QUICKSCORE is $\mathcal{O}(nm_1 2^{m_2})$ where n is the number of diseases, m_1 is the number of negative findings, and m_2 is the number of positive findings.

Unfortunately, many applications can not be represented properly by a singly-connected net of diameter 1. An example of a general multiply connected Bayesian network is given in Figure 1.6. Pearl [1986] presents *loop cutset conditioning* as an indirect method for inference in multiply connected networks. Selected variables (loop cutset) are instantiated to cut open all loops such that resultant singly connected networks can be solved by $\lambda - \pi$ message passing, and the results from the instantiations are combined.

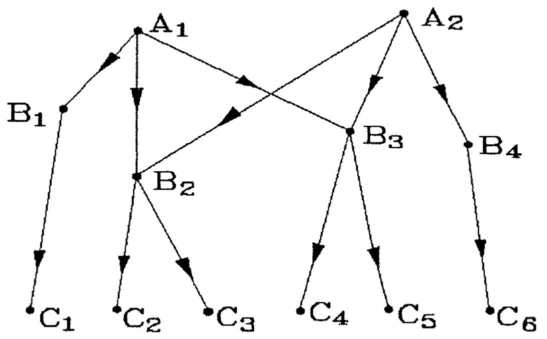


Figure 1.6: The DAG of a general multiply connected Bayesian network

Shachter [1986,1988a] applies a sequence of operations called *arc-reversal* and *barren node reduction* to an arbitrary multiply connected Bayesian net. The process continues until the network contains only those nodes whose posterior distributions are desired with the evidence nodes as immediate predecessors.

Baker and Boult [1990] extend the barren node reduction method to prune a Bayesian network relative to each query instance such that saving in computational complexity is

[2] This implies that the net topology is of diameter 1.

obtained when evidence comes in a batch.

Lauritzen and Spiegelhalter [1988] describe an algorithm based on a reformulation of a multiply connected network. The DAG is moralized and triangulated. Then cliques are identified to form a clique hypergraph of the DAG. The cliques are finally organized into a directed tree of cliques[3] which satisfies running intersection property. They provide an algorithm for the propagation of evidence within this secondary representation. The complexity of the algorithm is $\mathcal{O}(pr^m)$ where p is the number of cliques in the clique list, r is the maximum number of alternatives for a variable in the network, and m is the maximum number of variables in a clique. Pearl [1988] proposes a *clustering method* with many similarities but propagates evidence in a secondary directed tree by $\lambda - \pi$ message passing. The directed tree approach and the following junction tree approach all have the advantage of trading compile time with run time for 'reusable systems'. Here reusable systems mean those systems where the domain knowledge is captured once and is used for multiple cases, as opposed to the decision systems which are created for decision making in a non-repeatable situation.

Jensen, Lauritzen, and Olesen [1990] further improve the directed clique tree approach and they organize clique hypergraph into an (undirected) *junction tree* to allow more flexible computation. Similar work was done by Shafer and Shenoy [1988]. A close look at the junction tree approach is included in Chapter 4.

[3] The 'directed' property is indicated by Henrion [1990], Neapolitan [1990], and Shachter [1988b] since the cliques are ordered and this order is to be followed in belief propagation.

Chapter 2

THE FEASIBILITY OF FINITE TOTALLY ORDERED PROBABILITY ALGEBRA FOR PROBABLE REASONING

Medical diagnosis is featured by probable reasoning and a proper uncertainty calculus is thus crucial in knowledge representation for medical expert systems. A finite calculus is plausible since it is close to human language in communicating uncertainty. An investigation into the feasibility of using finite totally ordered probability models for probable reasoning was conducted under Aleliunas's Theory of Probabilistic Logic [Aleliunas 88]. In this investigation, the general form of the probability algebra of these models and the number of possible algebras given the size of probability set are derived. Based on this analysis, the problems of denominator-indifference and ambiguity-generation that arise in reasoning by cases and abductive reasoning are identified. The investigation shows that a finite probability model will be of very limited usage. This highlights infinite totally ordered probability algebras including probability theory as uncertainty management formalism for probable reasoning.

This chapter presents the major results of the investigation. The results are mainly taken from Xiang et al. [1991a]. Section 2.1 discusses the motivation and criteria of the investigation. Section 2.2 presents the mathematical structure of finite totally ordered probability models. Section 2.3 derives the inference rules under these models. Section 2.4 identifies the problems of these models both intuitively and quantitatively. Section 2.5 presents an experiment which exhaustively investigates models of a given size with an example.

2.1 Motivation

The investigation started in the early stage of this thesis research when two engineering applications of AI in neurology, EEG analysis and neuromuscular diagnosis, were under consideration. Probable reasoning is the common feature of both domains. When consulted about the formalism of representing uncertainty in EEG analysis, the experts claimed that they did not use numbers, but rather used a small number of terms to describe uncertainty. This motivated a desire for a formal finite non-numerical uncertainty calculus. In such a calculus, the domain expert's vocabulary about uncertainty could be used directly in encoding knowledge and in reasoning about uncertain information. This would facilitate knowledge acquisition and make the system's diagnostic suggestion and explanation more understandable.

There were few known finite calculus for general uncertainty management [Pearl 89, Halpern and Rabin 87], but Aleliunas' probabilistic logic [Aleliunas 88] was explored, because it seemed to be based on clear intuitions, and to allow measures of belief (probability values) to be summarized by values other than just real numbers.

Aleliunas [1988] presents an axiomatization for a theory of rational belief, the *Theory of Probabilistic Logic* (TPL). It generalizes classical probability theory to accommodate a variety of probability values rather than just $[0, 1]$. According to the theory, *probabilistic logic* is a scheme for relating a body of evidence to a potential conclusion (a hypothesis) in a rational way, using *probabilities* as *degrees of belief.* '$p(P|Q)$' stands for the conditional probability of proposition P given the evidence Q, where P and Q are sentences of some formal language $\mathbf{L}$ consisting of boolean combinations of propositions. TPL is chiefly concerned with identifying the characteristics of a family $\mathbf{F}$ of functions from $\mathbf{L} \times \mathbf{L}$ to the set of probabilities $\mathbf{P}$. The probability values $\mathbf{P}$ are not constrained to be just $[0, 1]$, but can be any values that conform to a set of reasonably intuitive axioms (Appendix B.1).

The semantics of TPL is given by 'possible worlds'. Each proposition P is associated with a set of *situations* or *possible worlds* $S(P)$ in which P holds. Given Q as evidence, the conditional probability $p(P|Q)$, whose value ranges over the set $\mathbf{P}$, is some measure of the fraction of the set $S(Q)$ that is occupied by the subset $S(P\&Q)$.

TPL provides minimum constraints for a rational belief model. For the application domain in question the following criteria are thought desirable:

R1 The domain experts do not express and communicate uncertainty using numerical values in their practice. Their language consists of a small set of terms 'likely', 'possibly', etc., used to describe the uncertainty in their domain. Thus a *finite* set of probability values is required.

R2 Any two probability values in a chosen model should be comparable. An essential task of a medical diagnostic system is to differentiate between a set of competing diagnoses given a patient's symptoms and history. It is felt as though *totally ordered* probabilities are needed in order to allow for totally ordered decisions when one has to act on the results of the diagnoses.

R3 Inference based on a TPL model should generate empirically intuitive results. That is, the inference outcomes generated with such a model should reflect, as far as possible, the *reasonable* outcomes reached by a human expert.

Although these criteria are formed from the point of application in question, it is believed that they are shared by many automated reasoning systems making decisions under uncertainty. Based on the first 2 criteria, the focus is placed on finite totally ordered probability models.

2.2 Finite Totally Ordered Probability Algebras

2.2.1 Characterization

To investigate the mathematical structure (*probability algebra*) of the probability space, the characterization of any finite totally ordered probability algebra under TPL axioms [Aleliunas 88] is given in the proposition below. For more about universal algebra, see Burris and Sankappannvar [1981] and Kuczkowski and Gersting [1977]. This proposition is a restriction of the Probability Algebra Theorem [Aleliunas 86] (Appendix B.2) to finite totally ordered sets.

The smallest element of $\mathbf{P}$ is denoted as 0, and the largest element of $\mathbf{P}$ as 1. There are a *finite* number of other values between 0 and 1.

Proposition 1 *A probability algebra defined on a totally ordered finite set* $\mathbf{P}$ *with ordering relation '<' satisfies TPL axioms iff*

Cond1 *An order preserving binary operation '$*$' (product) is well defined and closed on* $\mathbf{P}$.

Cond2 *'$*$' is commutative, i.e.,* $(\forall p, q \in \mathbf{P})\ p * q = q * p$.

Cond3 *'$*$' is associative, i.e.,* $(\forall p, q, r \in \mathbf{P})\ p * (q * r) = (p * q) * r$.

Cond4 $(\forall p, q, r \in \mathbf{P})\ (p * q = r) \Rightarrow (r \leq \min(p, q))$.

Cond5 *No non-trivial zero, i.e.,* $(\forall p, q \in \mathbf{P})\ p * q = 0 \Rightarrow (p = 0 \vee q = 0)$.

Cond6 $(\forall p, q \in \mathbf{P})\ p \leq q \Rightarrow (\exists r \in P)\ p = r * q$. *The* **solution** *will be denoted as* $r = p/q$.

Cond7 $(\forall p \in \mathbf{P})\ 0 \leq p \leq 1$.

Cond8 $(\forall p \in \mathbf{P})\ p * 1 = p$.

Cond9 *A monotone decreasing inverse function* $i[\cdot]$ *is well defined and closed on* $\mathbf{P}$*, i.e.,*

$$(\forall p < q \in \mathbf{P})\ i[p] > i[q].$$

Cond10 $(\forall p \in \mathbf{P})\ i[i[p]] = p$.

Proof:

One needs to show the equivalence of Cond1,...,Cond10, to T1,...,T7 of Probability Algebra Theorem (Appendix B.2). Cond4 and Cond10 are not obvious in the theorem and they are included in proposition 1 for the convenience of later use. They are proved here from TPL axioms AX1,...,AX12 (Appendix B.1) directly.

Cond1 and Cond3 are equivalent to T1. Cond2 is equivalent to T5.

Proof of Cond4. With Cond2, it suffices to prove $r \leq q$. By AX10, let $p = f(B|A)$ and $q = f(A|1)$. Then

$$\begin{aligned} p * q &= f(B|A) * f(A|1) \\ &\leq f(B|A) * f(A|A) \quad (f(A|1) \leq f(A|A)) \\ &= f(B\&A|A) \quad (AX8) \\ &= f(B|A) \quad (AX5) \\ &= p \end{aligned}$$

Cond5 is equivalent to T6. Cond6 is equivalent to T7. Cond7 is equivalent to T4. Cond8 and Cond4 are equivalent to T2. Cond9 is equivalent to T3. Cond10 is implied by AX3.

□

From now on any *Finite Totally Ordered Probability Algebra* satisfying proposition 1 is referred as *legal FTOPA*. The general form of all legal FTOPA is derived in section 2.2.2.

2.2.2 Mathematical Structure

Here only those probability algebras with at least 3 elements are considered[1]. A finite totally ordered probability set with size n is denoted as $\mathbf{P} = \{e_1, e_2, \ldots, e_{n-1}, e_n\}$, where $1 = e_1 > e_2 > \ldots > e_{n-1} > e_n = 0$. For example, $\mathbf{P} = \{e_1, e_2, e_3, e_4\}$ could stand for {certain, likely, unlikely, impossible}. This linguistic interpretation is left open.

The uniqueness of the inverse function $i[\cdot]$ of any legal FTOPA is given by the following lemma.

Lemma 1 *For a legal FTOPA with size n, the inverse is uniquely defined as*

$$i[e_k] = e_{n+1-k} \quad (1 \leq k \leq n).$$

Proof:

By Cond10, the inverse is symmetric and it suffices to prove for $k \leq n/2$.

For $k = 1$, $i[e_1] \geq e_n$ by Cond7. Assume $i[e_1] > e_n$. Then $i[e_n] > i[i[e_1]] = e_1$ which is contradictory to Cond7. Hence $i[e_1] = e_n$.

Suppose $i[e_j] = e_{n+1-j}$ for $j \leq n/2$. For $k = j + 1 \leq n/2 + 1$, assume $i[e_{j+1}] < e_{n-j}$ which implies $i[e_{j+1}] \leq e_{n-j+1}$. By Cond9, this further implies $e_{j+1} \geq i[e_{n-j+1}] = e_j$ which is contradictory to ordering $e_{j+1} < e_j$. Thus, $i[e_{j+1}] \geq e_{n-j}$.

Assume $i[e_{j+1}] > e_{n-j}$. Then $e_{j+1} < i[e_{n-j}]$, i.e. $i[e_{n-j}] \geq e_j$. By Cond9, $e_{n-j} \leq i[e_j] = e_{n+1-j}$ which is contradictory to $e_{n-j} > e_{n-j+1}$.

□

Thus given the size of a legal FTOPA, only the choice of the product function is left.

A probability $p \in P$ is *idempotent* if $p * p = p$. Idempotent elements play important roles in defining probability algebras as will be shown in a moment. The following 2 lemmas describe properties of idempotent elements. Aleliunas [1986] gives similar statement to that in lemma 3.

[1] Probability algebra with 2 elements is equivalent to propositional logic [Aleliunas 87].

Lemma 2 *Any legal FTOPA has at least 3 idempotent elements, namely e_1, e_{n-1} and e_n.*

Proof:

The proof is trivial for e_1 and e_n. $e_{n-1} * e_{n-1} \leq e_{n-1}$ by Cond4 and $e_{n-1} * e_{n-1} > e_n$ by Cond5. Therefore $e_{n-1} * e_{n-1} = e_{n-1}$.

□

Lemma 3 *For any legal FTOPA, if $p \in \mathbf{P}$ is idempotent, then $(\forall q \in P)\ p{*}q = \min(p, q)$.*

Proof:

Assume $q > p$. Then $(\exists r \in \mathbf{P})\ r * q = p$. Therefore $p * p = r * (q * p) = p$ implies $q * p \geq p$. But by Cond4, $q * p \leq p$ which proves $q * p = p$.

If $q < p$, then $(\exists r \in \mathbf{P})\ r * p = q$. Then $q * p = r * (p * p) = r * p = q$.

□

Proposition 2 *For a finite totally ordered set with size $n \geq 3$, there exists only one legal FTOPA with 3 idempotent elements. The '$*$' operation on it is defined as*

$$e_i * e_j = \begin{cases} e_n & \text{if } i \text{ or } j = n \\ e_{\min(i+j-1,n-1)} & \text{otherwise.} \end{cases}$$

Proof:

Let $M_{n,k}$ denote a legal FTOPA with size n and k idempotent elements.[2] Let $a_{i,j}$ denote $e_i * e_j$. The following proves the proposition constructively.

(1) In case of i or $j = n$, the proposition holds due to lemma 3. By non-trivial zero, zero part of the product table is entirely covered within this case.

[2] In general, for a pair of n and k, there may be more than one legal FTOPA. Thus $M_{n,k}$ does no necessarily stand for a unique model characterized by n and k.

$$\begin{array}{c|ccc|} & e_1 & e_2 & e_3 \\ \hline e_1 & e_1 & e_2 & e_3 \\ e_2 & e_2 & e_2 & e_3 \\ e_3 & e_3 & e_3 & e_3 \\ \hline \end{array} \qquad \begin{array}{c|cccc|} & e_1 & e_2 & e_3 & e_4 \\ \hline e_1 & e_1 & e_2 & e_3 & e_4 \\ e_2 & e_2 & e_3 & e_3 & e_4 \\ e_3 & e_3 & e_3 & e_3 & e_4 \\ e_4 & e_4 & e_4 & e_4 & e_4 \\ \hline \end{array}$$

$M_{3,3}$ $M_{4,3}$

(2) What is left is to prove the non-zero part of the product table (the second half of the product formula) which is bounded by two idempotent elements e_1 and e_{n-1}. For the completeness of the product table, the zero parts are still included in the following tables although they are not relevant to the remaining proof.

For $M_{3,3}$ and $M_{4,3}$ the proposition holds (see the product tables). It is not difficult to check that they satisfy proposition 1 and any change to these product tables will violate proposition 1 in one way or another.

Suppose a unique legal FTOPA $M_{m,3}$ exists with product defined as in the proposition. As for $M_{m+1,3}$ (table below), the product $a_{i,j}$ $(i+j \leq m)$ should be constructed in the same way as in $M_{m,3}$, i.e., the second half of product formula

$$a_{i,j} = e_{\min(i+j-1,m)} = e_{i+j-1}$$

applies within this portion as does in $M_{m,3}$. If this portion could be changed without violating proposition 1, the corresponding portion in $M_{m,3}$ could also be changed which is contradictory to the uniqueness assumption for $M_{m,3}$.

Further one can show the uniqueness of $a_{i,j}$ for all $(i, j < m < i+j)$.

$$\begin{array}{c|ccccccccc|}
 & e_1 & e_2 & e_3 & \cdots & & & e_{m-1} & e_m & e_{m+1} \\
\hline
e_1 & e_1 & e_2 & e_3 & \cdots & & & e_{m-1} & e_m & e_{m+1} \\
e_2 & e_2 & e_3 & \cdots & & & e_{m-1} & a_{2,m-1} & e_m & e_{m+1} \\
e_3 & e_3 & \cdots & & & e_{m-1} & a_{3,m-2} & a_{3,m-1} & e_m & e_{m+1} \\
 & & & & & & \cdots & & & \\
 & & & & \cdots & a_{j,m-j+1} & & & & \\
\vdots & \vdots & & \cdots & a_{j+1,m-j} & & & & \vdots & \vdots \\
 & & & & \cdots & & & & & \\
e_{m-1} & e_{m-1} & \cdots & & & & & ? & & \\
e_m & e_m & & & & & \cdots & & e_m & \\
e_{m+1} & e_{m+1} & & & & & \cdots & & & e_{m+1} \\
\hline
\end{array}$$

Note: '?' stands for product items to be chosen.

$M_{m+1,3}$

By associativity, one has

$$\begin{aligned}
& (e_j * e_2) * e_{m-j} = e_{j+1} * e_{m-j} \\
= \; & a_{j+1,m-j} \\
= \; & e_j * (e_2 * e_{m-j}) = e_j * e_{m-j+1} \\
= \; & a_{j,m-j+1} \qquad (2 \leq j \leq m-2) \quad (a)
\end{aligned}$$

Also one has

$$\begin{aligned}
& e_2 * (e_j * e_{m-1}) = e_2 * a_{j,m-1} \\
= \; & (e_2 * e_j) * e_{m-1} = e_{j+1} * e_{m-1} \\
= \; & a_{j+1,m-1} \qquad (2 \leq j \leq m-2) \quad (b)
\end{aligned}$$

From order preserving property of '$*$', one knows

$$a_{i,j} = e_{m-1} \quad \vee \quad a_{i,j} = e_m \quad (i, j < m < i + j).$$

Suppose $a_{2,m-1} = e_{m-1}$. Then from (b),

$$e_2 * a_{2,m-1} = e_2 * e_{m-1} = a_{2,m-1} = e_{m-1} = a_{3,m-1}.$$

Similarly, and from commutativity and order preserving, one has

$$a_{i,j} = e_{m-1} \quad (i, j < m < i + j).$$

This means that e_{m-1} is also an idempotent element which is contradictory to the 3 idempotent elements assumption. Therefore, $a_{2,m-1} = e_m$. Then from (a) and order preserving, one ends up with $a_{i,j} = e_m$ $(i, j < m < i + j)$.

□

The second part of the above proof for product bounded by e_1 and e_{n-1} does not involve the 0 element at all as already stated. Thus for any legal FTOPA with more than 3 idempotent elements, the proposition holds for each diagonal block of its product table bounded by two adjacent idempotent elements. The non-diagonal part of the product table is totally determined by lemma 3, the order preserving and solution existing property. Thus one has the following theorem 1. Given proposition 2 and above description, the proof is trivial.

Theorem 1 *Given a finite totally ordered set $P = \{e_1, e_2, \ldots, e_n\}$ with ordering relation $e_1 > e_2 > \ldots > e_n$ and a set I of indexes of all the idempotent elements on P, $I = \{i_1, i_2, \ldots, i_m\}$ where $i_1 < i_2 < \ldots < i_m$ there exists a unique legal FTOPA whose product function is defined as*

$$e_j * e_k = \begin{cases} \min(e_j, e_k) & \textit{if } \ j = i_l \\ e_{\min(j+k-i_l, i_{l+1})} & \textit{if } \ i_l < j, k \leq i_{l+1} \\ e_k & \textit{if } \ j \leq i_l < k \leq i_{l+1} \end{cases}$$

and whose inverse function is defined as

$$i[e_k] = e_{n+1-k}$$

Theorem 1 says that, given the set of idempotent elements, a legal FTOPA is totally defined. From theorem 1 and lemma 2 one can easily derive the following corollary.

Corollary 1 *The number of all the possible legal FTOPA of size $n \geq 3$ is*

$$\sum_{i=0}^{n-3} C_{n-3}^{i} = 2^{n-3}$$

where C_m^i is the number of combinations taking i elements out of m.

Theorem 1 and corollary 1 provide the possibility of exhaustive investigation for any legal FTOPA of a given size.

2.2.3 Solution and Range

Once a legal FTOPA is defined, its solution table is forced. Inverses to the operation $* : P \times P \to P$ will not be unique. For this reason, it is necessary to introduce a probability *range* denoted by $[l, u]$ representing all the probability values between lower bound l and upper bound u.

$$[l, u] = \{v \in P | l \leq v \leq u\}$$

$[v, v]$ is written as just v. One has the following corollary on single value probability solution. Its proof can be found in Appendix D.

Corollary 2 *Given a finite totally ordered set $P = \{e_1, e_2, \ldots, e_n\}$ with ordering relation $e_1 > e_2 > \ldots > e_n$ and a set I of indexes of all the idempotent elements on P, $I = \{i_1, i_2, \ldots, i_m\}$ where $i_1 < i_2 < \ldots < i_m$ the solution function (multiple value) of a legal FTOPA is forced to be:*

$$e_k/e_j = \begin{cases} e_k & \mathit{if}\ \ j = 1 \\ e_n & \mathit{if}\ \ k = n, j \neq n \\ e_{k-j+i_l} & \mathit{if}\ \ k > j, \\ & \quad i_l + 1 < k \leq i_{l+1} - 1, \\ & \quad i_l + 1 \leq j < i_{l+1} - 1 \\ [e_{i_l}, e_1] & \mathit{if}\ \ k = j, i_l < k < i_{l+1} \\ [e_{i_{l+1}}, e_{i_{l+1}+i_l-j}] & \mathit{if}\ \ k = i_{l+1}, i_l < j < i_{l+1} \\ [e_k, e_1] & \mathit{if}\ \ k = j = i_l, k \neq 1, n \\ e_k & \mathit{if}\ \ j \leq i_l < k \leq i_{l+1} \end{cases}$$

In Appendix C.1, the product and solution tables for three legal FTOPAs of size 8 are presented.

The solution of two single valued probabilities may become a range which will participate in further manipulation. Thus the product and solution of ranges should be considered before we can manipulate uncertainty in an inference chain.

Definition 1 *For any legal FTOPA, the product of two ranges $[a,b]$ $(a \leq b)$ and $[c,d]$ $(c \leq d)$ is defined as*

$$[a,b] * [c,d] = \{z | \exists x \in [a,b] \& \exists y \in [c,d] \& z = x * y\}.$$

And the solution of above two ranges with additional constraint $a \leq d$ is defined as

$$[a,b]/[c,d] = \{z | \exists x \in [a,b] \& \exists y \in [c,d] \& x = y * z\}.$$

One can prove the following proposition.

Proposition 3 *For any legal FTOPA, the product of two ranges $[a,b]$ $(a \leq b)$ and $[c,d]$ $(c \leq d)$ is*

$$[a,b] * [c,d] = [a * c, b * d].$$

And the solution of above two ranges with additional constraint $a \leq d$ is

$$[a,b]/[c,d] = \begin{cases} [LB(a/d), UB(b/c)] & \text{if } b \leq c \\ [LB(a/d), e_1] & \text{if } b > c \end{cases}$$

where LB and UB are lower and upper bounds of ranges.

It should be noted that, in general, product and solution of legal FTOPAs do not follow commutativity. For example, in model $M_{8,8}$,

$$(e_2 * e_5)/e_5 = [e_5, e_1] \neq e_2 * (e_5/e_5) = [e_5, e_2].$$

Thus the order of product and solution in evaluation of conditional probability

$$\begin{aligned} p(A|B\&C) &= p(A\&B|C)/p(B|C) \\ &= (p(B|A\&C) * p(A|C))/p(B|C) \end{aligned}$$

can not be changed arbitrarily.

2.3 Bayes Theorem and Reasoning by Case

Having derived the mathematical structure of legal finite totally ordered probability models, one needs deductive rules. In this investigation, DAG and conditional independence assumption made in Bayesian nets are adopted as the qualitative part of the knowledge representation. Instead of using probability theory to represent uncertainty as Bayesian nets, the legal FTOPAs are used. Call the resulting overall representation a *quasi-Bayesian net.* The quasi-Bayesian nets are implemented by PROLOG programs with the approach described by Poole and Neufeld [1988]. The inference rules thus required are *Bayes theorem* and *reasoning by cases.*

Bayes theorem provides a way of determining the possibility of certain causes from the observation of effects.[3] It takes the form:

$$p(P|Q\&C) = p(Q|P\&C) * p(P|C)/p(Q|C)$$

which is the same with the one introduced in section 1.3.1 for probability theory. But here the order of calculation is to be followed as it appears as discussed in section 2.2.3.

Reasoning by cases is an inference rule to compute a conditional probability by partitioning the condition into several mutually exclusive situations such that the estimation under each of them is more manageable. The simplest form considers the cases where B is true and where B is false:

$$p(A|C) = p((A\&B) \vee (A\&\overline{B})|C)$$

Under probability theory, it can be derived from marginalization rule in section 1.3.1:

$$p(A|C) = p(A|B\&C) \cdot p(B|C) + p(A|\overline{B}\&C) \cdot p(\overline{B}|C)$$

Using TPL, the $\cdot$ becomes $*$, and one does not have the $+$. This can, however, be simulated using product and inverse. The corresponding formula under TPL is given by the following propositions.

Proposition 4 *Let A, B, and C be three sentences. $p(A|C)$ can be computed using the following:*

$$\begin{aligned} f_1 &= i[p(A|\overline{B}\&C) * i[p(B|C)]] \\ f_2 &= i[p(A|B\&C) * p(B|C)/f_1] \\ p(A|C) &= i[f_1 * f_2]. \end{aligned}$$

[3] *Cause* and *effect* are used here in a broad sense.

Using quasi-Bayesian nets, the probability of a hypothesis given some set of evidence can be computed by applying the two inference rules, namely, Bayes theorem and reasoning by cases [Poole and Neufeld 88].

2.4 Problems With Legal Finite Totally Ordered Probability Models

2.4.1 Ambiguity-generation and Denominator-indifference

With the mathematical structure of legal finite totally ordered probability models and the form of relevant deductive rules derived, one can assess these probability models as to how well they fit in with the intuition.

To begin with, examine the solution of legal FTOPA $M_{n,n}$ which has all its elements idempotent. The solution takes the form of (compare to Appendix C.1)

$$e_k/e_j = \begin{cases} e_k & \text{if } \ j < k \\ [e_k, e_1] & \text{if } \ k = j \end{cases}$$

Note that e_j does not have direct influence on the result of the first case of the solution. Name this phenomenon as *denominator-indifference.* Also, name the emergence of range in the second case of the solution operation as *ambiguity-generation.*

To analyze the effect of denominator-indifference and ambiguity-generation on application of Bayes theorem, apply Bayes theorem to $M_{n,n}$.

$$\begin{aligned} & p(A|B\&C) \\ = \ & p(A\&B|C)/p(B|C) \\ = \ & \begin{cases} p(A\&B|C) & \text{if } \ p(A\&B|C) \neq p(B|C) \\ [p(A\&B|C), e_1] & \text{if } \ p(A\&B|C) = p(B|C) \end{cases} \end{aligned}$$

In the first case, the probability $p(B|C)$ does not affect the estimation of $p(A|B\&C)$ due to denominator-indifference. In the second case, ambiguity-generation produces a disjunct of all the probabilities larger than $p(B|C)$ which is a very rough estimation. Neither satisfies the requirement for empirically satisfactory probability estimates.

To analyze the effect of denominator-indifference and ambiguity-generation on reasoning by cases, consider applying proposition 4 to $M_{n,n}$.

$$p(A|C) = \begin{cases} \max(p(A\&B|C), p(A\&\overline{B}|C)) \\ \quad \text{if} \quad p(A\&B|C) \neq p(\overline{A\&\overline{B}}|C) \\ [\max(p(A\&B|C), p(A\&\overline{B}|C)), e_1] \\ \quad \text{if} \quad p(A\&B|C) = p(\overline{A\&\overline{B}}|C) \end{cases}$$

Here again, in the first situation, denominator-indifference forces a choice of outcome from one case or another instead of giving some combination of the two outcomes. One does not get an estimation larger than both which is contrary to the intuition. In the second situation, a very rough estimation appears because of ambiguity-generation. Note that, when $\max(p(A\&B|C), p(A\&\overline{B}|C))$ is small, $p(A|C)$ can span almost the whole range of probability set $\mathbf{P}$.

The analysis here is in terms of a model that has all of its values idempotent. The other case to consider is what happens at the values between the idempotent values.

Consider $M_{n,3}$ which has minimal number of idempotent elements. By proposition 2, its product is

$$e_i * e_j = \begin{cases} e_{\min(i+j-1,n-1)} & \text{if} \quad i,j \neq n \\ e_n & \qquad \text{otherwise.} \end{cases}$$

Its solution simplifies to (compare to Appendix C.1)

$$e_k/e_j = \begin{cases} e_n & \text{if} \quad k = n > j \\ e_{k-j+1} & \text{if} \quad j \leq k < n-1 \\ [e_{n-1}, e_{n-j}] & \text{if} \quad k = n-1 \geq j \end{cases}$$

In this algebra, it is quite easy for a manipulation to reach the probability value e_{n-1}:

1. Whenever one of the factors of product is e_{n-1}, the product will be e_{n-1} unless the other factor is e_n.

2. Whatever takes the value e_2, its inverse will be e_{n-1}.

3. Products of low or moderate probability tend to reach e_{n-1} due to quick decreasing of product.

4. $e_j/e_{j-1} = e_2$ for all $2 \leq j \leq n-2$.

Once e_{n-1} is reached, any solution will be ambiguous. This ambiguity will be propagated and amplified during further inference in Bayesian analysis or case analysis. Although e_{n-1} is a value one should try to avoid, there is no means to avoid it. Here one sees an interesting trade off between the two problems. In $M_{n,3}$, the denominator-indifference disappears. But, since manipulations under this model move probability values quickly, they tend to produce e_{n-1} more frequently and thus one suffers more from the ambiguity-generation.

As all finite totally ordered probability algebras can be seen as combinations of the above two cases, they must all suffer from the denominator-indifference and the ambiguity-generation. The question is how serious the problems are in an arbitrary model. This is to be answered in the next section.

2.4.2 Quantitative Analysis of the Problems

Given the constraint of legal FTOPA in choosing a probability model, one is free to select the model size n and to select among 2^{n-3} alternative legal FTOPAs once n is fixed. A few straightforward measurements are introduced to quantify the degree of suffering in a randomly chosen model.

The number of ranges in a model's solution table and the number of elements covered by each range mirror the problem of ambiguity-generation of the model. Define a measurement of the amount of ambiguity in a model as the number of elements covered by ranges in its solution table minus the number of ranges.

Definition 2 *Let $S = \{r_1, r_2, \ldots, r_m\}$ be the set of ranges in the solution table of a legal FTOPA. Let w_j be the number of values covered by range r_j. Let M be the number of different solution pairs in the solution table.*

The **amount of ambiguity** *of the algebra is defined as*

$$A = \sum_{j=1}^{m} w_j - 1.$$

The **relative ambiguity** *of the algebra is defined as*

$$R = A/M.$$

Example 1 The three legal FTOPAs with size 8 in Appendix C.1 all have $A = 21$ and $R = 0.6$.

One has the following proposition. The proof can be found in Appendix D.

Proposition 5 *The amount of ambiguity of any legal FTOPA with size n is*

$$A = (n-1)(n-2)/2.$$

The relative ambiguity of the algebra is

$$R = (n-2)/(n+2).$$

The number of solution pairs satisfying $e_j/e_k = e_j$ reflects the seriousness of denominator-indifference of the model. Define the order of denominator-indifference as this number minus the number of such e_js.

Definition 3 *Let d_j be the number of times $e_j/e_k = e_j$ for $1 \leq k \leq j$ in a legal FTOPA of size n. The* **order of denominator-indifference** *of the algebra is defined as*

$$O_d = \sum_{j=2}^{n-1} d_j - 1.$$

Example 2 The three legal FTOPAs $M_{8,3}$, $M_{8,8}$ and $M_{8,4}$ in Appendix C.1 have O_d values 0, 15 and 9 respectively.

Define the order of mobility of a model to express the chance with which a product or a solution transfers an operand to a different value. The higher this order, the more likely for a manipulation to generate an idempotent element and produce ambiguity afterwards.

Definition 4 *The* **order of mobility** *O_m of a legal FTOPA is defined as the number of distinct product pairs $a * b$ in its product table such that $a * b < \min[a, b]$.*

Example 3 The three legal FTOPAs $M_{8,3}$, $M_{8,8}$ and $M_{8,4}$ in Appendix C.1 have O_m values 15, 0 and 6 respectively.

One has the following proposition. The proof can be found in Appendix D.

Proposition 6 *For any legal FTOPA with size n and a set I of indexes of all its idempotent elements $I = \{i_1, i_2, \ldots, i_k\}$ where $i_1 < i_2 < \ldots < i_k$ its order of denominator-indifference is*

$$O_d = \sum_{m=2}^{k-2} (i_m - 1) \cdot (i_{m+1} - i_m),$$

its order of mobility is

$$O_m = \sum_{m=1}^{k-2} \sum_{j=1}^{i_{m+1}-i_m-1} j,$$

and

$$O_d + O_m = (n-2)(n-3)/2.$$

Example 4 The three legal FTOPAs with size 8 in Appendix C.1 all have $O_d+O_m = 15$.

Proposition 5 tells us that all the legal FTOPAs of same size have same amount of ambiguity. Increasing size *increases* R which approaches 1 as n approaches infinity.

Proposition 6 says that,

1. among legal FTOPAs of same size n, the order of denominator-indifference O_d changes from lower bound 0 at $M_{n,3}$ to upper bound $(n-2)(n-3)/2$ at $M_{n,n}$;

2. the upper bound of O_d as well as O_m increases with model size n;

3. given n, the sum $O_d + O_m$ remains constant and thus if a model suffers less from denominator-indifference, it must suffer more frequently from ambiguity-generation due to the increase in its mobility.

2.4.3 Can the Changes in Probability Assignment Help?

After explored model size and alternative models given size, the final freedom that remains is the assignment of probability values. From Corollary 2, it is apparent that, in general, denominator-indifference and ambiguity-generation happen only in certain regions of the solution table. So, is it possible, by choosing certain set of probability values as prior knowledge, to avoid intermediate results falling onto those unfavorable regions?

To help answer this question, a derivation of conditional probability $p(fire|smoke$ *& alarm*$)$[4] for a smoke-alarm problem in Figure 2.7 is given in Appendix C.2. The calculation involves 2 applications of Bayes theorem, and 3 of reasoning by cases. It requires 19 products, 9 solutions, and 14 inverses.

In general,

[4] The four nodes involved form a minimum set which has alternative hypotheses (fire and tampering) and allows accumulation of evidences (smoke + alarm).

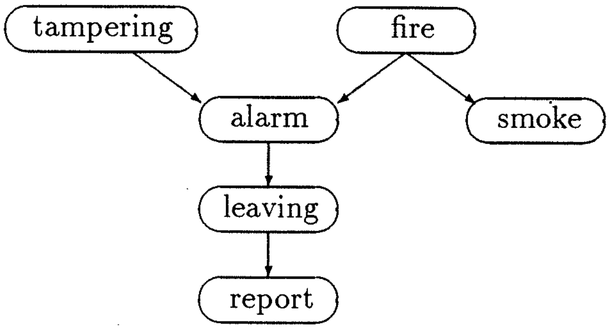


Figure 2.7: Smoke-alarm example

1. a product tends to decrease the probability value until an idempotent value is reached.

2. a solution tends to increase the probability value or cause a large range to occur (especially for idempotent values).

3. an inverse tends to transfer small value into big and vice versa.

Since many operations are required even in a small problem and each operation tends to move the intermediate value around the probability set, the compound effect of the operations are not generally controllable.

To summarize, in the context of legal FTOPA, there seems to be no way to get away with the problem of denominator-indifference and ambiguity-generation by means of clever assignment of probability values; increasing model size does no good in reducing the difficulty; selecting among different models trades one trouble with another.

In the next section, these problems are demonstrated by an experiment.

2.5 An Experiment

All the 32 legal FTOPAs with size 8 were implemented in PROLOG and their performance were tested by the smoke-alarm example (Figure 2.7) from Poole and Neufeld [1988]. The PROLOG program has basically the same structure, but inverse, product, solution, as well as Bayes theorem and reasoning by cases are redefined.

Table 2.1 lists the probabilities in the knowledge base together with numerical values used by Poole and Neufeld [1988] for comparison.

$p(fire) =$	e_6	0.01	$p(smoke\|fire) =$	e_2	0.9
$p(tampering) =$	e_6	0.02	$p(smoke\|\overline{fire}) =$	e_6	0.01
$p(alarm\|fire\&tampering) =$	e_4	0.5	$p(leaving\|alarm) =$	e_2	0.88
$p(alarm\|fire\&\overline{tampering}) =$	e_2	0.99	$p(leaving\|\overline{alarm}) =$	e_7	0.001
$p(alarm\|\overline{fire}\&tampering) =$	e_2	0.85	$p(report\|leaving) =$	e_3	0.75
$p(alarm\|\overline{fire}\&\overline{tampering}) =$	e_7	0.0001	$p(report\|\overline{leaving}) =$	e_6	0.01

Table 2.1: Probabilities for smoke-alarm example

The following posterior probabilities are calculated in all 32 possible legal FTOPAs with size 8 (using quasi-Bayesian nets) and in [0, 1] real number probability model (using Bayesian nets) as a comparison.

$$p(s|f), p(a|f), p(s|t), p(a|t), p(f|s), p(f|a), p(f|s\&a), p(t|s), p(t|a), p(t|s\&a)$$

The first four probabilities are deductive which, given cause acting, estimate the probabilities of effects appearing. The remaining six are abductive which, given effects observed, estimate the probability of each conceivable cause. The results are included in Appendix C.3.

- Among the 32 legal FTOPAs, eight of them produced identical value for the abductive cases:

  $$p(f|s) = p(f|a) = p(f|s\&a) = e_6,$$

  and 16 others produce the identical ranges for all the abductive cases about fire.

  According to Table 2.1, smoke does not necessarily relate to fire ($p(s|\overline{f}) = e_6$). Nor does alarm ($p(a|\overline{f}\&t) = e_2$). As a result, observing only one of smoke and alarm, one is not quite sure about fire. Intuitively, adding the positive evidence alarm to smoke should increase one's belief for fire. As well, adding to alarm the evidence smoke which is independent of tampering indicates higher chance of fire causing alarm. Thus this intuitive inference arrives at

  $$p(f|s\&a) > p(f|s) \quad \& \quad p(f|s\&a) > p(f|a)$$

  which the results obtained from the above mentioned 24 legal FTOPAs do not fit in with.

  To illustrate how this happens, evaluate $p(f|s\&a)$ in model $M_{8,4}$ with idempotent elements $\{e_1, e_5, e_7, e_8\}$.

  $$p(f|s\&a) = p(s|f\&a) * p(f|a)/p(s|a) = e_2 * e_6/e_4 = e_6/e_4 = e_6$$

  Pay attention to the solution in last step. The result is no larger than $p(f|a) = e_6$ due to denominator-indifference. One does not get extra evidence accumulating.

- One of the very useful results provided by $[0, 1]$ numerical probability is that although $p(f|s) = 0.48$ and $p(f|a) = 0.37$ are moderate, when both smoke and alarm are observed $p(f|s\&a) = 0.98$ is quite high which is more intuitive than the case above. According to Table 2.1, fire is the only event which can cause both smoke and alarm with high certainty ($p(s|f) = p(a|f) = e_2$). Thus observing both simultaneously one would expect a higher probability. But the remaining 8 legal FTOPAs give only ambiguous $p(f|s\&a)$ spanning at least half of the total probability range. Consider the evaluation of $p(f|s\&a)$ in model $M_{8,4}$ with idempotent elements $\{e_1, e_4, e_7, e_8\}$.

$$p(f|s\&a) = p(s|f\&a) * p(f|a)/p(s|a) = e_2 * e_5/e_5 = e_5/e_5 = [e_4, e_1]$$

Notice the solution in last step.

- In the deductive case, the situation is slightly better. Some models achieve the same tendency as $[0, 1]$ probability in deduction (e.g., $p(s|t) < p(a|t)$). Some achieve the same tendency with increased ambiguity. Others either produce identical ranges for different probabilities or do not reflect the correct trend. The slight improvement attributes to less operations required in deduction (only reasoning by cases but not Bayes theorem is involved). Since reasoning by cases needs the solution operation, it still creates denominator-indifference and generates ambiguity.

Our experiment is systematic with respect to legal FTOPAs of a particular size 8. Although a set of arbitrarily chosen probability is used in this presentation, it has been tried to vary them in a non-systematic way, but the outcomes are basically the same.

2.6 Conclusion

The motivation of this investigation is to find finite totally ordered probability models to automate reasoning under uncertainty and to facilitate knowledge acquisition and explanation in expert systems.

Under the theory of probabilistic logic [Aleliunas 88], the general form of finite totally ordered probability algebras is derived and the number of different models is deduced such that all the possible models can be explored systematically.

Two major problems of those models are analyzed: denominator-indifference, and ambiguity-generation. They are manifested during the processes of applying Bayes theorem and reasoning by cases. Changes in size, model and assignment of priors do not seem to solve the problems.

All the models with size 8 have been implemented in a PROLOG program and tested against a simple example. The results are consistent with the analysis.

The investigation reveals that under the TPL axioms, *finite* probability models will have limited usefulness. The premise of legal FTOPA is {TPL axioms, finite, totally ordered}. It is believed that TPL axioms represent the necessity of general inference under uncertainty. 'Totally ordered' seems to be necessary for a probability model to be useful. Thus it is conjectured that a useful uncertainty management mechanism can not be realized in a finite setting.

The result of the investigation does not leave the probability theory as the only choice for representation of probable reasoning. But it does highlight infinite totally ordered probability algebras including probability theory. Since the appeal of Bayesian networks for representing uncertain knowledge has become increasingly apparent; and substantial advances have been made in recent years on reasoning algorithms in Bayesian nets, Bayesian networks are adopted as the framework of this research.

Chapter 3

QUALICON: QUALITY CONTROL IN NERVE CONDUCTION STUDIES BY COUPLING BAYESIAN NETWORKS WITH SIGNAL PROCESSING PROCEDURES

Before more complicated diagnostic problem was tackled (Chapter 5), a pilot study on Bayesian networks was done with a problem of quality control in nerve conduction studies. The resultant system is QUALICON. The corresponding network has a small size (less than 20 variables and singly connected) and thus is a good testbed. While the solution to the problem contributes to the information gathering stage of neuromuscular diagnosis.

This chapter presents major issues in the implementation of QUALICON. The results are mainly taken from Xiang et al. [1992]. Section 3.1 introduces the QUALICON domain. Section 3.2 describes the overall structure of QUALICON. Section 3.3 discusses the knowledge representation of QUALICON. Section 3.4 presents a preliminary evaluation of QUALICON.

3.1 The Problem: Quality Control in Nerve Conduction Studies

Nerve conduction studies are now used routinely as part of the electrodiagnostic examination for neuromuscular diagnosis. Their clinical value is demonstrated in the examination of diseases or injuries which might be difficult to diagnose with (needle) EMG alone. The clinical procedures involve the placement of surface electrodes, delivering of stimulating impulse to nerve or muscle, and recording of nerve or muscle responses (action potentials).

The procedures are fairly simple to perform, are easily tolerated, and require little cooperation from the patient. Interpretation of the results, however, requires knowledge of the range of normal values, the morphology of normal potentials and, of equal importance, the sources of technical error which may affect the finding [Goodgold and Eberstein 83]. Since abnormalities can be due to technical error or disease, identification of technical error is a major element of quality control in nerve conduction studies.

Contemporary equipment used for nerve conduction studies is usually capable of computerized measurement of latency, amplitude, duration and area of nerve and muscle action potentials and resulting conduction velocities. However, computerized measurements assume that stimulating and recording characteristics and electrode placements are correct; they do not take cognizance of technical acceptability. The development of an appropriate technique for automating the quality control is addressed in this Chapter.

3.2 QUALICON: a Coupled Expert System in Quality Control

Since identification of technical errors are based on observations of recorded nerve and muscle responses, the problem is of the nature of diagnosis, i.e., given the abnormal outcomes, explaining the cause. Since stimuli are applied to and responses are generated through complex biological systems, namely, human bodies, much uncertainty is involved in the interpretation of abnormal responses in terms of possible technical errors. Thus the problem is in many aspects similar to a medical diagnostic problem, the limitation of conventional techniques and promise of application of AI technique apply as discussed in the beginning of this thesis. Since the recorded responses take the form of continuous signals, it is essential to 'couple' the expert system technique with the numerical procedures for signal processing in order to develop a feasible technique as a solution.

The term 'couple' needs some explanation. The concept of *coupled expert system* emerges, when rule-based systems are dominant, from a need to apply expert system techniques to domains where numerical information is largely involved and is processed by conventional numerical procedures. At that time, expert systems are considered as 'symbolic' since rule-based systems use rules for inference which are different than conventional 'numeric' computation. Therefore, the coupled expert system was defined as a system which links numeric and symbolic computing processes; has some knowledge of the numerical processes embedded; and reasons about the application or results of these numerical processes [Kowalik 86]. In light of newer expert system methodology for probable reasoning, namely Bayesian networks which compute extensively the (numerical) posterior probability using algorithms, it is not appropriate to characterize expert systems which do not deal with numerical information other than measurement of uncertainty as simply 'symbolic'. Thus, my suggestion is to define a coupled expert system as a system which links numerical processes not directly involving inference with computing processes (symbolic or numeric) for inference; has some knowledge of the (non-inferential) numerical processes embedded, and reasons about the application or results of these numerical processes.

A prototype expert system QUALICON coupling Bayesian networks with numerical procedures for signal processing has been developed in the thesis research, which automates quality control in nerve conduction studies. The system was developed in cooperation with Neuromuscular Disease Unit (NDU) of Vancouver General Hospital (VGH). Utilizing information for stimulating and recording parameters for a given nerve or muscle action potential, QUALICON compares specific characteristics with values from a normal database determining the qualitative nature of each feature. Probabilistic reasoning allows QUALICON to provide the user with recommendations on the acceptability of the potential. If it is not technically acceptable, the *most likely* explanation(s) of the problem is/are provided with information on how to correct it.

Figure 3.8 illustrates the 3 modules comprising the framework of QUALICON, namely: the Feature Extraction (FE) module, the Feature Partition (FP) module and Probabilistic Inference Engine (PIE) module. There are also 2 knowledge bases: a Normal Database (ND) and a Bayesian Net Specification (BNS).

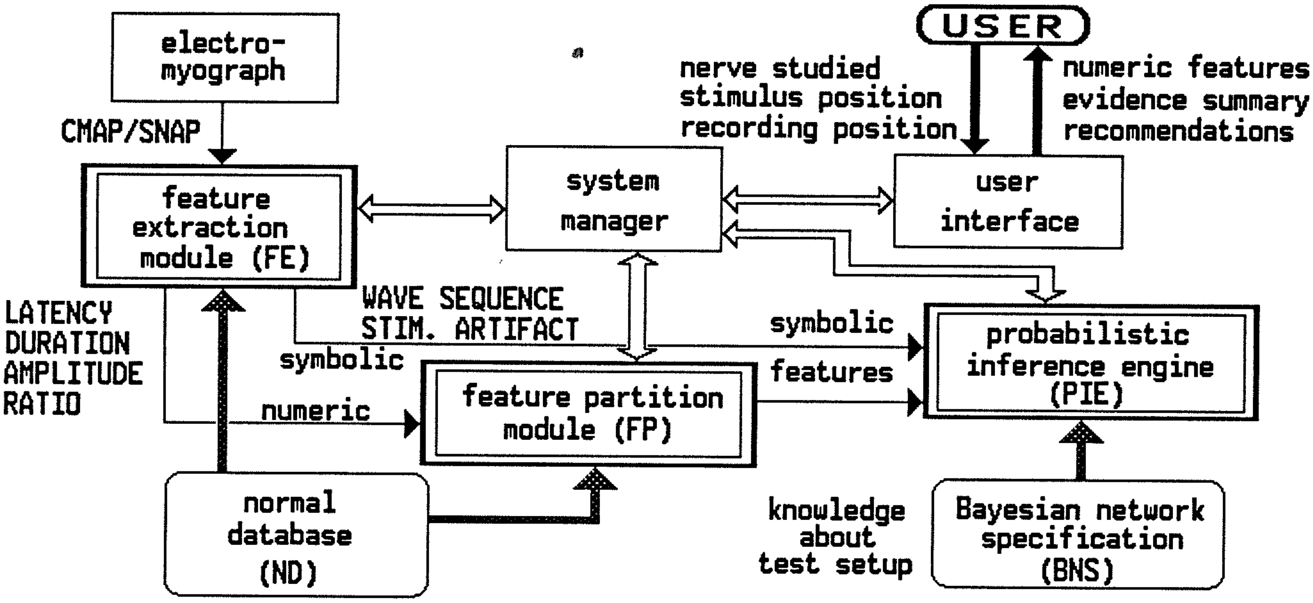


Figure 3.8: QUALICON system structure

FE consists of a set of numerical procedures extracting numeric or symbolic features from an action potential. These routines consult ND in determining their best heuristic strategy. ND contains the normal value ranges (in terms of means and standard deviations) of all the numeric features to be examined. Normal value ranges for *distal* and *proximal* are distinguished. It combines the statistics derived from several hundred normal studies in NDU of VGH, and the electromyographic "expertise" from neuromuscular specialist A. Eisen and registered EMG technician M. MacNeil. FP partitions the numeric features against ND translating them into symbolic descriptions of the potential. When all the features are available in symbolic form they are fed into PIE for probabilistic inference. PIE reasons with BNS as the background knowledge and with the symbolic features as new evidence to generate recommendations with respect to any technical error in test setup.

The FE and FP modules are programmed in C, and the PIE module is programmed in PROLOG.

3.3 Development of QUALICON

Important technical issues involved in the development of QUALICON is presented in this section, which concern knowledge representation, robust feature extraction, coupling, and an algorithm for probabilistic reasoning.

3.3.1 Feature Extraction and Partition

Feature Selection

A set of features of numerical data provides a basic interface between signal processing and Bayesian net components. Two criteria are used in the feature selection. (1) Utilization of human expertise as a resource in building QUALICON requires that the selected features be mainly composed of those used in daily practice. (2) The set of features should have sufficient differential power, i.e. they should enable different potentials to be characterized mostly by different sets of feature values.

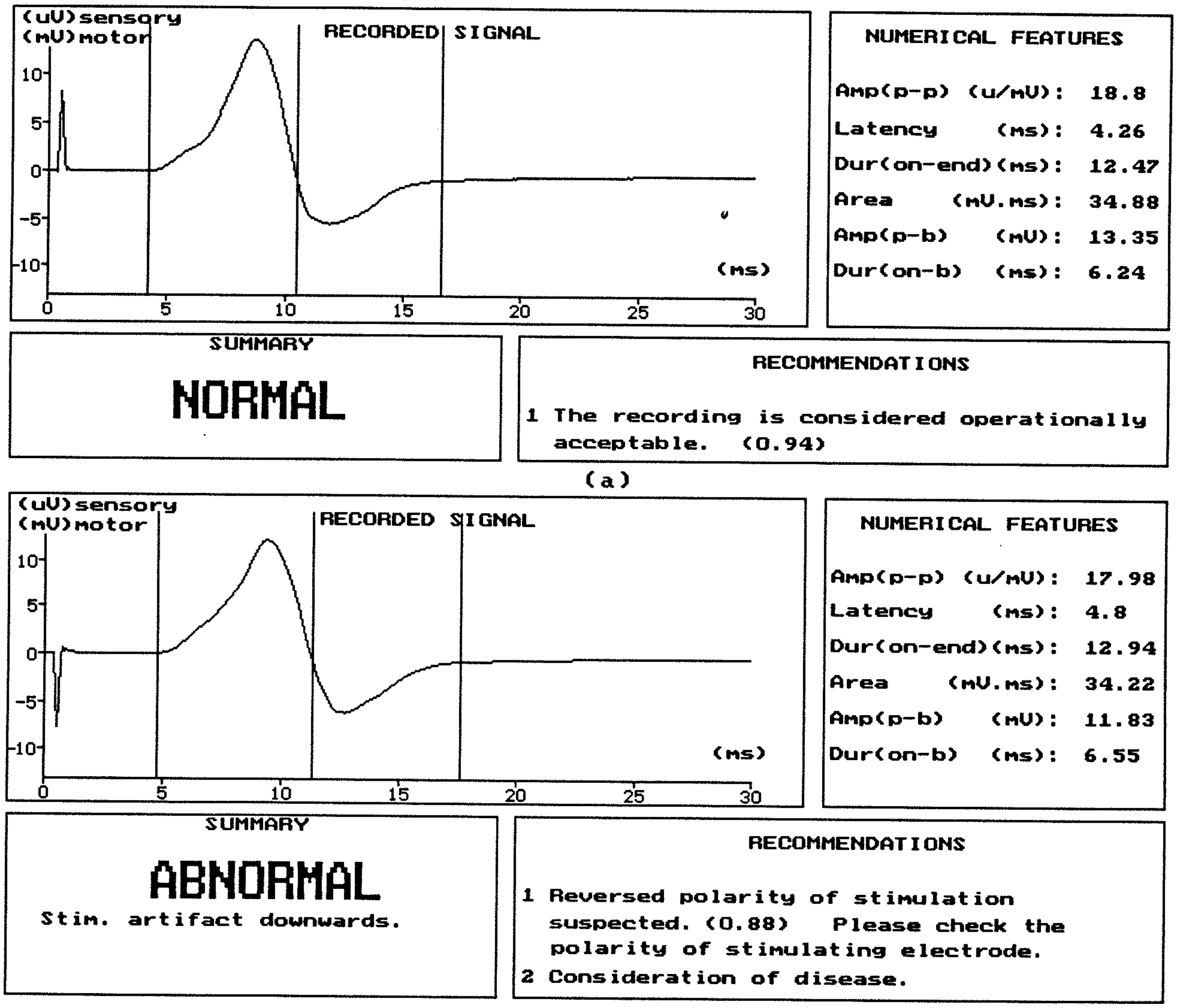
Based on the criteria, 6 variables are chosen: *LATENCY*, *DURATION*, *AMPLITUDE* (peak to peak), *WAVE SEQUENCE*, *RATIO*, and *STIMULATION ARTIFACT*. The first 3 are routinely used features. *WAVE SEQUENCE* is selected to characterize the basic morphology of the potentials. This feature can take 4 symbolic values: *bnpb* (baseline negative positive baseline), *bpnb*, *bpnpb*, and *ab_seq* (abnormal sequence) for CMAPs (Compound Muscle Action Potentials) and *pnp*, *npn*, *np*, and *ab_seq* for SNAPs (Sensory Nerve Action Potentials). *bnpb* (Figure 3.9(a)) and *pnp* are the wave sequences seen in normal potentials; *bpnb* and *npn* (Figure 3.10(a)) are most often created by reversed polarity of recording electrodes; *bpnbp* (Figure 3.10(b)) and *np* usually signify the misplacement of recording electrodes; and *ab_seq* encapsulates any wave sequences different from the above 3.

The feature *RATIO* is selected to complement *WAVE SEQUENCE*. It is defined as the ratio of negative peak amplitude over positive peak amplitude. Although not explicitly used in routine electromyography, this feature is employed implicitly in combination with the basic morphology. In Figure 3.10(b), the ratio is $3.75mV/(4.7-3.75)mV = 3.95$. This is larger than normal and therefore the entry in 'summary': "Positive dip too shallow".

STIMULATION ARTEFACT takes 3 symbolic values: *negative* (upwards), *positive* (downward), and *isoelectric*. An otherwise normal potential except *STIMULATION ARTEFACT* = *positive* could well be produced by reversing the stimulating polarity. The latency will be prolonged but could still be within normal range. Without using this feature, an abnormal setup might be overlooked (Figure 3.9(b)).

Determine values for feature variables

Some features like *STIMULATION ARTEFACT* and *WAVE SEQUENCE* are intrinsically symbolic. Others like *LATENCY* could be extracted in symbolic form from *rough* estimates of numeric values. The quality control task does not logically require any features in numeric form since all the features must be symbolic when entering the final symbolic computation stage, and computation savings could be achieved by abandoning *accurate* numerical measurement. The choice of accurate measurement of *LATENCY*



NUMERICAL FEATURES

Amp(p-p) (u/mV):	18.8
Latency (ms):	4.26
Dur(on-end)(ms):	12.47
Area (mV.ms):	34.88
Amp(p-b) (mV):	13.35
Dur(on-b) (ms):	6.24

SUMMARY

NORMAL

RECOMMENDATIONS

1 The recording is considered operationally acceptable. (0.94)

(a)

NUMERICAL FEATURES

Amp(p-p) (u/mV):	17.98
Latency (ms):	4.8
Dur(on-end)(ms):	12.94
Area (mV.ms):	34.22
Amp(p-b) (mV):	11.83
Dur(on-b) (ms):	6.55

SUMMARY

ABNORMAL

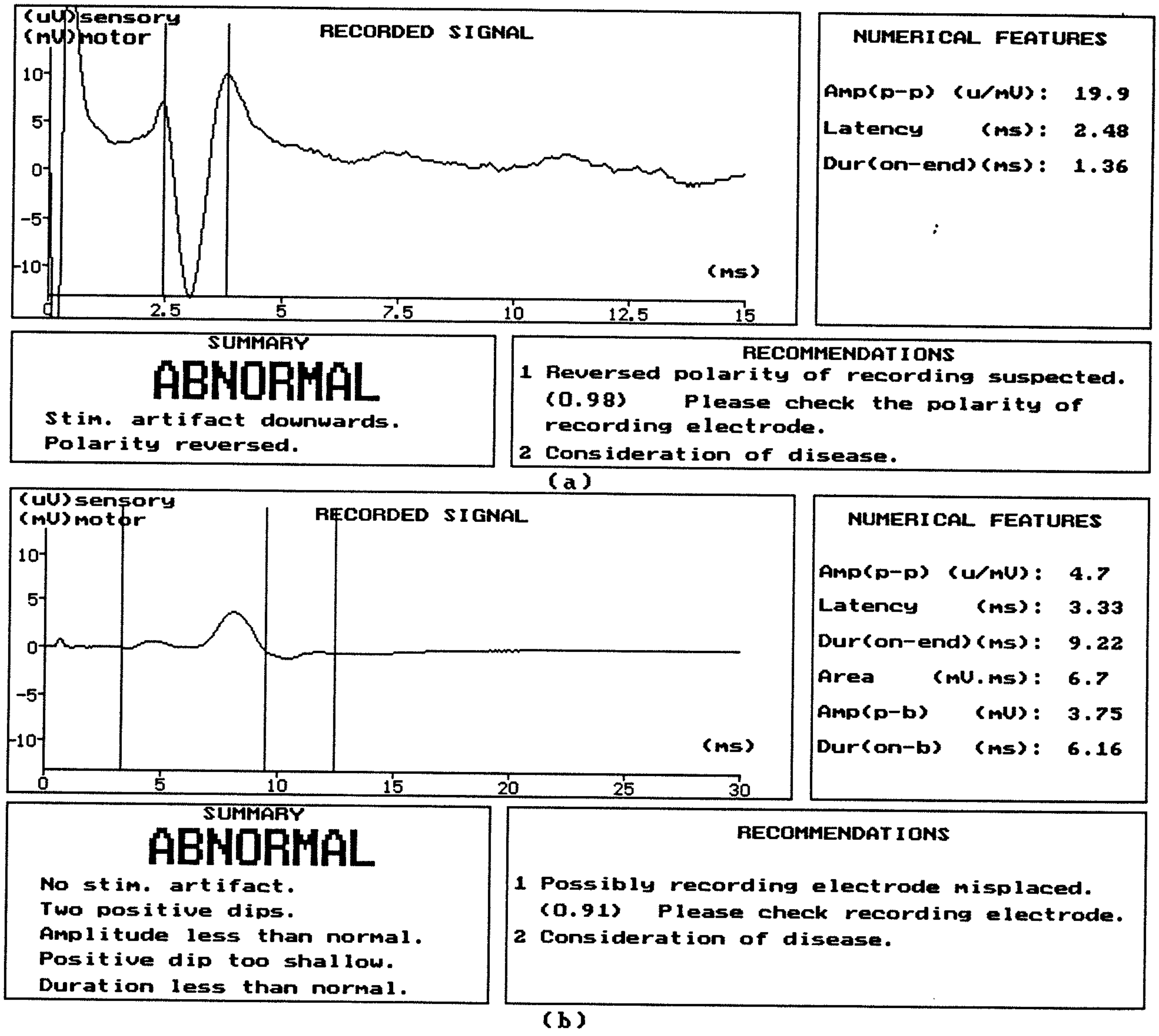
Stim. artifact downwards.

RECOMMENDATIONS

1 Reversed polarity of stimulation suspected. (0.88) Please check the polarity of stimulating electrode.
2 Consideration of disease.

(b)

Figure 3.9: QUALICON report: (a) Normal CMAP evoked by tibial nerve stimulation at ankle. Recorded at abd. hallucis; (b) Same but with a reversed stimulus polarity.



NUMERICAL FEATURES

Amp(p-p) (u/mV): 19.9
Latency (ms): 2.48
Dur(on-end)(ms): 1.36

SUMMARY

ABNORMAL

Stim. artifact downwards.
Polarity reversed.

RECOMMENDATIONS

1 Reversed polarity of recording suspected.
(0.98) Please check the polarity of recording electrode.
2 Consideration of disease.

(a)

NUMERICAL FEATURES

Amp(p-p) (u/mV): 4.7
Latency (ms): 3.33
Dur(on-end)(ms): 9.22
Area (mV.ms): 6.7
Amp(p-b) (mV): 3.75
Dur(on-b) (ms): 6.16

SUMMARY

ABNORMAL

No stim. artifact.
Two positive dips.
Amplitude less than normal.
Positive dip too shallow.
Duration less than normal.

RECOMMENDATIONS

1 Possibly recording electrode misplaced.
(0.91) Please check recording electrode.
2 Consideration of disease.

(b)

Figure 3.10: QUALICON report: (a) Median nerve SNAP recorded at wrist, with stimulation at finger 2 (normal subject). Recording polarity was reversed. (b) The CMAP is recorded from median nerve of a healthy person with stimulation at wrist and with recording electrode misplaced.

(and the other variables) is made in order to present the user with values with which they are familiar. The partition into symbolic values follows normally.

Two methods determining the onset and end of a CMAP are presented by Meyer and Hilfiker [1983]. 'Slope threshold' identifies the 2 critical points when 2% of the maximal slope are reached. 'Power threshold' signals the points corresponding to 0.1 and 0.999 of the integrated squared potential. Even though the methods perform well in 'semiautomatic' condition, they are sensitive to artifacts without human supervision. For the quality control task, a robust strategy is required.

The problem is treated by developing a technique named *guided filtering* which (1) introduces the *knowledge* about the wave morphology and guides the search for critical points; and (2) adopts noise rejection and suppression filtering. Figure 3.11 depicts the feature extraction strategy used by FE module. The following are the important steps employed.

- Detecting stimulation artifacts in distal CMAP. The first few samples are averaged and compared with 2 thresholds to determine the symbolic value of the variable. This approach cannot be applied to SNAPs since, when the stimulation polarity is reversed, only a short positive component appears in the stimulation artefact reflecting the high gain used (Figure 3.12). Thus the following rules are applied to the first T ms samples: (1) if successive positive samples of TP ms are found with their average greater than N1 μV, the *STIMULATION ARTEFACT* takes the value *downward*; (2) if the average over T ms is lower than -N2 μV, the value is *upwards*; (3) otherwise the value is *none*. The parameters T, TP, N1 and N2 depend on the electromyograph used and the sampling interval.

- ND contains the means and 2 standard deviations for *LATENCY* and *DURATION*. The linear combination of these defines a time period $P = [l, u]$ (Figure 3.13). The

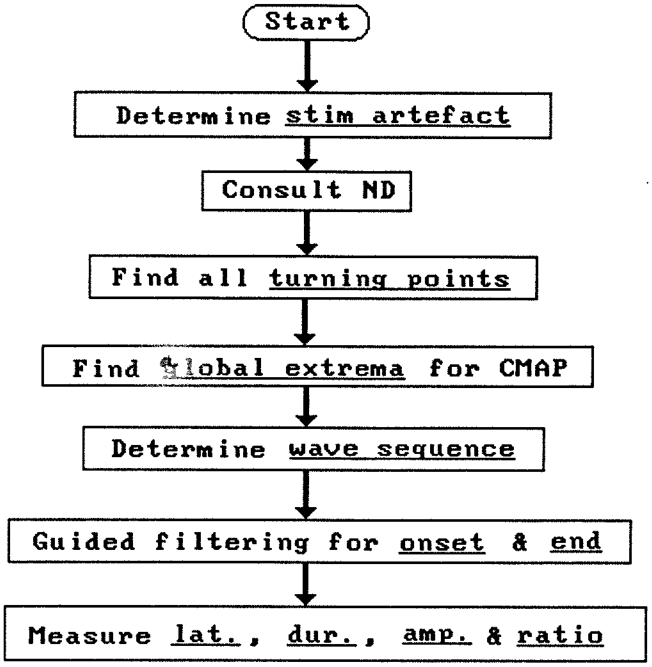

Start
Determine stim artefact
Consult ND
Find all turning points
Find global extrema for CMAP
Determine wave sequence
Guided filtering for onset & end
Measure lat., dur., amp. & ratio

Figure 3.11: Feature extraction strategy

most significant part of the nerve response lies within P with a probability close to 1. Most of the noise and artefact before the onset or after the end are outside P. The values for the remaining features are searched only through the samples in P.

- In determining turning points within P, another level of noise suppression is adopted to catch the turning points which define the basic patterns of wave sequences. Turning points corresponding to small perturbations are ignored. This technique is similar to that developed for EEG processing [Gotman and Gloor 76]. Two successive turning points (one a local maximum and the other minimum) with their difference of amplitudes less than a preset threshold are ignored.

- Once turning points are found, a pattern matching suffices to determine the value for the *WAVE SEQUENCE*.

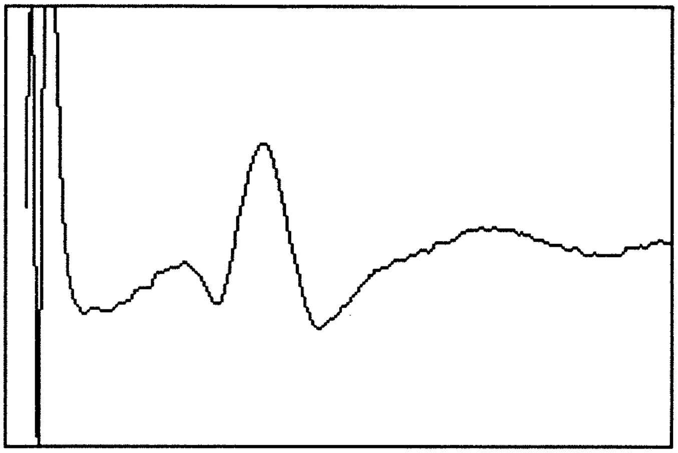


Figure 3.12: SNAP produced by reversing stimulation polarity

- The onset and end of SNAP coincide with turning points. Estimation of the onset and end of CMAP requires additional search. A digital filter (Figure 3.13) is used in the form:
$$y_i = \frac{\alpha + \sum_{k=-(w-1)}^{-1}(x_{i+k} - x_{i+k-1})^2}{\alpha + \sum_{k=1}^{w-1}(x_{i+k} - x_{i+k+1})^2}$$
where x is the sample data, i the time index, $2w + 1$ the window width and α a preset constant. The minimum (maximum) of y indicates the onset (end) of CMAP. α prevents denominator from being 0 when the window slides over flat baseline. The filtering over the window width smooths any small perturbation, and provides another level of noise suppression. Increasing the window width strengthens the noise suppression but decreases the sensitivity of the filter. The filtering is only applied to the period $[l, f(y_i)]$ for onset estimation. $f(y_i)$ is initialized to j the time index of the turning point corresponding to the negative peak of CMAP. When y_i drops below a threshold which signifies the onset is nearby, $f(y_i)$ takes the value $i + \beta$ where β is a preset constant.

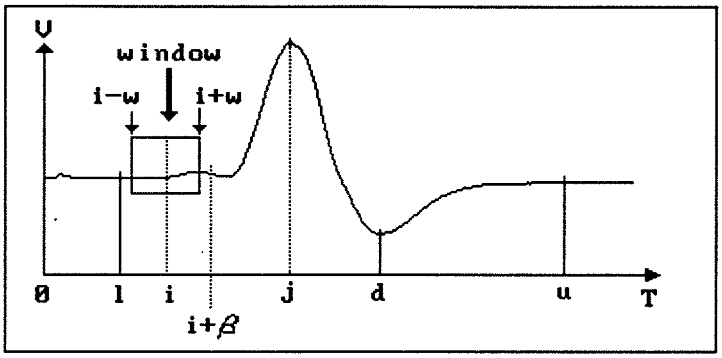


Figure 3.13: Illustration of Guided Filtering

For estimation of the end, the filtering is applied to $[d, d+\gamma]$ $(d+\gamma \leq u)$ where d is the time index of the turning point corresponding to the positive peak of CMAP, and γ is a preset constant determined by the estimation of the maximum possible length between d and the end.

In summary, this technique, named *guided filtering*, attempts to utilize fully the available knowledge and information up to the moment during its search process. Initially, the general knowledge about the nerve-stimulation pair guides the search. As the search progress, new information about turning points and wave sequence are absorbed to localize further search for the onset and end. It simulates human heuristic which starts with some expectation and search is guided by information obtained during the search. In this way, not only the overall processing is very efficient, but since in each step the search is kept local, the effect of artefact and noise is minimized.

Partitioning Numeric Features

Partition is also a feature interpretation process. It involves (1) straightforward mapping of numeric features against the normal database ND, for example, into *less_than_normal*, *normal* and *greater_than_normal*; and (2) capturing the dependency in feature interpretation. Two kinds of dependency required consideration in partition. The first is technically obvious. For instance, if *WAVE SEQUENCE* = *ab_seq*, no values could be assigned to the variables *AMPLITUDE*, *DURATION* etc. In this case, call these variables *inapplicable*.

Another kind of dependency reflects the experts' context dependent view of numeric features. For example, in the case of CMAP if the stimulus is applied proximally, the possibility of *LATENCY* = *less_than_normal* will not be considered. This is because individual variation in limb length precludes a standard interstimulus distance. Likewise when *AMPLITUDE* = *normal*, even if the *DURATION* is below normal range, it is still interpreted as *normal*. This is probably due to inability to accurately determine the end of the potential (where it returns to baseline). In both examples, the dimension of certain variables changes depending on the context.

3.3.2 Probabilistic Reasoning

Construction of Bayesian Networks

Due to the difference in nerve conduction studies with respect to CMAPs and SNAPs. Two separate Bayesian nets with similar topology are constructed for CMAPs and SNAPs respectively, and only one of them needs to be evaluated at a particular session. The net for CMAP is shown in Figure 3.14. Each box (a node) in the network represents a variable with its name underlined and its alternative values listed. The root is *STIMULUS POSITION* which affects the distribution of *OPERATION* which in turn determines the likelihood of observing different feature values.

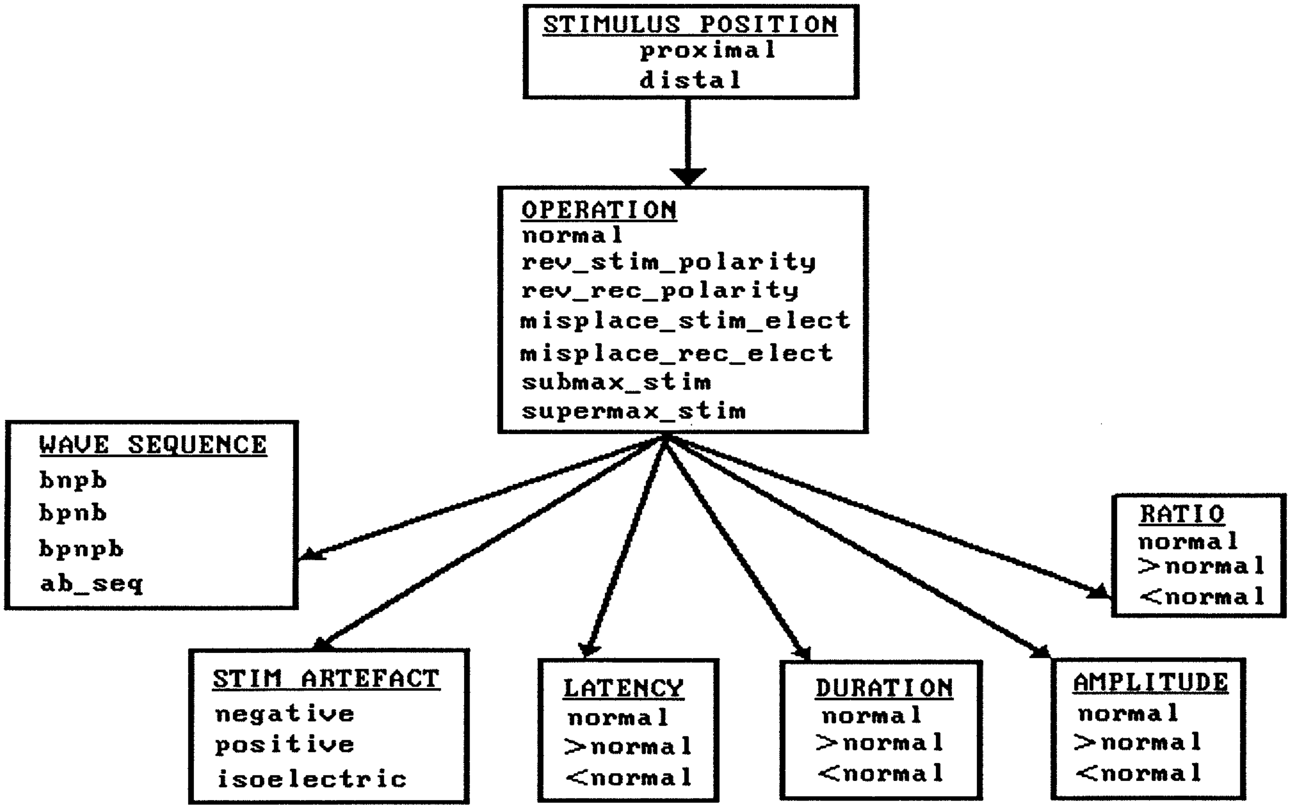


Figure 3.14: Bayesian subset for CMAP

A system dealing with technical errors must make assumptions about the knowledge and skill levels of potential users. QUALICON has intended users as residents, fellows and technologists learning in a hospital EMG laboratory. It is further assumed that the inadequate operations are mutually exclusive since it is unlikely that 2 or more could occur in combination given the intended user level. Assuming mutual exclusion, different technical errors can be represented by a single node *OPERATION*. In this way simplicity in the network structure is gained and an efficient evaluation algorithm (section 3.3.2) can be developed.

After the topology of the net is decided, the conditional probabilities for all links are acquired from subjective judgement of a domain expert in the form like *p(WAVE SEQUENCE = bpnb | OPERATION = reversed recording polarity)*.

Although not constructed for diagnostic purposes, QUALICON recognizes that abnormal potentials could result from disease rather than a technical error. When an abnormal potential is encountered QUALICON attempts to interpret the abnormality in terms of a technical error but also raises the possibility of disease (e.g. recommendation 2 in Figure 3.10). If an unacceptable recording could not be overcome by correcting the identified technical error then disease becomes the likely cause.

The net for CMAP shown in Figure 3.14 reflects the influence among relevant variables in the most general case. When the network is used in particular situation, it must adapt to the reality. For instance, when *STIMULUS POSITION = proximal*, *STIM ARTEFACT* usually can no longer be detected. QUALICON will remove the node *STIM ARTEFACT* together with its relation with *OPERATION* in this case. Likewise, when *WAVE SEQUENCE* is found to be *ab_seq*, *DURATION*, *AMPLITUDE*, and *RATIO* will be removed. This is a higher level parallel of *inapplicability* raised in the previous subsection. Heckerman, Horvitz, and Nathwani [1989] discuss this similar issue.

Inference Algorithm in QUALICON

In a QUALICON session, after the feature extraction and partition are finished, QUALICON is able to reason at the Bayesian net level where *STIMULUS POSITION* (Figure 3.14) and all other applicable feature variables are known. At this time, only the value of *OPERATION* is to be estimated (i.e., given the evidence, only the probabilities of *OPERATION* are to be evaluated). A simple and efficient algorithm to take advantage of this net structure is developed. With the algorithm, only the probability of one alternative value needs to be calculated in full length; the rest can be obtained by a negligible amount of computation. Figure 3.15 depicts the DAG used in QUALICON which is slightly simplified for illustration of the algorithm.

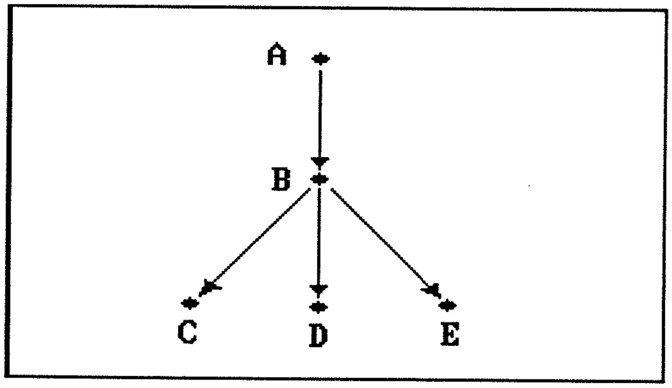


Figure 3.15: A DAG to illustrate the inference algorithm in QUALICON

Algorithm 1 *Suppose a set of evidence is given as $\{a, c, d\}$ with the value of E unknown. Assuming $B = \{b_1, \ldots, b_n\}$, the probability $p(b_i|a\&c\&d)$ $(i = 1, \ldots, n)$ can be computed in the following way.*

step 1 *For* $p(b_1|a\&c\&d) = p(b_1\&c\&d|a)/p(c\&d|a)$, *retrieve the knowledge base to calculate*

$$p(b_i\&c\&d|a) = p(c|b_i)p(d|b_i)p(b_i|a) \ \ (i = 1, \ldots, n)$$

$$p(c\&d|a) = \sum_{i=1}^{n} p(b_i\&c\&d|a)$$

and cache the intermediate results.

step 2 *For the rest of the probabilities to be evaluated, no knowledge base retrieval is needed at all. Fetch the cache content to obtain*

$$p(b_i|a\&c\&d) = p(b_i\&c\&d|a)/p(c\&d|a)$$

It can be easily derived that in the case there are m known children of B ($m = 2$ is illustrated above), the algorithm requires mn multiplication, n division, and $n - 1$ addition. That is, the time complexity of the algorithm is linear to the number of features and number of technical errors considered.

Out of the 7 alternative values of *OPERATION*, those with their posterior probabilities above 1/7 (*above-equal-likelihood*) are chosen as the basis of the recommendations for the user. The probabilities are attached to the corresponding recommendations (Figure 3.9, 3.10).

3.4 Assessment of QUALICON

3.4.1 Assessment Procedure

In order to test the validity of QUALICON, 84 muscle and nerve action potentials are elicited and recorded from *normal* volunteers using standard procedures (see Table 3.2). For recording compound muscle and nerve action potentials, evoked by either proximal or distal stimulation, filter settings are fixed at 2 Hz to 10 kHz and 20 Hz to 2 kHz, respectively. Five different types of technical error are introduced: reversal of stimulating

muscle or nerve	*recording condition*					
	normal	sti_rev	rec_rev	sti_mis	rec_mis	sub_max
median motor distal	2	1	2	2	2	1
median motor proximal	2	2		1		
median sensory	2	6	4		3	1
ulnar motor distal	2	1	2	2	2	1
ulnar motor proximal	2	1		2		1
ulnar sensory	3	1	2		4	2
post. tib. motor distal	2	2	2		3	1
post. tib. motor proximal		1		1		2
sural sensory	2	2	2	1	2	2
subtotal	17	17	14	9	16	11

Table 3.2: Potential recording conditions.
Technical errors introduced include reversal of stimulating (sti_rev) or recording (rec_rev) polarity, misplacement of stimulating (sti_mis) or recording (rec_mis) electrodes, and use of submaximal stimulating current (sub_sti). Empty entry: no potential recorded under corresponding condition.

(Sti_rev) or recording (Rec_rev) polarity, misplacement of stimulating (Sti_mis) or recording (Rec_mis) electrodes, and use of submaximal stimulating current (sub_max). Only 1 error is introduced at a time. *No* distribution about errors introduced is assumed. In analyzing the potentials QUALICON is blinded as to whether an error has been introduced or not. Two physicians, 1 technician and 1 resident also manually analyze the data. In doing so they are given information as to the nerve stimulated, stimulating and recording sites, and are allowed to select from: Normal, Sti_rev, Rev_rev, Stim_mis, Rec_mis, and Sub_max in making their interpretations. Multiple options of up to 3 are allowed when it is difficult to make a single choice. If an option matches the actual technical error, the interpretation is recognized as a *success*. The *success rate* is defined as the number of successes/the total number of interpretations made.

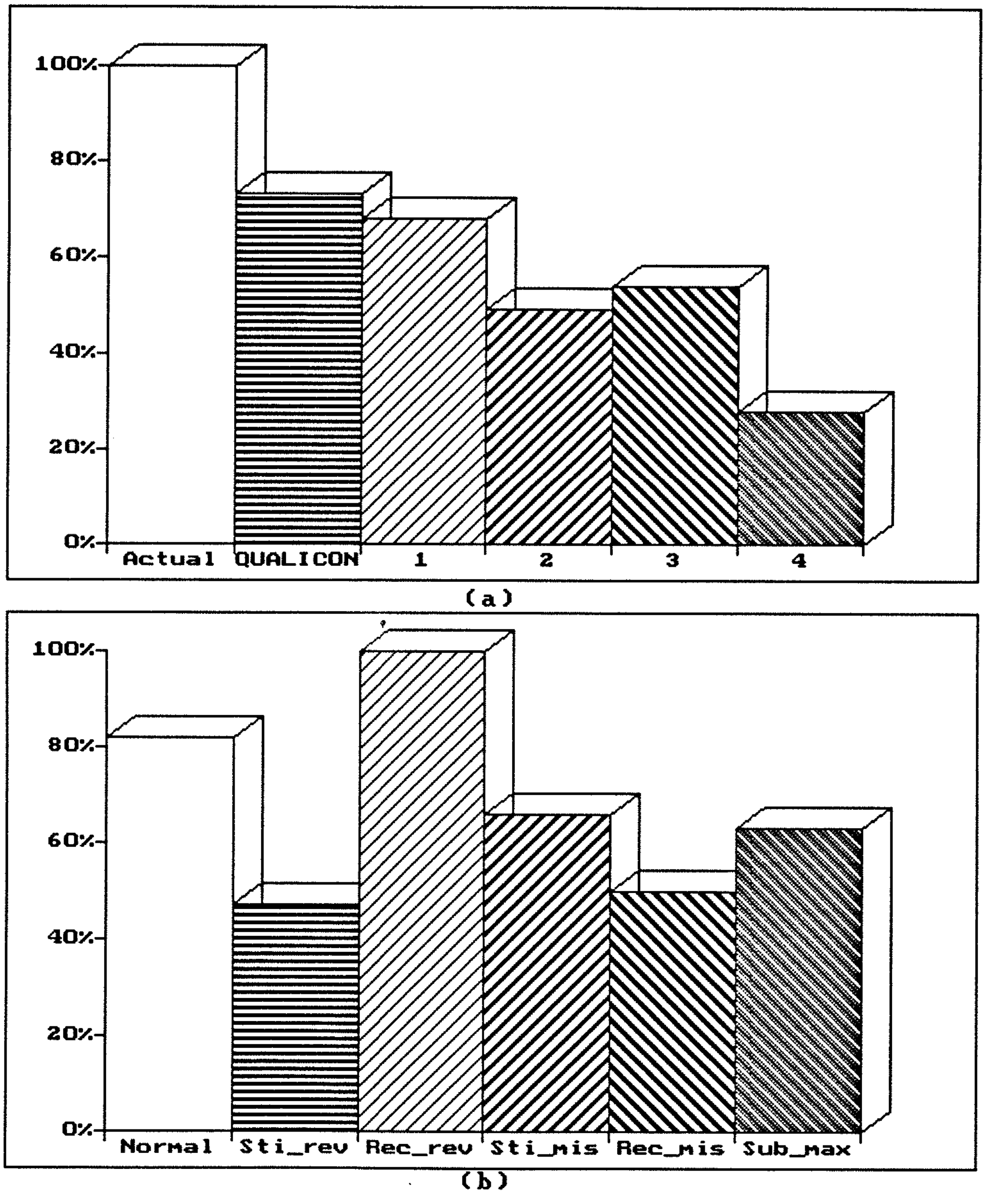
3.4.2 Assessment Results

The assessment is treated as Bernoulli trials (Appendix E). The success rate of QUALICON is 73% compared to average success rate 57% for manual assessment (excluding assessor 4 who is a resident in training) (Figure 3.16(a)). Given the result, it can be derived, using the standard statistic technique (Appendix E) that the 95% confidence intervals of success probabilities for QUALICON (p_Q) and 3 human average (p_h) are (0.62, 0.82) and (0.46, 0.68), respectively. Further more, the hypothesis $H_0 : p_Q = p_h$ (as opposed to $H_1 : p_Q > p_h$) is accepted at 0.01 level of significance but rejected at 0.05 level of significance. Thus it can be safely concluded that *QUALICON performed as well as, or better than human assessment did on the task.*

One would have noticed that the success rate (both QUALICON and human) is not high. The average inter-human agreement rate for the different, imposed, errors is depicted in Figure 3.16(b). Agreement is high for recognition of a normal potential and one in which the recording electrode is inverted. There is much poorer agreement between individuals for the other technical errors reflecting the limited amount of information that is provided which would have been necessary. In daily practice, a series of potentials are usually recorded on a nerve or muscle. Comparison among them is an important clue for detecting technical errors. This suggests the use of multiple potentials in a group in detecting errors in the future extension of QUALICON.

3.5 Remarks

The experienced electromyographer(EMGer) or technologist can usually detect a technical error as a cause of an abnormal potential rapidly and with ease. Doing so is a vital element in the quality control of electromyography. Contemporary equipment automates much of the requisite measurements needed, for example, in performing nerve conduction



(a)

(b)

Figure 3.16: (a) Manual success rates of human assessors compared to QUALICON. (b) Average agreement rate for manual assessment. (this excludes the assessor 4 who is a resident in training)

studies but does not take cognizance of error as a cause of abnormality. Artificial intelligence offers a possible solution to this problem as demonstrated through the coupled knowledge based prototype system QUALICON.

QUALICON is not the first attempt in computerized quality control in nerve conduction studies. With conventional signal processing and database techniques, deviations of features from reference values are used [Stalberg and Stalberg 89] to turn on a warning display. MUNIN [Andreassen et al. 89], currently under development by large research groups, is one system which gives a test guide in electromyography. MUNIN displays where the electrodes should be placed and other test setup information but does not check for technical errors. QUALICON is unique in that it tries to pinpoint what is the most likely technical error given the recorded abnormal potential.

The intended user of QUALICON are residents, fellows and technologists learning in a hospital EMG laboratory. Given the level of knowledge and experience in nerve conduction studies, QUALICON only considers the most common technical errors in this context. Also, development of appropriate technique rather than a complete system is emphasized in this work. Thus 6 kinds of technical errors are considered in QUALICON including reversal of stimulating or recording polarity, misplacement of stimulating or recording electrodes, and use of submaximal or supermaximal stimulating current.

QUALICON's task is to recognize abnormal potentials due to technical errors. An abnormal potential due to disease is defined, from QUALICON's point of view, as an abnormal potential which is technically correction-resistant. *No* attempt is made for QUALICON to differentiate among disease states. With the success in applying Bayesian network technique for recognition of technical errors the next step of my thesis research is to extend the technique to disease diagnosis.

Currently, QUALICON supports quality control for nerve conduction studies derived from median and ulnar, motor and sensory nerve, posterior tibial motor nerve and sural sensory nerve conductions. The CMAPs and SNAPs used by QUALICON can be acquired from any electromyograph capable of producing digitized potentials. With efficient signal processing and inference algorithms implemented for QUALICON, it takes 6 seconds to evaluate a potential in an IBM AT compatible. The short processing time with such a small computer means that QUALICON can be easily included in a computerized electromyograph.

In an assessment using 84 potentials, QUALICON's performance is compared with human professionals. QUALICON's 95% confidence interval of success probability is assessed as (0.62, 0.82), with corresponding interval for average human professional as (0.46, 0.68). The assessment suggests QUALICON performs as well as human professional with the given task. It is currently used as a teaching instrument at NDU of VGH.

Future extension to QUALICON can be expected in supporting other nerves routinely used in nerve conduction studies; and strengthening the knowledge base to include such factors as the effect of age. Although QUALICON detects only 6 types of technical errors, its success suggests that a more sophisticated system can be built using the same approach to deal with broader technical errors.

Chapter 4

MULTIPLY SECTIONED BAYESIAN NETWORKS AND JUNCTION FORESTS FOR LARGE EXPERT SYSTEMS

With QUALICON's performance being satisfactory, the neuromuscular diagnosis involving a painful or impaired upper limb was tackled using Bayesian networks. This is the PAINULIM project. Two major problems arose.

First, the PAINULIM project has a large domain. The Bayesian network representation contains 83 variables including 14 diseases and 69 features each of which has up to three possible outcomes. The network is multiply connected and has 271 arcs and 6795 probability values. When transformed into a junction tree representation (to be detailed below), the system contains 10608 belief values. To process the problem of this complexity with a reasonable system response time demands powerful computing equipment.

On the other hand, the tight schedule of the medical staff demands that the knowledge acquisition (including initial network construction, subsequent system testing and refinement) should be conducted within the hospital environment where most computing equipment consists of small personal computers.

These two conflicting demands motivates the development of a technique to reduce the computational complexity. The result is the general technique of Multiply Sectioned Bayesian Networks (MSBNs) and junction forests presented in this chapter. The MSBN technique is an extension to the d-separation concept [Pearl 88] and the junction tree technique [Andersen et al. 89, Jensen, Lauritzen and Olesen 90]. The application of MSBN to PAINULIM is presented in chapter 5.

Section 4.1 introduces an important observation called *localization* which is to be associated with large application domains. MSBN exploits localization. Section 4.2 reviews background research, particularly, the d-separation concept [Pearl 88] and the junction tree technique [Jensen, Olesen and Andersen 90, Jensen, Lauritzen and Olesen 90] [Andersen et al. 89] on which the MSBN technique is based. Section 4.3 explains why 'obvious' solutions to exploit localization do not work, and Section 4.4 gives an overview of the MSBN technique. These two sections serve to motivate and guide readers into the subsequent sections which present the mathematical theory necessary for the technique.

4.1 Localization

As Cooper (1990) has shown, probabilistic inference in a general Bayesian net is NP-hard. Several approaches have been pursued to reduce the computational complexity of inference in Bayesian nets; these are reviewed in section 1.3.4. Here the *exact* methods in section 1.3.4 are restated briefly and the problems that remain are discussed.

Efficient algorithms have been developed for inference in Bayesian nets with special topologies [Pearl 86, Heckerman 90a]. Unfortunately many domain models can not be represented by these special types of Bayesian nets. For sparse nets, Lauritzen and Spiegelhalter [1988] have created a secondary directed tree structure to achieve efficient computation. Jensen, Lauritzen and Olesen [1990] and Shafer and Shenoy [1988] have created an undirected tree structure. Creating secondary structures offers also the advantage of trading compile time (the computation time spent in transforming a Bayesian net into a secondary structure) with run time (the computation time spent in inference). This advantage is particularly relevant to reusable systems (e.g., expert systems).

However, for large applications, the run time overhead (both space and time) is still forbidding. Pruning Bayesian nets with respect to each query instance is yet another exact method with savings in computational complexity [Baker and Boult 90]. A portion S of a Bayesian net may not be relevant given a set of evidence and a set of queries. It can be pruned away before computation. In light of a piece of new evidence, S may become relevant but can not be restored within the pruning algorithm. If networks have to be pruned for each set of queries, the advantage of trading compile time with run time will also be lost. The MSBN technique can be viewed as a way to retain the advantage of the pruning algorithm, but to overcome its limitations.

This chapter addresses domains representable by general but sparse networks and characterized by incremental evidence. It addresses *reusable* systems where the probabilistic knowledge can be captured once and be used for multiple cases. Current Bayesian net representations do not consider structure in the domain and lump all variables into a *homogeneous* network. For small applications, this is appropriate. But for a large application domain where evidence arrives incrementally, in practice one often directs attention to only part of the network within a period of time, i.e., there is 'localization' of queries and evidence. More precisely, 'localization' means 2 things. For an average query session, first, only certain parts of a large network are interesting[1]; second, *new* evidence and queries are directed to a small part of a large network repeatedly within a period of time. When this is the case, the homogeneous network is inefficient since newly arrived evidence has to be propagated to the overall network before queries can be answered.

[1] 'Interesting' is more restrictive than 'relevant'. One may not be interested in something even though it is relevant.

The following observation of the PAINULIM domain based on the practice in Neuromuscular Disease Unit (NDU), Vancouver General Hospital (VGH) illustrates localization.

A neurologist, examining a patient with a painful impaired upper limb, may temporarily consider only his findings' implication on a set of diseases candidates. He may not consider the diagnostic significance of each available laboratory test until he has finished the clinical examination. After each clinical finding (about 5 findings on an average patient), he dynamically changes his choice of the most likely disease candidates. Based on this he chooses the best question to ask the patient next or the best examination to perform on the patient next. After the clinical examination of the patient, findings highlight certain disease candidates and make others less likely, which may suggest that further nerve conduction studies are of no help at all, while (needle) EMG tests are diagnostically beneficial (about 60% of patients have only EMG tests, and about 27% of patients have only nerve conduction studies). Since EMG tests are usually not comfortable for the patients, the neurologist would not perform a test unless it is diagnostically necessary. Thus he would perform each test depending on results in previous ones (about 6 EMG tests performed on an average patient, and 4 for nerve conduction studies).

The above scenario and above statistics illustrate both aspects of localization. During the clinical examination, only clinical findings and disease candidates are of current interests to the neurologist; and during EMG tests, only EMG test results and their implications on a subset of the diseases are under the neurologist's attention; furthermore, for a large percentage (87%) of patients, only one of EMG or nerve conduction studies is required. If the neurologist is assisted by a Bayesian network based system, the evidence and queries would repeatedly (about 5 times) involve a 'small' part of the network during each diagnostic period (the clinical or the EMG). For 87% of patients certain parts of the network (either the EMG or the nerve conduction) may not be of interest at all. If the Bayesian net representation is homogeneous, each batch of clinical findings has to be propagated to all the EMG and nerve conduction variables which are not *relevant* at the moment. Likewise, after each batch of EMG tests, the overall net has to be updated even though the neurologist is only *interested* in planning the next EMG test.

A large application domain can often be partitioned naturally in terms of localization. In the PAINULIM domain, a natural subdomain can be formed from the knowledge about the clinical symptoms and a set of diseases. Another natural subdomain can be formed from the knowledge about EMG results and a subset of diseases. The third natural subdomain can be formed from the knowledge about nerve conduction study results and a different subset of diseases. One problem of current Bayesian net representations is that they do not provide means to distinguish variables according to natural subdomains. Heckerman [1990b] partitions Bayesian nets into small groups of naturally related variables to ease the construction of large networks. But once the construction is finished, the run time representation is still homogeneous.

If groups of naturally related variables in a domain can be identified and represented, the run time computation can be restricted to one group at any given stage of a query session due to localization. In particular, one may not need to propagate new evidence beyond the current group. Along with the arrival of new evidence, attention can be shifted from one group to another. Chunks of knowledge not required with respect to current attention remain inactive (not being thrown away) until they are activated. This way, the *run time* overhead is governed by the size of the group of naturally related variables, not the size of application domain. Computational savings can be achieved. As demonstrated by Heckerman [1990b], grouping of variables can also help in ease and accuracy in construction of Bayesian networks.

Partitioning a large domain into separate knowledge bases and coordinating them in problem solving have a history in rule-based expert systems termed *blackboard architectures* [Nii 86a, Nii 86b]. However a proper parallel for Bayesian network technology has not yet appeared.

This chapter derives constraints which can often be satisfied such that a natural (localization preserving) partition of a domain and its representation by separate Bayesian subnets are possible. Such a representation is termed a *multiply sectioned Bayesian network* (MSBN). In order to perform efficient evidential reasoning in a sparse network, the set of subnets are transformed into a set of junction trees as a secondary representation which is termed a *junction forest.* The junction forest becomes the permanent representation for the reusable system in which incremental evidential reasoning takes place. Since the junction trees preserve localization, each of them stands as a computational object which can be used alone during reasoning. Multiple linkages between the junction trees are introduced to allow evidence acquired from previously active junction trees to be absorbed into a newly active junction tree of current interest. In this way, localization naturally existing in the domain can be exploited and the above illustrated idea is realized.

4.2 Background

This section reviews the background research particularly related to the MSBN technique which is not included in section 1.3.

4.2.1 Operations on Belief Tables

The following introduces the belief table representation of probability distribution on a set of variables, and the mathematical operations on belief tables.

A *belief table* [Andersen et al. 89, Jensen, Olesen and Andersen 90], or *potential*

[Lauritzen and Spiegelhalter 88] denoted as $B()$ is a non-normalized probability distribution. It can be viewed as a function from the space of a set of variables to the reals. For example, the belief table $B(X)$ of a set X of variables maps $\Psi(X)$ (the space of X defined in section 1.3.3) to the reals. If $\mathbf{x} \in \Psi(X)$, the belief value of $\mathbf{x}$ is denoted by $B(\mathbf{x})$. Denote a set X of variables and corresponding belief table $B(X)$ with an ordered pair $(X, B(X))$ and call the pair a *world*.

For $Y \subseteq X$, the *projection* $\mathbf{y} \in \Psi(Y)$ of $\mathbf{x} \in \Psi(X)$ to the space $\Psi(Y)$ is denoted as $Prj_{\Psi(Y)}(\mathbf{x})$. Denote the *marginalization* of $B(X)$ to $Y \subseteq X$ by $\sum_{X \setminus Y} B(X)$ which specifies a belief table on Y. The operation is defined as the following. If $B(Y) = \sum_{X \setminus Y} B(X)$, then for all $\mathbf{y} \in \Psi(Y)$,

$$B(\mathbf{y}) = \sum_{Prj_{\Psi(Y)}(\mathbf{x})=\mathbf{y}} B(\mathbf{x}).$$

Similarly denote the *multiplication* of $B(X)$ and $B(Y)$ by $B(X) * B(Y)$ which specifies a belief table on $X \cup Y$. If $B(X \cup Y) = B(X) * B(Y)$, then for all $\mathbf{z} \in \Psi(X \cup Y)$, $B(\mathbf{z}) = B(\mathbf{x}) * B(\mathbf{y})$ where $\mathbf{x} = Prj_{\Psi(X)}(\mathbf{z})$ and $\mathbf{y} = Prj_{\Psi(Y)}(\mathbf{z})$. Denote the *division* of $B(X)$ over $B(Y)$ by $B(X)/B(Y)$ which specifies a belief table on $X \cup Y$. If $B(X \cup Y) = B(X)/B(Y)$, then for all $\mathbf{z} \in \Psi(X \cup Y)$, $B(\mathbf{z}) = B(\mathbf{x})/B(\mathbf{y})$ if $B(\mathbf{y}) \neq 0$ where $\mathbf{x} = Prj_{\Psi(X)}(\mathbf{z})$ and $\mathbf{y} = Prj_{\Psi(Y)}(\mathbf{z})$.

4.2.2 Transform a Bayesian Net into a Junction Tree

The MSBN technique is an extension to the junction tree technique [Andersen et al. 89, Jensen, Lauritzen and Olesen 90] which transforms a Bayesian net into an equivalent secondary structure where inference is conducted (Figure 4.19). Because of this restructuring, belief propagation in multiply connected Bayesian nets can be performed in a similar manner as can in singly connected nets. The following briefly summarizes the junction tree technique. Readers not familiar with the terminologies should refer to

Appendix A.

Moralization Transform the DAG into its moral graph.

Triangulation Triangulate the moral graph. Call the resultant graph a *morali-triangulated* graph.

Clique hypergraph formation Identify cliques of the morali-triangulated graph and obtain a clique hypergraph.

Junction tree construction Organize the clique hypergraph into a junction tree of cliques.

Node assignment Assign each node in the DAG to a clique in the junction tree of cliques.

Belief universes formation For each clique C_i in the junction tree of cliques, obtain its belief table $B(C_i)$ by multiplication of the conditional probability tables of its assigned nodes. Call the $(C_i, B(C_i))$ a *belief universe.* When it is clear from the context no distinction is made between a junction tree of cliques and a junction tree of belief universes.

Inference is conducted through the junction tree representation. An *absorption* operation is defined for local belief propagation. Global belief propagation is achieved by a forward propagation procedure *DistributeEvidence* and a backward propagation procedure *CollectEvidence.*

Belief initialization Before any evidence can be entered to the junction tree, the belief tables are made *consistent* by CollectEvidence and DistributeEvidence such that the *prior* marginal probability for a variable (of the original Bayesian net) can be obtained by marginalization of the belief table in any universe which contains it.

Evidential reasoning When evidence about a set of variables (of the original Bayesian net) is available, the evidence is entered to universes which contain the variables. Then the belief tables of the junction tree are made consistent again by CollectEvidence and DistributeEvidence such that the *posterior* marginal probability for a variable can be obtained from any universe containing the variable.

The computational complexity of evidential reasoning in junction trees is about the same as the reasoning method by Lauritzen and Spiegelhalter [1988] which can be viewed as performed on a (secondary) directed tree structure [Shachter 88, Neapolitan 90]. But junction trees are undirected and allow more flexible computation. The junction tree representation is explored in this chapter since its flexibility is of crucial importance to the MSBN extension.

4.2.3 d-separation

The concept of d-separation introduced by Pearl (1988, page 116-118) is fundamental in probabilistic reasoning in Bayesian networks. It permits easy determination, by inspection, of which sets of variables are considered independent of each other given a third set, thus making any DAG an unambiguous representation of dependence and independence. It plays an important role in our partitioning of Bayesian networks.

Definition 5 (d-separate [Pearl 88]) *If X, Y, and Z are 3 disjoint subsets of nodes in a DAG, then Z is said to* **d-separate** *X from Y, if there is no path between a node in X and a node in Y along which 2 conditions hold: (1) every node with converging arcs (***head-to-head node***) is in Z or has a descendent in Z and (2) every other node (***non-head-to-head node***) is outside Z.*

A path satisfying the conditions above is said to be active*; otherwise it is said to be* blocked *by Z.*

For example, in the DAG Θ of Figure 4.17, $\{F_1\}$ d-separates $\{F_2\}$ from $\{H_1, H_2\}$. $\{H_2, H_3, H_4\}$ d-separates $\{E_1, E_2, E_3, E_4\}$ from the rest. The path between A_3 and E_4 is blocked by H_4.

The importance of d-separation lies in that in a Bayesian network X and Y are conditionally independent given Z iff Z d-separates X from Y [Geiger, Verma and Pearl 90].

4.3 'Obvious' Ways to Explore Localization

Recall that, the 'localization' means (1) for an average query session, only certain parts of a large network are interesting; and (2) *new* evidence and queries are directed to small part of a large network repeatedly within a period of time. An obvious way to explore localization in multiply connected networks is to preserve localization within subtrees of a junction tree by clever choice in triangulation and junction tree construction. If this can be done, the junction tree can be split and each subtree can be used as a separate computational object. The following example shows that this is not always a workable solution.

Consider the DAG Θ in Figure 4.17. Suppose variables in the DAG form three groups naturally related which satisfy localization:

$$\begin{aligned} G_1 &= \{A_1, A_2, A_3, A_4, H_1, H_2, H_3, H_4\} \\ G_2 &= \{F_1, F_2, H_1, H_2\} \\ G_3 &= \{E_1, E_2, E_3, E_4, H_2, H_3, H_4\} \end{aligned}$$

One would like to construct a junction tree of it which preserves localization within three subtrees. The graph Υ in Figure 4.17 is the moral graph of Θ. Only the cycle $A_3 - H_3 - E_3 - E_4 - H_4 - A_3$ needs to be triangulated. There are six distinct ways of triangulation out of which only two do not mix nodes in different groups. The two triangulations have a link (H_3, H_4) in common and they do not make significant difference

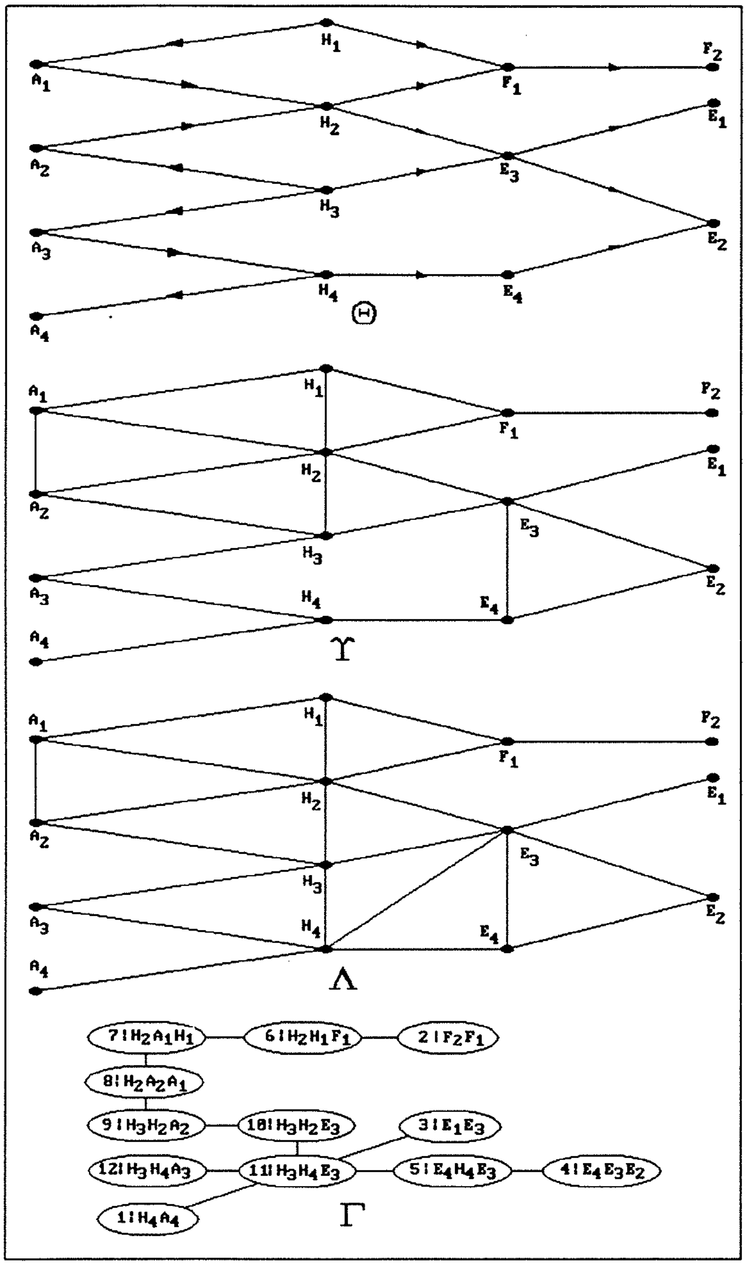


Figure 4.17: A DAG Θ, its moral graph Υ, one of Υ's triangulated graph Λ, and the corresponding junction tree Γ. Each clique in Γ is numbered (the number is separated from clique members by a '|').

in the following analysis. The Λ in Figure 4.17 shows one of the two triangulations. The nodes of graph Γ are all the cliques in Λ.

The junction tree Γ does not preserve localization since cliques 7, 8, 9, 12 and 1 correspond to group G_1 but are connected via cliques 10 and 11 which contains E_3 from group G_3. Examine the morali-triangulated graph Λ to see why this is unavoidable. When there is evidence towards A_1 or A_2 in Λ, updating belief in group G_3 requires passing the joint distribution of H_2 and H_3. But updating the belief in A_3 and A_4 requires passing only the marginal distribution of H_3. That is to say, updating the belief in A_3 and A_4 needs less information than group G_3. In the junction tree representation which insists on a single information channel between any two cliques, this becomes a path from cliques 7, 8, and 9 to clique 12 or 1 via cliques 10 and 11.

In general, let X and Y be two sets of variables in a same natural group, and let Z be a set of variables in a distinct neighbor group. Suppose the information exchange between pairs of them requires the exchange of distribution on sets I_{XY}, I_{XZ} and I_{YZ} of variables respectively. Sometime I_{XY} is a subset of both I_{XZ} and I_{YZ}. When this is the case, a junction tree representation will always indirectly connect cliques corresponding to X and Y through cliques corresponding to Z if the method by Andersen et al. [1989], and Jensen, Lauritzen and Olesen [1990] is followed.

However, there is a way around the problem with a brute force method. In the above example, when there is evidence towards A_1 or A_2, the brute force method pretends that updating the belief in A_3 and A_4 needs as much information as G_3. What one does is to add a dummy link (H_2, A_3) to the moral graph Υ in Figure 4.17. Then triangulating the augmented graph gives the graph Λ' in Figure 4.18. The resultant junction tree Γ' in Figure 4.18 does have 3 subtrees which correspond to the 3 groups desired. However, the largest cliques now have size 4 instead of 3 as before. In binary case, the size of total state space is 92 instead of 84 as before.

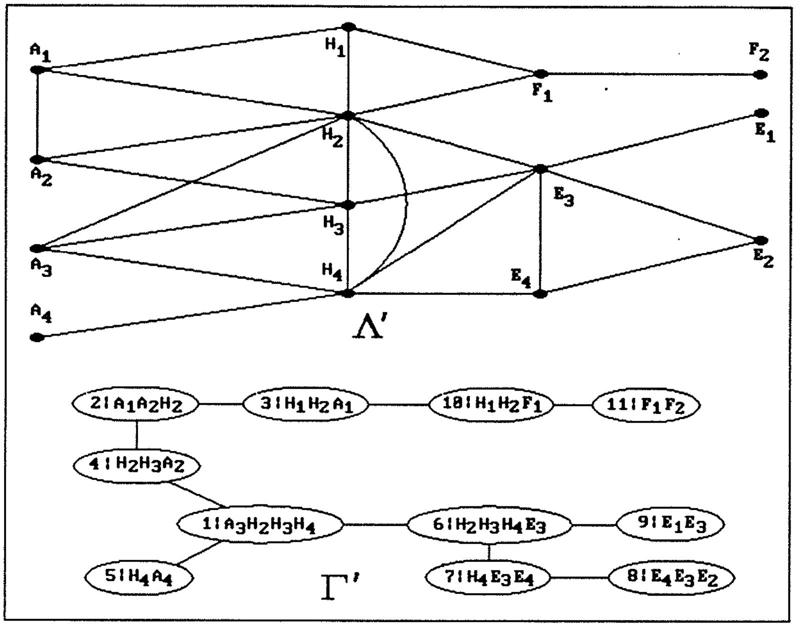


Figure 4.18: Λ' is a triangulated graph. Γ' is a junction tree of Λ'.

In general, the brute force method preserves natural localization by congregation of the set of interfacing nodes (nodes H_2, H_3, H_4 above) between natural groups. In this way, the joint distribution on interfacing nodes[2] can be passed between groups through a single channel, and preservation of localization and preservation of tree structure can be compatible. However, in a large application domain with the original network sparse, this will greatly increase the amount of computation in each group due to the exponential enlargement of the clique state space. The computation amount increased could outweigh the savings gained by exploring localization in general.

The trouble illustrated in the above 2 situations can be traced to the tree structure of junction tree representation which insists on single path between any 2 cliques in the tree. In the normal triangulation case, one has small cliques but loses localization. In the brute

[2] It will be shown later that when the set of interfacing nodes possesses certain property, the joint distribution on the set is the minimum information to be exchanged.

force case, one preserves localization but does not have small cliques. To summarize, the preservation of natural localization and small cliques can *not* coexist by the method of Andersen et al. [1989], and Jensen, Lauritzen and Olesen [1990]. It is claimed here that this is due to a single information channel between local groups of variables. In the following, it is shown that by introducing multiple information channels between groups and by exploring conditional independence, one can pass the joint distribution on a set of interfacing variables between groups by passing only marginal distributions on subsets of the set.

4.4 Overview of MSBN and Junction Forest Technique

As demonstrated in the previous section, in order to explore localization, tree structure and single channel requirement have to be relaxed. Since the computational advantage offered by tree structure has also been demonstrated repeatedly, it is not desirable totally to abandon tree structure. Rather, it is desirable to keep tree structure within each natural group, but allow multiple channels between groups. The MSBNs and junction forests representations extend the d-separation concept and the junction tree technique to implement this idea. This section outlines the development of these representations. Each major step involved is described in terms of its functionality. The problems possibly encountered and the hints for solutions are discussed. The details are presented in the subsequent sections. Since the technique extends the junction tree technique reviewed in section 4.2.2, the parallels and the differences are indicated. Figure 4.19 illustrates the major steps in transformation of the original representation into the secondary representation for both techniques.

The d-sepset The purpose is to partition a large domain according to natural localization into subdomains such that each can be represented separately by a Bayesian subnet;

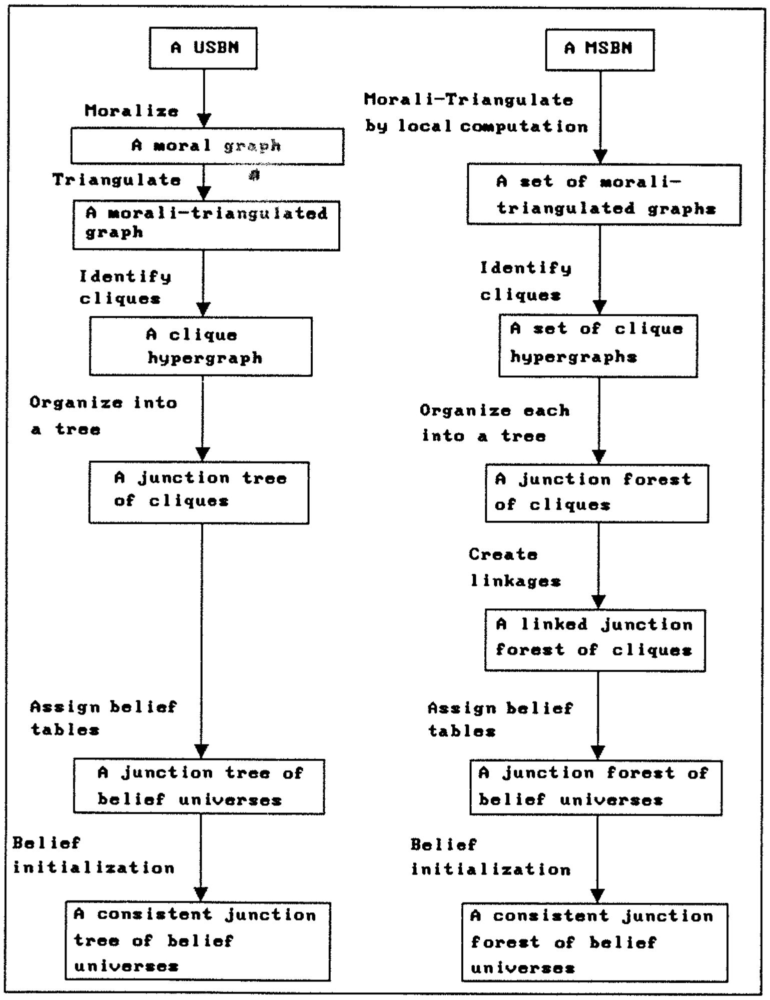

A USBN
Moralize
A moral graph
Triangulate
A morali-triangulated graph
Identify cliques
A clique hypergraph
Organize into a tree
A junction tree of cliques
Assign belief tables
A junction tree of belief universes
Belief initialization
A consistent junction tree of belief universes
A MSBN
Morali-Triangulate by local computation
A set of morali-triangulated graphs
Identify cliques
A set of clique hypergraphs
Organize each into a tree
A junction forest of cliques
Create linkages
A linked junction forest of cliques
Assign belief tables
A junction forest of belief universes
Belief initialization
A consistent junction forest of belief universes

Figure 4.19: Left: major steps in transformation of a USBN (UnSectioned Bayesian Network) into a junction tree. Right: major steps in transformation of a MSBN into a junction forest.

and that these subnets can cooperate with each other during inference by exchanging minimum amount of information between them. In order to do that, one needs to find out the technical constraints which have to be followed during the partition. This can be formulated conceptually in the opposite direction. Suppose the domain has been represented with a homogeneous network. The task is to find necessary technical constraints to be followed when the net is partitioned into subnets according to natural localization. Section 4.5 defines *d-sepsets* whose joint distribution is the minimum information to be exchanged to keep neighbor subnets informed. It is shown that in the junction tree representation of the homogeneous net, the nodes in d-sepsets actually serve as information passageways between nodes in different subnets. Thus the d-sepset is a constraint on the interface between each pair of subnets.

Sectioning Continuing in the conceptual direction, section 4.6 describes how to *section* a homogeneous Bayesian net into subnets called *sects*, based on d-sepsets. The collection of these sects forms a *MSBN*. It is described how probability distribution should be assigned to sects relative to the distribution in homogeneous network. Particularly, it is necessary to assign the original probability table of a d-sepnode to a unique sect which contains the d-sepnode and all its parent nodes, and assign to the same d-sepnode in other sects a uniform table. This is necessary, for one thing, because if a sect does not contain a d-sepnode's all parents as in the homogeneous net, the size of probability table of the d-sepnode must be decreased; for another, because this assignment will guarantee the joint system belief constructed later to be proportional to the joint probability distribution of the homogeneous net.

In order to perform efficient inference in a general but sparse network, one wants to transform each sect into a separate junction tree which will stand as an inference entity. When doing so, it is necessary to preserve the intactness of the clique hypergraph resulted from corresponding homogeneous net. That is, one has to ensure each clique in the original hypergraph will find at least one host sect. This imposes another constraint, termed *soundness of sectioning*, on the overall organization of sects. Actually the soundness of sectioning plus d-sepset interface imposes conditional independence constraints at a macro level, i.e., at the level of sects (as opposed to conditional independence at the level of nodes). In addition to a necessary and sufficient condition for soundness, a guiding rule called *covering subDAG* is provided to ensure soundness. Its repeated application forms a second rule called *hypertree* which can be used for creating sophisticated MSBNs whose sectioning is sound. Although there exists MSBNs of sound sectioning which do not follow the 2 rules, it is shown that computational advantages are obtained in MSBNs sectioned according to the rules. Further discussion will therefore only be directed to MSBNs satisfying the 2 rules.

Moralization and triangulation To transform a MSBN into a set of junction trees requires moralization and triangulation as reviewed in section 4.2.2. However in the MSBN context, an operational option is available, i.e., transformation can be performed globally or by local computation at the level of sects. The global computation performs moralization and triangulation in the same way as in the junction tree technique with care to be taken not to mix nodes in distinct sects into one clique. An additional mapping of the resultant morali-triangulated graph into subgraphs corresponding to sects is needed. But when *space saving* is concerned, local computation is desired. The pitfalls and procedures in moralization and triangulation by *local* computation is discussed.

Since the number of parents for a d-sepnode may be different in different sects, the moralization in MSBN can not be achieved by 'pure' local computation in each sect. Communication between sects is required to ensure parent d-sepnodes are moralized identically in different sects.

The criterion of triangulation in the MSBN is to ensure the 'intactness' of resulting hypergraph from the corresponding homogeneous net. Problems arise if one insists in triangulation by local computation at the level of sects. One problem is that an intersect cycle will be triangulated in the homogeneous net, but the cycle can not be realized by viewing locally in each sect involved. Another problem is that cycles involving d-sepnodes may be triangulated differently in different sects. The solution is to let sects communicate during triangulation. Since moralization and triangulation both involve adding links and both require communication between sects, the corresponding local operations in each sect can be performed together and messages to other sects can be sent together. Therefore, operationally, moralization and triangulation in MSBN are not separate steps as in the junction tree technique. The corresponding integrated operation is termed *morali-triangulation* to conceptually reflect this reality.

In section 4.7.1, the above concept of 'intactness' of hypergraph is formalized in terms of *invertibility* of morali-triangulation. it is shown that if the sectioning of a MSBN is sound, then there exists an invertible morali-triangulation such that the 'intactness' of hypergraph is preserved. Section 4.7.1 provides an algorithm for an invertible morali-triangulation assuming a covering subDAG.

Next steps in the junction tree technique In the junction tree technique, after triangulation, further steps of transformation are identification of cliques in the morali-triangulated graph (clique hypergraph formation) and junction tree construction. In MSBNs, these steps are performed in the similar way for each sect as the junction tree technique. A MSBN is thus transformed into a set of junction trees called a *junction forest of cliques.* See Andersen et al. [1989], and Jensen, Lauritzen and Olesen [1990] for technique details involving these steps.

Linkage formation An important extension of MSBNs and junction forests to the junction tree technique is the formation of multiple information channels between junction trees (in a junction forest) such that a joint distribution on a d-sepset can be passed between a pair of junction trees by passing through marginal distributions on subsets of the d-sepset. In this way, the exponential enlargement of the clique state space caused by brute force method (section 4.3) can be avoided. These channels are termed *linkages* (section 4.7.2). Each linkage is a set of d-sepnodes which links 2 cliques each in one of the 2 junction trees involved. During inference, if evidence is obtained from previously active junction tree, it can be propagated to newly active neighbor junction tree through linkages between them.

As can be imagined, multiple linkages can cause redundant information passage or confuse the information receiver. The problem can be avoided by coordination among linkages during information passing. Since the problem manifests differently during belief initialization and evidential reasoning, it has to be treated differently. In both cases, information passing is performed one linkage at a time. During initialization, (redundant) information already passed through other linkages is removed from the linkage belief table before the latter is passed over. Operationally, one orders linkages. The intersection of a linkage with linkages ordered before it is defined as the *redundancy set* of the linkage. The redundancy set tells a linkage what portion of information has to be removed during information passing. During evidential reasoning, a DistributeEvidence is performed after each information passing to avoid confusion in the receiving junction tree. With linkages and redundancy sets created, one has a *linked junction forest of cliques.*

Formation of joint system belief of junction forest The joint system belief of the junction forest is defined (section 4.7.3) in terms of the belief on each junction trees, the belief on linkages and redundancy sets such that it is proportional to the joint probability distribution of the homogeneous net. With the joint system belief defined, one has a *junction forest of belief universes.* When it is clear from the context, only 'junction forest' is used without differentiating between its different stages.

Consistency and separability of junction forest As in the case of the junction tree technique, one would like to obtain marginal probability of a variable by marginalization of the belief in any belief universe of any junction tree which contains the variable, i.e., by local computation. In the case of the junction tree technique, this requires the *consistency* property which can be satisfied by performing the DistributeEvidence and the CollectEvidence as reviewed in section 4.2.2. In the context of a junction forest, an additional property called *separability* is required (section 4.8) due to multiple linkages between junction trees. It imposes a *host composition* constraint on the composition of linkage host cliques. The function of linkages is to pass the joint belief of the corresponding d-sepset. When linkage hosts are ill-composed, what is passed over is not a correct version of joint belief on the d-sepset. The marginal probabilities thus obtained by local computation will also be incorrect. It is shown if all the junction trees in a junction forest satisfies the host composition condition, then separability is guaranteed. Why these conditions usually hold naturally is explained. The remedy when the condition does not hold is also discussed. With a junction forest structure satisfying the separability, and with a set of operations performed to bring the forest into consistency, one is guaranteed to obtain marginal probabilities by local computation.

Belief initialization Belief initialization (section 4.9.3) in a junction forest is achieved by first bringing the belief universes in each junction tree into consistency, and then exchanging prior belief between junction trees to bring the junction forest into global consistency. When exchanging beliefs, care is to be taken on (1) non-trivial information (recall that d-sepnodes in some sects are assigned uniform tables during sectioning) could be contained in either side of the 2 junction trees involved; and (2) not to pass redundancy information through multiple linkages. Section 4.9 defines several levels of operations to initialize belief of a junction forest by local computation.

Evidential reasoning Only 1 junction tree in a junction forest needs to be active due to localization and great computational savings are possible when repeated computation of the junction tree is required. Whenever new evidence becomes available to the currently active junction tree, it is entered and the tree is made consistent such that queries can be answered. Thus the computation complexity of a MSBN/junction forest is $1/\beta$ where β is the number of sects in the MSBN. When the user shifts attention, a new junction tree replaces the currently active tree and all previously acquired evidence is absorbed through the operation ShiftAttention. The operation requires only a chain of neighbor junction trees to be updated. During the inter-junction tree updating, one needs to ensure no confusion is resulted from multi-linkage information passing.

4.5 The d-sepset and the Junction Tree

4.5.1 The d-sepset

As discussed in section 4.4, the problem of partitioning a Bayesian net by natural localization can be conceptually formulated as though the domain has been represented with a homogeneous network. The task is to find out the technical constraint to partition the net into subnets such that the subnets can be used separately and cooperatively during inference with minimum amount of information exchange. This section defines the most important concept for partitioning, namely, d-sepset. Then some insights are provided into its implication in the secondary structure of DAGs.

Definition 6 (d-sepset) *Let $D = D^1 \sqcup D^2$ be a DAG. The set of nodes $I = N^1 \cap N^2$ is a* **d-sepset** *between subDAG D^1 and D^2 if the following condition holds.*

For every $A_i \in I$ with its parents π_i in D, either $\pi_i \subseteq N^1$, or $\pi_i \subseteq N^2$.

Elements of a d-sepset are called **d-sepnodes**. *When the above condition holds, D is said to be* **sectioned** *into $\{D^1, D^2\}$.*

Note in general a DAG $D = D^1 \sqcup D^2$ does not imply that D is sectioned into $\{D^1, D^2\}$ since the intersection of the corresponding 2 sets of nodes may not be a d-sepset.

Lemma 4 *Let a DAG D be sectioned into $\{D^1, D^2\}$ and $I = N^1 \cap N^2$ be a d-sepset. I d-separates $N^1 \setminus I$ from $N^2 \setminus I$.*

Proof:

It suffices to prove every path between $N^1 \setminus I$ and $N^2 \setminus I$ is blocked by I. Every path between $N^1 \setminus I$ and $N^2 \setminus I$ has at least one d-sepnode. From definition of d-separate, if one of the d-sepnode in a path is a non-head-to-head node, the path is blocked.

In the case of a single d-sepnode path, by definition, the d-sepnode must be a non-head-to-head node. In case of multiple d-sepnode path, for any 2 adjacent d-sepnodes on the path, one of them must be a non-head-to-head node.

□

The lemma can be generalized into the following theorem which states that d-sepsets d-separate a subDAG from the rest of the DAG. Note when a d-sepset is indexed with 2 superscripts, their order is immaterial.

Theorem 2 (completeness of d-sepset) *Let a DAG D be sectioned into $\{D^1, \ldots, D^\beta\}$ and $I^{ij} = N^i \cap N^j$ be the d-sepset between D^i and D^j. $\cup_{j \neq i} I^{ij}$ d-separates $N^i \setminus \cup_{j \neq i} I^{ij}$ from $N \setminus N^i$.*

The theorem implies that the joint distribution on d-sepsets is the minimum information to be exchanged between a Bayesian subnet and the rest of the network. Consider a DAG (N, E) sectioned into $\beta > 1$ subDAGs. Let A denote the set of all the non-d-sepnodes in one of the subDAGs, C denote the set union of all the d-sepsets between the chosen subDAG and its neighbor subDAGs, and B denote the set $N \setminus A \setminus C$. From the above theorem, C d-separates A from B. Thus $p(ABC) = p(AC)p(BC)/p(C)$. When evidence is available such that some of the variables in A are instantiated, one can update $p(AC)$ into $p'(AC) = p'(A)p(C|A)$. One can update $p(C)$ into $p'(C)$ by marginalization on $p'(AC)$. Then it follows that $p'(BC) = p(B|C)p'(C)$, and that the updated joint distribution $p'(ABC) = p'(AC)p'(BC)/p'(C)$. Note that updating $p'(BC)$ is through replacing $p(C)$ by $p'(C)$ - the posterior joint distribution on the d-sepset union.

Example 5 The DAG Θ in Figure 4.17 is sectioned into $\{\Theta^1, \Theta^2, \Theta^3\}$ in Figure 4.20. $I^{12} = \{H_1, H_2\}$ is the d-sepset between Θ^1 and Θ^2; $I^{13} = \{H_2, H_3, H_4\}$ is the d-sepset between Θ^1 and Θ^3; and $I^{23} = \{H_2\}$ is the d-sepset between Θ^2 and Θ^3. $I^{12} \cup I^{13} = \{H_1, H_2, H_3, H_4\}$ d-separates the rest of Θ_1 from the rest of Θ_2 and Θ_3.

There is a close relation between d-sepset and usual graph separator given in proposition 7. If Z is the graph separator of X and Y, then the removal of the set Z of nodes from the graph (together with their associated links) would render the nodes in X disconnected from those in Y.

Proposition 7 *Let a DAG D be sectioned into $\{D^1, D^2\}$. The set of nodes $I = N^1 \cap N^2$ is a d-sepset between D^1 and D^2 iff I is a graph separator in the moral graph of D.*

Proof:

Suppose $I = N^1 \cap N^2$ is a d-sepset. For any $A_1 \in N^1 \setminus I$ and $A_2 \in N^2 \setminus I$, moralization will not create links between A_1 and A_2. Hence I is a graph separator in the moral graph of D.

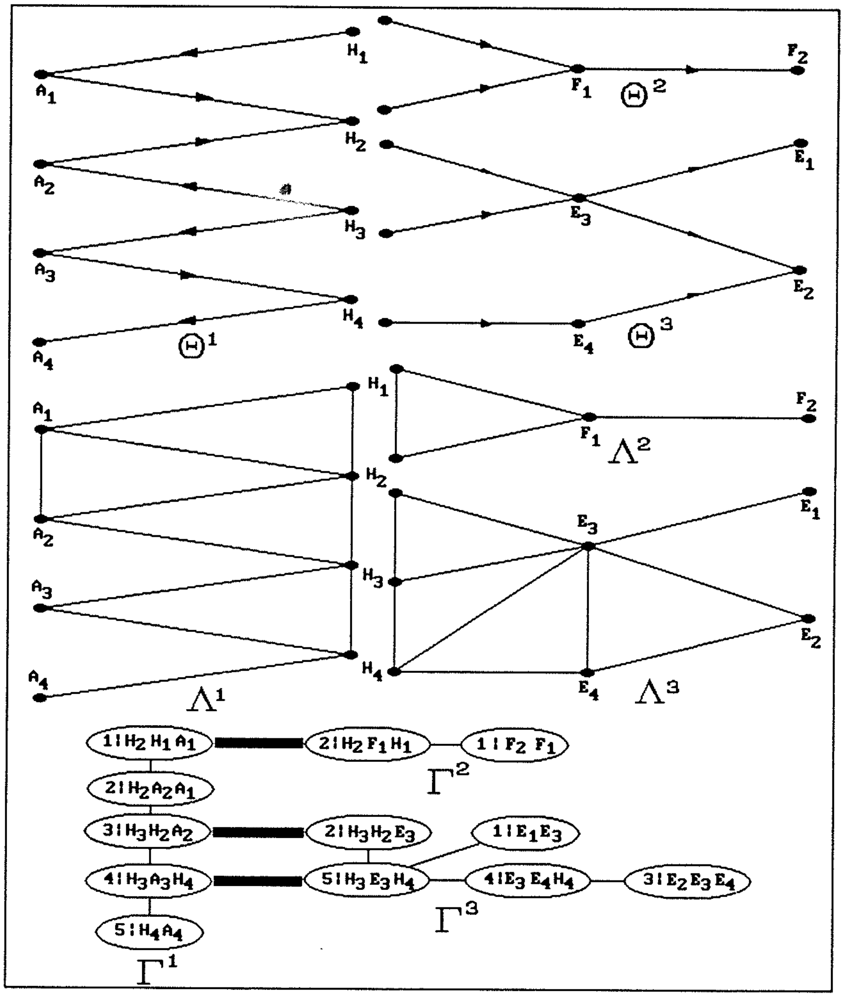


Figure 4.20: The set $\{\Theta^1, \Theta^2, \Theta^3\}$ of 3 subDAGs (top) forms a sectioning of Θ in Figure 4.17. $\{\Lambda^1, \Lambda^2, \Lambda^3\}$ (middle) is the set of morali-triangulated graphs of $\{\Theta^1, \Theta^2, \Theta^3\}$, and $\Xi = \{\Gamma^1, \Gamma^2, \Gamma^3\}$ (bottom) is the corresponding junction forest obtained by Algorithm 2. The ribbed bands indicate linkages.

On the other hand, suppose I separate $N^1 \setminus I$ from $N^2 \setminus I$ in moral graph M of D, and I is not a d-sepset. Then there exists $A \in I$ such that A has a parent $A_1 \in N^1 \setminus I$ and a parent $A_2 \in N^2 \setminus I$. But then there would have been a link between A_1 and A_2 by moralization. Thus I is not a separator in M. Contradiction.

□

The properties of d-separation in the DAG representation of Bayesian networks have been studied extensively [Pearl 88, Geiger, Verma and Pearl 90]. It can be used to derive Pearl's propagation algorithm in singly-connected Bayesian nets [Neapolitan 90]. But to my knowledge, its implication in secondary structure has not been examined. The definition of the d-sepset now allows to do so.

4.5.2 Implication of d-sepset in Junction Trees

By representing a multiply connected Bayesian network in its secondary structure - junction tree, flexible and efficient belief propagation can be achieved. With the d-sepset concept defined, one would like to know how information is passed in the junction tree between nodes separated by the d-sepset in the original Bayesian network.

Lemma 5 *Let a DAG D be sectioned into $\{D^1, D^2\}$ and $I = N^1 \cap N^2$ be the d-sepset. A junction tree T can be constructed from D, such that the following statement is true.*

> *For all pairs of nodes $A_1 \in N^1 \setminus I$ and $A_2 \in N^2 \setminus I$, if A_1 is contained in clique C_1 and A_2 in C_2, then on the unique path between C_1 and C_2 in T, there exists a clique sepset Q containing only d-sepnodes.*

Proof:

[Step 1] First, prove T can be constructed such that $C_1 \neq C_2$. By definition of the d-sepset, A_1 and A_2 are not adjacent in D. Cliques are formed through moralization and triangulation. If A_1 and A_2 both are adjacent to a d-sepnode H, by definition of d-sepset, A_1 and A_2 can not both be H's parents. Thus moralization does not create a link between A_1 and A_2.

Enforce the following rule for triangulation: if a cycle contains 2 or more non-adjacent d-sepnodes, add a link between 2 such d-sepnodes first.

After moralization, if there is only 1 path between A_1 and A_2, triangulation will not add a link between them. If there are more than 1 path, at least 1 cycle is formed. For any such cycle including A_1 and A_2, 2 paths between A_1 and A_2 can be identified. On each of the 2 paths, there must be at least 1 d-sepnode. With the above rule followed, a link between 2 such d-sepnodes is always added such that there is no need to link A_1 and A_2 to triangulate D.

Since A_1 and A_2 are not linked initially; and not during moralization and triangulation, they will not be contained in the same clique in T.

[Step 2] Proceed from C_1 to C_2 along the unique path in T connecting them to find Q. Examine C_1's neighbor clique C_x.

Case 1: $C_x \subseteq I$. Then $Q = C_1 \cap C_x$.

Case 2: C_x contains nodes in $N^2 \setminus I$. By step 1, C_1 does not contain nodes in $N^2 \setminus I$ and C_x does not contain nodes in $N^1 \setminus I$. However, $C_1 \cap C_x \neq \phi$. Therefore $Q = C_1 \cap C_x$.

Otherwise C_x contains nodes in $N^1 \setminus I$. In this case, proceed to C_x's neighbor C_y and above examination is repeated. Since the other end of the path is C_2 which contains no node in $N^1 \setminus I$, some point along the path one will eventually hit either case 1 or case 2.

□

The lemma can be generalized to the case of any finite number of subDAGs. This is the following proposition. Its proof is similar to the lemma.

Proposition 8 (belief relay) *Let a DAG D be sectioned into $\{D^1, \ldots, D^\beta\}$ and $I = \cup_{j \neq i} I^{ij}$ be the union of d-sepsets between D^i and its neighbor subDAGs. A junction tree T can be constructed from D, such that the following statement is true.*

> *For all pairs of nodes $A_1 \in N^i \setminus I$ and $A_2 \in N \setminus N^i$, if A_1 is contained in clique C_1 and A_2 in C_2, then on the unique path between C_1 and C_2 in T, there exists a clique sepset Q containing only d-sepnodes in I.*

Example 6 Recall the DAG Θ in Figure 4.17 which is sectioned into $\{\Theta^1, \Theta^2, \Theta^3\}$ in Figure 4.20 with $I = \{H_2, H_3, H_4\}$ being the d-sepset between Θ^1 and Θ^3. Consider the node A_3 in clique $\{H_3, H_4, A_3\}$ and the node E_2 in clique $\{E_4, E_3, E_2\}$ in the junction tree Γ in Figure 4.17. In the path between the 2 cliques, the sepset $\{H_3, H_4\}$ between cliques $\{H_3, H_4, A_3\}$ and $\{H_3, H_4, E_3\}$ contains only d-sepnodes.

When new evidence is available, it can be propagated to the overall junction tree through sepsets between cliques [Jensen, Lauritzen and Olesen 90]. Therefore, the above proposition means that a junction tree can be constructed such that evidence in $N^i \setminus I$ must pass through at least 1 sepset containing only nodes in I in order to be propagated to nodes in $N \setminus N^i$.

Theorem 2 and Proposition 8 suggest that one can organize the clique hypergraph such that the cliques corresponding to different subDAGs separated by d-sepsets can be organized into different junction trees. Communication between them can be accomplished through d-sepsets. This idea is formalized below.

4.6 Multiply Sectioned Bayesian Nets

4.6.1 Definition of MSBN

Definition 7 (MSBN) *Let $S = (N, E, P)$ be a Bayesian network; $D = (N, E)$ be sectioned into $\{D^1, \ldots, D^\beta\}$ where $D^i \equiv (N^i, E^i)$; and $I^{ij} = N^i \cap N^j$ be the d-sepset between D^i and D^j $(1 \leq i, j \leq \beta; i \neq j)$. Assign d-sepnodes to subDAGs in the following way.*

> *For each d-sepnode A, if the in-degree η^i of A in subDAG D^i satisfies $\eta^i \geq \eta^j$ $(j = 1, \ldots, \beta)$ then assign A to subDAG D^i and break ties arbitrarily.*

Assign probability distribution P^i for subDAG D^i $(i = 1, \ldots, \beta)$ in the following way.

> *For all node $A \in N^i$, if A is a d-sepnode and A is not assigned to D^i, assign to A a* **uniform** *probability table.*[3] *Otherwise assign to A an identical probability table as that in (N, E, P).*

Call $S^i = (D^i; P^i) \equiv (N^i, E^i; P^i)$ a **sect** *and call the set of sects $\{S^1, \ldots, S^\beta\}$ a* **Multiply Sectioned Bayesian Network (MSBN).**

The original Bayesian net S is called an 'UnSectioned Bayesian Network (USBN)'. Note that the sectioning of a Bayesian network is essentially determined by the sectioning of the corresponding DAG D. It doesn't matter which way to break ties. There will be no significant difference in further processing.

Example 7 Suppose the variables in DAG Θ in Figure 4.17 are all binary. Associate the probability distribution P given in Table 4.3 with Θ. (Θ, P) is an USBN.

Given the USBN (Θ, P), and corresponding 3 subDAGs Θ^1, Θ^2 and Θ^3, a 3-sect MSBN $\{(\Theta^1, P^1), (\Theta^2, P^2), (\Theta^3, P^3)\}$ can be constructed. First assign d-sepnodes $H_1, \ldots,$ H_4 to the subDAGs. H_2 and H_4 must be assigned to Θ^1. H_1 can be assigned to either

[3] This is necessary, for one thing, because if a sect does not contain a d-sepnode's all parents as in the homogeneous net, the size of probability table of the d-sepnode must be decreased; for another, because this assignment will guarantee that the joint system belief constructed in section 4.7.3 is proportional to the joint probability distribution P.

$p(h_{11}) =$	.15	$p(a_{31} \mid h_{31}) =$	.3	$p(e_{21} \mid e_{31} e_{41}) =$	.9789
$p(h_{21} \mid a_{21} a_{11}) =$	.8696	$p(a_{31} \mid h_{32}) =$	.8	$p(e_{21} \mid e_{31} e_{42}) =$	.8
$p(h_{21} \mid a_{21} a_{12}) =$	.7	$p(a_{41} \mid h_{41}) =$	.9	$p(e_{21} \mid e_{32} e_{41}) =$	.9
$p(h_{21} \mid a_{22} a_{11}) =$	.6	$p(a_{41} \mid h_{42}) =$	.2	$p(e_{21} \mid e_{32} e_{42}) =$	.05
$p(h_{21} \mid a_{22} a_{12}) =$	.08	$p(f_{11} \mid h_{11} h_{21}) =$	.7895	$p(e_{31} \mid h_{21} h_{31}) =$	.7702
$p(h_{31}) =$	.3	$p(f_{11} \mid h_{11} h_{22}) =$	.5	$p(e_{31} \mid h_{21} h_{32}) =$	.35
$p(h_{41} \mid a_{31}) =$	.25	$p(f_{11} \mid h_{12} h_{21}) =$	.6	$p(e_{31} \mid h_{22} h_{31}) =$	.65
$p(h_{41} \mid a_{32}) =$	.4	$p(f_{11} \mid h_{12} h_{22}) =$	.05	$p(e_{31} \mid h_{22} h_{32}) =$	.01
$p(a_{11} \mid h_{11}) =$	.8	$p(f_{21} \mid f_{11}) =$	.4	$p(e_{41} \mid h_{41}) =$	.8
$p(a_{11} \mid h_{12}) =$	.1	$p(f_{21} \mid f_{12}) =$	.75	$p(e_{41} \mid h_{42}) =$	.15
$p(a_{21} \mid h_{31}) =$	.8	$p(e_{11} \mid e_{31}) =$	.2		
$p(a_{21} \mid h_{32}) =$	.1	$p(e_{11} \mid e_{32}) =$	.7		

Table 4.3: Probability distribution associated with DAG Θ in Figure 4.17.

Θ^1 or Θ^2, and H_3 can be assigned to either Θ^1 or Θ^3. Here it is chosen to assign all 4 d-sepnodes to Θ^1. Based on this assignment and P given, the probability distribution for each sect can be determined (Table 4.4). Note the uniform probability tables assigned to d-sepnodes in Θ_2 and Θ_3.

P^1		P^2		P^3	
$p(h_{11}) =$	.15	$p(h_{11}) =$	.5	$p(h_{11}) =$	.5
$p(h_{21} \mid a_{21} a_{11}) =$	.8696	$p(h_{21}) =$	.5	$p(h_{21}) =$	.5
$p(h_{21} \mid a_{21} a_{12}) =$	.7	$p(f_{11} \mid h_{11} h_{21}) =$	.7895	$p(h_{31}) =$	.5
$p(h_{21} \mid a_{22} a_{11}) =$	.6	$p(f_{11} \mid h_{11} h_{22}) =$	.5	$p(e_{11} \mid e_{31}) =$	.2
$p(h_{21} \mid a_{22} a_{12}) =$	.08	$p(f_{11} \mid h_{12} h_{21}) =$	.6	$p(e_{11} \mid e_{32}) =$	.7
$p(h_{31}) =$	.3	$p(f_{11} \mid h_{12} h_{22}) =$	.05	$p(e_{21} \mid e_{31} e_{41}) =$	.9789
$p(h_{41} \mid a_{31}) =$	.25	$p(f_{21} \mid f_{11}) =$	.4	$p(e_{21} \mid e_{31} e_{42}) =$	.8
$p(h_{41} \mid a_{32}) =$	.4	$p(f_{21} \mid f_{12}) =$	.75	$p(e_{21} \mid e_{32} e_{41}) =$	.9
$p(a_{11} \mid h_{11}) =$	.8			$p(e_{21} \mid e_{32} e_{42}) =$	.05
$p(a_{11} \mid h_{12}) =$	.1			$p(e_{31} \mid h_{21} h_{31}) =$	.7702
$p(a_{21} \mid h_{31}) =$	.8			$p(e_{31} \mid h_{21} h_{32}) =$	.35
$p(a_{21} \mid h_{32}) =$	.1			$p(e_{31} \mid h_{22} h_{31}) =$	.65
$p(a_{31} \mid h_{31}) =$	.3			$p(e_{31} \mid h_{22} h_{32}) =$	.01
$p(a_{31} \mid h_{32}) =$	.8			$p(e_{41} \mid h_{41}) =$	.8
$p(a_{41} \mid h_{41}) =$	.9			$p(e_{41} \mid h_{42}) =$	.15
$p(a_{41} \mid h_{42}) =$	.2				

Table 4.4: Probability distribution associated with subDAGs Θ^1, Θ^2 and Θ^3 in Figure 4.20.

4.6.2 Soundness of Sectioning

In order to perform efficient inference computation in a multiply connected Bayesian net, the junction tree technique transforms the Bayesian net into a clique hypergraph through moralization and triangulation. Then the hypergraph is organized into a junction tree, and efficient inference can take place. Because of the computational advantage of junction trees, in the context of a MSBN, one would like to transform each sect into a junction tree representation. The immediate question is: for an arbitrary MSBN, what is the condition in the transformation such that correct inference can take place in the resultant set of junction trees. The following reviews the major theoretical results related to this question.

Lauritzen et al. [1984] show that the clique hypergraph of a graph is decomposable iff the graph is triangulated. Jensen [1988] proves that a hypergraph has a junction tree iff it is decomposable. Maier [1983] proves the same in the context of relational database. Jensen, Lauritzen and Olesen [1990] and Pearl [1988] show that a junction tree representation of a Bayesian net is an equivalent representation in the sense that the information about joint probability distribution can be preserved. Finally, a more flexible algorithm (compared to [Lauritzen and Spiegelhalter 88]) is devised on the junction tree representation of multiply connected Bayesian nets [Jensen, Lauritzen and Olesen 90].

The above results highlight the importance of clique hypergraphs resulted from triangulation of original graphs. Thus, when one tries to transform each sect in a MSBN into a junction tree, it is necessary to preserve the intactness of the clique hypergraph resulted from corresponding USBN. This is possible only if the sectioning of DAG D of the original USBN is sound defined formally below.

Definition 8 (soundness of sectioning) *Let a DAG D be sectioned into $\{D^1, \ldots, D^\beta\}$. If there exists a clique hypergraph from D such that for every clique C_k in the hypergraph* *there is at least 1 subDAG D^i satisfying $C_k \subseteq N^i$, then the sectioning is* **sound**. *Call D^i the* **host subDAG** *of clique C_k.*

Although the soundness of sectioning is defined in DAGs, the concept is useful only in the context of MSBNs. Therefore when the sectioning of DAG is sound, it is said that the sectioning of the corresponding USBN into the MSBN is sound.

If the sectioning of a DAG D is unsound, then one will find no host subDAG for at least 1 clique in all possible hypergraphs from D. If a MSBN is based on this sectioning of DAG, it is impossible to maintain the autonomous status of sects in the secondary representation.

Example 8 In Figure 4.21, $\{D^1, D^2, D^3\}$ is an unsound sectioning of D. The clique hypergraph for D must have clique $\{A, B, C\}$ which finds no host subDAG from D^1, D^2, and D^3.

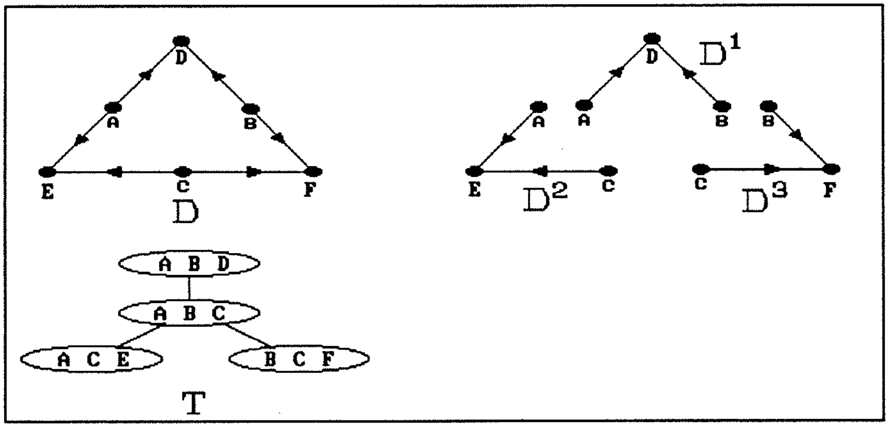


Figure 4.21: Top left: A DAG D. Top right: The set of subDAGs from an unsound sectioning of D. Bottom left: The junction tree T from D.

The following develops necessary and sufficient condition for soundness of sectioning.

Lemma 6 *Let $A_1 - \ldots - A_{i-1} - B_1 - \ldots - B_{j-1} - C_1 - \ldots - C_{k-1} - \ldots - A_1$ be a cycle consisting of nodes from 3 or more sets: $X = \{A_1, \ldots, A_{i-1}, B_1\}$, $Y = \{B_1, \ldots, B_{j-1}, C_1\}$, and so on. The nodes from a same set are adjacent in the cycle, and 2 adjacent sets have a node in common. Then triangulation of this cycle must create a triangle with its 3 nodes not belonging to any single set.*

Proof:

The proof is inductive. The lemma is trivially true if the cycle involves only 3 nodes.

Assume the lemma is true when the cycle involves $n = 3$ sets and the cycle O is $A_1 - \ldots - A_{i-1} - B_1 - \ldots - B_{j-1} - C_1 - \ldots - C_{k-1} - A_1$, i.e., there are i nodes in set 1, j in set 2, and k in set 3. If 1 more node is added to set 3 ($k' = k+1$) such that a node C_k is added between C_{k-1} and A_1 of cycle O, then one can first add a chord between C_{k-1} and A_1, and triangulate the rest of the cycle. Since the remaining untriangulated cycle is exactly the cycle O, the lemma is true by assumption. Since the 3 sets are symmetric and the nodes in any set above can be augmented by nodes from any sets other than the above 3, the proof is valid in general.

□

If a MSBN has only 2 sects, the sectioning is *always* sound. Unsoundness can arise only when there are 3 or more sects. The following shows *exactly* the case where a sectioning is unsound.

Theorem 3 (inter-subDAG cycle) *A sectioning of a DAG D to a set of 3 or more subDAGs is sound iff there exists no (undirected) cycle in D which consists of nodes from 3 or more distinct subDAGs such that the nodes from each subDAG are adjacent on the cycle.*

Proof:

Let D be a DAG sectioned into $\{D^1, \ldots, D^\beta\}$ ($\beta \geq 3$). Suppose, relative to this sectioning, D has no cycle which consists of nodes from 3 or more distinct subDAGs such that the nodes from each subDAG are adjacent on the cycle. By definition of d-sepset, moralization of D may triangulate existing cycles but will not create new cycle. Thus by assumption, in moral graph of D, all cycles consisting of nodes from more than 1 subDAG such that the nodes from each subDAG are adjacent on the cycle can involve nodes from at most 2 subDAGs. By the argument in Step 1 of the proof for Lemma 5, all the 2-subDAG crossing cycles can be triangulated without creating cliques containing non-d-sepnodes in both subDAGs. Thus the sectioning is sound.

On the other hand, suppose, relative to the sectioning, there is a cycle in D which consists of nodes from 3 or more distinct subDAGs such that the nodes from each subDAG are adjacent on the cycle. Then, by lemma 6, the triangulation of this cycle must create a triangle with its 3 nodes not belonging to any subDAG. Hence the sectioning is unsound.

□

Given a DAG and a sectioning, search for inter-subDAG cycles relative to the sectioning is expensive, especially by local computation when space is concerned. Even if a sectioning is decided to be sound, it may not be computationally desirable at later reasoning stages as will be discussed at corresponding sections. Furthermore, each sect in a large system (MSBN) is constructed one at a time. If a sectioning is not sound and it can only be discovered after all sects have been constructed, the overall revision would be disastrous. Thus one would like to develop simple guidelines for sound sectioning which could be followed during incremental construction of MSBNs. The following covering subDAG rule is one of such guidelines.

Theorem 4 (covering subDAG) *Let a DAG D be sectioned into $\{D^1, \ldots, D^\beta\}$. Let $I^{jk} = N^j \cap N^k$ be the d-sepset between D^j and D^k. If there is a subDAG D^i such that*

$N^i \supseteq \cup_{j \neq k} I^{jk}$ then the sectioning is sound. The subDAG D^i is called the **covering subDAG** *relative to the sectioning.*

In the context of a MSBN, call the sect corresponding to the covering subDAG as the *covering sect.* Note that the covering sect rule actually imposes a conditional independence constraint at a macro level.

Proposition 9 *Let S^i and S^j be any 2 sects in a MSBN with a covering sect S^k ($i \neq k$, $j \neq k$). The 2 sets of variables N^i and N^j are conditionally independent given N^k.*

Example 9 Consider the 3-sect MSBN $\{(\Theta^1, P^1), (\Theta^2, P^2), (\Theta^3, P^3)\}$ constructed. (Θ^1, P^1) is the covering sect.

Note, in general, the covering sect of a MSBN may not be unique. As far as the soundness is concerned, one is as good as the others. Practically, the one to be consulted most often or the one with the least size is preferred for the sake of computational efficiency which will be clear later.

The covering sect is usually formed naturally. For example (Chapter 5), in a neuromuscular diagnosis system, the sect containing knowledge about clinical examination contains all the disease hypotheses considered by the system. The EMG sect or nerve conduction sect contains only a subset of the disease hypotheses based on diagnostic importance of these tests to each disease. Thus the clinical sect is a natural covering sect with all the disease hypothesis as d-sepnodes interfacing the sect with other sects.

The covering subDAG rule can be repeatedly used to create sophisticated MSBNs which are sound. When doing so, a global covering subDAG requirement is replaced by a local covering subDAG requirement. A *local covering subDAG* is a subDAG interfacing two or more subDAGs and including all d-sepsets among them in its domain.

Theorem 5 (hypertree) *Any MSBN created by the following procedure is sound.*

> *Start with any single subDAG. Recursively add a new subDAG D^i to previous subDAGs such that if D^i has nonempty d-sepsets with more than one previous subDAGs, then one of the previous interfacing subDAGs must be a local covering subDAG which covers all the d-sepsets resultant from the addition.*

The following example illustrate the hypertree rule. It also explains that the sectioning determined by the procedure is sound.

Example 10 Figure 4.22 depicts part of a MSBN constructed by the hypertree rule. Each box represents a subDAG with boundaries between boxes representing d-sepsets. The superscripts of subDAGs represent the order of their creation. D^1, D^4, D^5 are local covering subDAGs.

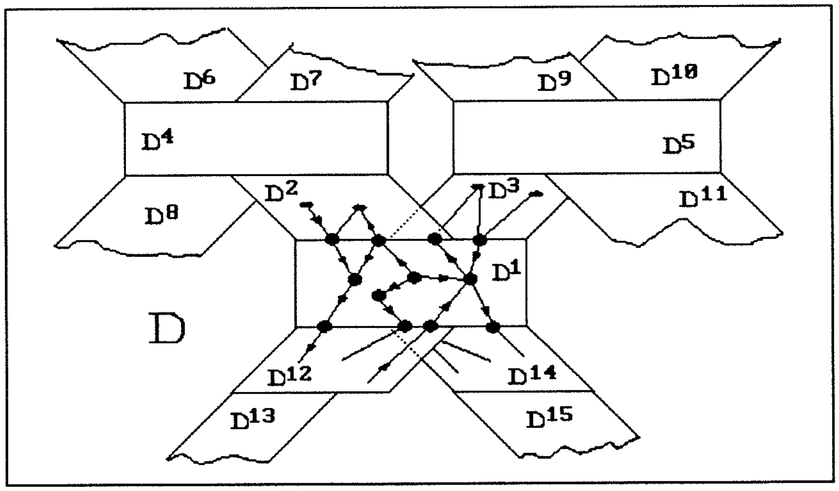


Figure 4.22: A MSBN with a hypertree structure.

It is easy to see that the inter-subDAG cycle as described in theorem 3 can not happen in this MSBN due to its hypertree structure, and hence the sectioning is sound.

Note that the hypertree rule also imposes a conditional independence constraint at a macro level.

Proposition 10 *Let S^i and S^j be any 2 sects with an empty d-sepset in a MSBN sectioned by the hypertree rule. Let S^k be any sect on the unique route mediating S^i and S^j on the hypertree. The 2 sets of variables N^i and N^j are conditionally independent given N^k.*

It should be indicate that the covering subDAG rule and the hypertree rule do not cover every case where sectioning is sound.

Example 11 The 3-sect MSBN $\{D^1, D^2, D^3\}$ in Figure 4.23 has no covering subDAG. But the sectioning is sound.

Note that, although the sectioning of the MSBN in Figure 4.23 is sound, this kind of structure is restricted. For example, one can add arcs between A and B in D^1, between A and C in D^2, but as soon as one adds 1 more arc between B and C in D^3, the theorem 3 is violated and the sectioning become unsound. That is, when n subDAGs ($n \geq 3$) are interfaced in this style, there can be *at most* $n-1$ of them being multiply connected. Further computational problems with such structure will be discussed in the appropriate latter sections.

Since MSBNs constructed by the covering subDAG rule or the hypertree rule have sound sectioning, are less restricted, and have extra computational advantages (to be discussed in latter sections) over the MSBNs which do not follow these rules, the following study is directed to only the former MSBNs.

Conceptually, all MSBNs constructed by the hypertree rule can be viewed as MSBNs with covering subDAGs when attention is directed to local structures. For example, if one pays attention to D^1 in Figure 4.22 and its surrounding subDAGs, one can view

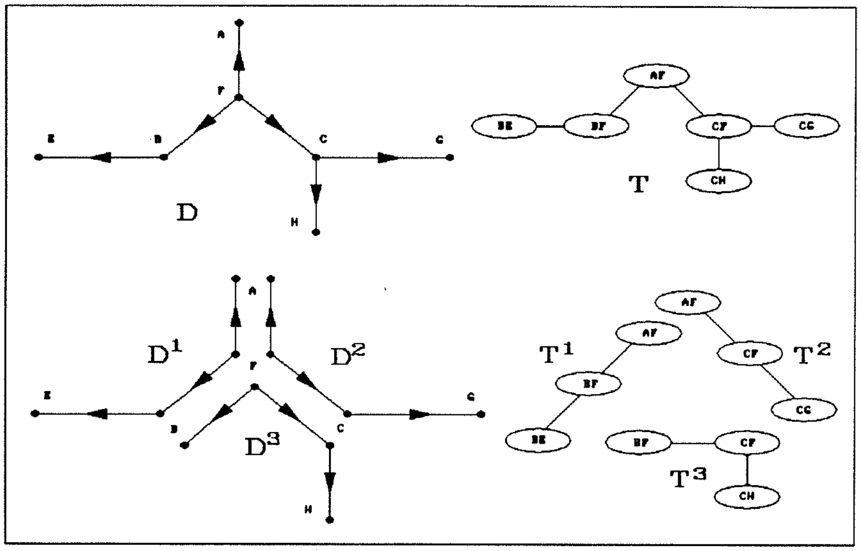


Figure 4.23: Top left: A DAG D. Top right: A junction tree T from D. Bottom left: $\{D^1, D^2, D^3\}$ forms a sound sectioning of D. Bottom right: The junction trees from the MSBN in Bottom left.

D^2, D^4, D^6, D^7, D^8 as 1 subDAG, $D^3, D^5, D^9, D^{10}, D^{11}$ as another, D^{12}, D^{13} and D^{14}, D^{15} as 2 others. Thus the MSBN is viewed as one with a global covering subDAG D^1. Likewise, when one is concerned with relation between D^{14} and D^{15}, the MSBN can be viewed as one satisfying the covering subDAG rule with $\beta = 2$. Therefore, the computation required for a MSBN of a hypertree structure is just the repetition of the computation required for a MSBN with a global covering subDAG. On the other hand, a MSBN with a global covering subDAG is a special case of the hypertree structure. Hence, the following study is often simplified by considering only one of the 2 cases.

4.7 Transform MSBN into Junction Forest

In order to perform efficient inference in a general but sparse network, it is desirable to transform each sect of a MSBN into a junction tree which will stand as an inference entity (Section 4.4). The transformation takes several steps to be discussed in this section. The set of subDAGs of the MSBN are morali-triangulated into a set of morali-triangulated graphs from which a set of clique hypergraphs are formed. Then the set of clique hypergraphs are organized into a set of junction trees of cliques. Afterwards, the linkages between the junction trees are created. Finally, belief tables are assigned to cliques and linkages and a junction forest of belief universes is constructed.

4.7.1 Transform SubDAGs into Junction Trees by Local Computation

The key issue is morali-triangulating subDAGs of a MSBN into a set of morali-triangulated graphs. Once this is done, the formation of clique hypergraph and the organization of junction tree for each subDAG are performed the same way as in the case of a USBN and a single junction tree [Andersen et al. 89, Jensen, Lauritzen and Olesen 90]. As mentioned before, the criterion in morali-triangulation of a set of subDAGs of a MSBN into a set of clique hypergraphs is to preserve the 'intactness' of the clique hypergraph resulted from the corresponding USBN. The concept of 'intactness' is formalized below.

Definition 9 (invertible morali-triangulation) *Let D be a DAG sectioned into $\{D^1, \ldots, D^\beta\}$ where D^i has domain N^i. If there exists a morali-triangulated graph G of D, with the clique hypergraph H, such that $G = \sqcup_{i=1}^{\beta} G^i$ where G^i is the subgraph of G induced by N^i, and $H = \sqcup_{i=1}^{\beta} H^i$ where H^i is the clique hypergraph of G^i, then the set of morali-triangulated graphs $\{G^1, \ldots, G^\beta\}$ is* **invertible**. *Also the transformation of $\{D^1, \ldots, D^\beta\}$ into $\{G^1, \ldots, G^\beta\}$ is said to be an invertible morali-triangulation.*

A morali-triangulated graph G is equivalent to the corresponding clique hypergraph H in that given one of them the other is completely determined. By definition 8, if the sectioning of a MSBN is sound, then there exists a clique hypergraph H whose every clique is a subset of the domain of at least 1 subDAG of the MSBN. Thus one has the following theorem.

Theorem 6 (existence of invertible morali-triangulation) *There exists an invertible morali-triangulation for $\{D^1, \ldots, D^\beta\}$ sectioned from a DAG D, iff the sectioning is sound.*

One could construct a set of invertible morali-triangulated graphs of a MSBN by first performing a global computation (moralization and triangulation) on D to find G, and then determining its subgraphs relative to the sectioning of the MSBN. The moralization and triangulation would be the same as in the junction tree technique with care to be taken not to mix nodes in different subDAGs into one clique. However, when space requirement is of concern, MSBNs offer the possibility of morali-triangulation by local computation at the level of subDAGs of sects. Each subDAG in a MSBN is morali-triangulated separately (message passing may be involved) such that the collection of them is invertible. The following discusses how this can be achieved.

Example 12 In the example depicted in Figure 4.17 and 4.20, Θ is sectioned into $\{\Theta^1, \Theta^2, \Theta^3\}$ by a sound sectioning and $\{\Lambda^1, \Lambda^2, \Lambda^3\}$ is a set of invertible morali-triangulated graphs relative to the sectioning. The desire is to find Λ^i $(i = 1, 2, 3)$ from Θ^i $(i = 1, 2, 3)$ by local computation.

Since subDAGs of a MSBN are interfaced through d-sepsets, the focus of finding a set of invertible morali-triangulated graphs by local computation is to decide whether each pair of d-sepnodes is to be linked. Coordination between neighbor subDAGs is necessary to ensure correct decisions. The following considers this systematically.

Call a link between 2 d-sepnodes a *d-link*. Call a simple path $(A_1, A_2, \ldots, A_k)$ a *d-path* if $A_1, \ldots, A_i$ and $A_j, \ldots, A_k$ $(1 \leq i, i+1 \leq j, j \leq k)$ are all d-sepnodes, while all the other nodes on the path are non-d-sepnodes. A d-link is a trivial d-path. Each morali-triangulated graph G^i from an invertible morali-triangulation may generally contain 6 types of d-links.

Arc type inherited from the subDAG. That is, if 2 d-sepnodes are connected originally in the subDAG, there is a d-link between them in G^i. Decision on this type of d-links is trivial.

ML type created by local moralization. For example, the d-links (H_1, H_2) in Λ^2 and (H_2, H_3) in Λ^3.

ME type created by moralization in neighbor subDAGs. For example, the d-links (H_1, H_2) and (H_2, H_3) in Λ^1. Decision on this type of d-links requires communication between neighbor subDAGs.

Cy type created to triangulate inter-subDAG cycles. For example, the d-link (H_3, H_4) in Λ^1 and Λ^3. Decision on this type of d-links requires communication between neighbor subDAGs.

TL type created during local triangulation. After the above 4 types of d-links have been introduced to the moral graph of a subDAG, there may still be un-triangulated cycles within the moral graph involving 4 or more d-sepnodes. The example used above is too simple to illustrate this and the next type.

TE type created by local triangulation in neighbor subDAGs. The triangulation of a cycle of length > 3 involving only d-sepnodes is not unique. If 2 neighbor subDAGs triangulate such a cycle by local computation without coordination, they may triangulate in different ways and result in different set of cliques for the nodes in the d-sepset. Therefore communication is required such that a subDAG may adopt the d-links introduced by triangulation in neighbor subDAGs. The argument also applies to the case of triangulating cycles consisting of general d-paths.

An algorithm for morali-triangulation of subDAGs of a MSBN into a set of invertible triangulated graphs under the covering subDAG assumption is given below.

Algorithm 2 (morali-triangulation with a covering subDAG) *Let D^1 be the covering subDAG in the MSBN.*

1. *For each subDAG D^i, do the following: (1) moralize D^i, and add d-links due to moralization to ML^i (a set of node pairs); (2) search for pairs of d-sepnodes connected by a d-path in the moral graph of D^i, and add the pairs found to Cy^i (a set of node pairs).*
2. *For D^1, do the following: (1) for each pair of nodes of D^1 contained in one of ML^i $(i > 1)$, connect the pair in the moral graph of D^1; and (2) for each pair of d-sepnodes contained in both Cy^1 and one of Cy^j $(j > 1)$, connect the 2 nodes in the moral graph of D^1; (3) triangulate the augmented moral graph of D^1 (the morali-triangulation of D^1 is completed); and (4) add d-links to $DLINK$ (a set of node pairs).*
3. *For D^i $(i = 2, \ldots, \beta)$, do the following: (1) for each pair of nodes of D^i contained in $DLINK$, connect the pair in the moral graph of D^i; and (2) triangulate the augmented moral graph of D^i. The morali-triangulation of D^i is completed.*

Note that the above process has 2 passes through all the subDAGs. The following theorem shows the invertibility of the morali-triangulation.

Theorem 7 (invertibility of Algorithm 2) *The morali-triangulation by Algorithm 2 is invertible.*

Proof:

The key issue is to decide on d-links. The algorithm considered only neighbor relations between the covering subDAG and other subDAGs. The first step is to show this is sufficient.

Let d_a and d_b be 2 d-sepnodes between 2 subDAGs D^a and D^b. Let D^1 be the covering subDAG. It is claimed that if d_a and d_b join 2 simple paths in D^a and D^b respectively to form an inter-subDAG cycle, then they must join 2 simple paths in D^a and D^1 as well. Since d_a and d_b both are contained in D^1 which is connected, it is certainly true.

Similarly, if there is a cycle involving d-sepnodes in the d-sepset between D^a and D^b, these d-sepnodes are also in the d-sepset between D^a and D^1.

Now it is a simple matter to show: the ME type d-links are introduced by step 2 (1) and step 3 (1); the Cy type d-links are introduced by step 2 (2) and step 3 (1); and the TE type d-links are introduced by step 2 (2) and step 3 (1).

□

Example 13 The following is a recount for morali-triangulation of $\sqcup_{i=1}^{3}\Theta^i$ in Figure 4.20 by Algorithm 2.

1. After step 1 of the algorithm, $ML^1 = \phi$, $ML^2 = \{(H_1, H_2)\}$, and $ML^3 = \{(H_2, H_3)\}$; $Cy^1 = \{(H_1, H_2), (H_2, H_3), (H_3, H_4)\}$, $Cy^2 = \{(H_1, H_2)\}$, and $Cy^3 = \{(H_2, H_3), (H_2, H_4), (H_3, H_4)\}$.
2. After step 2, the morali-triangulated graph Λ^1 of Θ^1 is completed by adding d-links $(H_1, H_2), (H_2, H_3), (H_3, H_4)$ to Θ^1's moral graph, and then triangulating (nothing is added). $DLINK$ will contain $\{(H_1, H_2), (H_2, H_3), (H_3, H_4)\}$.

3. After step 3, the morali-triangulated graph Λ^2 of Θ^1 is completed without change to its moral graph; the morali-triangulated graph Λ^3 of Θ^3 is completed by adding the d-link (H_3, H_4) to its moral graph, and then triangulating (with the link (E_3, H_4) added).

As mentioned in section 4.4, after the morali-triangulation, the other steps in transformation of a MSBN into a set of junction trees of cliques are: identifying cliques of the morali-triangulated graphs to form a set of clique hypergraphs and then organizing each hypergraph into a junction tree. These steps are performed in the same way as in the junction tree technique. Throughout the rest of the chapter, it is assumed that junction trees are obtained through a set of invertible triangulated graphs, and it is said that the junction trees are obtained by invertible transformation.

Call a set of junction trees of cliques from an invertible transformation of subDAGs of a MSBN as a *junction forest of cliques* denoted by $F = \{T^1, \ldots, T^\beta\}$ where T^i is the junction tree from the subDAG D^i.

4.7.2 Linkages between Junction Trees

Just as d-sepsets interface subDAGs, linkages interface junction trees transformed from subDAGs and serve as information channels between junction trees during inference. The extension of the MSBN technique to the junction tree technique is to allow multiple linkages between pairs of junction trees in a junction forest such that localization can be preserved within junction trees and the exponential explosion of the sizes of clique state spaces associated with the brute force method (section 4.3) can be avoided.

Definition 10 (linkage set) *Let I be the d-sepset between 2 subDAGs D^a and D^b. Let T^a and T^b be the junction trees transformed from D^a and D^b respectively. A* **linkage** *of T^a relative to T^b is a set l of nodes such that the following 2 conditions hold.*

1. *Boundary: there exists a clique $C_x \in T^a$ such that $l = C_x \cap I$. C_x is called a* **host clique** *of l;*

2. *Maximum: there is no subset of l that is also a linkage.*

In general there may be more than one linkage between a pair of junction trees. Define L^{ab} to be the set of all linkages of T^a relative to T^b.

Proposition 11 (identity of linkages) *Let T^a and T^b be the junction trees from sub-DAGs D^a and D^b respectively. If L^{ab} is the set of linkages of T^a relative to T^b and L^{ba} is the set of linkages of T^b relative to T^a, then $L^{ab} = L^{ba}$.*

Proof:

A linkage consists of the d-sepnodes which are pairwise connected. In an invertible morali-triangulation, d-sepnodes are connected identically in both morali-triangulated graphs involved.

□

Example 14 In Figure 4.20, linkages between junction trees are indicated with ribbed bands connecting the corresponding host cliques. The 2 linkages between Γ^1 and Γ^3 are $\{H_3, H_2\}$ and $\{H_3, H_4\}$.

Given a set of linkages between a pair of junction trees, the concept of a redundancy set can be defined. As mentioned in section 4.4, redundancy sets provide structures which allow redundant information to be removed during inter-junction tree information passing. The concept will be used for defining joint system belief in section 4.7.3 and defining the operation NonRedundancyAbsorption in section 4.9.2. To define the redundancy set, one needs to index linkages such that the redundancy sets defined based on the indexing possess certain desirable property which will become clear below. Index a set L^{ab} of linkages by the following procedure.

Procedure 1

1. Pick one of the junction trees in the pair, say T^a. Create a tree G with nodes labeled by linkages in L^{ab}. Connect 2 nodes in G by a link if either the hosts of corresponding linkages are directly connected in T^a, or the hosts of corresponding linkages are (indirectly) connected in T^a by a path on which all intermediate cliques are not linkage hosts. Call this tree a **linkage tree**.
2. Index the nodes (linkages in L^{ab}) of G into $L_1, L_2, \ldots$ in any order that is consistent with G, i.e., for every $i > j$ there is a unique predecessor $j(i) < i$ such that $L_{j(i)}$ is adjacent to L_i in G.

With linkages indexed this way, the redundancy set can be defined as the following.

Definition 11 (redundancy set) *Let a set of linkages $L^{ab} = \{L_1, \ldots, L_g\}$ be indexed by procedure 1. Then for this set of indexed linkages, a* **redundancy set** *R_i for index i is defined as*

$$R_i = \begin{cases} \phi & \text{if } i = 1 \\ L_i \cap L_{j(i)} & i > 1;\ j(i) < i \text{ and } L_{j(i)} \text{ adjacent to } L_i \text{ in linkage tree } G \end{cases}$$

Note that a linkage tree is itself a junction tree, and redundancy sets are sepsets of the linkage tree.

Example 15 There are 2 linkages between Γ^1 and Γ^3 in Figure 4.20. Consider junction tree Γ^3. The linkage tree G has 2 connected nodes, one labeled by the linkage $\{H_3, H_2\}$ and the other by $\{H_3, H_4\}$. An indexing $L_1 = \{H_3, H_2\}$ and $L_2 = \{H_3, H_4\}$ defines 2 redundancy sets $R_1 = \phi$ and $R_2 = \{H_3\}$.

With linkages and redundancy sets constructed, one has a *linked junction forest of cliques*.

4.7.3 Joint System Belief of Junction Forest

Let (D, P) be a USBN, $S = \{S^1, \ldots, S^\beta\}$ be a corresponding MSBN with a covering sect S^1, and $F = \{T^1, \ldots, T^\beta\}$ be the junction forest from an invertible transformation. Let T^i be the junction tree of D^i with cliques C^i and sepsets Q^i. Let I^i $(i > 1)$ be the d-sepset between S^i and S^1. Let L^i $(i > 1)$ be the set of linkages between T^i and T^1; and R^i $(i > 1)$ be the corresponding set of redundancy sets.

Construct a joint system belief for the junction forest through an assignment of belief tables to each clique, clique sepset, linkage and redundancy set in the junction forest. First, for each junction tree T^i in F, do the following.

- Assign each node $n_k \in N^i$ to a unique clique $C_x \in C^i$ such that C_x contains n_k and its parents π_k. Break ties arbitrarily.
- Let P_k denote the probability table associated with node n_k. For each clique C_x that has assigned nodes $n_k, \ldots, n_l$, associate it with belief table $B(C_x) = P_k * \ldots * P_l$.
- For each clique sepset $Q_y \in Q^i$, associate it with constant belief table $B(Q_y)$.

Then for each set of linkage L^i, do the following.

- Associate each linkage $L_z \in L^i$ with constant belief table $B(L_z)$.

Define the belief table for redundancy set $R_z \in R^i$ as

$$B(R_z) = \sum_{L_z \setminus R_z} B(L_z)$$

Define the belief table for d-sepset I^i as

$$B(I^i) = \frac{\prod_{L_z \in L^i} B(L_z)}{\prod_{R_z \in R^i} B(R_z)}$$

Define the belief table for each junction tree T^i as

$$B(T^i) = \frac{\prod_{C_x \in C^i} B(C_x)}{\prod_{Q_y \in Q^i} B(Q_y)}$$

Here the notation $B(T^i)$ is used in stead of $B(N^i)$ to emphasize that it is related to the junction tree. Note $B(I^i)$ and $B(T^i)$ are mathematical objects which do not have corresponding data structures in the knowledge base. Comparing the form of joint probability distribution for an USBN (section 1.3.3) and the assignment of probability tables for nodes in a sect (Definition 7), it is easy to see that $B(T^i)$ is proportional to the joint probability distribution of S^i relative to that assignment.

Define the *joint system belief* for the junction forest F as

$$B(F) = \frac{\prod_{i=1}^{\beta} B(T^i)}{\prod_{i=2}^{\beta} B(I^i)}$$

The notation $B(F)$ in stead of $B(N)$ is used for the same reason as above. With this definition, one has the following lemma.

Lemma 7 *The joint belief $B(F)$ of a MSBN is proportional to the joint probability distribution P of the corresponding USBN.*

To see this is true, it suffices to indicate that each d-sepnode, appearing in at least 2 sects, carries its original probability table as in (N, E, P) exactly once by Definition 7, and carries uniform table for the rest.

Example 16 Table 4.5 lists the constructed belief tables for belief universes of junction forest $F = \{\Gamma^1, \Gamma^2, \Gamma^3\}$ in Figure 4.20. The belief tables for sepsets, linkages, and redundancy sets are all constant tables at this stage.

Having constructed belief tables for cliques, sepsets, linkages, redundancy sets and junction forest, using the definition of world of section 4.2.1, one can talk about *belief*

$B(\Gamma^1)$		$B(\Gamma^2)$		$B(\Gamma^3)$	
Clique	*NodeAss.*	*Clique*	*NodeAss.*	*Clique*	*NodeAss.*
$\{H_2, H_1, A_1\}$	H_1, A_1	$\{F_2, F_1\}$	F_2	$\{E_1, E_3\}$	E_1
Config.	$B()$	*Config.*	$B()$	*Config.*	$B()$
$\{h_{21}, h_{11}, h_{11}\}$	.12	$\{f_{21}, f_{11}\}$	.4	$\{e_{11}, e_{31}\}$	.2
$\{h_{21}, h_{11}, h_{12}\}$	.03	$\{f_{21}, f_{12}\}$	.75	$\{e_{11}, e_{32}\}$	.7
$\{h_{21}, h_{12}, h_{11}\}$	.085	$\{f_{22}, f_{11}\}$	.6	$\{e_{12}, e_{31}\}$	.8
$\{h_{21}, h_{12}, h_{12}\}$	.765	$\{f_{22}, f_{12}\}$	.25	$\{e_{12}, e_{32}\}$	.3
$\{h_{22}, h_{11}, h_{11}\}$	.12	*Clique*	*NodeAss.*	*Clique*	*NodeAss.*
$\{h_{22}, h_{11}, h_{12}\}$	.03	$\{H_2, F_1, H_1\}$	H_2, F_1, H_1	$\{H_3, H_2, E_3\}$	H_3, H_2, E_3
$\{h_{22}, h_{12}, h_{11}\}$	.085	*Config.*	$B()$	*Config.*	$B()$
$\{h_{22}, h_{12}, h_{12}\}$	.765	$\{h_{21}, f_{11}, h_{11}\}$	.7895	$\{h_{31}, h_{21}, e_{31}\}$	.7702
Clique	*NodeAss.*	$\{h_{21}, f_{11}, h_{12}\}$	.6	$\{h_{31}, h_{21}, e_{32}\}$	.2298
$\{H_2, A_2, A_1\}$	H_2	$\{h_{21}, f_{12}, h_{11}\}$	.2105	$\{h_{31}, h_{22}, e_{31}\}$	.65
Config.	$B()$	$\{h_{21}, f_{12}, h_{12}\}$	.4	$\{h_{31}, h_{22}, e_{32}\}$	.35
$\{h_{21}, a_{21}, a_{11}\}$	.8696	$\{h_{22}, f_{11}, h_{11}\}$	.5	$\{h_{32}, h_{21}, e_{31}\}$	.35
$\{h_{21}, a_{21}, a_{12}\}$	.7	$\{h_{22}, f_{11}, h_{12}\}$	.05	$\{h_{32}, h_{21}, e_{32}\}$	.65
$\{h_{21}, a_{22}, a_{11}\}$	.6	$\{h_{22}, f_{12}, h_{11}\}$	.5	$\{h_{32}, h_{22}, e_{31}\}$	.01
$\{h_{21}, a_{22}, a_{12}\}$	.08	$\{h_{22}, f_{12}, h_{12}\}$	.95	$\{h_{32}, h_{22}, e_{32}\}$	.99
$\{h_{22}, a_{21}, a_{11}\}$	.1304			*Clique*	*NodeAss.*
$\{h_{22}, a_{21}, a_{12}\}$	.3			$\{E_2, E_3, E_4\}$	E_2
$\{h_{22}, a_{22}, a_{11}\}$	.4			*Config.*	$B()$
$\{h_{22}, a_{22}, a_{12}\}$	.92			$\{e_{21}, e_{31}, e_{41}\}$	.9789
Clique	*NodeAss.*			$\{e_{21}, e_{31}, e_{42}\}$	.8
$\{H_3, H_2, A_2\}$	H_3, A_2			$\{e_{21}, e_{32}, e_{41}\}$	.9
Config.	$B()$			$\{e_{21}, e_{32}, e_{42}\}$	.05
$\{h_{31}, h_{21}, a_{21}\}$	.24			$\{e_{22}, e_{31}, e_{41}\}$	.0211
$\{h_{31}, h_{21}, a_{22}\}$	.06			$\{e_{22}, e_{31}, e_{42}\}$	.2
$\{h_{31}, h_{22}, a_{21}\}$	.24			$\{e_{22}, e_{32}, e_{41}\}$	.1
$\{h_{31}, h_{22}, a_{22}\}$	.06			$\{e_{22}, e_{32}, e_{42}\}$	.95
$\{h_{32}, h_{21}, a_{21}\}$	.07			*Clique*	*NodeAss.*
$\{h_{32}, h_{21}, a_{22}\}$	.63			$\{E_3, E_4, H_4\}$	E_4, H_4
$\{h_{32}, h_{22}, a_{21}\}$	.07			*Config.*	$B()$
$\{h_{32}, h_{22}, a_{22}\}$	.63			$\{e_{31}, e_{41}, h_{41}\}$	.8
Clique	*NodeAss.*			$\{e_{31}, e_{41}, h_{42}\}$	.15
$\{H_3, A_3, H_4\}$	A_3, H_4			$\{e_{31}, e_{42}, h_{41}\}$	.2
Config.	$B()$			$\{e_{31}, e_{42}, h_{42}\}$	.85
$\{h_{31}, a_{31}, h_{41}\}$	.075			$\{e_{32}, e_{41}, h_{41}\}$	.8
$\{h_{31}, a_{31}, h_{42}\}$	.225			$\{e_{32}, e_{41}, h_{42}\}$	.15
$\{h_{31}, a_{32}, h_{41}\}$	.28			$\{e_{32}, e_{42}, h_{41}\}$	.2
$\{h_{31}, a_{32}, h_{42}\}$	.42			$\{e_{32}, e_{42}, h_{42}\}$	.85
$\{h_{32}, a_{31}, h_{41}\}$	.2			*Clique*	*NodeAss.*
$\{h_{32}, a_{31}, h_{42}\}$	.6			$\{H_3, E_3, H_4\}$	
$\{h_{32}, a_{32}, h_{41}\}$	.08			*Config.*	$B()$
$\{h_{32}, a_{32}, h_{42}\}$	.12			$\{h_{31}, e_{31}, h_{41}\}$	1
Clique	*NodeAss.*			$\{h_{31}, e_{31}, h_{42}\}$	1
$\{H_4, A_4\}$	A_4			$\{h_{31}, e_{32}, h_{41}\}$	1
Config.	$B()$			$\{h_{31}, e_{32}, h_{42}\}$	1
$\{h_{41}, a_{41}\}$	.9			$\{h_{32}, e_{31}, h_{41}\}$	1
$\{h_{41}, a_{42}\}$	.1			$\{h_{32}, e_{31}, h_{42}\}$	1
$\{h_{42}, a_{41}\}$	.2			$\{h_{32}, e_{32}, h_{41}\}$	1
$\{h_{42}, a_{42}\}$	.8			$\{h_{32}, e_{32}, h_{42}\}$	1

Table 4.5: Constructed belief tables for belief universes of junction forest $F = \{\Gamma^1, \Gamma^2, \Gamma^3\}$ in Figure 4.20. Config: Configuration. Node Ass: Nodes Assigned.

universes, sepset worlds, linkage worlds, redundancy worlds, and junction forest of belief universes. These terms will be used below.

The preceding has an assumption of a covering sect. The joint system belief of a junction forest with a hypertree structure can be defined in the similar way. As there is no need to consider the d-sepset/linkages between non-covering sects above, in the hypertree case, there is no need to consider the d-sepset/linkages between neighbor sects covered by a local covering sect. Therefore in practice, these linkages are never created. One sees another computational advantage of the covering sect rule and the hypertree rule.

4.8 Consistency and Separability of Junction Forest

Part of the goal is to propagate the information stored in different belief universes in different junction trees of a junction forest to the whole system such that marginal probability of variables can be obtained from any universes containing them with local computation.[4] The information to be propagated can be the prior knowledge in the form of products of original probability tables from the corresponding USBN. The information to be propagated can also be evidence entered from a set of universes possibly in different junction trees. The following defines consistency and separability that are the properties of junction forests which guarantee the achievement of this goal.

4.8.1 Consistency of Junction Forest

The property of consistency partly guarantees the validity of obtaining marginal probabilities by local computation. In the context of the junction forest, 3 levels of consistency can be identified.

[4] Obtaining marginals by local computation is what the junction tree technique is developed for. More is obtained from junction forests, namely, exploiting localization.

Definition 12 (local consistency) *Neighbor universes $(C_i, B(C_i))$ and $(C_j, B(C_j))$ in a junction tree T^i with sepset world $(Q_k, B(Q_k))$ are consistent if*

$$\sum_{C_i \backslash C_j} B(C_i) \propto B(Q_k) \propto \sum_{C_j \backslash C_i} B(C_j)$$

where '$\propto$' reads 'proportional to'. When the relation holds among all neighbor universes, the junction tree T^i is said to be consistent. When all junction trees are consistent, a junction forest F is said to be **locally consistent**.

Definition 13 (boundary consistency) *Host universes $(C^i_x, B(C^i_x))$ and $(C^j_y, B(C^j_y))$ of linkage world $(L_k, B(L_k))$ are consistent if*

$$\sum_{C^i_x \backslash C^j_y} B(C^i_x) \propto B(L_k) \propto \sum_{C^j_y \backslash C^i_x} B(C^j_y)$$

When the relation holds among all linkage host universes, a junction forest is said to have reached **boundary consistency**.

Definition 14 (global consistency) *A junction forest is said to be* **globally consistent** *if for any 2 belief universes (possibly in different junction trees) $(C^i_x, B(C^i_x))$ and $(C^j_y, B(C^j_y))$*

$$\sum_{C^i_x \backslash C^j_y} B(C^i_x) \propto \sum_{C^j_y \backslash C^i_x} B(C^j_y)$$

Theorem 8 (consistent junction forest) *A junction forest is globally consistent iff it reaches both local consistency and boundary consistency.*

Proof:

The necessity is obvious. The sufficiency is proven below.

Let $(C^i_x, B(C^i_x))$ and $(C^j_y, B(C^j_y))$ be 2 universes in junction trees T^i and T^j of junction forest F respectively. Let I^{ij} be the d-sepset between the 2 corresponding sects. One has

$Z = C_x^i \cap C_y^j \subseteq I^{ij}$. By definition of linkages, they have the maximum property. Therefore, there exists a linkage $L_w \supseteq Z$ with its hosts being $(C_w^i, B(C_w^i))$ and $(C_w^j, B(C_w^j))$.

Since $C_w^i \supseteq L_w \supseteq Z$, Z is contained in all cliques in the unique path between C_x^i and C_w^i. Because S is locally consistent,

$$\sum_{C_x^i \setminus Z} B(C_x^i) \propto \sum_{C_w^i \setminus Z} B(C_w^i).$$

Similarly,

$$\sum_{C_w^j \setminus Z} B(C_w^j) \propto \sum_{C_y^j \setminus Z} B(C_y^j).$$

Because T also reached boundary consistency,

$$\sum_{C_w^i \setminus Z} B(C_w^i) \propto \sum_{C_w^j \setminus Z} B(C_w^j)$$

Therefore

$$\sum_{C_x^i \setminus Z} B(C_x^i) \propto \sum_{C_y^j \setminus Z} B(C_y^j)$$

$\Box$

4.8.2 Separability of Junction Forests

In the junction tree technique, consistency is all that is required in order to obtain marginals by local computation. In junction forests, this is not sufficient due to the existence of multiple linkages. The function of multiple linkages between a pair of junction trees is to pass the joint distribution on the d-sepset by passing marginal distributions on subsets of the d-sepset. By doing so, one avoids the exponential increase in clique state space sizes as outlined in section 4.3. When breaking down the joint into the marginals, one must ensure the joint can be reassembled from the marginals, i.e., one must pass a correct version of the joint. Otherwise, the correctness of local computation is not guaranteed. Since passing the marginals is achieved by passing the belief tables on linkages and redundancy sets, the structure of linkage hosts is the key factor. The following defines separability of junction forests in terms of the correctness of local computation. Then the structural condition of linkage hosts is given under which the separability holds.

Definition 15 (separability) *Let $F = \{T^i | 1 \leq i \leq \beta\}$ be a junction forest with domain N and joint system belief $B(F)$. F is said to be* **separable** *if, when it is globally consistent, for any T^i over subdomain N^i*

$$\sum_{N \setminus N^i} B(F) \propto B(T^i)$$

The following lemma is quoted from Jensen for latter proof of proposition 12.

Lemma 8 *[Jensen 88]*

Let T be a junction tree from clique hypergraph $(N, \mathbf{C})$. Let C_1 and C_2 be 2 adjacent cliques in T. Let T' be the graph resulting from T by union of C_1 and C_2 into one clique, and by keeping the original sepsets. Then T' is a junction tree for clique hypergraph $(N, (\mathbf{C} \setminus \{C_1, C_2\}) \cup \{C_1 \cup C_2\})$.

The following is the structural condition for separability to be proved below.

Definition 16 (host composition) *Let a MSBN by sound sectioning be transformed into a junction forest. Let S^i be a sect in the MSBN; T^i be the junction tree of S^i; I be the d-sepset between S^i and any distinct sect; and L be the set of linkages between T^i and the junction tree for the other sect.*

Recursively remove from T^i every leaf clique which is not a linkage host relative to L. Call the resultant junction tree as a **host tree***.*

T^i satisfies a **host composition** *condition relative to L, whenever the following is true.*

1. *If a non-d-sepnode is contained in any linkage host in T^i, then it is contained in exactly 1 linkage host; and*
2. *if a set of non-d-sepnodes are contained in any non-host clique in the host tree, and each element appears in some host clique, then the set must be contained in exactly 1 linkage host.*

Example 17 The host composition condition is violated in the host trees of Figure 4.24. The following shows the violation and the resultant problem. Assume both trees are consistent.

First consider the top tree. Let L consist of linkages $L_1 = \{A, D\}$ and $L_2 = \{A, E\}$. Let their hosts be $C_1 = \{A, B, D\}$ and $C_2 = \{A, B, E\}$ which are adjacent in the tree. B is a common non-d-sepnode - a violation of part 1 of the host composition condition. Even if all the belief tables are consistent, in general,

$$\sum_{B} \frac{B(ABD)B(ABE)}{B(AB)} \not\propto \frac{B(AD)B(AE)}{B(A)}$$

That is, the joint distribution on the d-sepset $\{A, D, E\}$ constructed from belief tables on linkages and redundancy sets is inconsistent in general.

Consider the bottom tree. Let L and C_1 be the same. Let the host $C_2 = \{A, E, G\}$ which is connected to C_1 through a non-host $C_3 = \{A, B, G\}$. $\{B, G\}$ is a set of non-d-sepnodes violating the part 2 of the host composition condition. If C_1 and C_3 are united as described in lemma 8, the resultant graph is still a junction tree. If let

$$B(C_{13}) = B(C_1)B(C_3)/B(Q_{13})$$

where $B(Q_{13})$ is the original belief for sepset between C_1 and C_3, the joint belief for the new tree is exactly the same as before and the new tree is consistent. Now the common node G in C_{13} and C_2 creates the same problem illustrated above.

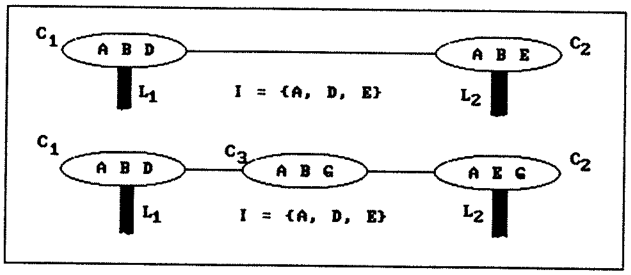


Figure 4.24: Two example trees for violation of the host composition condition. I: the d-sepset. The ribbed bands indicate linkages.

Proposition 12 *Let a MSBN by sound sectioning be transformed into a junction forest F. Let S^i be a sect in the MSBN; T^i be the junction tree of S^i in F; I be the d-sepset between S^i and any distinct sect; and L be the set of linkages between T^i and the junction tree for the other sect. Let all belief tables be defined as in section 4.7.3.*

When F is globally consistent, $B(I)$ satisfies

$$B(I) \propto \sum_{N^i \setminus I} B(T^i)$$

iff T^i satisfies the host composition condition relative to L.

Proof:

[Sufficiency] The proof is constructive. Denote the host tree of T^i relative to L by T' and denote T''s domain by N'. Marginalize $B(T^i)$ with respect to $N^i \setminus N'$. This is done by marginalization of $B(T^i)$ recursively with respect to unique variables in each leaf clique not in T'. This results in

$$\sum_{N^i \setminus N'} B(T^i) = B(T')$$

where $B(T')$ is the joint belief for the consistent junction tree T'. Note $N^i \setminus N'$ contains no d-sepnodes.

Second, unite non-host cliques into linkage hosts in T'. Suppose C_i is a non-linkage-host with host clique neighbor(s). Choose the neighbor host C_j containing all the non-d-sepnodes of C_i which appear in host cliques. Since T' is a junction tree and the host composition condition holds, one is guaranteed to find such a C_j. By lemma 8, the graph resulting from T' by union of C_i, C_j into a new clique C_k, and by keeping the original sepsets is still a junction tree on domain N'. The host composition condition still holds in the new graph. If let

$$B(C_k) = B(C_i)B(C_j)/B(Q_{ij})$$

where $B(Q_{ij})$ is the original belief for sepset between C_i and C_j, the joint belief for the new junction tree $T(1)$ is exactly the same as $B(T')$ and $T(1)$ is consistent. Repeat the union operation for all non-hosts, one ends up with a consistent junction tree T'' with only linkage hosts (possibly new composition) and every non-d-sepnode in N' appears in exactly 1 host. If for every clique in T'', marginalize each clique belief with respect to its (unique) non-d-sepnodes; remove these non-d-sepnodes from the clique; assign the marginalized belief to the new clique and keep the sepset belief invariant, one ends up with a new consistent junction tree T''' with its joint belief

$$BT''' \propto \sum_{N^i \setminus I} B(T^i).$$

Note that T''' is just the linkage tree in the procedure 1. That is, all the cliques in T''' are linkages, and all the sepsets are redundancy sets. Therefore,

$$B(I) \propto B(T''').$$

[Necessity] Suppose there are 2 or more linkages in L, and the host composition condition does not hold in T^i. Obtain the host tree T' in the same way as in the sufficiency proof. Recursively unite each non-host clique with a neighbor host clique if there is one, and assign the belief for the new clique in the same way as in the sufficiency proof. The resultant is a consistent junction tree T''. Since the host composition condition does not hold in T', and the uniting process does not remove non-d-sepnodes from any host, there is at least 2 neighboring cliques C_1 and C_2 in T'' such that they have a set of common non-d-sepnodes. Denote the 2 cliques as $C_1 = X \cup Y \cup W$ and $C_2 = X \cup Z \cup W$ (Figure 4.25) with $L_1 = C_1 \cap I = X \cup Y$ and $L_2 = C_2 \cap I = X \cup Z$. That is, C_1 and C_2 share a set of d-sepnodes X and a set of non-d-sepnodes W. Even if all the belief tables are consistent, in general,

$$\sum_W \frac{B(C_1)B(C_2)}{B(Q_{12})} = \sum_W \frac{B(X \cup Y \cup W)B(X \cup Z \cup W)}{B(X \cup W)}$$
$$\not\propto \frac{B(X \cup Y)B(X \cup Z)}{B(X)} = \frac{B(L_1)B(L_2)}{B(R_i)}$$

where R_i is the intersection of L_1 and L_2 (a redundancy set). That is, the distribution on $I \cap (C_1 \cup C_2)$ is not consistent with the distribution constructed from the belief tables on the corresponding linkages and redundancy set in general. The above C_1 and C_2 are chosen not including unique non-d-sepnodes in each of them. If this is not the case, these non-d-sepnodes can always be removed by marginalization at each of the 2 cliques, and the result is the same. If $L = \{L_1, L_2\}$, the proof is complete.

Consider the case L contains more than 2 linkages. In this case, the the left side of the above equation represents a correct version of the marginal distribution on a subset of d-sepset I, while the right side is an inconsistent version of the same distribution defined in terms of belief of linkages and redundancy sets (section 4.7.3). Now, it suffices to indicate that, in general, if the marginal distribution is inconsistent, the joint is also inconsistent.

□

The proposition shows that the belief table $B(I)$ defined in section 4.7.3 is indeed the joint belief on d-sepset I when the host composition condition is satisfied and the forest

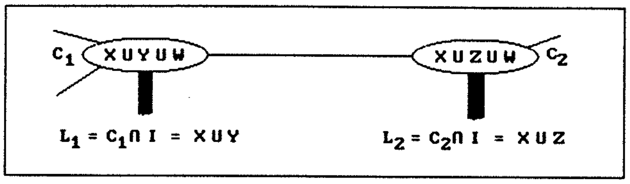


Figure 4.25: Part of a host tree violating the host composition condition. I: the d-sepset. The ribbed bands indicate linkages.

is consistent. Now it is ready for the following result on separability.

Theorem 9 (host composition) *Let $\{S^1, \ldots, S^\beta\}$ be a MSBN satisfying the hypertree condition. Let $F = \{T^i | 1 \leq i \leq \beta\}$ be a corresponding junction forest with joint system belief $B(F)$. F is separable iff the host composition condition is satisfied in all pairs of junction trees with linkages constructed.*[5]

Proof:

Assume F has reached global consistency. First, unite all the cliques in each junction tree into a huge clique with the belief table for the junction tree as its belief. Connect the huge cliques as they are connected in the hypertree (that is, if the linkages between 2 trees are not created as discussed in section 4.7.3, do not connect the 2 corresponding huge cliques), and assign the original $B(I)$ to their sepset. The resultant graph is a junction tree due the hypertree structure of the MSBN. By proposition 12, the tree is consistent with joint belief being $B(F)$, and for any T^i over subdomain N^i

$$\sum_{N \setminus N^i} B(F) \propto B(T^i)$$

iff the host composition condition is satisfied.

□

[5] Recall that when a MSBN has a hypertree structure, no linkage is created between junction trees whose corresponding sects are covered by a local covering sect.

The host composition condition can usually be satisfied naturally in an application system. Since d-sepsets are the only media for information exchange between sects, d-sepnodes usually involve many inter-subDAG cycles. The consequence is that they will be heavily connected during morali-triangulation and form several large cliques in the clique hypergraph as well as some small ones. On the other hand, non-d-sepnodes rarely form connections with so many d-sepnodes simultaneously and hence will rarely be the elements of these large cliques. To be an element of more than 1 such large clique is even more unlikely. Because linkages are defined to be maximal, these large cliques will become linkage hosts.

For example, in the PAINULIM application system (Chapter 5), there are 3 sects and correspondingly 3 junction trees. The host composition condition is satisfied naturally in all 3 trees. Figure 4.26 gives one of them. The 4 linkage hosts contain no non-d-sepnode at all.

When the host composition condition can not be satisfied naturally, one can add dummy links between d-sepnodes in the moral graph before triangulation such that linkage hosts will be enlarged and the condition is satisfied. Hence, given a MSBN, a separable junction forest can always be realized. The penalty of added links is increased amount of computation during belief propagation due to increased sizes of cliques and linkages. In the worst case, one resorts to the brute force method discussed in section 4.3 in order to satisfy the host composition condition for certain pairs of junction trees. If the system is large, sectioning may still yield computational savings on the whole even if cliques are enlarged at a few junction trees.

One of the key results now follows.

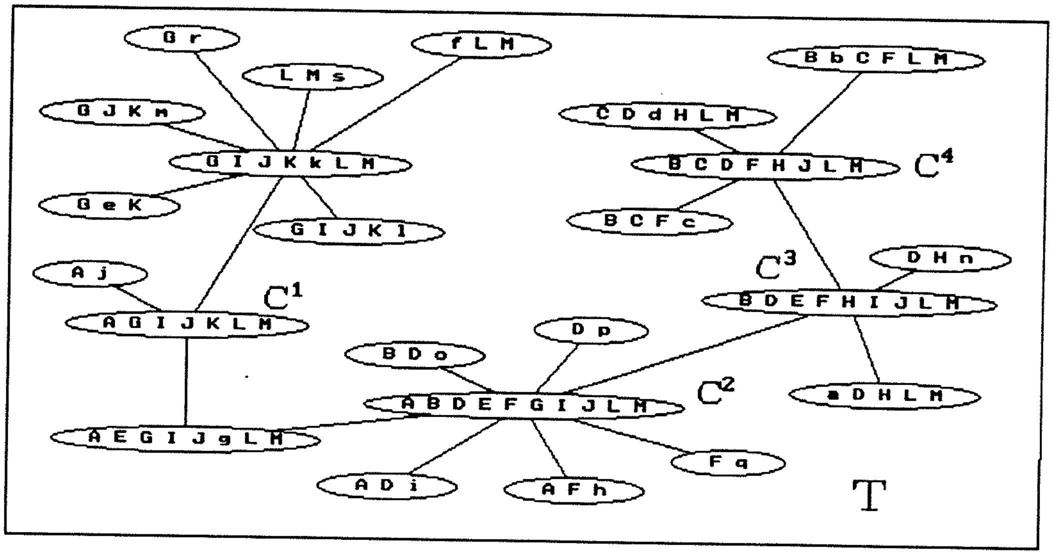


Figure 4.26: T is a junction tree in a junction forest taken from an application system PAINULIM with variable names revised to simplify. An upper case letter in a clique represents a d-sepnode member, and a lower case letter represents a non-d-sepnodes member. The cliques C_1, C_2, C_3, C_4 are linkage hosts.

Theorem 10 (local computation) *Let F be a consistent and separable junction forest with domain N and joint system belief $B(F)$. Let $(C_x, B(C_x))$ be any universe in F. Then*

$$\sum_{N \setminus C_x} B(F) \propto B(C_x)$$

Proof:

Let $(C_x, B(C_x))$ be a universe in one of the junction tree T^x of F, Let the subdomain of T^x be N^x and its belief be $B(T^x)$. Since F is consistent and separable, by definition of separability one has

$$\sum_{N \setminus N^x} B(F) = B(T^x)$$

and T^x is a consistent junction tree. Jensen, Lauritzen and Olesen [1990] have proved that the belief of a consistent junction tree marginalized to any of its universes is proportional to the belief of that universe. Hence

$$\sum_{N\setminus C_x} B(F) \propto \sum_{N^x\setminus C_x} (\sum_{N\setminus N^x} B(F)) \propto B(C_x)$$

□

With the above theorem, the marginal belief of any variable in a consistent and separable junction forest can be computed by marginalization of the belief table of any universe which contains the variable. In this respect, a consistent and separable junction forest behaves the same as a consistent junction tree [Jensen, Lauritzen and Olesen 90]. It will be seen that, in the context of junction forests, additional computational advantage is available, i.e., the global consistency is not necessary for obtaining marginal belief by local computation.

4.9 Belief Propagation in Junction Forests

Given the importance of consistency of junction forests, a set of operations are introduced which bring a junction forest into consistency. Since the purpose is to exploit localization, only operations for local computation at the junction tree level are considered. That is, at any moment, there is only 1 junction tree resident in the memory. This junction tree is said to be *active*.

4.9.1 Supportiveness

Jensen, Lauritzen and Olesen [1990] introduced the concept of supportiveness. Let $(Z, B(Z))$ be a world. The *support* of $B(Z)$ is defined as

$$\Delta(B(Z)) = \{z \in \Psi(Z)|\text{belief of z} > 0\}.$$

A junction tree is *supportive*, if, for any universe $(C_i, B(C_i))$ and for any neighboring sepset world $(Q_j, B(Q_j))$, $\Delta(B(Ci)) \subseteq \Delta(B(Q_j))$. The underlying intuition is that, when beliefs are propagated in a supportive junction tree, non-zero belief values will not be turned into zeros.

Here the concept is extended to junction forests. A junction forest is supportive, if all its junction trees are supportive, and if, for any linkage host $(C_i, B(C_i))$ and corresponding linkage world $(L_j, B(L_j))$, $\Delta(B(Ci)) \subseteq \Delta(B(Lj))$.

The construction in section 4.7.3 results in a supportive junction forest.

4.9.2 Basic Operations

Operations for Consistency within a Junction Tree

The following operation brings a pair of belief universes into consistency.

Operation 1 (AbsorbThroughSepset) *[Jensen, Lauritzen and Olesen 90]* *Let $(C_0, B(C_0))$ and its neighbors $(C_1, B(C_1)), \ldots, (C_k, B(C_k))$ be belief universes in a junction tree. Let $(Q_1, B(Q_1)), \ldots, (Q_k, B(Q_k))$ be the corresponding sepset worlds. Suppose $\Delta(B(C_0)) \subseteq \Delta(B(Q_i))$ $(i = 1, \ldots, k)$. Then the* **AbsorbThroughSepset** *of $(C_0, B(C_0))$ from $(C_i, B(C_i))$ $(i = 1, \ldots, k)$ changes $B(Q_i)$ and $B(C_0)$ into $B'(Q_i)$ and $B'(C_0)$.*

$$\begin{aligned} B'(Q_i) &= \sum_{C_i \setminus C_0} B(C_i) \quad i = 1, \ldots, k \\ B'(C_0) &= B(C_0) \prod_{i=1}^{k} B'(Q_i)/B(Q_i) \end{aligned}$$

This operation is performed at the level of belief universe. Jensen, Lauritzen and Olesen [1990] show that the operation changes neither the supportiveness of a junction tree nor the joint system belief in the context of a junction tree. In the context of a junction forest, the supportiveness is also invariant after AbsorbThroughSepset. This is because the operation does not increase the support of any linkage host and does not change linkage beliefs directly. The invariance of joint system belief for the junction forest is obvious given the definition of joint system belief and the invariance of beliefs for junction trees.

The following are three high level operations which bring a junction tree into consistency.

Operation 2 (DistributeEvidence) *[Jensen, Lauritzen and Olesen 90]*
Let $(C_0, B(C_0))$ be a universe in a junction tree. When **DistributeEvidence** *is called in $(C_0, B(C_0))$, it performs AbsorbThroughSepset to absorb from the caller if the caller is a neighbor and calls DistributeEvidence in all its neighbors except the caller.*

Suppose a junction tree is originally consistent. If evidence is entered[6] in one of its belief universes and DistributeEvidence is initiated from that universe, then the resulting junction tree is still consistent.

Operation 3 (CollectEvidence) *[Jensen, Lauritzen and Olesen 90]*
Let $(C_0, B(C_0))$ be a universe in a junction tree. When **CollectEvidence** *is called in $(C_0, B(C_0))$, it calls CollectEvidence in all its neighbors except the caller, and when they have finished their CollectEvidence, $(C_0, B(C_0))$ performs AbsorbThroughSepset to absorb from them.*

DistributeEvidence and CollectEvidence are operations performed at the level of belief universes. Since they are composed of AbsorbThroughSepset, they do not change the supportiveness and joint system belief of junction forest. The combination of these 2 operations yields the operation UnifyBelief which brings a supportive junction tree into consistency and is performed at the level of junction trees.

[6] The concept of evidence entering is defined later.

Operation 4 (UnifyBelief) UnifyBelief *can be initiated at any universe* $(C_0, B(C_0))$ *in a junction tree.* $(C_0, B(C_0))$ *calls CollectEvidence in all its neighbors and when they have finished their CollectEvidence,* $(C_0, B(C_0))$ *calls DistributeEvidence in them.*

Operations for Belief Exchange in Belief Initialization

Belief initialization brings a junction forest into global consistency before any evidence is available. One problem arises when there are multiple linkages between junction trees. Care is to be taken not to count the same information passed on different linkages multiple times. The following two operations pass information through multiple linkages during belief initialization. NonRedundancyAbsorption is performed at the level of linkage hosts. ExchangeBelief calls NonRedundancyAbsorption and is performed at the level of junction trees. ExchangeBelief ensures the exchange between junction trees of prior distribution on d-sepsets without redundant information passing.

Operation 5 (NonRedundancyAbsorption) *Let* $(C^a_x, B(C^a_x))$ *and* $(C^b_x, B(C^b_x))$ *be 2 linkage host universes in junction trees* T^a *and* T^b *respectively. Let* $(L_x, B(L_x))$ *and* $(R_x, B(R_x))$ *be the worlds for corresponding linkage and redundancy set. Suppose*

$$\Delta(B(C^a_x)) \subseteq \Delta(B(L_x)).$$

The **NonRedundancyAbsorption** *of* $(C^a_x, B(C^a_x))$ *from* $(C^b_x, B(C^b_x))$ *through linkage* L_x *changes* $B(L_x)$, $B(R_x)$ *and* $B(C^a_x)$ *into* $B'(L_x)$, $B'(R_x)$ *and* $B'(C^a_x)$ *respectively.*

$$\begin{aligned} B'(L_x) &= \sum_{C^b_x \setminus L_x} B(C^b_x) \\ B'(R_x) &= \sum_{L_x \setminus R_x} B'(L_x) \\ B'(C^a_x) &= B(C^a_x) * \frac{B'(L_x)/B'(R_x)}{B(L_x)/B(R_x)} \end{aligned}$$

The factor $1/B'(R_x)$ *above has the function of* **redundancy removal.**

At initialization, the belief tables for linkages and redundancy sets are in the state of construction, i.e., constant. Hence, $B(L_x)$ and $B(R_x)$ above are constant tables. If $B'(L_x)$ is constant, which is possible because constant probability tables are assigned to d-sepnodes in some sects in Definition 7, then after the operation

$$\sum_{C_x^a\backslash L_x} B'(C_x^a) \propto \sum_{C_x^a\backslash L_x} B(C_x^a).$$

That is, if C_x^b has no information to offer, then C_x^a will not change its belief. If $\sum_{C_x^a\backslash L_x} B(C_x^a)$ is constant, then after the operation

$$\sum_{C_x^a\backslash L_x} B'(C_x^a) \propto \left(\sum_{C_x^b\backslash L_x} B(C_x^b)\right) \Big/ \left(\sum_{C_x^b\backslash R_x} B(C_x^b)\right).$$

That is, if C_x^b has new information and C_x^a contains no non-trivial information, then the belief of C_x^b will be copied with redundancy removed. If none of $B'(L_x)$ and $\sum_{C_x^a\backslash L_x} B(C_x^a)$ is constant, then after the operation

$$\sum_{C_x^a\backslash L_x} B'(C_x^a) \propto \sum_{C_x^a\backslash L_x} \left(B(C_x^a) * \left(\sum_{C_x^b\backslash L_x} B(C_x^b)\right) \Big/ \left(\sum_{C_x^b\backslash R_x} B(C_x^b)\right)\right).$$

That is, if none of the above cases is true, the belief from both sides will be combined with redundancy removed. Since

$$\Delta(B(C_x^b)) \subseteq \Delta(\sum_{C_x^b\backslash L_x} B(C_x^b)) = \Delta(B'(L_x))$$

the supportiveness of the junction forest is invariant under NonRedundancyAbsorption. Since

$$\frac{B'(C_x^a)}{B'(L_x)/B'(R_x)} = \frac{B(C_x^a)}{B(L_x)/B(R_x)}$$

the joint system belief is invariant under NonRedundancyAbsorption. This operation is equipped at each linkage host.

Operation 6 (ExchangeBelief) *Let L be the set of linkages between junction trees T^a and T^b. When T^a initiates* **ExchangeBelief** *with T^b, the NonRedundancyAbsorption is performed at all linkage host universes in T^a.*

Since ExchangeBelief is composed of NonRedundancyAbsorption, the supportiveness of the junction tree and its joint system belief are invariant under the ExchangeBelief. The operation is equipped at the level of junction trees. After ExchangeBelief, the non-trivial content of joint distribution on d-sepset at T^b will be passed onto T^a without redundancy.

Operations for Belief Update in Evidential Reasoning

Evidential reasoning propagates evidence obtained in one junction tree to the junction forest. A junction tree receiving updated belief on the d-sepset from a neighbor junction tree may be confused due to multiple linkage evidence passing. The following 2 operations handle the evidence propagation between junction trees. AbsorbThroughLinkage propagates evidence through one linkage and is performed at the level of linkage hosts. UpdateBelief calls AbsorbThroughLinkage to propagate evidence from one junction tree to another, and is performed at the level of junction trees. It is used during evidential reasoning when both junction trees are consistent themselves but may not reach boundary consistency between them.

Operation 7 (AbsorbThroughLinkage) *Let $(C_x^a, B(C_x^a))$ and $(C_x^b, B(C_x^b))$ be 2 linkage host universes in junction trees T^a and T^b respectively. Let $(L_x, B(L_x))$ be the corresponding linkage world. Suppose $\Delta(B(C_x^a)) \subseteq \Delta(B(L_x))$. The* **AbsorbThroughLinkage** *of $(C_x^a, B(C_x^a))$ from $(C_x^b, B(C_x^b))$ changes $B(C_x^a)$ and $B(L_x)$ into $B'(C_x^a)$ and $B'(L_x)$*

as the following.

$$\begin{aligned} B'(L_x) &= \sum_{C_x^b \setminus L_x} B(C_x^b) \\ B'(C_x^a) &= B(C_x^a) * B'(L_x)/B(L_x) \end{aligned}$$

After AbsorbThroughLinkage,

$$\sum_{C_x^a \setminus L_x} B'(C_x^a) = \sum_{C_x^b \setminus L_x} B(C_x^b)$$

Since

$$\Delta(B(C_x^b)) \subseteq \Delta(\sum_{C_x^b \setminus L_x} B(C_x^b)) = \Delta(B'(L_x))$$

the supportiveness of a junction forest is invariant under AbsorbThroughLinkage. The operation makes the belief of C_x^a up-to-date with respect to the belief of C_x^b on their common variables.

Operation 8 (UpdateBelief) *Let $L = \{L_1, \ldots, L_k\}$ be the set of linkages between junction trees T^a and T^b with corresponding linkage hosts $C_1^a, \ldots, C_k^a$. When T^a initiates* **UpdateBelief** *with T^b, the operation pair AbsorbThroughLinkage and then DistributeEvidence is performed at $C_1^a, \ldots, C_k^a$.*

Since the operation is composed of AbsorbThroughLinkage and DistributeEvidence, the supportiveness of the junction forest is invariant under the operation.

Since after UpdateBelief,

$$\sum_{C_x^a \setminus L_x} B'(C_x^a) = \sum_{C_x^b \setminus L_x} B(C_x^b) \quad x = 1, \ldots, k$$

and T^a is consistent, the effect of the operation is

$$B'(T^a) = B(T^a) * B'(I)/B(I)$$

where I is the d-sepset between S^a and S^b or equivalently

$$B'(T^a)/B'(I) = B(T^a)/B(I)$$

which implies the joint system belief is invariant under the operation.

Note that after each AbsorbThroughLinkage in operation UpdateBelief, a DistributeEvidence is performed. This is used to avoid confusion in the information receiving junction tree possibly resulted from multiple linkages information passing. The following simple example exemplifies this necessity.

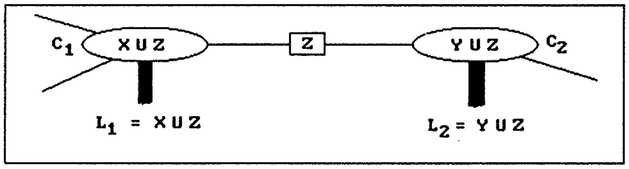


Figure 4.27: An example illustrating the operation UpdateBelief.

Example 18 Let junction tree T^i (Figure 4.27) have 2 linkage host $C_1 = L_1 = X \cup Z$ and $C_2 = L_2 = Y \cup Z$ where X, Y, Z are 3 disjoint sets of nodes. Let $B(C_1)$, $B(C_2)$ and $B(Z)$ be the belief tables of the 2 hosts and their sepset respectively. Suppose new information is passed over to T^i through the 2 linkages from its neighbor junction tree. If AbsorbThroughLinkage is performed at L_1 and then L_2 Without a DistributeEvidence between the 2 operations, then the belief on 2 host cliques will be updated to $B'(C_1)$, $B'(C_2)$, while $B(Z)$ is unchanged. If C_1 initiates an AbsorbThroughSepset from C_2 in the process of propagating the new information to the rest of T^i, the belief on C_1 will become

$$B''(C_1) = B'(C_1)(\sum_{Y} B'(C_2)/B(Z)) \not\propto B'(C_1)$$

which is not expected. This is because $\sum_Y B'(C_2) \not\propto B(Z)$.

With a DistributeEvidence between the 2 AbsorbThroughLinkage operations, there is $B'(Z) = \sum_Y B'(C_2)$. The result of AbsorbThroughSepset becomes

$$B''(C_1) = B'(C_1)(\sum_Y B'(C_2)/B'(Z)) \propto B'(C_1)$$

which is correct.

4.9.3 Belief Initialization

Before any evidence is available, an internal representation of beliefs is to be established. The establishment of this representation is termed *initialization* by Lauritzen and Spiegelhalter [1988] for their method. The function of initialization in the context of junction forests is to propagate the prior knowledge stored in different belief universes of different junction trees to the rest of the forest such that (1) prior marginal probability distribution for any variable can be obtained in any universe containing the variable, and (2) subsequent evidential reasoning can be performed.

The following defines operations DistributeBelief and CollectBelief which are analogous to DistributeEvidence and CollectEvidence but at the junction tree level. The operation BeliefInitialization is then defined in terms of DistributeBelief and CollectBelief just as UnifyBelief is defined in terms of DistributeEvidence and CollectEvidence but at the junction forest level.

Two junction trees in a junction forest is called *neighbors* if the d-sepset between the 2 corresponding sects is nonempty.

Operation 9 (DistributeBelief) *Let T^i be a junction tree in a junction forest. When* **DistributeBelief** *is called in T^i, it performs UpdateBelief with respect to the caller if the caller is a junction tree and then calls DistributeBelief in all its neighbors except the caller and caller's neighbors.*

Operation 10 (CollectBelief) *Let T^i be a junction tree in a junction forest. When* **CollectBelief** *is called in T^i, it calls CollectBelief in all its neighbors except the caller and caller's neighbors. when they have finished, T^i performs ExchangeBelief with respect to each of them followed by a UnifyBelief on T^i.*

Operation 11 (BeliefInitialization) BeliefInitialization *can be initiated at any junction tree T^i in a junction forest if T^i is transformed from a local covering sect. T^i calls CollectBelief in all its neighbors, and when they have finished T^i calls DistributeBelief in them.*

All 3 operations do not change the supportiveness and joint system belief. Thus one has the following theorem.

Theorem 11 (belief initialization with hypertree) *Let $\{S^1, \ldots, S^\beta\}$ be a MSBN with a hypertree structure. Let $F = \{T^1, \ldots, T^\beta\}$ be a junction forest with T^i being the junction tree of S^i. Let $B(F)$ be the joint system belief constructed as section 4.7.3. After BeliefInitialization, the junction forest is globally consistent.*

Note that the BeliefInitialization does not involve direct information passing between neighbors of a local covering sect. This is also the case during evidential reasoning to be discussed latter. As assumed in section 4.7.3, the linkages between these neighbors are not created in the first place. This saving is possible only if the MSBN has a hypertree structure.

Example 19 BeliefInitialization is initiated at Γ^1 in Figure 4.20. It calls CollectBelief in Γ^2 and Γ^3. Since the latter do not have neighbors other than the caller and caller's neighbor, only UnifyBelief is performed in Γ^2 and Γ^3. Table 4.6 lists the belief tables for belief universes in junction trees Γ^2 and Γ^3 after their UnifyBeliefs.

Table 4.7 and Table 4.8 list the belief tables for belief universes of the junction forest and the (prior) marginal probabilities for all variables of the corresponding MSBN respectively after the completion of BeliefInitialization. The marginal probabilities are identical to what would be derived from the USBN (Γ, P) with Γ in Figure 4.17 and P in Table 4.3. The marginals are obtained by marginalization of belief universes which contain the corresponding variables.

Once belief initialization is completed, the junction forest becomes the permanent representation which will be reused for each query session.

4.9.4 Evidential Reasoning

The joint system belief defined in section 4.7.3 is proportional to the *prior* joint distribution representing the background domain knowledge. Initialization allows one to obtain prior marginal probabilities with efficient local computation. When evidence about a particular case becomes available, one wants the prior distribution to change into the posterior distribution.

Evidence is represented in terms of evidence functions. Two types of evidence are considered here as by Jensen, Olesen and Andersen [1990]. The first type has a value range of $\{0, 1\}$ where '0' stands for that the corresponding outcome is impossible and '1' stands for that the corresponding outcome is still possible with relative belief strength remaining the same. The second type is a restriction. It has the same function value range but the function assigns '1' to only one outcome. This type of evidence arises when the corresponding evidential variables are directly observable. Both types of evidence functions can be entered to junction forests by multiplying the prior distribution with the evidence function.

Call the overall process of entering evidence and propagating evidence as *evidential*

$B(\Gamma^2)$		$B(\Gamma^3)$	
Clique	*NodeAss.*	*Clique*	*NodeAss.*
$\{F_2, F_1\}$	F_2	$\{E_1, E_3\}$	E_1
Config.	$B()$	*Config.*	$B()$
$\{f_{21}, f_{11}\}$	.7758	$\{e_{11}, e_{31}\}$	.7121
$\{f_{21}, f_{12}\}$	1.545	$\{e_{11}, e_{32}\}$	3.108
$\{f_{22}, f_{11}\}$	1.164	$\{e_{12}, e_{31}\}$	2.848
$\{f_{22}, f_{12}\}$	.5151	$\{e_{12}, e_{32}\}$	1.332
Clique	*NodeAss.*	*Clique*	*NodeAss.*
$\{H_2, F_1, H_1\}$	H_2, F_1, H_1	$\{H_3, H_2, E_3\}$	H_3, H_2, E_3
Config.	$B()$	*Config.*	$B()$
$\{h_{21}, f_{11}, h_{11}\}$	.7895	$\{h_{31}, h_{21}, e_{31}\}$	1.54
$\{h_{21}, f_{11}, h_{12}\}$	.6	$\{h_{31}, h_{21}, e_{32}\}$	.4596
$\{h_{21}, f_{12}, h_{11}\}$	.2105	$\{h_{31}, h_{22}, e_{31}\}$	1.3
$\{h_{21}, f_{12}, h_{12}\}$	.4	$\{h_{31}, h_{22}, e_{32}\}$	.7
$\{h_{22}, f_{11}, h_{11}\}$	.5	$\{h_{32}, h_{21}, e_{31}\}$	.7
$\{h_{22}, f_{11}, h_{12}\}$	.05	$\{h_{32}, h_{21}, e_{32}\}$	1.3
$\{h_{22}, f_{12}, h_{11}\}$	.5	$\{h_{32}, h_{22}, e_{31}\}$	.02
$\{h_{22}, f_{12}, h_{12}\}$	.95	$\{h_{32}, h_{22}, e_{32}\}$	1.98
		Clique	*NodeAss.*
		$\{E_2, E_3, E_4\}$	E_2
		Config.	$B()$
		$\{e_{21}, e_{31}, e_{41}\}$	1.656
		$\{e_{21}, e_{31}, e_{42}\}$	1.495
		$\{e_{21}, e_{32}, e_{41}\}$	1.898
		$\{e_{21}, e_{32}, e_{42}\}$	.1165
		$\{e_{22}, e_{31}, e_{41}\}$	.0356
		$\{e_{22}, e_{31}, e_{42}\}$	.3738
		$\{e_{22}, e_{32}, e_{41}\}$	.2109
		$\{e_{22}, e_{32}, e_{42}\}$	2.214
		Clique	*NodeAss.*
		$\{E_3, E_4, H_4\}$	E_4, H_4
		Config.	$B()$
		$\{e_{31}, e_{41}, h_{41}\}$	1.424
		$\{e_{31}, e_{41}, h_{42}\}$	.2670
		$\{e_{31}, e_{42}, h_{41}\}$	.3560
		$\{e_{31}, e_{42}, h_{42}\}$	1.513
		$\{e_{32}, e_{41}, h_{41}\}$	1.776
		$\{e_{32}, e_{41}, h_{42}\}$	.3330
		$\{e_{32}, e_{42}, h_{41}\}$	.4440
		$\{e_{32}, e_{42}, h_{42}\}$	1.887
		Clique	*NodeAss.*
		$\{H_3, E_3, H_4\}$	
		Config.	$B()$
		$\{h_{31}, e_{31}, h_{41}\}$	1.42
		$\{h_{31}, e_{31}, h_{42}\}$	1.42
		$\{h_{31}, e_{32}, h_{41}\}$	.5798
		$\{h_{31}, e_{32}, h_{42}\}$	.5798
		$\{h_{32}, e_{31}, h_{41}\}$	.36
		$\{h_{32}, e_{31}, h_{42}\}$	.36
		$\{h_{32}, e_{32}, h_{41}\}$	1.64
		$\{h_{32}, e_{32}, h_{42}\}$	1.64

Table 4.6: Belief tables for belief universes in junction trees Γ^2 and Γ^3 in Figure 4.20 after CollectBelief during BeliefInitialization.

$B(\Gamma^1)$		$B(\Gamma^2)$		$B(\Gamma^3)$	
Clique	*NodeAss.*	*Clique*	*NodeAss.*	*Clique*	*NodeAss.*
$\{H_2, H_1, A_1\}$	H_1, A_1	$\{F_2, F_1\}$	F_2	$\{E_1, E_3\}$	E_1
Config.	$B()$	*Config.*	$B()$	*Config.*	$B()$
$\{h_{21}, h_{11}, h_{11}\}$	.8203	$\{f_{21}, f_{11}\}$	1.160	$\{e_{11}, e_{31}\}$	.564
$\{h_{21}, h_{11}, h_{12}\}$	.08166	$\{f_{21}, f_{12}\}$	5.324	$\{e_{11}, e_{32}\}$	5.026
$\{h_{21}, h_{12}, h_{11}\}$	.5810	$\{f_{22}, f_{11}\}$	1.741	$\{e_{12}, e_{31}\}$	2.256
$\{h_{21}, h_{12}, h_{12}\}$	2.082	$\{f_{22}, f_{12}\}$	1.775	$\{e_{12}, e_{32}\}$	2.154
$\{h_{22}, h_{11}, h_{11}\}$	.3797	*Clique*	*NodeAss.*	*Clique*	*NodeAss.*
$\{h_{22}, h_{11}, h_{12}\}$	.2183	$\{H_2, F_1, H_1\}$	H_2, F_1, H_1	$\{H_3, H_2, E_3\}$	H_3, H_2, E_3
$\{h_{22}, h_{12}, h_{11}\}$	.2690	*Config.*	$B()$	*Config.*	$B()$
$\{h_{22}, h_{12}, h_{12}\}$	5.568	$\{h_{21}, f_{11}, h_{11}\}$	.7121	$\{h_{31}, h_{21}, e_{31}\}$	1.444
Clique	*NodeAss.*	$\{h_{21}, f_{11}, h_{12}\}$	1.598	$\{h_{31}, h_{21}, e_{32}\}$	.4310
$\{H_2, A_2, A_1\}$	H_2	$\{h_{21}, f_{12}, h_{11}\}$	.1899	$\{h_{31}, h_{22}, e_{31}\}$	.731
Config.	$B()$	$\{h_{21}, f_{12}, h_{12}\}$	1.065	$\{h_{31}, h_{22}, e_{32}\}$	.3936
$\{h_{21}, a_{21}, a_{11}\}$	.5526	$\{h_{22}, f_{11}, h_{11}\}$	.2990	$\{h_{32}, h_{21}, e_{31}\}$	.5915
$\{h_{21}, a_{21}, a_{12}\}$	1.725	$\{h_{22}, f_{11}, h_{12}\}$	.2918	$\{h_{32}, h_{21}, e_{32}\}$	1.098
$\{h_{21}, a_{22}, a_{11}\}$	.8487	$\{h_{22}, f_{12}, h_{11}\}$	.2990	$\{h_{32}, h_{22}, e_{31}\}$	.0531
$\{h_{21}, a_{22}, a_{12}\}$	.4388	$\{h_{22}, f_{12}, h_{12}\}$	5.545	$\{h_{32}, h_{22}, e_{32}\}$	5.257
$\{h_{22}, a_{21}, a_{11}\}$	.08289			*Clique*	*NodeAss.*
$\{h_{22}, a_{21}, a_{12}\}$	.7394			$\{E_2, E_3, E_4\}$	E_2
$\{h_{22}, a_{22}, a_{11}\}$	.5658			*Config.*	$B()$
$\{h_{22}, a_{22}, a_{12}\}$	5.047			$\{e_{21}, e_{31}, e_{41}\}$	1.020
Clique	*NodeAss.*			$\{e_{21}, e_{31}, e_{42}\}$	1.422
$\{H_3, H_2, A_2\}$	H_3, A_2			$\{e_{21}, e_{32}, e_{41}\}$	2.182
Config.	$B()$			$\{e_{21}, e_{32}, e_{42}\}$	.2378
$\{h_{31}, h_{21}, a_{21}\}$	1.763			$\{e_{22}, e_{31}, e_{41}\}$	.02194
$\{h_{31}, h_{21}, a_{22}\}$	.1120			$\{e_{22}, e_{31}, e_{42}\}$	.3555
$\{h_{31}, h_{22}, a_{21}\}$	.6366			$\{e_{22}, e_{32}, e_{41}\}$	.2424
$\{h_{31}, h_{22}, a_{22}\}$	.4880			$\{e_{22}, e_{32}, e_{42}\}$	4.518
$\{h_{32}, h_{21}, a_{21}\}$	.5143			*Clique*	*NodeAss.*
$\{h_{32}, h_{21}, a_{22}\}$	1.176			$\{E_3, E_4, H_4\}$	E_4, H_4
$\{h_{32}, h_{22}, a_{21}\}$	.1857			*Config.*	$B()$
$\{h_{32}, h_{22}, a_{22}\}$	5.124			$\{e_{31}, e_{41}, h_{41}\}$	.7622
Clique	*NodeAss.*			$\{e_{31}, e_{41}, h_{42}\}$	.2801
$\{H_3, A_3, H_4\}$	A_3, H_4			$\{e_{31}, e_{42}, h_{41}\}$	.1906
Config.	$B()$			$\{e_{31}, e_{42}, h_{42}\}$	1.587
$\{h_{31}, a_{31}, h_{41}\}$	.225			$\{e_{32}, e_{41}, h_{41}\}$	1.658
$\{h_{31}, a_{31}, h_{42}\}$	.675			$\{e_{32}, e_{41}, h_{42}\}$	.7662
$\{h_{31}, a_{32}, h_{41}\}$	.84			$\{e_{32}, e_{42}, h_{41}\}$	.4145
$\{h_{31}, a_{32}, h_{42}\}$	1.26			$\{e_{32}, e_{42}, h_{42}\}$	4.342
$\{h_{32}, a_{31}, h_{41}\}$	1.4			*Clique*	*NodeAss.*
$\{h_{32}, a_{31}, h_{42}\}$	4.2			$\{H_3, E_3, H_4\}$	
$\{h_{32}, a_{32}, h_{41}\}$	.56			*Config.*	$B()$
$\{h_{32}, a_{32}, h_{42}\}$	.84			$\{h_{31}, e_{31}, h_{41}\}$	.7723
Clique	*NodeAss.*			$\{h_{31}, e_{31}, h_{42}\}$	1.403
$\{H_4, A_4\}$	A_4			$\{h_{31}, e_{32}, h_{41}\}$	.2927
Config.	$B()$			$\{h_{31}, e_{32}, h_{42}\}$	.5319
$\{h_{41}, a_{41}\}$	2.723			$\{h_{32}, e_{31}, h_{41}\}$	.1805
$\{h_{41}, a_{42}\}$	.3025			$\{h_{32}, e_{31}, h_{42}\}$	.4641
$\{h_{42}, a_{41}\}$	1.395			$\{h_{32}, e_{32}, h_{41}\}$	1.780
$\{h_{42}, a_{42}\}$	5.58			$\{h_{32}, e_{32}, h_{42}\}$	4.576

Table 4.7: Belief tables for belief universes of junction forest $F = \{\Gamma^1, \Gamma^2, \Gamma^3\}$ in Figure 4.20 obtained after the completion of BeliefInitialization.

$p(h_{11}) =$	.15	$p(a_{11}) =$	.205	$p(f_{11}) =$	.2901	$p(e_{11}) =$	.559
$p(h_{21}) =$	.3565	$p(a_{21}) =$	.31	$p(f_{21}) =$	.6485	$p(e_{21}) =$	.4862
$p(h_{31}) =$	.3	$p(a_{31}) =$	.65			$p(e_{31}) =$	.282
$p(h_{41}) =$	.3025	$p(a_{41}) =$	.4118			$p(e_{41}) =$	.3466

Table 4.8: Prior probabilities from junction forest $F = \{\Gamma^1, \Gamma^2, \Gamma^3\}$ in Figure 4.20 obtained after the completion of BeliefInitialization.

reasoning. After a batch of evidence is entered to a junction tree, UnifyBelief can be performed to bring the junction tree into consistency. This is the same in the context of junction forests as in the junction tree technique. However, in order to obtain posterior marginal distributions on variables in the current active junction tree, the global consistency of the junction forest is *not* necessary. Before this is formally treated, several concepts are to be defined.

Here only junction forests transformed from MSBNs with hypertree structures are considered. When a user wants to obtain marginal distributions on variables not contained in the currently active junction tree, it is said that there is an *attention shift.* The junction tree which contains the desired variables is called the *destination tree.*

Definition 17 (intermediate tree) *Let S^i, S^j, S^k be 3 sects in a MSBN with a hypertree structure, and T^i, T^j, T^k be their junction trees respectively in the corresponding junction forest. T^j is the* intermediate tree *between T^i and T^k if the removal of S^j from the MSBN would render the hypertree disconnected with S^i and S^k in different parts.*

Due to the hypertree structure, one has the following lemma.

Lemma 9 *Let a junction forest be transformed from a MSBN with a hypertree structure. Let T^i and T^j be 2 junction trees in the forest. The set of intermediate junction trees between T^i and T^j is unique.*

The following defines an operation ShiftAttention at the junction forest level. It is performed when the user's attention shifts.

Operation 12 (ShiftAttention) *Let F be a junction forest whose corresponding MSBN has a hypertree structure. Let T^{j_0} and $T^{j_{m+1}}$ be the currently active tree and destination tree in F respectively. Let $\{T^{j_1}, \ldots, T^{j_m}\}$ be the set of m intermediate trees between T^{j_0} and $T^{j_{m+1}}$ such that $T^{j_0}, T^{j_1}, \ldots, T^{j_m}, T^{j_{m+1}}$ form a chain of neighbors.*

For $i = 1$ to $m + 1$, T^{j_i} performs UpdateBelief with respect to $T^{j_{i-1}}$.

Before each attention shift, several batches of evidence can be entered to the currently active tree. When an attention shift happens, ShiftAttention swaps in and out of memory sequentially only the intermediate trees between the currently active tree and destination tree without the participation of the rest of the forest. The following theorem shows that this is sufficient in order to obtain the marginal distributions in the destination tree.

Theorem 12 (attention shift) *Let F be a consistent junction forest whose corresponding MSBN has a hypertree structure. Start with any active junction tree. Repeat the following cycle for finite times:*

1. *repeatedly enter evidence to the currently active tree followed by UnifyBelief for finite times;*
2. *use ShiftAttention to shift attention to any destination tree.*

The marginal distributions obtained in the final active tree are identical as would be obtained when the forest is globally consistent.

Proof:

Before any evidence is entered, the forest is consistent by assumption. Before each ShiftAttention, the currently active tree is consistent due to UnifyBelief. The supportiveness of the forest is invariant under the execution of the cycle. The joint system belief changes (correctly) under only evidence entering but remains invariant under other operations of the cycle.

Transform the initial consistent forest into a junction tree F' of huge universes as in the proof of theorem 9. Each huge universe in F' corresponds to a tree in F. The active tree and destination trees corresponds to an 'active' universe and destination universe. The evidence entering in F and the subsequent UnifyBelief correspond to the evidence entering in F'. The ShiftAttention in F corresponds to performing a series of AbsorbThroughSepsets starting at the neighbor of the 'active' universe down to the destination universe in F'.

Note that after each series of AbsorbThroughSepset operations in F', the belief table of each sepset of F' is the marginalization of the belief in the neighbor more distant from the destination universe in the junction tree F'. Therefore, at the end of cycles in F, if a DistributeEvidence is performed in F', it will be consistent and the belief in the destination universe does not undergo further change. That is, the belief on the destination tree at the end of the cycles is identical to what would obtained when the forest is globally consistent.

□

4.9.5 Computational Complexity

Theorem 12 shows the most important characterization of the MSBN and junction forests, namely, the capability of exploitation of localization to reduce the computational complexity.

Due to localization, the user interest and new evidence will remain in the sphere of one junction tree for a period of time. The judgments obtained is at the knowledge level of overall junction forest while the computation required is at the level of the currently active tree. Compared to the USBN and single junction tree representation where the evidence has to be propagated to the overall system, this leads to great savings in terms of time and space requirement when localization is valid.

When the user subsequently shifts interest into another set of variables contained in a destination tree, only the intermediate trees need to be updated. The amount of computation required is linear to the number of intermediate trees and to the number of linkages between each pair of neighbors. No matter how large is the overall junction forest, the amount of computation for attention shift is fixed once the destination tree and mediating trees are fixed. For example, in a MSBN with a covering sect, no matter how many sects are in the MSBN, the attention shift updates maximum 2 sects.

Given the analysis, the computational complexity of a MSBN with β sects is about $1/\beta$ of the corresponding USBN system when localization is valid. The actual time and space requirement is little more than $1/\beta$ due to the repetition of d-sepnodes and the computation required for attention shift. The computational savings obtained in PAINULIM system is discussed in the chapter 5.

Example 20 Complete the example on junction forest $F = \{\Gamma^1, \Gamma^2, \Gamma^3\}$ with evidential reasoning. Suppose the outcome of variable E_3 in Γ^3 is found to be e_{31}. Table 4.9 lists $B'(\Gamma^3)$ which is obtained after evidence entering and UnifyBelief; $B'(\Gamma^1)$ and $B'(\Gamma^2)$ both of which are obtained after a ShiftAttention with destination Γ^2. Table 4.10 lists the posterior marginal probabilities after the ShiftAttention.

Before closing this section, the following example demonstrates the computational advantage, during attention shift, provided by covering sects.

Example 21 In figure 4.23, D is sectioned into $\{D^1, D^2, D^3\}$ by sound sectioning without a covering sect. The MSBN is transformed into the junction forest $\{T^1, T^2, T^3\}$ by an invertible transformation. If evidence about E, and then about G comes, the first piece of evidence will be entered to T^1, and then T^1 will send message to T^2. After entering the second piece of evidence, T^2 will be the only one up-to-date. Now if one

$B'(\Gamma^3)$	
Clique	*NodeAss.*
$\{E_1, E_3\}$	E_1
Config.	$B()$
$\{e_{11}, e_{31}\}$	.5640
$\{e_{11}, e_{32}\}$	0
$\{e_{12}, e_{31}\}$	2.256
$\{e_{12}, e_{32}\}$	0
Clique	*NodeAss.*
$\{H_3, H_2, E_3\}$	H_3, H_2, E_3
Config.	$B()$
$\{h_{31}, h_{21}, e_{31}\}$	14.44
$\{h_{31}, h_{21}, e_{32}\}$	0
$\{h_{31}, h_{22}, e_{31}\}$	7.31
$\{h_{31}, h_{22}, e_{32}\}$	0
$\{h_{32}, h_{21}, e_{31}\}$	5.915
$\{h_{32}, h_{21}, e_{32}\}$	0
$\{h_{32}, h_{22}, e_{31}\}$	.5310
$\{h_{32}, h_{22}, e_{32}\}$	0
Clique	*NodeAss.*
$\{E_2, E_3, E_4\}$	E_2
Config.	$B()$
$\{e_{21}, e_{31}, e_{41}\}$	1.020
$\{e_{21}, e_{31}, e_{42}\}$	1.422
$\{e_{21}, e_{32}, e_{41}\}$	0
$\{e_{21}, e_{32}, e_{42}\}$	0
$\{e_{22}, e_{31}, e_{41}\}$	.02194
$\{e_{22}, e_{31}, e_{42}\}$	.3555
$\{e_{22}, e_{32}, e_{41}\}$	0
$\{e_{22}, e_{32}, e_{42}\}$	0
Clique	*NodeAss.*
$\{E_3, E_4, H_4\}$	E_4, H_4
Config.	$B()$
$\{e_{31}, e_{41}, h_{41}\}$	7.622
$\{e_{31}, e_{41}, h_{42}\}$	2.801
$\{e_{31}, e_{42}, h_{41}\}$	1.906
$\{e_{31}, e_{42}, h_{42}\}$	15.87
$\{e_{32}, e_{41}, h_{41}\}$	0
$\{e_{32}, e_{41}, h_{42}\}$	0
$\{e_{32}, e_{42}, h_{41}\}$	0
$\{e_{32}, e_{42}, h_{42}\}$	0
Clique	*NodeAss.*
$\{H_3, E_3, H_4\}$	
Config.	$B()$
$\{h_{31}, e_{31}, h_{41}\}$	7.723
$\{h_{31}, e_{31}, h_{42}\}$	14.03
$\{h_{31}, e_{32}, h_{41}\}$	0
$\{h_{31}, e_{32}, h_{42}\}$	0
$\{h_{32}, e_{31}, h_{41}\}$	1.805
$\{h_{32}, e_{31}, h_{42}\}$	4.641
$\{h_{32}, e_{32}, h_{41}\}$	0
$\{h_{32}, e_{32}, h_{42}\}$	0

$B'(\Gamma^1)$	
Clique	*NodeAss.*
$\{H_2, H_1, A_1\}$	H_1, A_1
Config.	$B()$
$\{h_{21}, h_{11}, h_{11}\}$	4.105
$\{h_{21}, h_{11}, h_{12}\}$	.5036
$\{h_{21}, h_{12}, h_{11}\}$	2.908
$\{h_{21}, h_{12}, h_{12}\}$	12.84
$\{h_{22}, h_{11}, h_{11}\}$	.4627
$\{h_{22}, h_{11}, h_{12}\}$	.2661
$\{h_{22}, h_{12}, h_{11}\}$	.3278
$\{h_{22}, h_{12}, h_{12}\}$	6.785
Clique	*NodeAss.*
$\{H_2, A_2, A_1\}$	H_2
Config.	$B()$
$\{h_{21}, a_{21}, a_{11}\}$	3.732
$\{h_{21}, a_{21}, a_{12}\}$	11.65
$\{h_{21}, a_{22}, a_{11}\}$	3.281
$\{h_{21}, a_{22}, a_{12}\}$	1.696
$\{h_{22}, a_{21}, a_{11}\}$	.4190
$\{h_{22}, a_{21}, a_{12}\}$	3.737
$\{h_{22}, a_{22}, a_{11}\}$	.3715
$\{h_{22}, a_{22}, a_{12}\}$	3.313
Clique	*NodeAss.*
$\{H_3, H_2, A_2\}$	H_3, A_2
Config.	$B()$
$\{h_{31}, h_{21}, a_{21}\}$	13.58
$\{h_{31}, h_{21}, a_{22}\}$	.8623
$\{h_{31}, h_{22}, a_{21}\}$	4.138
$\{h_{31}, h_{22}, a_{22}\}$	3.172
$\{h_{32}, h_{21}, a_{21}\}$	1.800
$\{h_{32}, h_{21}, a_{22}\}$	4.115
$\{h_{32}, h_{22}, a_{21}\}$	.01857
$\{h_{32}, h_{22}, a_{22}\}$	.5124
Clique	*NodeAss.*
$\{H_3, A_3, H_4\}$	A_3, H_4
Config.	$B()$
$\{h_{31}, a_{31}, h_{41}\}$	1.632
$\{h_{31}, a_{31}, h_{42}\}$	4.895
$\{h_{31}, a_{32}, h_{41}\}$	6.091
$\{h_{31}, a_{32}, h_{42}\}$	9.137
$\{h_{32}, a_{31}, h_{41}\}$	1.289
$\{h_{32}, a_{31}, h_{42}\}$	3.867
$\{h_{32}, a_{32}, h_{41}\}$	.5157
$\{h_{32}, a_{32}, h_{42}\}$	.7735
Clique	*NodeAss.*
$\{H_4, A_4\}$	A_4
Config.	$B()$
$\{h_{41}, a_{41}\}$	8.575
$\{h_{41}, a_{42}\}$	.9528
$\{h_{42}, a_{41}\}$	3.734
$\{h_{42}, a_{42}\}$	14.94

$B'(\Gamma^2)$	
Clique	*NodeAss.*
$\{F_2, F_1\}$	F_2
Config.	$B()$
$\{f_{21}, f_{11}\}$	5.523
$\{f_{21}, f_{12}\}$	10.79
$\{f_{22}, f_{11}\}$	8.285
$\{f_{22}, f_{12}\}$	3.598
Clique	*NodeAss.*
$\{H_2, F_1, H_1\}$	H_2, F_1, H_1
Config.	$B()$
$\{h_{21}, f_{11}, h_{11}\}$	3.638
$\{h_{21}, f_{11}, h_{12}\}$	9.450
$\{h_{21}, f_{12}, h_{11}\}$	.9702
$\{h_{21}, f_{12}, h_{12}\}$	6.300
$\{h_{22}, f_{11}, h_{11}\}$	.3644
$\{h_{22}, f_{11}, h_{12}\}$	.3556
$\{h_{22}, f_{12}, h_{11}\}$	.3644
$\{h_{22}, f_{12}, h_{12}\}$	6.757

Table 4.9: Belief tables for $F = \{\Gamma^1, \Gamma^2, \Gamma^3\}$ in Figure 4.20 in evidential reasoning. $B'(\Gamma^3)$ is obtained first after $E_3 = e_{31}$ is entered to Γ^3 and the evidence is propagated to the junction tree. $B'(\Gamma^1)$ and $B'(\Gamma^2)$ are obtained afterwards by ShiftAttention.

$p(h_{11}) =$	.1893	$p(a_{11}) =$	.2767	$p(f_{11}) =$	.4897	$p(e_{11}) =$	.2
$p(h_{21}) =$	.7219	$p(a_{21}) =$	.6928	$p(f_{21}) =$	.5786	$p(e_{21}) =$	.8661
$p(h_{31}) =$	.7714	$p(a_{31}) =$	.4143			$p(e_{31}) =$	1
$p(h_{41}) =$	.3379	$p(a_{41}) =$	.4365			$p(e_{41}) =$	.3696

Table 4.10: Posterior probabilities from junction forest $F = \{\Gamma^1, \Gamma^2, \Gamma^3\}$ in Figure 4.20 after evidence $E_3 = e_{31}$ is propagated by ShiftAttention.

is interested in the belief on H, the belief tables on $\{B, F\}$ and $\{C, F\}$ in T^3 have to be updated. However T^2 can not provide distribution on $\{B, F\}$. Thus, T^2 has to send message to T^3 about $\{C, F\}$, then send message to T^1. Then T^1 can becomes up-to-date and send distribution on $\{B, F\}$ to T^3. One sees 3 message passings are necessary, and linkages between each pair of junction trees have to be created and maintained. More message passings and more linkages are needed when there are more sects organized in this structure. When n sects are inter-connected and there is a covering sect, only n-1 sets of linkages need to be created; and maximum 2 message passings are needed to update the belief in any destination tree.

4.10 Remarks

This chapter presents MSBNs and junction forests as flexible and efficient knowledge representation and inference formalisms to exploit localization naturally existing in large knowledge-based systems. The systems which can benefit from the technique are those reusable, representable by general but sparse networks, and characterized by incremental evidential reasoning.

The MSBNs allow the partition of a large application domain into smaller natural subdomains such that each of them can be represented as a Bayesian subnetwork - a sect, and can be tested and refined individually. This makes the representation of a complex domain easier and more precise for knowledge engineers and makes the resultant system more natural and more understandable to system users. The resultant modularity facilitates implementation of large systems in an incremental fashion. The constraints technically imposed by MSBNs on the partition are found to be d-sepsets and soundness of sectioning.

Two important guidelines for sound sectioning are derived. The covering subDAG rule is suitable for partition according to categories in the same abstraction level. While the hypertree structure is suitable for partition of domain with hierarchical nature. It provides a formalism to represent and reason at different levels of abstraction. MSBNs following the rules allow multiply connected sects, do not require expensive computation for validation of soundness of sectioning, and have additional computational advantage during attention shift in evidential reasoning.

Each sect in the MSBN is transformed into a junction tree such that the MSBN is transformed into a junction forest representation where evidential reasoning takes place. The constraints on transformation are found to be the invertibility of morali-triangulation and separability.

Each sect/junction tree in the MSBN/junction forest stands as a separate computational object. Since the technique allows transformation of sects into junction trees through local computation at the sect level, and allows reasoning to be conducted with junction trees as units, the space requirement is governed by the size of 1 sect/junction tree. Hence large applications can be built and run on relatively smaller computers whenever hardware resource is of concern.

For large application domain, an average case may involve only a portion of the total knowledge encoded in a system, and one portion may be used repeatedly over a period of time. A MSBN and junction forest representation allows the 'interesting' or 'relevant' sect/junction tree to be loaded while the rest of the junction forest remains inactive and stays in the secondary storage. However, the judgments made on variables in the active junction tree are consistent with all the knowledge available, including both prior knowledge and new evidence, embedded in the overall junction forest. When user's attention shifts, inactive junction trees can be made active and previous accumulation of evidence is preserved. This is achieved through transfer of joint belief on d-sepsets. The amount of computation needed for this shift of attention is that required by the operation DistributeEvidence in the newly active junction tree times the number of linkages between the 2 junction trees when there exists a covering sect. Therefore, when localization is valid during consultation, the time and space requirements of a MSBN system of β sects with a covering sect or with a hypertree structure are little more than $1/\beta$ of that required by a USBN system, assuming equal size of sects. The larger a MSBN system, the more computational savings can be obtained.

Chapter 5

PAINULIM: AN EXPERT SYSTEM FOR NEUROMUSCULAR DIAGNOSIS

This chapter discusses the implementation of a prototype expert system called PAINULIM. The system assists EMGers in neuromuscular diagnosis involving the painful or impaired upper limb. Section 5.1 introduces the PAINULIM domain, and compares PAINULIM with MUNIN. Section 5.2 discusses how localization is exploited in PAINULIM using the MSBN technique (Chapter 4). Section 5.3 discusses other issues in knowledge representation of PAINULIM. Section 5.4 introduces a expert system shell WEBWEAVR which is used in development of PAINULIM. Section 5.5 provides a query session with PAINULIM to show its capability. Section 5.6 presents an evaluation of PAINULIM.

5.1 PAINULIM Application Domain and Design Criteria

As is reviewed in section 1.1, several expert systems in the neuromuscular diagnosis field have appeared since the mid 80's. Satisfaction in system testing with constructed cases have been reported, while only one of them (ELECTRODIAGNOSTIC ASSISTANT [Jamieson 90]) reported clinical evaluation using 15 cases of 78% agreement rate with electromyographers (EMGers). Most of these systems are rule-based systems which are reviewed in Chapter 1. MUNIN [Andreassen et al. 89] as a large system has attracted much attention and it can be used as a yard stick against which to compare PAINULIM.

MUNIN project started in the mid 80's in Denmark as part of the European ESPRIT program. MUNIN was to be a 'full expert system' for neuromuscular diagnosis. Functionalities to be included would be test-planning, test-guide, test-set-up, signal processing of test results, diagnosis, and treatment recommendation. The intended users of MUNIN would range from novice to experienced practitioners. The knowledge base would ultimately include full human neuroanatomy. MUNIN adopts Bayesian networks to represent probabilistic knowledge. Substantial contributions to Bayesian network techniques were made (e.g., [Lauritzen and Spiegelhalter 88, Jensen, Lauritzen and Olesen 90]).

The MUNIN system is to be developed in 3 stages. In the first stage a 'nanohuman' model with 1 muscle and 3 possible diseases has been developed. In the second stage a 'microhuman' system with 6 muscles and corresponding nerves is to be developed. The last stage would correspond to a model of the full human neuroanatomy. The only clinical experience with MUNIN is contained in the phrase: "we expect semi-clinical trials of the 'microhuman' to give the first indication of clinical usefulness" [Andreassen et al. 89].

PAINULIM started in 1990 at University of British Columbia in the Department of Electrical Engineering with cooperation from the Department of Computer Science and at the Neuromuscular Disease Unit (NDU) of Vancouver General Hospital (VGH). Expertise needed to build PAINULIM's knowledge base was provided by experienced electromyographer Andrew Eisen and Bhanu Pant. Rather than attempting to cover the full range of diagnosis as does MUNIN, PAINULIM sets to cover the more modest, but realizable, goal of diagnosis for patients suffering from a painful or impaired upper limb due to diseases of spinal cord and/or peripheral nervous system. About 50% of the patient entering NDU of VGH can be diagnosed with PAINULIM. The 14 most common diseases considered by PAINULIM include: Amyotrophic Lateral Sclerosis, Parkinsons disease, Anterior horn cell disease, Root diseases, Intrinsic cord disease, Carpal tunnel syndrome, Plexus lesions, Median, Ulnar and Radial nerve lesions. PAINULIM uses features from clinical examination, (needle) EMG studies and nerve conduction studies to make diagnostic recommendations.

PAINULIM requires from the users specific knowledge and experience:

- minimum competence in clinical medicine especially in neuromuscular diseases;
- basic knowledge of nerve conduction study techniques; and
- minimum experience of EMG patterns in common neuromuscular diseases.

PAINULIM will benefit the following users:

- students and residents in neurology, physical medicine and neuromuscular diseases;
- doctors who are practicing EMG and nerve conduction in their offices;
- experienced EMGers as a formal peer review (self evaluation); and
- hopefully different labs to adapt uniform procedures and criteria for diagnosis.

Clinical diagnosis is performed in steps as anatomical, pathological and etiological. PAINULIM currently works at the anatomical level only. Extension to other levels is considered as one of the future research topics.

Given the level of knowledge and experience of intended user of PAINULIM, the system does not attempt to represent explicitly the neuroanatomy involved in the painful or impaired upper limb. Rather, PAINULIM chooses to represent explicitly the clinically significant disease-feature relations which is one of the most important part of the expertise of experienced neurologists. This choice allow rapid development of PAINULIM which can work directly at the clinical level.

PAINULIM is based on Bayesian belief networks as is MUNIN. The PAINULIM project has benefited from the research results of MUNIN, especially the junction tree technique (section 4.2.2); but, PAINULIM is not limited to duplicating the techniques developed in MUNIN; rather the Multiply Sectioned Bayesian Networks and Junction Forests technique presented in chapter 4 is the extension to the junction tree technique in an effort to achieve more flexible knowledge representation and more efficient inference computation.

5.2 Exploiting Localization in PAINULIM Using MSBN Technique

5.2.1 Resolve Conflict Demands by Exploiting Localization

The PAINULIM project has a large domain. The Bayesian network representation contains 83 variables including 14 diseases and 69 features each of which has up to 3 possible outcomes. The network is multiply connected and has 271 arcs and 6795 probability values. When transformed into a junction tree representation, the system contains 10608 belief values. During the system development, the tight schedule of medical staff demands (1) knowledge acquisition and system testing within hospital environment where most computing equipments are personal computers; and (2) short response time in system testing and refinement. Implementation in hospital equipments will also facilitate the adoption of the system when it is completed. On the other hand, the space and time complexity of PAINULIM system tends to slow down the response and to demand more powerful computing equipments not available in the hospital lab.

This conflict demands motivated the improvement on current Bayesian network representation in order to reduce the computational complexity. The 'localization' naturally existing in the PAINULIM domain (section 4.1) inspired the development of the MSBN technique (chapter 4). The following briefly summarizes the description on localization in the PAINULIM domain.

Localization means that, during a particular consultation session in a large application domain represented by a Bayesian net, (1) new evidence and queries are directed to small part of a large network repeatedly within a period of time; and (2) certain parts of the network may not be of interest to users at all. An EMGer bases his diagnosis on 3 major information sources: clinical examination, (needle) EMG studies and nerve conduction studies. The examination/studies are performed in sequence. Based on the practice in NDU of VGH, about 60% of patients have only EMG studies, and about 27% of patients have only nerve conduction studies. The number of clinical findings on an average patient is about 5. The number of EMG and nerve conduction studies performed on an average patient are about 6 and 4 respectively. All findings are obtained incrementally.

5.2.2 Using MSBN Technique in PAINULIM

Based on localization, PAINULIM uses the MSBN representation. The domain is partitioned into 3 natural localization preserving subdomains (clinical, EMG and nerve conduction) which are separately represented by 3 sects (CLINICAL, EMG, and NCV) in PAINULIM (Figure 5.28). The (sub)domain of each sect contains the corresponding feature variables and relevant disease variables. Using this representation, the 3 sects are created and tested separately. This modularity allows the EMGer to concentrate on one natural subdomain at a time during network construction rather than to manage the overall complexity at the same time. This eases the task of knowledge acquisition in both the part of the EMGer and the part of the knowledge engineer. Problems in each sect can thus be isolated and corrected quickly and easily.

The 3 sects are interfaced by common disease variables. The d-sepset between CLINICAL and EMG contains 12 diseases, and the d-sepset between CLINICAL and NCV contains 10 diseases. All the disease variables have no parent variables and thus the d-sepset constraint (section 4.5) is satisfied. The CLINICAL sect has all the 14 disease variables considered in PAINULIM, and it becomes the covering sect (section 4.6.2). Thus the soundness of sectioning is guaranteed in PAINULIM. This representation is

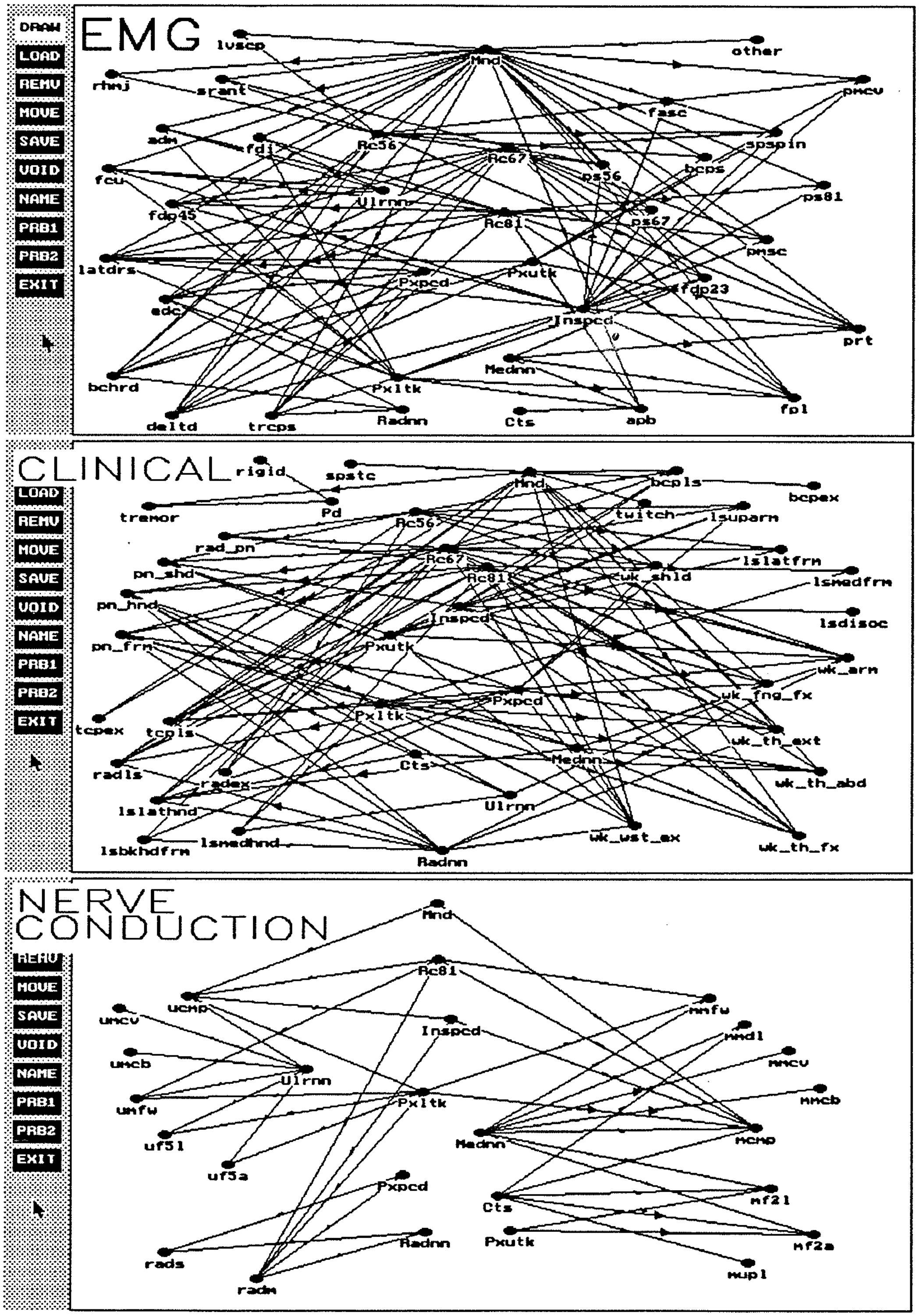


Figure 5.28: The 3 sects of PAINULIM: CLINICAL, EMG, and NCV

natural because clinical examination is the stage where all the disease candidates are subject to consideration.

After the 3 sects are created, they are transformed into a junction forest (Figure 5.29). The MSBN technique allows the transformation to be conducted by local computation at the level of sects (section 4.7.1). Thus the space requirement for transformation is governed by the size of only 1 sect.

Multiple linkages are created between junction trees such that evidence can be propagated from one to another during evidential reasoning. There are 3 linkages between CLINICAL tree and EMG tree (thick lines in Figure 5.29). The sizes of their state spaces are 768, 1536, and 1536 respectively. Without introducing multiple linkages (using the brute force method in section 4.3), the d-sepset would form a clique with state space size 6144, which would greatly increase the computational complexity in each sect. This is because this large clique would have all the other cliques in the same junction tree as neighbors. During evidential reasoning, the belief table of this large clique would be marginalized and updated as many times as the number of neighbor cliques.

With the 3 junction trees linked and joint system belief initialized, the resultant consistent junction forest becomes the permanent representation of the PAINULIM domain where evidential reasoning takes place. The original MSBN still serves as the user interface while the computation for inference is solely performed in the junction forest.

Since the junction forest representation preserves localization, the run time computation of PAINULIM can be restricted to only one junction tree. Thus the time and space requirements are governed by the size of one sect, not the size of the forest. If PAINULIM were represented by a USBN, the size of total state space of the junction tree would be 10608. This overall junction tree has to be updated for each batch of evidence. Using the MSBN technique, the sizes of total state space of each junction tree are 7308, 6498 and 2178 (in order of CLINICAL, EMG and NCV) respectively. The largest size 7308

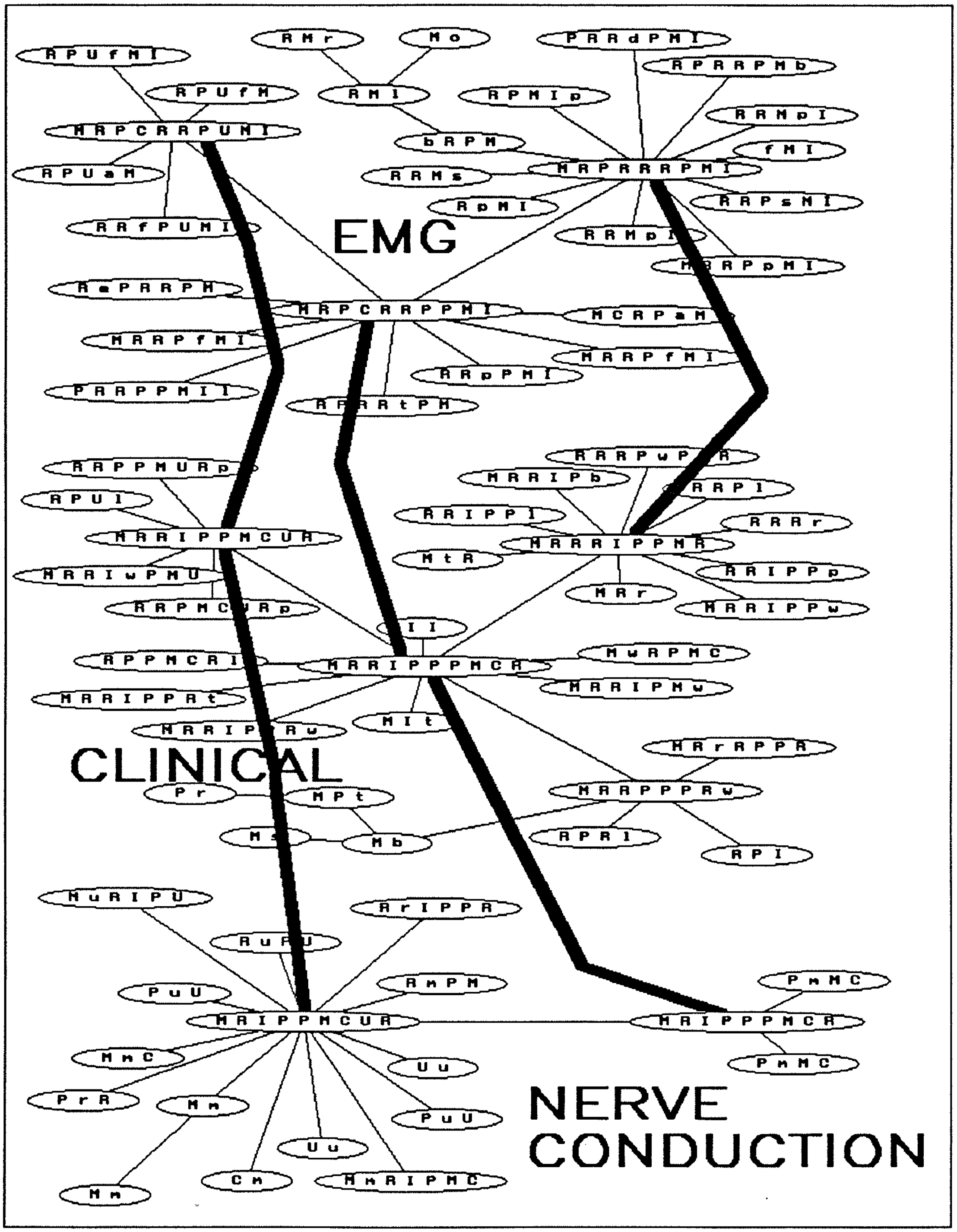


Figure 5.29: The linked junction forest of PAINULIM. Each letter in a clique represents a member variable with the letter being the initial of the variable name.

will govern the space requirement and will put an upper bound on time requirement. The mean size 5328 gives an average time requirement which is about half of the amount required by a USBN system with size 10608. Due to localization, each tree will have to be computed about 5 times (recall that findings in 3 subdomains come in 5, 6 and 4 batches respectively on an average patient) before an attention shift. Thus the overall time saving is about half compared to a USBN system.

When a user's attention is shifted from the current active tree to another tree, the latter is swapped from secondary storage into main memory and all previously acquired evidence is absorbed. The posterior distributions obtained are always based on all the available knowledge and evidence embedded in the overall system. Thus the savings in space and time do not sacrifice accuracy. The computational savings thus obtained translate immediately to smaller hardware requirement and quicker response time. With the MSBN technique, it has been possible to use hospital equipments (IBM AT compatible computers) to construct, refine and run PAINULIM interactively with EMGers right in VGH lab. EMGer's before-, inter-, and after-patient time can be utilized for knowledge acquisition and system refinement. The inference time for each batch of evidence takes from 12sec in NCV to 28sec in CLINICAL. The attention shift takes from 24sec to 106sec depending on the currently active tree and the destination tree. The efficiency in cooperation with EMGers gained greatly speeded up the development of PAINULIM (less than a year).

5.3 Other Issues in Knowledge Acquisition and Representation

5.3.1 Multiple Diseases

Many probability-based medical expert systems have assumed that diseases are mutually exclusive, for example, PATHFINDER [Heckerman, Horvitz and Nathwani 89] and

MUNIN's nanohuman system [Andreassen et al. 89]. A few did not, for example, QMR [Heckerman 90a]. When this assumption is valid, diseases can be lumped into 1 variable in the Bayesian network which simplifies the network topology.

PAINULIM considers 14 most common diseases in patients presenting with a painful or impaired upper limb. Since a patient could suffer from multiple neuromuscular diseases, the assumption of mutually exclusive diseases is not valid in the PAINULIM domain. PAINULIM has therefore represented each disease by a separate node.

Although this representation is acceptable for most of the 14 diseases, there is an exception: Amyotrophic lateral sclerosis (Als), and Anterior horn cell diseases (Ahcd). Both are disorders of the motor system. Ahcd involves only the lower motor neuronal system (between the spinal cord and the muscle), but Als additionally involves the upper motor neuron (between the brain and the spinal cord). However when one speaks of Als it is not considered as an Ahcd plus disease, but an entity by itself. Therefore, conceptually, an EMGer would never diagnose a patient to have both Als and Ahcd. This conceptual exclusion is represented by combining the 2 into 1 variable which is Motor Neuron Disease (Mnd). The variable has 3 exclusive and exhaustive outcomes: Als, Ahcd, Neither.

All the disease variables and most of the feature variables are represented as binary. That is, only 'positive' or 'negative' outcomes are allowed. 'Positive' corresponds to 'severe' or 'moderate' degree of severity, and 'negative' corresponds to 'mild' or 'absent'. This choice is made for simplicity in both knowledge acquisition and inference computation. More refined representation is planned when the system's performance is satisfactory at the current grade level.

5.3.2 Acquisition of Probability Distribution

In PAINULIM, a feature variable can have up to 7 parent disease nodes. For example, wk_wst_ex can be caused by Mnd, Rc67, Rc81, Pxutk, Pxltk, Pxpcd, and Radnn. It requires 384 numbers to fully specify the conditional probability distribution at wk_wst_ex. It would be frustrating if all these numbers have to be elicited from a human EMGer.

The *leaky noisy OR gate* model [Pearl 88, Henrion 89] is found to be a powerful tool for distribution acquisition in PAINULIM. When a symptom can be caused by *n* explicitly represented diseases, the model assumes (1) each disease has a probability to produce the symptom in the absence of all other diseases; (2) the symptom-causing probability of each disease is independent of the presence of other diseases; and (3) due to the possibility of unrepresented diseases there is a non-zero probability that the symptom will manifest in the absence of any of the diseases represented explicitly.

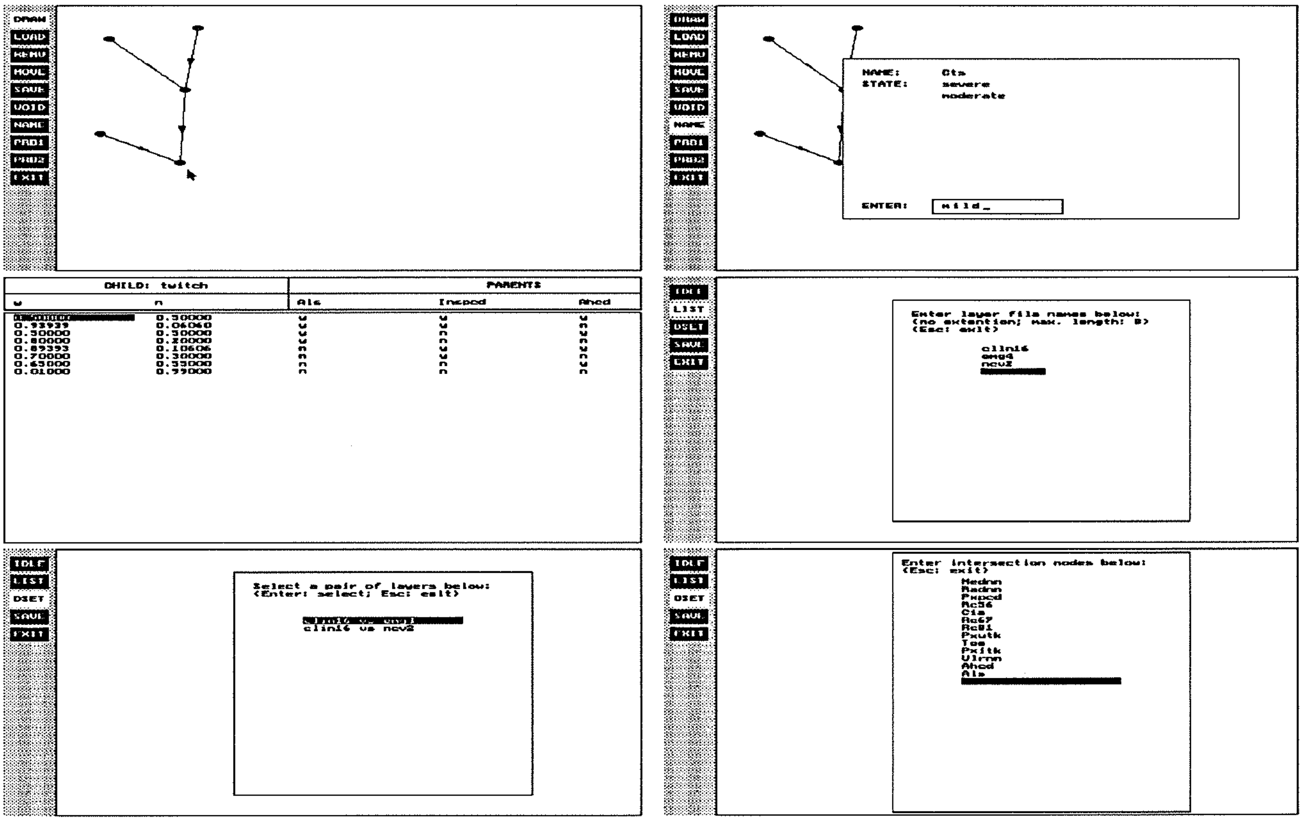
In discussion with EMGers, it is found that the above assumptions are quite valid in the PAINULIM domain. A symptom will occur in any given disease with a unique frequency. Should there exist more than one disease that could cause the same symptom, the frequency of occurrence of this particular symptom will be heightened. Using the leaky noise OR gate model, the above distribution for wk_wst_ex is assessed by eliciting only 8 numbers. Seven of them takes the form

$$p(wk\_wst\_ex = yes|Radnn = yes\ and\ every\ other\ disease = no)$$

and one of them takes the form

$$p(wk\_wst\_ex = yes|all\ 7\ parent\ disease = no)$$

5.4 Shell Implementation



NAME: Cts
STATE: severe
moderate

ENTER: mild_

CHILD: twitch		PARENTS		
w	n	Als	Inspcd	Ahcd
0.50000	0.50000	w	w	w
0.93939	0.06060	w	w	n
0.50000	0.50000	w	n	w
0.80000	0.20000	w	n	n
0.89393	0.10606	n	w	w
0.70000	0.30000	n	w	n
0.65000	0.35000	n	n	w
0.01000	0.99000	n	n	n

Enter layer file names below:
(no extention; max. length: 8)
(Esc: exit)
clini6
emg4
ncv2

Select a pair of layers below:
<Enter: select; Esc: exit>
clini6 vs emg4
clini6 vs ncv2

Enter intersection nodes below:
(Esc: exit)
Medmn
Radmn
Pxpcd
Rc56
Cts
Rc67
Rc81
Pxutk
Tos
Pxltk
Ulrmn
Ahcd
Als

Figure 5.30: Top left: drawing a sect with mouse operation; top right: naming a variable and specifying its outcomes; middle left: specifying the conditional probability distribution for a variable; middle right: specifying the sects composing the MSBN of PAINULIM; bottom left: specifying the d-sepset to be entered next; bottom right: specifying a d-sepset.

An expert system shell WEBWEAVR is implemented which incorporates the MSBN technique and leaky noisy OR gate model. This shell is in turn used to construct the PAINULIM expert system.

WEBWEAVR shell is written in *C* and is implemented in an IBM PC to suit the computing environment at the NDU, VGH where PAINULIM is constructed. It can be run in XT, AT or 386 although AT or above is recommended.

The shell consists of a graphical editor (EDITOR), a structure transformer (TRANSNET), and a consultation inference engine (DOCTR). EDITOR allows users to construct MSBNs in a visually intuitive manner. TRANSNET transforms constructed MSBNs into junction forests.

DOCTR does the evidence entering and evidential reasoning. It can work in 2 different modes: *interactive* or *batch processing.* The interactive mode allows user to enter evidence and obtain posterior distributions in an interactive and incremental way. This mode is used during consultation session. The batch processing mode allows user to enter all the evidence to a file for a patient case. Then DOCTR makes diagnosis based on the file. This mode is majorly used for system evaluation such that large amount of cases can be processed without human supervision.

DOCTR provides 2 modes for screen layout: *network* and *user-friendly.* The network mode (Figure 5.28) displays the full sect topology such that parent-child relation can be traced along with the marginal distributions. The user-friendly mode (Figure 5.31) does not display arcs of the network but labels variables with names closer to medical terminology. The former layout provides richer information while the latter gives neat screen.

Figure 5.30 illustrates the WEBWEAVR shell with 6 screen dumps. In the upper left screen, a sect is drawn using a mouse. In the upper right screen, the name of a variable and its possible outcomes are entered. In the middle left screen, the conditional probability distribution for a child variable is entered. Each sect can be constructed in this way separately. In the middle right screen, the composition of the MSBN for PAINULIM is specified. In the bottom left screen, a menu is displayed which allows a user to specify the d-sepset to be entered next. In the bottom right screen, the d-sepset between CLINICAL and EMG sects is specified.

WEBWEAVR supports the construction of any MSBN which has a covering sect.

Porting it into SUN workstation and X-Window is under consideration.

5.5 A Query Session with PAINULIM

In this section, a query session with PAINULIM in the diagnosis of a particular patient is illustrated with snapshots of major steps.

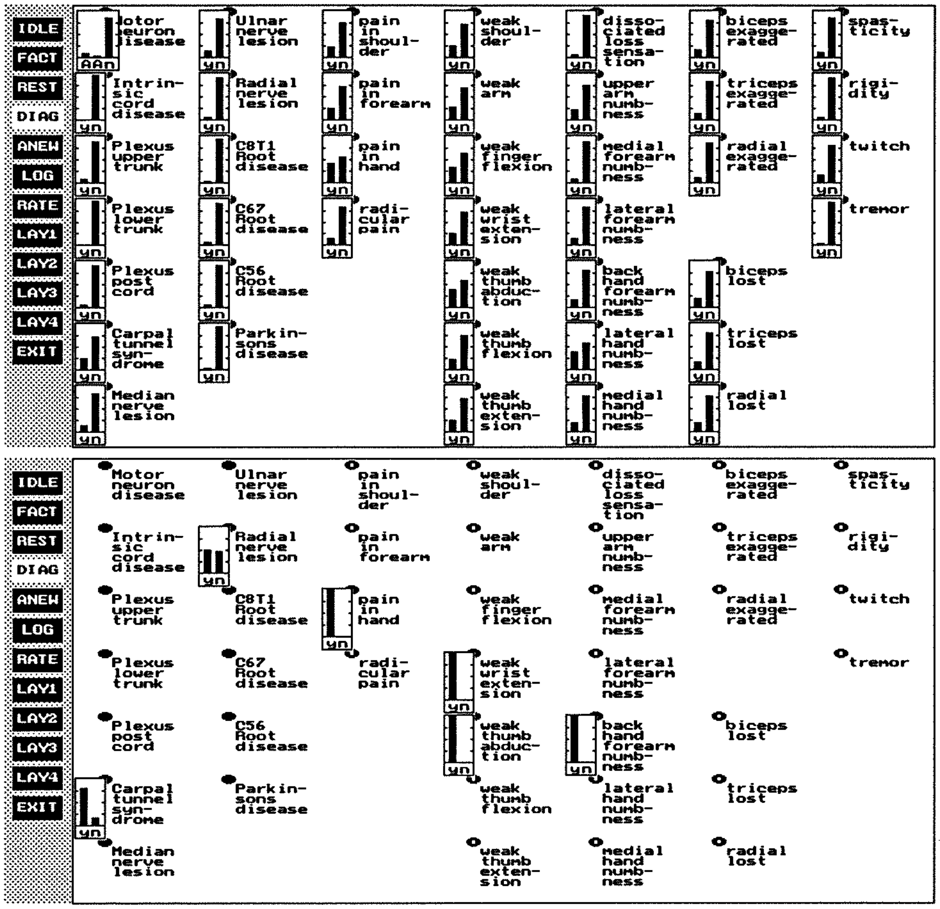


Figure 5.31: Top: CLINICAL sect with prior distributions; bottom: posterior distributions after clinical evidence is entered.

Since clinical examination is always the first stage in the diagnosis of a patient, PAINULIM begins with the CLINICAL subnet. Before any evidence is available, the prior distributions for all the diseases and symptoms can be obtained which reflect the background knowledge about the patient population in NDU of VGH. The top screen in Figure 5.31 shows the CLINICAL sect with prior distributions displayed in histograms. The user-friendly display mode is used here for better illustration.

The patient presents with tingling in the hand, weakness of thumb abduction, weakness of wrist extension, and loss of sensation in the back of the hand. After entering the evidence into the CLINICAL sect, the impression is Cts (0.808) and Radial nerve lesion (0.505). The bottom screen in Figure 5.31 highlights the diseases and features with posterior probability greater than 0.1.

After attention is shifted to NCV sect, PAINULIM suggests (Figure 5.32 top) that the most likely abnormalities on NCV are from the Median sensory study (0.785), the Median to ulnar palmar latency difference (0.777), the Median motor distal latency (0.58) and the Radial motor and sensory study (0.43 and 0.51 respectively).

After values for Median, Radial and Ulnar nerves are entered, the revised impression (Figure 5.32 bottom) is Cts (0.912) and Radial nerve lesion (0.918).

With attention further shifts to EMG sect, PAINULIM (Figure 5.33 top) prompts that the most likely EMG abnormalities are in the Edc (0.791), the Triceps (0.789), the Apb (0.827) and the Brachioradialis (0.840) muscles.

Data entered is that the Apb and the Edc are abnormal, while the Fdi is normal. The final impression (Figure 5.33 bottom) reads as Cts (0.992) and Radial lesion (0.922).

The above snapshots illustrate the *diagnostic* capability of PAINULIM. Another important usage is *education*. Given a patient with Cts and Radnn, Figure 5.34 displays highly expected (above 80% likelihood) positive features for the patient which can be used in training.

For the patient case in the above diagnosis, out of 6 highly expected CLINICAL features 4 are positive and 2 negative; out of 4 EMG features 2 are positive and 2

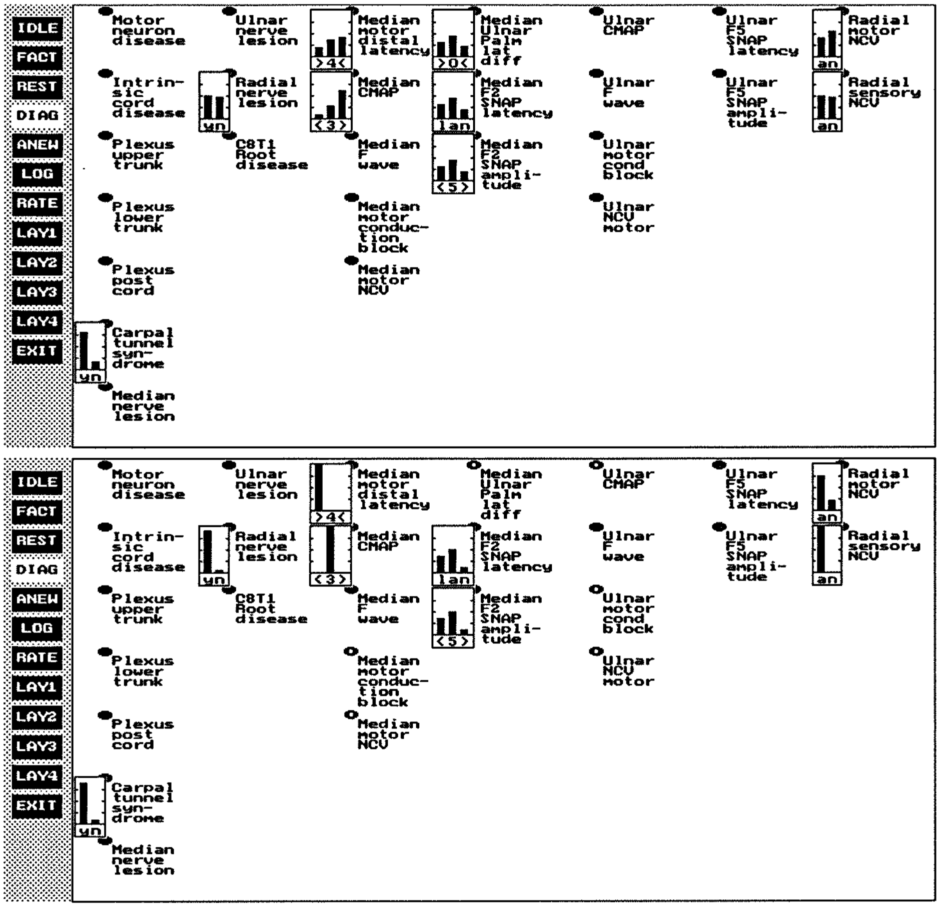


Figure 5.32: Top: NCV sect with evidence from CLINICAL and EMG sects absorbed; bottom: final diagnosis after nerve conduction studies are finished.

unchecked; out of 5 NCV features 1 is positive, 3 unchecked and 1 negative. Thus the majority of checked features matches the expectation for multiple disease Cts and Radnn, which explains the diagnosis reached above from one perspective.

5.6 An Evaluation of PAINULIM

This section presents the procedure and results of a preliminary evaluation of PAINULIM.

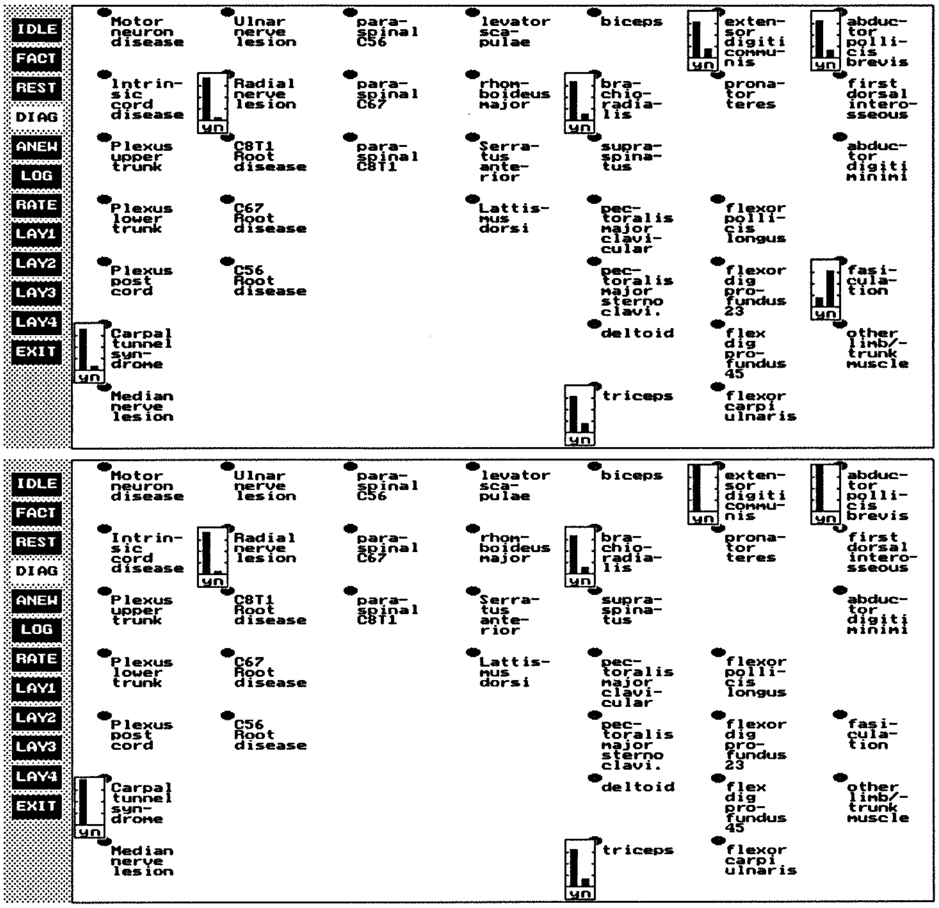


Figure 5.33: Top: EMG sect with evidence from CLINICAL sect absorbed; bottom: posterior distributions after EMG findings are entered.

5.6.1 Case Selection

76 patient cases in NDU of VGH are selected. They have been diagnosed by EMGers before used in the evaluation. The selection is conducted such that there is a balanced distribution among diseases considered by PAINULIM. Table 5.11 lists numbers of cases involved for each disease. If a case involves multiple diseases, that is, either the patient was diagnosed as suffering from multiple diseases or differentiation among several competing disease hypotheses could not be made at the time of diagnosis, then the count of

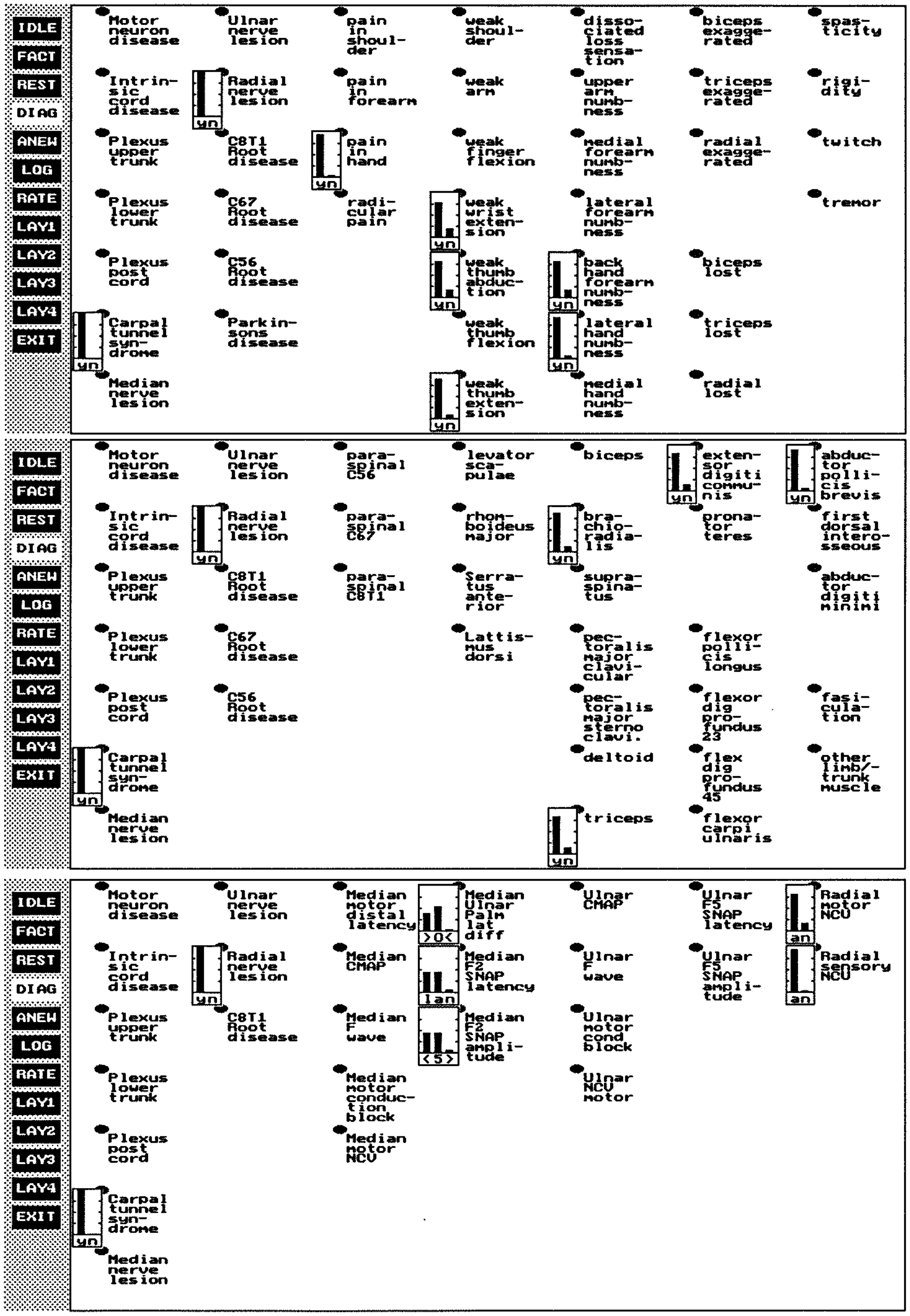


Figure 5.34: Highly expected feature presentation of a patient with both Cts and Radnn. Top: CLINICAL sect. Middle: EMG sect. Bottom: NCV sect.

each disease involved will be increased by 1. The total number of count is 124. Thus the ratio 124/76 serves as an indication of multiple-disease-ness of the case population. 3 cases not considered by PAINULIM but presenting with a painful or impaired upper limb are also included in the evaluation to test PAINULIM's performance at its limitation.

disease	count	disease	count	disease	count	disease	count
Als	5	Inspcd	2	Cts	16	Pd	3
Ahcd	1	Pxutk	6	Mednn	5	Normal	12
Rc56	8	Pxltk	8	Ulrnn	16	Other	3
Rc67	19	Pxpcd	8	Radnn	6		
Rc81	6						
subtotal	39		24		43		18

Table 5.11: Number of cases involved for each disease considered in PAINULIM

5.6.2 Performance Rating

Unlike the evaluation for QUALICON (chapter 3), in the PAINULIM domain, there is no absolutely certain way to know the 'real' disease(s) given a patient case. The best one can do is to compare the expert system with the human expert and to take the human judgment as the golden standard. Since no human expert is perfect, the evaluation conducted this way may have its limitation. Both the system and human may make the same error, or the system may be correct while the human makes an error.

In evaluating PATHFINDER, Ng and Abramson [1990] uses 'classic' cases with known diagnoses. The agreement between the known diagnosis and the disease with top probability is used as an indicator of PATHFINDER's performance. Heckerman [1990b] asks the expert to compare PATHFINDER's posterior distributions with his own and to give a rating between 0 to 10.

In the PAINULIM domain, human diagnosis can be single disease or multiple diseases, and can have different degrees of severity. Sometimes even the human diagnosis is unsure because the test studies are not well designed or incomplete. Thus a more sophisticated rating method than the one used by Ng and Abramson [1990] is desired for the evaluation of PAINULIM. A more objective rating than the one used by Heckerman [1990b] is also targeted such that the rating can be verified by persons other than the evaluator.

For each case, the patient documentation is examined and the values for feature variables are entered to PAINULIM to made a diagnosis. The posterior marginal probabilities of diseases produced by PAINULIM are compared with the EMGer's original diagnosis. 5 rating scales (EX (excellent), GD (good), FR (fair), PR (poor), and WR (wrong)) are informally defined. The definition is not exhaustive but serves as a guideline for the evaluation.

For each disease involved (possibly a disease not considered by PAINULIM as will be clear below), PAINULIM's performance is rated by the following rules.

1. For a disease judged by the EMGer as severe or moderate, the rating is EX if its posterior probability (PP) by PAINULIM falls in [0.8, 1] (GD: [0.6, 0.8); FR: [0.4, 0.6); PR: [0.2, 0.4); WR: [0, 0.2)).

2. For a disease judged by the EMGer as mild, the rating is EX if its PP falls in [0, 0.4] (GD: (0.4, 0.6]; FR: (0.6, 0.7]; PR: (0.7, 0.8]; WR: (0.8, 1]).

3. For a disease judged by the EMGer as absent, the rating is EX if its PP falls in [0, 0.2] (GD: (0.2, 0.4]; FR: (0.4, 0.6]; PR: (0.6, 0.8]; WR: (0.8, 1]).

4. For a disease judged by the EMGer as uncertain as to its likelihood because of evidence which cannot be easily explained, the rating is EX if PAINULIM suggests the same disease or other disease(s) whose anatomical site(s) is/are close to that suspected by the EMGer, with PP falling in [0, 0.6] (GD: (0.6, 0.7]; FR: (0.7, 0.8];

PR: (0.8, 0.9]; WR: (0.9, 1]).

5. For a disease not represented by PAINULIM but judged by the EMGer as the single disease diagnosis of the case in question, the rating is EX if the PPs of all represented diseases fall in [0, 0.2] (GD: (0.2, 0.4]; FR: (0.4, 0.6]; PR: (0.6, 0.8]; WR: (0.8, 1]).

6. If PAINULIM provides an alternative diagnosis of PP in [0.70, 1]; and the alternative is agreed upon by a second EMGer to whom only the evidence as presented to PAINULIM is available, then the rating is EX. The rating is also EX, if PAINULIM provides a diagnosis of PP in [0.3, 0.7) based on evidence ignored by the original EMGer, but which the second EMGer considers significant and agrees with PAINULIM.

The above 1, 2, and 3 are used when the EMGer has a confident diagnosis. 4 is used when the EMGer has a unsure diagnosis. 5 is used to rate PAINULIM's performance at its limitation. 6 is used when human diagnosis is biased by nonanatomical evidence not available to PAINULIM, or when PAINULIM behaves superior to the human diagnostician. After the rating for each disease is assigned, the lowest rating is given as the rating of PAINULIM' performance for the case in question.

5.6.3 Evaluation Results

The evaluation of 76 patient cases has the following outcomes: EX: 65, GD: 6, FR: 2, PR: 1, WR: 2. The excellent rate is 0.86 with 95% confidence interval being (0.756, 0.925) using the standard statistic technique (Appendix E). The good or excellent rate is 0.93 with 95% confidence interval being (0.853, 0.978). A closer look at the cases evaluated is worthwhile.

Among the 76 cases, there are 38 cases where the EMGer is confident about the diagnosis. The performance of PAINULIM is EX for 36 of them, and GD for the other 2. Thus for cases where evidence is complete, PAINULIM performs as well as the EMGer.

There are 8 cases where the EMGer is either confident about the diagnosis of a single mild disease, or is unsure about a single mild disease. For these 8 cases, PAINULIM's posterior probabilities for all disease hypotheses are below 0.2, which says none of the diseases is severe or moderate. Thus PAINULIM performs equally well as the EMGer in the case of a single mild disease.

For at least 7 cases, PAINULIM indicates a second disease which is not mentioned by the EMGer. Careful examination of the feature presentation would show that the corresponding features are indeed not explained by the EMGer's diagnosis. PAINULIM's behavior in these cases is considered superior to that of the EMGer. For several other cases where the EMGer's diagnosis is unsure about the possible diseases, PAINULIM indicates several candidates (with probability greater than 0.3) best matching the available evidence which provide hints to human users for a more complete test study.

For the 2 cases diagnosed as a disease in the spinal cord or peripheral nervous system but not represented in PAINULIM, PAINULIM's performance is EX for one and GD for the other. The posterior probabilities of all disease variables are below 0.3. This is interpreted as saying the patient is not in a severe or moderate disease state for any represented disease (not to be interpreted as normal). As stated earlier, PAINULIM represented 14 most common diseases of spinal cord and/or peripheral nervous system presented with a painful or impaired upper limb. The diseases in this category which are not represented in PAINULIM include: Posterior interosseous nerve lesions, Anterior interosseous nerve lesions, Axillary nerve lesions, Musculocutaneous nerve lesions, Suprascapular nerve lesions, and Long thoracic nerve lesions. They are not represented in current version of PAINULIM because their combined incidence is less than 2%. The evaluation shows that outside its representation domain, PAINULIM does not confuse such a disease with represented diseases whenever the unrepresented disease has its unique feature pattern.

On the other hand, the evaluation reveals several limitations of PAINULIM. One limitation relates to unrepresented diseases. One case in evaluation is diagnosed by the EMGer as Central Sensory Loss which is a disorder in the central nerval system. It is outside the PAINULIM domain but also characterized by a painful impaired upper limb. When restricted within the PAINULIM domain, the disease presentation is similar to Radnn. PAINUILM could not differentiate and gives probability 0.62 to Radnn (the rating is WR).

PAINULIM works with limited variables. For example in the clinical sect, the variable 'lslathnd' which represents loss of sensation in the lateral hand and/or fingers, does not allow distinguishing between the front of the hand (Median nerve, Cts or Rc67) and the back of the hand (Radial nerve, Plexus Posterior cord, Rc67). When evidence comes towards one of them but not the other, instantiation of lslathnd will enforce (incorrectly) a group of diseases.

One of the other important lesson is on the assessment of numerical data in the PAINULIM representation. PAINULIM represents each disease with 2 states. 'Positive' corresponds to 'severe' or 'moderate' degree of severity, and 'negative' corresponds to 'mild' or 'absent'. This choice is made for the reason explained in section 5.6.1. When assessing conditional probabilities for PAINULIM, the above mapping has to be kept consistently. When a probability

$$p(wk\_wst\_ex = yes|Radnn = yes\ and\ every\ other\ disease = no)$$

is assessed, the condition 'Radnn = yes' means either severe or moderate Radnn. Without emphasizing the mapping, the EMGer could include mild Radnn in his mind which increases the population considered and lower the assessed probability.

Another source of inaccuracy in the numerical probabilities come from limited experience for certain diseases. When a disease is very rare (for example, Inspcd), the EMGer giving the probabilities may have rare experience with the disease and thus provides inaccurate assessment. A few cases rated as FR or PR or WR are due to inappropriate assessment of conditional probabilities. Further fine tuning is planned for the next version of PAINULIM.

5.7 Remarks

The development of PAINULIM has shown that the MSBNs and junction forests technique provides a natural representation and an efficient inference formalism. Using the technique, the computational complexity of PAINULIM is reduced by half with no reduction of accuracy. The development of PAINULIM has thus benefited from efficient cooperation with medical staff, and rapid system construction and refinement.

The evaluation of PAINULIM's performance using 76 patient cases shows the good or excellent rate is 0.93 with 95% confidence interval being (0.853, 0.978). The case population is not selected sequentially (taking whatever patient case coming in a sequence) in order to have a balanced distribution among diseases considered by PAINULIM. If sequential population is used, even better performance can be expected. This is because normal or mild cases, or Cts cases will occupy even larger percentage than in the population used for the evaluation, and PAINULIM has performed excellently for these cases.

The deficiencies of current PAINULIM are recognized in (1) not sufficiently elaborated feature variables; (2) room for further fine tuning of numerical parameters; and (3) limitations with central nerval system disorder presenting with a painful or impaired upper limb. The improvement calls for further refinement and extension in PAINULIM's representation.

An important application of PAINULIM would be the assistance of test design. A correct and efficient neuromuscular diagnosis depends largely on good test design which will allow the gathering of minimum amount of but sufficient diagnostic information. However, when faced with difficult cases (initial evidence points to different directions) and many test alternatives, an EMGer may not be able to come up with an optimal test design. This has been true in several cases used in the evaluation. Without good test design and constrained by time and patient, an EMGer would have to make a diagnosis (after test) with limited and incomplete information.

There has been evidence that human thinking is often biased in making judgment under uncertainty [Tversky and Kahneman 74]. Thus an expert system like PAINULIM capable of reasoning rationally under uncertainty under complex conditions (multiply connected network and evidence pointing to different directions with different degrees of support to various hypotheses) would be a very helpful peer in test design. The benefit would be a better test design and an improved diagnostic quality. As demonstrated in section 5.5, given current available evidence, PAINULIM can prompt the EMGer the most likely disease(s) and most likely features. The confirmation of these features would lend further support to the disease(s); and the negative test results for these features will tend to exclude the disease(s) for further investigation. Patil et al. [1982] discuss several strategies in test design. This functionality has not been fully explored and made sufficiently explicit in the current version of PAINULIM. It is a future research topic. Since the likelihood of features depends on the current likelihood of diseases, the performance of PAINULIM in the diagnosis suggests promising potentials of its extension to the test design.

Explanation capability is important in communication between the system and its user both in diagnosis and in test design, and can add great usefulness to PAINULIM.

Section 5.5 demonstrates certain explanation functions available in the current version of PAINULIM. A sophisticated explanation facility would give reasons why a particular disease is likely; and it would give further reasons why this disease is more likely than another one. Such a facility would be a future research topic.

Chapter 6

A LEARNING ALGORITHM FOR SEQUENTIALLY UPDATING PROBABILITIES IN BAYESIAN NETWORKS

As described in section 1.3.1, in building medical expert systems, the probabilities necessary for the construction of Bayesian networks usually have to be elicited from medical experts. The values thus obtained are liable to be inaccurate. This has also been the experience with PAINULIM (Chapter 5). The sensitivity analysis of a medical expert system PATHFINDER shows that probabilities play an important role in PATHFINDER's performance and minor changes in all of its probabilities have a substantial impact on performance [Ng and Abramson 90]. Therefore, methodologies which allow the representation of uncertainty of probabilities and improvement of their accuracy are needed.

This chapter presents an *Algorithm for Learning by Posterior Probabilities (ALPP)* for sequential updating of probabilities in Bayesian networks. The results are mainly taken from Xiang, Beddoes and Poole [1990b]. Section 6.1 reviews 3 representations of uncertainty of probabilities: *interval, auxiliary variable, ratio* and corresponding methods for updating probabilities. Section 6.2 presents the idea of learning from expert's posterior probabilities in order to overcome the limitation of the existing updating method for the ratio representation. Section 6.3 presents ALPP. Section 6.4 proves the convergence of ALPP. Section 6.5 presents the performance of ALPP in simulation.

6.1 Background

Several formalisms for representation of uncertainty of probabilities in Bayesian networks have been proposed. Some of them have corresponding methodologies incorporated to improve the accuracy of probabilities. One formalism represents probabilities by intervals [Fertig and Breese 90]. The size of an interval signifies the degree of uncertainty to the probability it represents. No known method *directly* uses this representation for improvement of accuracy of probabilities.

Another formalism represents the uncertainty of probabilities by probabilities of probabilities [Spiegelhalter 86, Neapolitan 90, Spiegelhalter and Lauritzen 90]. Auxiliary variables (nodes) are added to Bayesian networks. Their outcomes are the probabilities of variables in the original Bayesian net. The distributions of these auxiliary variables, therefore, are the probabilities of probabilities. With this representation, the updating of probabilities can be performed within the probabilistic inference process.

Yet another formalism represents probabilities by ratios of imaginary sample sizes [Cheeseman 88a], [Spiegelhalter, Franklin and Bull 89]. A probability $p(symptomA|$ $diseaseB)$ is represented as x/y where y is an imaginary patient population with disease B and among these patients x of them show symptom A. The less certain the probability $p(symptomA|\ diseaseB)$ is, the smaller the integer y. Spiegehalter, Franklin and Bull [1989] present a procedure for sequentially updating probabilities represented as ratios. The procedure is presented in the form directly applicable to Bayesian networks of diameter 1 with a single parent node (possibly multiple children). Here, a single child case $(p(symptomA|diseaseB))$ is used to illustrate the idea. When a new patient with disease B is observed, the sample size y is increased by 1, and the sample size x is increased by 1 or 0 depending on if the patient shows symptom A. With more and more patient cases processed, the probability approaches the correct value. The improvement of this method is the focus of this chapter.

The limitation of the updating method for the ratio representation is the underlying assumption that when updating $p(symptomA|diseaseB)$, whether disease B is true or false *is known with certainty* (thus the method will be referred as $\{0,1\}$ *distribution learning*). The assumption is not realistic in many applications. A doctor would not always be 100% confident about a diagnosis he or she made of a patient. This argument is expanded in the section below.

6.2 Learning from Posterior Distributions

The spirit of $\{0,1\}$ distribution learning is to improve the precision of probabilities elicited from the human expert by learning from available data. What else does one really have in medical practice in addition to patients' symptoms? It may be possible, in some medical domain, that diagnoses can be confirmed with certainty. But this is not commonplace. A successful treatment is not always an indication of correct diagnosis. A disease can be cured by a patient's internal immunity or by a drug with wide disease spectrum. One subtlety of medical diagnosis comes from the unconfirmability for each individual patient case.

For most medical domains, the available data beside patients' symptoms are physician's subjective posterior probabilities (PPs) of disease hypotheses given the *overall* pattern of patient's symptoms. They are not distributions with values from $\{0,1\}$, but rather distributions from $[0,1]$[1]. The diagnoses appearing in patients' files are typically not the diagnoses that have been concluded definitely; they are only the top ranking diseases with physician's subjective PP omitted. The assumption of $\{0,1\}$ posterior disease distribution may, naively, be interpreted as an approximation to [0, 1] distribution with 1

[1] Note that $\{0,1\}$ denotes a set containing only elements 0 and 1, and $[0,1]$ is a domain of real numbers between 0 and 1 inclusive.

substituting top ranking PP, and 0 substituting the rest. This approximation loses useful information. Thus a way of learning directly from [0, 1] posterior distribution seems more natural and anticipates better performance.

In dealing with *learning* problem in a Bayesian network setting, three 'agents' are concerned: the real world (D_r, P_r), the human expert (D_e, P_e), and the artificial system (D_s, P_s). It is assumed that all 3 can be modeled by Bayesian networks. As the building of an expert system involves specifying both the topology of DAG D and probability distribution P, the improvement can also be separated into the two aspects. For the purpose of this chapter, D_r, D_e, and D_s are assumed identical, leaving to be improved only the accuracy of quantitative assignment of P_s.

As reviewed in section 1.3.3, a medical expert system based on Bayesian nets usually directs its arcs from disease nodes to symptom nodes, thus encoding quantitative knowledge by priors of diseases and conditional probabilities (CPs) of symptom given diseases. This results in the ease of DAG construction, simplicity of net topology, and portability of the system. Given that a Bayesian net is constructed with such directionality, and the desire to improve accuracy of CPs by PPs, a question which arises is whether PPs are any better in quality compared to CPs also supplied by the human expert. To avoid possible confusion, it is emphasized that the CP here is the conditional probabilities of a symptom variable given its disease parent variables, and the PP is the joint probabilities of a set of (relevant) disease variables given the outcomes of a set of symptom variables. The former is stored explicitly in Bayesian networks as parameter, while the latter is not explicit parameter even if the arcs in the network were directed from symptoms to diseases.

In my cooperation with medical staff, it was found that the causal network is a natural model to view the domain, however, the task of estimating CPs is more artificial than natural to them. Forming posterior judgments is their daily practice. An expert is an expert in that he/she is skilled at making diagnosis (posterior judgement), not necessarily skilled at estimating CPs. It is the expert's posterior judgment that is the behavior one wants the expert system to simulate.

An excellent argument supporting the idea of using PPs to improve CPs has been published by Neapolitan [1990]:

> For example, sometimes it is easier for a person to ascertain the probability of a cause given an effect than that of an effect given a cause. Consider the following situation: If a physician had worked at the same clinic for a number of years, he would have seen a large population of similar people with certain symptoms. Since his job is to reason from the symptoms to determine the likelihood of diseases, through the years he may have become adept at judging the probabilities of diseases given symptoms for the population of people who attend this clinic. On the other hand, a physician does not have the task of looking at a person with a known disease and judging whether a symptom is present. Hence it is not likely that the physician would have acquired the ability from his experience to judge the probabilities of symptoms given diseases. He does ordinarily learn something about this probability in medical school. However, if a particular disease were rare in the population, whereas a particular symptom of the disease were common, and the physician had not studied the disease in medical school, he would certainly not be able to determine the probability of the symptom given the disease. On the other hand, he could readily determine the probability of the disease given the symptom for this particular population. We see then that in this case it is easier to ascertain the probabilities of causes given effects. Notice that these conditional probabilities are only valid for the population of people who attend

> that particular clinic, and thus we would not want to incorporate them into an expert system. We could, however, use the definition of conditional probability to determine the probabilities of symptoms given diseases form these conditional probabilities obtained from the physician. These latter probabilities could then be used in an expert system which would be applicable at any clinic.

The task imposed on expert by the method presented here is actually more natural than that argued by Neapolitan. Experts will be asked for PPs given a set of symptoms (not just one as is seen in a moment), and thus the task will be close to their daily practice.

Furthermore, Kuipers and Kassier [1984] has been cited by Shachter and Heckerman [1987] to show "experts are more comfortable when their beliefs are elicited in the causal direction"; Tversky and Kahneman [1980] has been cited by Pearl [1988] to show "people often prefer to encode experiential knowledge in causal schemata, and as a consequence, rules expressed in causal forms are assessed more reliably". Neither Shachter and Heckerman nor Pearl made explicit distinction between 'qualitative' and 'quantitative' knowledge. Careful examination of the studies by Kuipers and Kassier and Tversky and Kahneman reveals the following two points. Both experts and ordinary people prefer to encode their qualitative knowledge in terms of causal schemata. People often give higher probabilities in the causal direction than those dictated by probability theory. These studies support the reliability of causal elicitation of qualitative knowledge. These studies do not support the reliability of causal elicitation of quantitative (probabilistic) knowledge. The reliability issue is left open.

Finally, Spiegehalter et al. [1989] has reported that the assessment of CPs by human experts is generally reliable, but has a tendency to be too extreme (too close to 0 or 1).

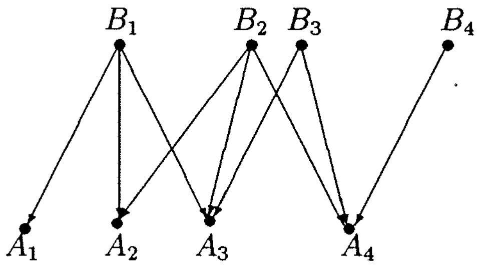


Figure 6.35: An example of D(1)

If one could assume that the human expert carries a mental Bayesian network and PPs are produced by the network, it is postulated that the CPs the expert articulates, which consists of P_s of the system, could be a distorted version of those in P_e. Also, P_e may differ from P_r in general. Thus, 4 categories of probability distributions are distinguished: P_r, P_e, P_s, and the PPs produced by P_e (written as p_e). One's access to only P_s and $p_e(hypotheses|evidence)$ is assumed. The goal is to use the latter to improve P_s such that the system's behavior will approach that of expert.

How can PP be utilized in the updating? The basic idea is: instead of updating imaginary sample sizes by 1 or 0, increase them by the measure of certainty of the corresponding diseases. The expert's PP is just such a measure. Formal treatment is given below.

6.3 The Algorithm for Learning by Posterior Probability (ALPP)

The following notation is used:

$D(1)$ DAGs of diameter 1 (The *diameter* is the length of the longest directed path in the DAG. An example of D(1) is given in Figure 6.35.);

$(D(1), P)$ Bayesian net with diameter 1 and underlying distribution P;

$B_i \in \mathcal{B}_i = \{b_{i1}, \ldots, b_{in_i}\}$ the ith parent variable in $D(1)$ with sample space $\mathcal{B}_i$;

$A_j \in \mathcal{A}_j = \{a_{j1}, \ldots, a_{jm_j}\}$ the jth child variable in $D(1)$ with sample space $\mathcal{A}_j$;

$A \in \Psi(A)$ the set of all child variables in $D(1)$ with space $\Psi(A)$;

$\mathbf{a}_{jl}$ element of $\Psi(A)$ with A_j instantiated as a_{jl};

$y_{k_1 k_2 \ldots k_n}$ the imaginary sample size for joint event $b_{1k_1} b_{2k_2} \ldots b_{nk_n}$ being true;

$x_{l_j k_1 k_2 \ldots k_n}$ the imaginary sample size for joint event $a_{jl_j} b_{1k_1} \ldots b_{nk_n}$ being true;

$\delta^c_{l_j}$ impulse function which equals 1 if for the cth fresh case A_j's outcome is a_{jl_j}, and equals 0 otherwise (superscripts denote the orders of fresh cases);

$p_r(), p_e(), p_s()$ probabilities contained or generated by $(D_r(1), P_r)$, $(D_e(1), P_e)$ and $(D_s(1), P_s)$ respectively.

A Bayesian net $(D(1), P)$ [2] is considered where the underlying distribution is composed via

$$p(B_1 \ldots B_N A_1 \ldots A_M) = \prod_{i=1}^{N} p(B_i) \prod_{j=1}^{M} p(A_j | \pi_j)$$

where π_j is the set of parents of A_j.

Each of the CPs is internally represented in the system as a ratio of 2 imaginary sample sizes. For child node A_1 having its parent nodes $B_1 \ldots B_Q$ $(Q \geq 1)$, a corresponding CP is

$$p^c_s(a_{1l_1} | b_{1k_1} \ldots b_{Qk_Q}) = x^c_{l_1 k_1 \ldots k_Q} / y^c_{k_1 \ldots k_Q}$$

where the superscript c signifies the cth updating. Only the real numbers $x^c_{l_1 k_1 \ldots k_Q}$ and $y^c_{k_1 \ldots k_Q}$ are stored. The prior probabilities for A_1's parents can be derived as

[2] Whether it is a subnet or a net by itself is irrelevant.

$$p^c_s(b_{ik_i}) = \frac{\sum_{k_1,\ldots,k_{i-1},k_{i+1},\ldots,k_Q} y^c_{k_1\ldots k_Q}}{\sum_{k_1,\ldots,k_Q} y^c_{k_1\ldots k_Q}}$$

For a $(D(1), P)$ with M children and with all variables binary, the number of numbers to be stored in this way is upper bounded by $2\sum_{i=1}^{M} 2^{\eta_i}$ where η_i is the in-degree of child node i. Storage saving can be achieved when different child nodes share a common set of parents.

Updating P is done one child node at a time through updating xs and ys associated with the node as illustrated above. Once the xs and ys are updated, the updated CPs and priors can be derived. The order in which child nodes are selected for updating is irrelevant.

Without losing generality, we describe the updating with respect to above mentioned child node A_1. For the cth fresh case where $\mathbf{a}^c$ is the symptoms observed, the expert provides the PP distribution $p_e(b_{1k_1}\ldots b_{Nk_N}|\mathbf{a}^c)$. This is transformed into

$$p_e(b_{1k_1}\ldots b_{Qk_Q}|\mathbf{a}^c) = \sum_{b_{Q+1},\ldots,b_N} p_e(b_{1k_1}\ldots b_{Nk_N}|\mathbf{a}^c)$$

The sample sizes are updated by

$$x^c_{l_1k_1\ldots k_Q} = x^{c-1}_{l_1k_1\ldots k_Q} + \delta^c_{l_1} p_e(b_{1k_1}\ldots b_{Qk_Q}|\mathbf{a}^c)$$

$$y^c_{k_1\ldots k_Q} = y^{c-1}_{k_1\ldots k_Q} + p_e(b_{1k_1}\ldots b_{Qk_Q}|\mathbf{a}^c).$$

6.4 Convergence of the Algorithm

An expert is called *perfect* if $(D_e(1), P_e)$ is identical to $(D_r(1), P_r)$.

Theorem 13 *Let a Bayesian network $(D_s(1), P_s)$ be supported by a perfect expert equipped with $(D_e(1), P_e)$. No matter what initial state P_s is in, it will converge to P_e by ALPP.*

Proof:

Without losing generality, consider the updating with respect to A_1 in section 6.3.

(1) Priors. Let $\{\mathbf{a}(1), \mathbf{a}(2), \ldots\}$ be the set of all possible conjuncts of evidence. Let $u(t)$ be the number of times at which event $\mathbf{a}(t)$ is true in c cases; and $\sum_t u(t) = c$. From the prior updating formula of ALPP,

$$\begin{aligned}
&\lim_{c\to\infty} p_s^c(b_{ik_i}) \\
&= \lim_{c\to\infty} \left(\frac{\sum_{k_1,\ldots,k_{i-1},k_{i+1},\ldots,k_Q} y^0_{k_1\ldots k_Q} + \sum_{k_1,\ldots,k_{i-1},k_{i+1},\ldots,k_Q} \sum_{w=1}^{c} p_e(b_{1k_1}\ldots b_{Qk_Q}|\mathbf{a}^w)}{c + \sum_{k_1,\ldots,k_Q} y^0_{k_1\ldots k_Q}} \right) \\
&= \lim_{c\to\infty} \frac{1}{c} \left(\sum_{k_1,\ldots,k_{i-1},k_{i+1},\ldots,k_Q} \sum_t p_e(b_{1k_1}\ldots b_{Qk_Q}|\mathbf{a}(t)) u(t) \right) \\
&= \sum_{k_1,\ldots,k_{i-1},k_{i+1},\ldots,k_Q} \sum_t p_e(b_{1k_1}\ldots b_{Qk_Q}|\mathbf{a}(t)) p_r(\mathbf{a}(t)) \\
&= \sum_{k_1,\ldots,k_{i-1},k_{i+1},\ldots,k_Q} \sum_t p_e(b_{1k_1}\ldots b_{Qk_Q}|\mathbf{a}(t)) p_e(\mathbf{a}(t)) \qquad \text{(perfect expert)} \\
&= \sum_{k_1,\ldots,k_{i-1},k_{i+1},\ldots,k_Q} p_e(b_{1k_1}\ldots b_{Qk_Q}) = p_e(b_{ik_i})
\end{aligned}$$

(2) CPs. Let $u_{1l_1}(t)$ be the number of times at which event $\mathbf{a}_{1l_1}(t)$ is true in c cases. Following ALPP, one has

$$\begin{aligned}
\lim_{c\to\infty} p_s^c(a_{1l_1}|b_{1k_1}\ldots b_{Qk_Q}) &= \lim_{c\to\infty} \frac{x^0_{l_1k_1\ldots k_Q} + \sum_{w=1}^{c} \delta^w_{l_1} p_e(b_{1k_1}\ldots b_{Qk_Q}|\mathbf{a}^w)}{y^0_{k_1\ldots k_Q} + \sum_{v=1}^{c} p_e(b_{1k_1}\ldots b_{Qk_Q}|\mathbf{a}^v)} \\
&= \lim_{c\to\infty} \frac{\frac{1}{c}\sum_{w=1}^{c} \delta^w_{l_1} p_e(b_{1k_1}\ldots b_{Qk_Q}|\mathbf{a}^w)}{\frac{1}{c}\sum_{v=1}^{c} p_e(b_{1k_1}\ldots b_{Qk_Q}|\mathbf{a}^v)} \\
&= \frac{\lim_{c\to\infty} \frac{1}{c}\sum_t p_e(b_{1k_1}\ldots b_{Qk_Q}|\mathbf{a}_{1l_1}(t)) u_{1l_1}(t)}{\lim_{c\to\infty} \frac{1}{c}\sum_z p_e(b_{1k_1}\ldots b_{Qk_Q}|\mathbf{a}(z)) u(z)} \\
&= \frac{\sum_t p_e(b_{1k_1}\ldots b_{Qk_Q}|\mathbf{a}_{1l_1}(t)) p_r(\mathbf{a}_{1l_1}(t))}{\sum_z p_e(b_{1k_1}\ldots b_{Qk_Q}|\mathbf{a}(z)) p_r(\mathbf{a}(z))} \\
&= \frac{\sum_t p_e(b_{1k_1}\ldots b_{Qk_Q}|\mathbf{a}_{1l_1}(t)) p_e(\mathbf{a}_{1l_1}(t))}{\sum_z p_e(b_{1k_1}\ldots b_{Qk_Q}|\mathbf{a}(z)) p_e(\mathbf{a}(z))} \qquad \text{(perfect expert)} \\
&= \frac{p_e(b_{1k_1}\ldots b_{Qk_Q} a_{1l_1})}{p_e(b_{1k_1}\ldots b_{Qk_Q})} = p_e(a_{1l_1}|b_{1k_1}\ldots b_{Qk_Q})
\end{aligned}$$

□

A *perfect* expert is never available. One needs to know the behavior of ALPP when supported by an imperfect expert. This leads to the following theorem.

Theorem 14 *Let p_s^c be any resultant probability in $(D_s(1), P_s)$ after c updating by ALPP. p_s^c converges to a continuous function of P_e.*[3]

Proof:

(1) Continuity of priors.

Following the proof of theorem 13, the prior $p_s^c(b_{ik_i})$ converges to

$$f = \sum_{k_1,\ldots,k_{i-1},k_{i+1},\ldots,k_Q} \sum_t p_e(b_{1k_1}\ldots b_{Qk_Q}|\mathbf{a}(t))\frac{u(t)}{c}$$

where $p_e(b_{1k_1}\ldots b_{Qk_Q}|\mathbf{a}(t))$ is an elementary function of P_e, and so does f. Therefore, $p_s^c(b_{ik_i})$ converges to a continuous function of P_e.

(2) Continuity of CP.

From theorem 13, $p_s^c(a_{1l_1}|b_{1k_1}\ldots b_{Qk_Q})$ converges to

$$f = \frac{\sum_t p_e(b_{1k_1}\ldots b_{Qk_Q}|\mathbf{a}_{1l_1}(t))\frac{u_{1l_1}(t)}{c}}{\sum_z p_e(b_{1k_1}\ldots b_{Qk_Q}|\mathbf{a}(z))\frac{u(z)}{c}}$$

where $p_e(b_{1k_1}\ldots b_{Qk_Q}|\mathbf{a}(z))$ is an elementary function of P_e.

□

Theorem 14, together with Theorem 13, says that when the discrepancy between P_e and P_r is small, the discrepancy between P_s and P_r (P_e as well) will be small after enough learning trials. The specific form of the discrepancy is left open.

The absolute value of PPs is not really important in many applications but the posterior ordering of diseases be. A set of PPs defines such a posterior ordering. Claim a 100% *behavior match* between (D, P_1) and (D, P_2) if for any possible set of symptoms the

[3] By 'F is a function of P_e', it is meant that F takes probability variables in P_e as its independent variables which in turn themselves have [0,1] as their domain.

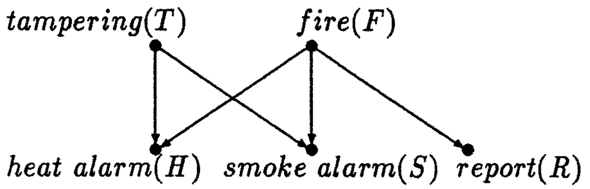


Figure 6.36: Fire alarm example for simulation

two give the same ordering. The minimum difference between successive PPs of (D, P_1) defines a threshold. Unless the maximum difference between corresponding PPs from 2 (D, P)s exceeds the threshold, 100% behavior match is guaranteed. Thus as long as the discrepancy between P_e and P_r is within some $(D_r(1), P_r)$ dependent threshold, a 100% match between the behavior of P_s and that of P_e is anticipated.

6.5 Simulation Results

Several simulations have been run using the example in Figure 6.36. It is a revised version of the smoke-alarm example in Poole and Neufeld [1988]. Here *heat alarm, smoke alarm* and *report* are used as evidence to estimate the likelihood of joint event *tampering* and *fire*. Each variable, denoted by uppercase letters, takes binary values. For example, F has value f or $\overline{f}$ which signify the event *fire* being *true* or *false*.

The simulation set-up is illustrated in Figure 6.37. Logical sampling [Henrion 88] is used in the real world model $(D_r(1), P_r)$ to generate scenarios $\{T_r, F_r, H_r, S_r, R_r\}$. The observed evidence $\{H_r, S_r, R_r\}$ is fed into $(D_e(1), P_e)$. The posterior distribution $p_e(TF| H_rS_rR_r)$ is computed by the expert model and is forwarded to update system model $(D_s(1), P_s)$.

To compare the performance between ALPP and $\{0, 1\}$ distribution learning, a control model $(D_c(1), P_c)$ is constructed in the set-up. It has the same DAG structure and initial

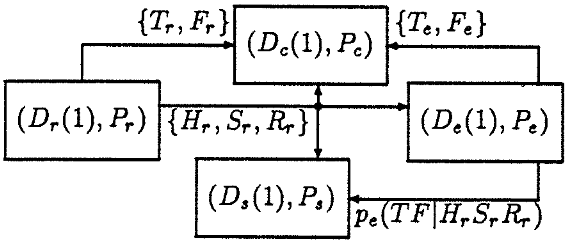


Figure 6.37: Simulation set-up

probability distribution as $(D_s(1), P_s)$ but is updated by $\{0,1\}$ distribution learning.[4] Two different sets of diagnoses are utilized in different simulation runs by $(D_c(1), P_c)$ for the purpose of comparison. In simulation 1, 2 and 3 to be described below, the top diagnosis $\{T_e, F_e\}$ made by $(D_e(1), P_e)$ is used. In simulation 4, the scenario $\{T_r, F_r\}$ is used. The former simulates the situation where posterior judgments could not be fully justified. The latter simulates the case where such justification is indeed available.

For all the simulations P_r is the following distribution.

$p(h\|ft)$	0.50	$p(s\|ft)$	0.60
$p(h\|f\bar{t})$	0.90	$p(s\|f\bar{t})$	0.92
$p(h\|\bar{f}t)$	0.85	$p(s\|\bar{f}t)$	0.75
$p(h\|\bar{f}\bar{t})$	0.11	$p(s\|\bar{f}\bar{t})$	0.09
$p(r\|f)$	0.70	$p(f)$	0.25
$p(r\|\bar{f})$	0.06	$p(t)$	0.20

P_s and P_c are distributions with the maximal error 0.3 relative to P_r. The initial imaginary sample size for each joint event FT is set to 1. Such setting is mainly for the purpose of demonstrating the convergence of ALPP under poor initial condition. The distribution error should generally be smaller and initial sample sizes be much larger in case of practical application where the convergence will be a slowly evolving process.

[4] The original form of $\{0,1\}$ distribution learning is extended to the form applicable to D(1).

$(D_e(1), P_e)$		$(D_s(1), P_s)$		$(D_c(1), P_c)$	
trial No.	diag. rate	behv. mat. rate	max. err. S-E	behv. mat. rate	max. err. C-E
0			0.30		0.30
1~25	68%	60%	0.14	48%	0.21
26~50	76%	96%	0.10	12%	0.25
51~100	80%	100%	0.06	36%	0.27
101~200	76%	100%	0.03	33%	0.28

Table 6.12: Summary for simulation 1. $|P_e - P_r| = 0$.

Simulation 1 is run with P_e being the same as P_r which assumes a perfect expert. The results are depicted in Table 6.12. The diagnostic rate of $(D_e(1), P_e)$ is defined as A/N where N is the base number of trials and A is the number of trials where the top diagnosis agrees with $\{T_r, F_r\}$ simulated by $(D_r(1), P_r)$. The behavior matching rate of $(D_s(1), P_s)$ relative to $(D_e(1), P_e)$ is defined as B/N where B is the number of trials in which $(D_s(1), P_s)$'s diagnoses have the same ordering as $(D_e(1), P_e)$. The behavior matching rate of $(D_c(1), P_c)$ to $(D_e(1), P_e)$ is similarly defined.

The results show convergence of probability values in P_s to those in P_e (maximum error(S-E) $\rightarrow 0$). The behavior matching rate of $(D_s(1), P_s)$ increases along with the convergence of probabilities and finally $(D_s(1), P_s)$ behaves exactly the same as $(D_e(1), P_e)$. An interesting phenomenon is that, despite $P_e = P_r$, the diagnostic rate of $(D_e(1), P_e)$ is only 76% in the total 200 trials. Though the rate is dependent on the particular (D, P), it is expected to be less than 100% in general. In terms of medical diagnosis, this is because some disease may manifest through unlikely symptoms, making other diseases more likely. In an uncertain world with limited evidence, mistakes in diagnoses are unavoidable. More importantly, P_s converges to P_e under the guidance of this 76% correct diagnoses while P_c does not. The maximum error of P_c remains about the same throughout the 200 trials and the behavior matching rate of $(D_c(1), P_c)$ is low. Similar performance of $(D_c(1), P_c)$ is seen in the next two simulations. This shows that under the circumstances where good experts are available but confirmations to diagnoses are not available, ALPP is robust while {0,1} distribution learning will be misled by the errors in diagnoses. This is not surprising since the assumption underlying {0,1} distribution learning is violated. One will gain more insight into this from the results of simulation 4 below.

An imperfect expert is assumed in simulation 2 (Table 6.13). The distribution P_e differed from P_r up to 0.05. Because of this error, P_s converges to neither P_e (as shown in Table 6.13) nor P_r. But the error between P_s and P_e approaches a small value (about 0.07) such that after 200 trials the behavior of P_s matches that of P_e perfectly.

$(D_e(1), P_e)$		$(D_s(1), P_s)$		$(D_c(1), P_c)$	
trial No.	diag. rate	behv. mat. rate	max. err. S-E	behv. mat. rate	max. err. C-E
0			.300		.300
1∼100	84%	82%	.058	32%	.272
101∼200	86%	92%	.122	43%	.287
201∼300	80%	100%	.067	32%	.290
301∼400	83%	100%	.076	36%	.292

Table 6.13: Summary of simulation 2. $|P_e - P_r| = 0.05$.

If the discrepancy between P_s and P_r is further increased so that the threshold discussed in last section is crossed, $(D_s(1), P_s)$ will no longer converge to $(D_e(1), P_e)$. This is the case in simulation 3 (Table 6.14) where the maximum error and root mean square error (rms) between P_e and P_r are 0.15 and 0.098 respectively. The rms error is calculated over all the priors and conditional probabilities of P_e and P_r. Introduction of rms error for interpretation of simulation 3 is because maximum error itself, when not approaching to 0, does not give good indication of the distance between the two.

$(D_e(1),P_e)$		$(D_s(1),P_s)$				
trial No.	diag. rate	diag. rate	behv. mat. rate	rms err. S-E	rms err. S-R	max. err. S-E
0				.170	.169	.39
1∼25	80%	84%	20%	.086	.079	.17
26∼75	74%	74%	40%	.071	.068	.11
76∼175	73%	73%	53%	.050	.083	.087
176∼375	79%	79%	46%	.059	.072	.095
376∼475	78%	78%	43%	.061	.071	.119
$(D_e(1),P_e)$		$(D_c(1),P_c)$				
trial No.	diag. rate	diag. rate	behv. mat. rate	rms err. C-E	rms err. C-R	max. err. C-E
0				.170	.169	.39
1∼25	80%	80%	32%	.110	.091	.20
26∼75	74%	74%	38%	.110	.092	.16
76∼175	73%	73%	38%	.098	.092	.15
176∼375	79%	79%	23%	.100	.090	.15
376∼475	78%	78%	26%	.096	.084	.15

Table 6.14: Summary of simulation 3. $|P_e - P_r| = 0.15$.

The simulation shows that the behavior matching rate of P_s and P_e is quite low (43% after 475 trials). Since the diagnostic rate of P_e is also lower (77%), one could ask which one is better. One way of viewing this is to compare the diagnostic rates. It is observed that, among P_s, P_c and P_e, no one is superior than others if *only* top diagnosis is concerned. More careful examination can be obtained by comparison of distances among models. It turns out that the distance (S-E) and distance (S-R) are smaller than the distance (E-R) with corresponding rms errors 0.061, 0.071 and 0.098 respectively.

The above 3 simulations assume that only the subjective posterior judgments are available. In simulation 4, it is assumed that the correct diagnosis is also accessible. This time, $(D_c(1),P_c)$ is supplied with the scenario generated by $(D_r(1),P_r)$. P_e is the same as P_r.

The results (Table 6.15) shows that ALPP converges much quicker than {0,1} distribution learning even the latter has access to 'true' answers to the diagnostic problem. After 1500 trials, $(D_s(1), P_s)$ reduces its maximum error from $(D_e(1), P_e)$ to 0.041 and matches the latter's behavior perfectly, while $(D_c(1), P_c)$ is still on its way of convergence with its error about 2 times larger and its behavior matching rate 80%.

$(D_e(1), P_e)$		$(D_s(1), P_s)$		$(D_c(1), P_c)$	
trial No.	diag. rate	behv. mat. rate	max. err. S-E	behv. mat. rate	max. err. C-E
0			.300		.300
1~100	88%	95%	.130	60%	.375
101~600	78%	98%	.048	72%	.045
601~1100	78%	93%	.052	61%	.075
1101~1500	79%	100%	.041	80%	.079
1501~1700	81%	100%	.025	85%	.093

Table 6.15: Summary of simulation 4. $|P_e - P_r| = 0$ and 'true' scenario is accessible to $\{0, 1\}$ distribution learning (P_c).

Real world scenarios could be distinguished as being *common* or *exceptional*. An expert with knowledge about the real world tends to catch the common and to ignore the exceptional. Thus the diagnostic rate will never be 100%. This is the best one could do given the limited evidence. The PPs provided by the expert contain the information about the entire domain, while a scenario contains only the information about this particular scene. Thus, although both $(D_s(1), P_s)$ and $(D_c(1), P_c)$ converge, the former converges quicker. This difference in convergence speed is expected to emerge wherever the diagnosis is difficult and the diagnostic rate of the expert is low (for example, in some area where disease mechanism is not well understood and diagnostic criteria are not well established).

6.6 Remarks

Several features of ALPP can be appreciated through the theoretical analysis and simulation results presented.

- ALPP provides an alternative way of sequentially updating conditional probabilities in Bayesian nets when confirmed diagnosis is not available.

- Under ideal conditions (perfect expert for ALPP and confirmed diagnosis for {0,1} distribution learning), both ALPP and {0,1} distribution learning converge to the real world model. However, ALPP converges faster than {0,1} distribution learning due to the richer information contained in expert's posterior judgments.

- When human expert's judgment is the only available source and the expert is not perfect but fairly good (his mental model is different from but close to the real world model), ALPP still converges to expert's posterior behavior and improve the system model towards real world model up to a small error. On the other hand, {0,1} distribution learning will be misled by unavoidable mistakes made in expert's diagnoses due to the violation of its underlying assumption. Consequently, it could only simulate expert's behavior up to top diagnosis, both posterior ordering and model parameters (probability values) are far out.

- As is argued at the beginning of this chapter, expert's diagnoses are indeed the only available source in many applications. Thus to use {0,1} distribution learning in these domain one must simplify the expert's posterior distribution to a {0,1} distribution. On the other hand, to use ALPP one has to obtain from the expert the overall real value distribution, which may not be practical. A proper compromise might be to ask the expert to provide a few top posterior probabilities and assign the remaining value uniformly to other probabilities.

- ALPP is directly applicable to Bayesian networks of diameter 1 as {0,1} distribution learning would. Although the topology is not feasible for many applications, there are domains the topology is adequate, for example, the Bayesian net version of QMR (its precursor is INTERNIST, one of the first expert systems in internal medicine) [Heckerman 90a, Heckerman 90b] and PAINULIM, among others.

Chapter 7

SUMMARY

The thesis research aims to construct medical expert systems which will be of practical use. The attitude adopted is to identify practical problems in the engineering practice and to solve the problems scientifically (as opposed to ad hoc approaches). The accomplishments of this research include the engineering part and scientific part which are closely related.

Contributions to theoretical knowledge

- The limitation of finite totally ordered probability algebras has been shown theoretically. This highlights the use of infinite totally ordered probability algebras including probability theory.

- The technique of multiply sectioned Bayesian networks (MSBN) and junction forests has been developed. This technique allows the exploitation of localization naturally existing in a large domain such that the construction and deployment of large expert systems using Bayesian networks can be practical.

- An algorithm for learning by posterior probabilities (ALPP) has been developed. This algorithm is useful for sequential updating conditional probabilities in Bayesian networks in order to improve the accuracy of (quantitative) knowledge representation. The algorithm removes the assumption that diagnoses must be confirmed.

Accomplishments in engineering

- WEBWEAVR, an expert system shell has been implemented in IBM compatibles which embeds the MSBN technique.
- PAINULIM, an expert neuromuscular diagnostic system for patients with a painful or impaired upper limb has been constructed. The utilization of the MSBN technique has made possible the knowledge acquisition and system refinement of PAINULIM being conducted within hospital environment. This has resulted in more efficient cooperation with medical experts and greatly speeded up the development of PAINULIM.
- QUALICON, a coupled expert system in technical quality control of nerve conduction studies has been constructed.

Limitations of the work

- The MSBN technique is only used in a single domain. Only the sectioning with a covering sect is used in the application. The applicability of the technique to many other domains requires future experience.
- The performance of ALPP has been shown through simulation but it has not been evaluated through real application.

Topics for future research

- Test design in a multiply sectioned Bayesian network. System prompt for acquisition of evidence most beneficial to a diagnosis given current state of belief has been the topic of expert system research. In chapter 4, the attention shift is described as originated by the user. Under the context of a MSBN, system prompt for acquisition of evidence from neighbor sect can be another reason for attention shift. How to generate the prompt for the target sect for attention shift is an open question. The test design within a sect also deserves further exploration.

- Explanation of inference in a multiply sectioned Bayesian network. Qualitative and intuitive explanation of inference is important for the acceptance of expert systems. The MSBN technique allow the representation of categorical and hierarchical knowledge. How to frame an explanation in terms of evidence distributed in different categories or hierarchies requires future research.

- Only the marginal distribution is obtained in the MSBN technique (in the junction tree technique as well). Is there a way to obtain the most likely combination of hypotheses?

- The representation incorporating decision analysis with Bayesian nets is usually termed influence diagram. Incorporating decision analysis with a MSBN (for, i.e., treatment recommendation) introduces new problems and possibilities.

Bibliography

[Aleliunas 86] R. Aleliunas, "Models of reasoning based on formal deductive probability theories," *Draft unpublished*, 1986.

[Aleliunas 87] R. Aleliunas, "Mathematical models of reasoning - competence models of reasoning about propositions in English and their relationship to the concept of probability," *Research Report CS-87-31, Univ. of Waterloo*, 1987.

[Aleliunas 88] R. Aleliunas, "A new normative theory of probabilistic logic," *Proc. CSCSI-88*, 67-74, 1988.

[Andersen et al. 89] S.K. Andersen, K.G. Olesen, F.V. Jensen and F. Jensen, "HUGIN - a shell for building Bayesian belief universes for expert systems", *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, Detroit, Michigan, Vol. 2, 1080-1085, 1989.

[Andreassen, Andersen and Woldbye 86] S. Andreassen, S.K. Andersen and M. Woldbye, "An expert system for electromyography", *Proc. of Novel Approaches to the Study of Motor Systems*, Banff, Alberta, Canada, 2-3, 1986.

[Andreassen et al. 89] S. Andreassen, F.V. Jensen, S.K. Andersen, B. Falck, U. Kjerulff, M. Woldbye, A.R. Sorensen, A. Rosenfalck and F. Jensen, "MUNIN - an expert EMG assistant", J.E. Desmedt, (Edt), *Computer-Aided Electromyography and Expert Systems*, Elsevier, 255-277, 1989.

[Baker and Boult 90] M. Baker and T.E Boult, "Pruning Bayesian networks for efficient computation", *Proc. of the Sixth Conference on Uncertainty in Artificial Intelligence*, Cambridge, Mass., 257-264, 1990.

[Blinowska and Verroust 87] A. Blinowska and J. Verroust, "Building an expert system in electrodiagnosis of neuromuscular diseases", *Electroenceph. Clin. Neurophysiol.*, 66: S10, 1987.

[Buchanan and Shortliffe 84] B.G. Buchanan and E.H. Shortliffe, (Edt), *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*, Addison-Wesley, 1984.

[Burris and Sankappannvar 81] S. Burris and H. P. Sankappannvar, *A course in universal algebra*, Springer-Verlag, 1981.

[Cheeseman 86] P. Cheeseman, "Probabilistic versus fuzzy reasoning", L.N. Kanal and J.F. Lemmer, (Edt), *Uncertainty in Artificial Intelligence*, Elsevier Science, 85-102, 1986.

[Cheeseman 88a] P. Cheeseman, "An inquiry into computer understanding", *Computational Intelligence*, 4: 58-66, 1988.

[Cheeseman 88b] P. Cheeseman, "In defense of An inquiry into computer understanding", *Computational Intelligence*, 4: 129-142, 1988.

[Cooper 84] G.F. Cooper, *NESTOR: A computer-based medical diagnostic aid that integrates causal and probabilistic knowledge*, Ph.D. diss., Stanford University, 1984.

[Cooper 90] G.F. Cooper, "The computational complexity of probabilistic inference using Bayesian belief networks", *Artificial Intelligence*, 42: 393-405, 1990.

[Desmedt 89] J.E. Desmedt, (Edt), *Computer-Aided Electromyography and Expert Systems*, Elsevier, 1989.

[Duda et al. 76] R.O. Duda, P.E. Hart and N.J. Nilsson, "Subjective Bayesian methods for rule-based inference systems", Technical Report 124, Stanford Research Institute, Menlo Park, California, 1976.

[Fertig and Breese 90] K.W. Fertig and J.S. Breese, "Interval Influence Diagrams", M. Henrion et al., (Edt), *Uncertainty In Artificial Intelligence 5*, 149-161, 1990.

[First et al. 82] M.B. First, B.J. Weimer, S. McLinden and R.A. Miller, "LOCALIZE: Computer assisted localization of peripheral nervous system lesions", *Comput. Biomed. Res.*, 15: 525-543, 1982.

[Fuglsang-Frederiksen and Jeppesen 89] A. Fuglsang-Frederiksen and S.M. Jeppesen, "A rule-based EMG expert system for diagnosing neuromuscular disorders", J.E. Desmedt, (Edt), *Computer-Aided Electromyography and Expert Systems*, Elsevier, 289-296, 1989.

[Gallardo et al. 87] R. Gallardo, M. Gallardo, A. Nodarse, S. Luis, R. Estrada, L. Garcia and O. Padron, "Artificial intelligence in EMG diagnosis of cervical roots and brachial plexus lesions", *Electroenceph. Clin. Neurophysiol.*, 66: S37, 1987.

[Geiger, Verma and Pearl 90] D. Geiger, T. Verma and J. Pearl, "d-separation: from theorems to algorithms. In *Uncertainty in Artificial Intelligence 5*. Edited by M. Henrion, R.D. Shachter, L.N. Kanal and J.F. Lemmer. Elsevier Science Publishers, 139-148, 1990.

[Gibbons 85] A. Gibbons, *Algorithmic Graph Theory*, Cambridge University Press, 1985.

[Golumbic 80] M.C. Golumbic, *Algorithmic graph theory and perfect graphs*, Academic Press, NY, 1980.

[Gotman and Gloor 76] J. Gotman and P. Gloor, "Automatic recognition and quantification of interictal epileptic activity in the human scalp EEG", *Electroenceph. clin. Neurophysiol.*, 41: 513-529, 1976.

[Goodgold and Eberstein 83] J. Goodgold and A. Eberstein, *Electrodiagnosis of Neuromuscular Diseases*, Williams and Wilkins, 1983.

[Halpern and Rabin 87] J.Y. Halpern and M.O. Rabin, "A logic to reason about likelihood," *Artificial Intelligence*, 32: 379-405, 1987.

[Heckerman 86] D. Heckerman, "Probabilistic interpretations for MYCIN's certainty factors", L.N. Kanal and J.F. Lemmer, (Edt), *Uncertainty in Artificial Intelligence*, Elsevier Science, 167-196, 1986.

[Heckerman and Horvitz 87] D.E. Heckerman and E.J. Horvitz, "On the expressiveness of rule-based systems for reasoning with uncertainty", *Proc. AAAI*, Seattle, Washington, 1987.

[Heckerman, Horvitz and Nathwani 89] D.E. Heckerman, E.J. Horvitz and B.N. Nathwani, "Update on the Pathfinder project", *13th Symposium on computer applications in medical care*, Baltimore, Maryland, U.S.A., 1989.

[Heckerman 90a] D. Heckerman, "A tractable inference algorithm for diagnosing multiple diseases", M. Henrion, R.D. Shachter, L.N. Kanal and J.F. Lemmer. (Edt), *Uncertainty in Artificial Intelligence 5*, Elsevier Science Publishers, 163-171, 1990.

[Heckerman 90b] D. Heckerman, *Probabilistic Similarity Networks*, Ph.D. Thesis, Stanford University, 1990.

[Henrion 88] M. Henrion, "Propagating uncertainty in Bayesian networks by probabilistic logic sampling", J. F. Lemmer and L. N. Kanal (Edt), *Uncertainty in Artificial Intelligence 2*, Elsevier Science Publishers, 149-163, 1988.

[Henrion 89] M. Henrion, "Some practical issues in constructing belief networks", L.N. Kanal, T.S. Levitt and J.F. Lemmer (Eds), *Uncertainty in Artificial Intelligence 3*, Elsevier Science Publishers, 161-173, 1989.

[Henrion 90] M. Henrion, "An introduction to algorithms for inference in belief nets", M. Henrion, et al. (Edt), *Uncertainty in Artificial Intelligence 5*, Elsevier Science, 129-138, 1990.

[Howard and Matheson 84] R.A. Howard and J.E. Matheson, "Influence diagrams", *Readings in Decision Analysis*, R.A. Howard and J.E. Matheson (Edt), Strategic Decisions Group, Menlo Park, CA, Ch 38, 763-771, 1984.

[Jamieson 90] P.W. Jamieson, "Computerized interpretation of electromyographic data", *Electroencephalogr Clin Neurophysiol*, 75: 390-400, 1990.

[Jensen 88] F.V. Jensen, "Junction tree and decomposable hypergraphs", *JUDEX Research Report*, Aalborg, Denmark, 1988.

[Jensen, Lauritzen and Olesen 90] , F.V. Jensen, S.L. Lauritzen and K.G. Olesen, "Bayesian updating in causal probabilistic networks by local computations", *Computational Statistics Quarterly.* 4: 269-282, 1990.

[Jensen, Olesen and Andersen 90] , F.V. Jensen, K.G. Olesen and S.K. Andersen, "An algebra of Bayesian belief universes for knowledge based systems", *Networks*, Vol. 20, 637-659, 1990.

[Kim and Pearl 83] J.H. Kim and J. Pearl, "A computational model for causal and diagnostic reasoning in inference engines", *Proc. of 8th IJCAI*, Karlsruhe, West Germany, 190-193, 1983.

[Kowalik 86] J.S. Kowalik, (Edt.), *Coupling Symbolic and Numerical Computing in Expert Systems*, Elsevier, 1986.

[Kuczkowski and Gersting 77] J.E. Kuczkowski and J.L. Gersting, *Abstract Algebra*, Marcel Dekker, 1977.

[Kuipers and Kassier 84] B. Kuipers and J. Kassier, "Causal reasoning in medicine: Analysis of a protocol", *Cognitive Science*, 8: 363-385, 1984.

[Larsen and Marx 81] R.J. Larsen and M.L. Marx, *An Introduction to Mathematical Statistics and Its Applications*, Prentice-Hall, NJ, 1981.

[Lauritzen et al. 84] S.L. Lauritzen, T.P. Speed and K. Vijayan, "Decomposable graphs and hypergraphs", *Journal of Australian Mathematical Society*, Series A, 36: 12-29, 1984.

[Lauritzen and Spiegelhalter 88] S.L. Lauritzen and D.J. Spiegelhalter, "Local computation with probabilities on graphical structures and their application to expert systems", *Journal of the Royal Statistical Society*, Series B, 50: 157-244, 1988.

[Maier 83] D. Maier, *The Theory of Relational Databases*, Computer Science Press, 1983.

[Meyer and Hilfiker 83] M. Meyer and P. Hilfiker, "Computerized motor neurography", J.E. Desmedt, (Edt), *Computer Aided Electromyography*, Karger, 1983.

[Miller et al. 76] A.C. Miller, M.M. Merkhofer, R.A. Howard, J.E. Matheson and T.R. Rice, *Development of Automated Aids for Decision Analysis*, Stanford Research Institute, Menlo Park, California, 1976.

[Neapolitan 90] R.E. Neapolitan, *Probabilistic Reasoning in Expert Systems*, John Wiley and Sons, 1990.

[Ng and Abramson 90] Keung-Chi Ng and B. Abramson, "A sensitivity analysis of PATHFINDER", *Proc. of the Sixth Conference on Uncertainty in Artificial Intelligence*, Cambridge, Mass., 204-212, 1990.

[Nii 86a] H.P. Nii, "Blackboard systems: the blackboard model of problem solving and the evolution of blackboard architectures", *AI Magazine*, Vol. 7, No. 2, 38-53, 1986.

[Nii 86b] H.P. Nii, "Blackboard systems: blackboard application systems, blackboard systems from a knowledge engineering perspective", *AI Magazine*, Vol. 7, No. 3, 82-106, 1986.

[Patil, Szolovits and Schwartz 82] R.S. Patil, P. Szolovits and W.B. Schwartz, "Modeling knowledge of the patient in acid-base and electrolyte disorders", P. Szolovits, (Edt), *Artificial Intelligence in Medicine*, Westview, 191-225, 1982.

[Pearl 86] J. Pearl, "Fusion, propagation, and structuring in belief networks", *Artificial Intelligence*, 29: 241-288, 1986.

[Pearl 88] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, 1988.

[Pearl 89] J. Pearl, "Probabilistic semantics for nonmonotonic reasoning: A survey", *Proc., First intl. conf. on principles of knowledge representation and reasoning*, 1989.

[Poole and Neufeld 88] D. Poole and E. Neufeld, "Sound probabilistic inference in Prolog: an executable specification of influence diagrams," *I SIMPOSIUM INTERNACIONAL DE INTELIGENCIA ARTIFICIAL*, 1988.

[Pople 82] H.E. Pople, Jr., "Heuristic methods for imposing structure on ill-structured problems: the structuring of medical diagnostics", P. Szolovits, (Edt), *Artificial Intelligence in Medicine*, Westview, 119-190, 1982.

[Rose et al. 76] D.J. Rose, R.E. Tarjan and G.S. Lueker, "Algorithmic aspects of vertex elimination on graphs", *SIAM J. Comput.*, 5: 266-283, 1976.

[Savage 61] L.J. Savage, "The foundations of statistics reconsidered", G. Shafer and J. Pearl, (Edt), *Readings in Uncertainty Reasoning*, Morgan Kaufmann, 1990.

[Shafer 76] G. Shafer, *A Mathematical Theory of Evidence*, Princeton University Press, Princeton, New Jersey, 1976.

[Shafer 90] G. Shafer, "Introduction for Chapter 2: The meaning of probability", G. Shafer and J. Pearl, (Edt), *Readings in Uncertainty Reasoning*, Morgan Kaufmann, 1990.

[Shafer and Shenoy 88] G. Shafer and P.P. Shenoy, *Local computation in hypertrees* School of Business Working Paper No. 201, University of Kansas, U.S.A, 1988.

[Shachter 86] R.D. Shachter, "Evaluating influence diagrams", *Operations Research*, 34, No. 6, 871-882, 1986.

[Shachter 88a] R.D. Shachter, "Probabilisitc inference and influence diagrams", *Operations Research*, 36, No. 4, 589-604, 1988.

[Shachter 88] R.D. Shachter, Discussion published in (Lauritzen and Spiegelhalter 88), 1988.

[Shachter and Heckerman 87] R.D. Shachter and D.E. Heckerman, "Thinking backward for knowledge acquisition", *AI Magazine*, 8:55-62, 1987.

[Spiegelhalter 86] D.J. Spiegelhalter, "Probabilistic reasoning in predictive expert systems", L.N. Kanal and J.F. Lemmer, (Edt), *Uncertainty in Artificial Intelligence*, Elsevier Science, 47-67, 1986.

[Spiegelhalter, Franklin and Bull 89] D.J. Spiegelhalter, R.C.G. Franklin, and K. Bull, "Assessment, criticism and improvement of imprecise subjective probabilities for a medical expert system," *Procc Fifth workshop on uncertainty in artificial intelligence*, 335-342, Windsor, Ontario, 1989.

[Spiegelhalter and Lauritzen 90] D.J. Spiegelhalter and S.L. Lauritzen, "Sequential updating of conditioanl probabilities on directed graphical structures", *Networks*, Vol. 20, 5: 579-606, 1990.

[Stalberg and Stalberg 89] E. Stalberg and S. Stalberg, "The use of small computers in the EMG lab", J.E. Desmedt, (Edt), *Computer-Aided Electromyography and Expert Systems*, Elsevier, 1-32, 1989.

[Szolovits and Pauker 78] P. Szolovits and S.G. Pauker, "Categorical and probabilistic reasoning in medical diagnosis", *Artificial Intelligence 11*, 115-144, 1978.

[Szolovits 82] P. Szolovits, "Artificial intelligence and medicine", P. Szolovits, (Edt), *Artificial Intelligence in Medicine*, Westview, 1-19, 1982.

[Tarjan and Yannakakis 84] R.E. Tarjan and M. Yannakakis, "Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs", *SIAM J. Comput.*, 13: 566-579, 1984.

[Tversky and Kahneman 74] A. Tversky and D. Kahneman, "Judgment under uncertainty: heuristics and biases", *Science*, 185: 1124-1131, 1974.

[Tversky and Kahneman 80] A. Tversky and D. Kahneman, "Causal schemata in judgments under uncertainty", M. Fishbein, (Edt), *Progress in Social Psychology*, Lawrence Erlbaum, Vol. 1, 49-72, 1980.

[Vila et al. 85] A. Vila, D. Ziebelin and F. Reymond, "Experimental EMG expert system as an aid in diagnosis", *Electroenceph. Clin. Neurophysiol.*, 61: S240, 1985.

[Xiang, Beddoes and Poole 90a] Y. Xiang, M. P. Beddoes and D. Poole, "Can uncertainty management be realized in a finite totally ordered probability algebra?", M. Henrion et al., (Edt), *Uncertainty In Artificial Intelligence 5*, 41-57, 1990.

[Xiang, Beddoes and Poole 90b] Y. Xiang, M. P. Beddoes and D. Poole, "Sequential updating conditional probability in Bayesian networks by posterior probability", *Proc. 8th Biennial Conference of the Canadian Society for Computational Studies of Intelligence*, Ottawa, 21-27, 1990.

[Xiang et al. 91b] Y. Xiang, B. Pant, A. Eisen, M.P. Beddoes, and D. Poole, "PAINULIM: An expert neuromuscular diagnostic system based on multiply sectioned Bayesian belief networks", *Proc. ISMM International Conference on Mini and Microcomputers in Medicine and Healthcare*, Long Beach, CA, 64-69, 1991.

[Xiang, Poole and Beddoes 91] Y. Xiang, D. Poole and M.P. Beddoes, "Multiply sectioned Bayesian networks and junction forests for large knowledge-based systems", submitted to *Computational Intelligence*, 1991.

[Xiang et al. 92] Y. Xiang, A. Eisen, M. MacNeil and M. P. Beddoes], "Quality control in nerve conduction studies with coupled knowledge based system approach", *Muscle and Nerve*, Vol.15, No.2, 180-187, 1992.

[Zadeh 65] L. Zadeh, "Fuzzy sets", *Information and Control*, 8: 338-353, 1965.

Appendix A

Background on Graph Theory

Graph theory plays an important role in characterizing Bayesian networks and developing efficient inference algorithms for Bayesian networks. In this Appendix, the basic concepts of graph theory relevant to this thesis are introduced. For formal treatment of the graph theoretical concepts introduced, see [Golumbic 80, Gibbons 85, Jensen 88, Lauritzen et al. 84].

A.1 Graphs

A graph G is a pair (N, E) where $N = \{A_1, \ldots, A_\alpha\}$ is a set of nodes and $E = \{(A_i, A_j) | A_i, A_j \in N; i \neq j\}$ is a set of links between pairs of nodes in N. A *directed graph* is a graph where links in E are ordered pairs and an *undirected graph* is a graph where links in E are unordered pairs. Call the links in directed graphs by *arcs* when concerned with their directions. A *subgraph* of a graph (N, E) is any graph (N^k, E^k) satisfying $N^k \subset N$ and $E^k \subset E$. Given a subset of nodes $N^l \subset N$ of a graph (N, E), the subgraph *induced* by N^l is (N^l, E^l) where $E^l = \{(A_i, A_j) \in E | A_i \in N^l \ \& \ A_j \in N^l\}$. The *union graph* of subgraphs $G^1 = (N^1, E^1)$ and $G^2 = (N^2, E^2)$ is the graph $(N^1 \cup N^2, E^1 \cup E^2)$ denoted $G^1 \sqcup G^2$.

Example 22 Figure A.38 shows eight examples. $G^1 = (N^1, E^1)$ is a undirected graph where $N^1 = \{A_1, A_2, \ldots, A_6\}$ and

$$E^1 = \{(A_1, A_2), (A_1, A_4), (A_2, A_3), (A_3, A_4), (A_4, A_5), (A_4, A_6)\}$$

is a set of unordered pairs. $G^4 = (N^4, E^4)$ is a directed graph where $N^1 = N_1$ and $E^4 = \{(A_1, A_2), (A_1, A_4), (A_3, A_2), (A_4, A_3), (A_5, A_4), (A_4, A_6)\}$ is a set of ordered pairs.

G^1 is the union graph of subgraphs G^2 and G^3 ($G^1 = G^2 \sqcup G^3$). Likewise, $G^4 = G^5 \sqcup G^6$. Both G^2 and G^3 are also subgraphs of G^7 but $G^7 \neq G^2 \sqcup G^3$ since the link (A_1, A_5) in G^7 is not contained in either subgraph.

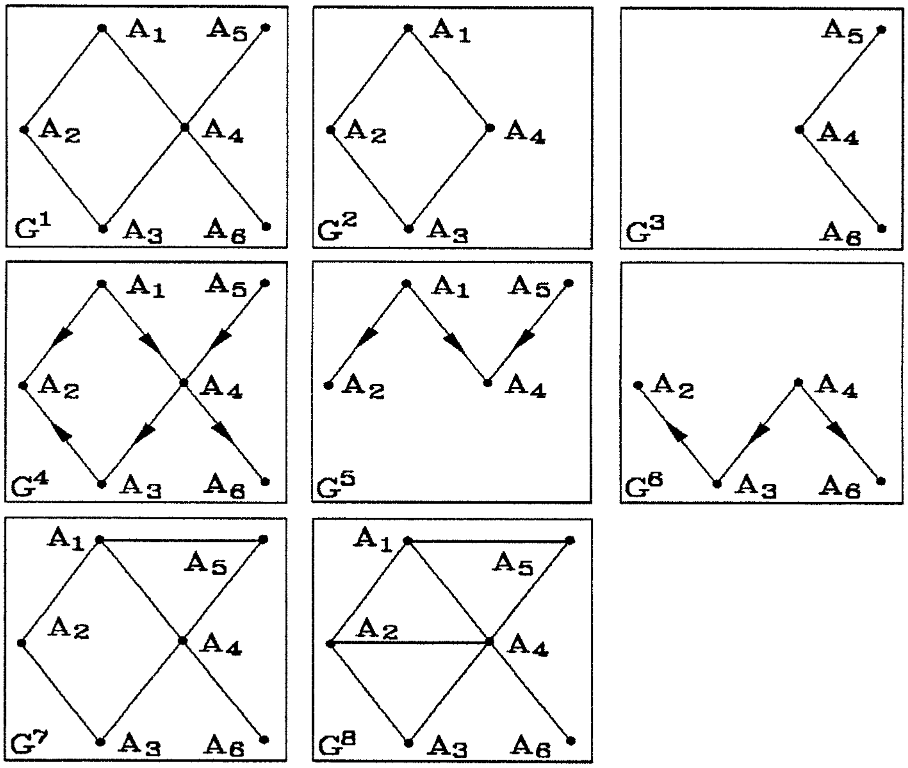


Figure A.38: Examples of graphs

A *path* in graph (N, E) is a sequence of nodes $A_1, A_2, \ldots, A_k$ ($k > 1$) such that $(A_i, A_{i+1}) \in E$. A path in a directed graph can be directed or undirected (i.e., each arc is considered undirected). A *simple path* is a path with no repeated node except that A_1 is allowed to equal A_k. A *cycle* is a simple path with $A_1 = A_k$. Directed graphs without directed cycle are called DAGs (directed acyclic graphs). Define the *diameter* of a DAG as the number of arcs of the longest directed path in the DAG. A graph (N, E) is *connected* if for any pair of nodes in N there is an undirected path between them. A graph is *singly connected* or is a *tree* if there is a unique undirected path between any pairs of nodes. If a graph consists of several unconnected trees, call the graph a *forest*. A graph is *multiply connected* if more than one undirected path exists between a pair of nodes. Note: a graph can be multiply connected but NOT connected!

Example 23 In Figure A.38, $A_1 - A_2 - A_3 - A_4 - A_6$ is a path in G^1. It is also an undirected path in G^4. $A_5 \rightarrow A_4 \rightarrow A_3 \rightarrow A_2$ is a directed path in G^4.

$A_6 - A_4 - A_1 - A_2 - A_3 - A_4 - A_5$ is not a simple path in G^1, but $A_1 - A_2 - A_3 - A_4 - A_6$ is. $A_1 - A_2 - A_3 - A_4 - A_1$ is a undirected cycle in G^4. There is no directed cycle in G^4. There would be one in G^4 if the arc (A_1, A_2) were reversed. Therefore, G^4 is a DAG.

G^3 is a singly connected graph or is a tree. G^5 is a singly connected DAG or is a tree. G^4 is a multiply connected DAG.

The diameters of DAGs G^5, G^6, and G^4 are 1, 2, and 3 respectively.

Only connected DAGs are considered in this thesis since an unconnected DAG can always be treated as several connected ones. Define a *subDAG* of a DAG $D = (N, E)$ as any connected subgraph of D. A DAG D is the *union DAG* of subDAG D^1 and D^2 if $D = D^1 \sqcup D^2$.

Example 24 In Figure A.38, G^4 is the union DAG of subDAGs G^5 and G^6 ($G^4 = G^5 \sqcup G^6$).

If there is an arc (A_1, A_2) from node A_1 to A_2, A_1 is called a *parent* of A_2, and A_2 a *child* of A_1. Similarly, if there is a directed path from A_1 to A_k, the two nodes are called, respectively, *ancestor* and *descendant*, relative to each other. The *in-degree* of a node (denoted by η) is defined as the number of its parents.

Example 25 In G^4 of Figure A.38, A_1 and A_5 are the parents of A_4 and A_4 is their child. A_2 has 4 ancestors namely A_1, A_3, A_4 and A_5. The in-degree of A_4 is 2 and the in-degree of A_1 is 0.

For each node in a DAG, if links are added between all its parents and the directions on arcs are dropped, the graph thus formed is the *moral graph* of the DAG. A graph is *triangulated* if every cycle of length > 3 has a chord. A *chord* is a link connecting 2 non-adjacent nodes. A maximal set of nodes all of which are pairwise linked is called a *clique.* Algorithms for triangulating graphs have been developed, for example, the *Lexicographic search* [Rose et al. 76] with time complexity $\mathcal{O}(ne)$ where n is the number of nodes and e the number of links, and the *maximum cardinality search* [Tarjan and Yannakakis 84] with time complexity $\mathcal{O}(n+e)$.

Example 26 In Figure A.38, G^7 is the moral graph of G_4. G_8 is a triangulated graph of G^7.

A.2 Hypergraphs

A *hypergraph* is a pair $(N, \mathbf{C})$ where N is a set and $\mathbf{C} \subseteq 2^N$ is a set of subsets of N. Define the union of hypergraphs similarly to the union of graphs. The *union hypergraph* of $(N^1, \mathbf{C}^1)$ and $(N^2, \mathbf{C}^2)$ is $(N^1 \cup N^2, \mathbf{C}^1 \cup \mathbf{C}^2)$ denoted $(N^1, \mathbf{C}^1) \sqcup (N^2, \mathbf{C}^2)$. Let (N, E) be a graph, and $\mathbf{C}$ be the set of cliques of (N, E). Then $(N, \mathbf{C})$ is a *clique hypergraph* of graph (N, E).

If a clique hypergraph is organized into a tree where the nodes of the tree are labeled with cliques such that for any pair of cliques, their intersection is contained in each of the cliques on the unique path between them, then the tree is called a *junction tree* or *join tree.* The intersection of 2 adjacent cliques in a junction tree is called the *sepset* of the 2 cliques. Given a hypergraph, its junction trees can be characterized as *maximum-weight* *spanning-tree* if it has one [Jensen 88]. The algorithm for constructing maximum -weight spanning-trees can be found in, e.g., [Gibbons 85].

Example 27 Consider the graph G^8 in Figure A.38 where $N^8 = \{A_1, A_2, \ldots, A_6\}$. The set of cliques of G^8 is $\mathbf{C} = \{\{A_1, A_2, A_4\}, \{A_2, A_3, A_4\}, \{A_1, A_4, A_5\}, \{A_4, A_6\}\}$. Therefore $(N^8, \mathbf{C})$ is a clique hypergraph of G^8. Figure A.39 is a junction tree of the hypergraph $(N^8, \mathbf{C})$, where nodes are labeled with cliques (in ovals) and links are labeled with sepsets of cliques (in squares).

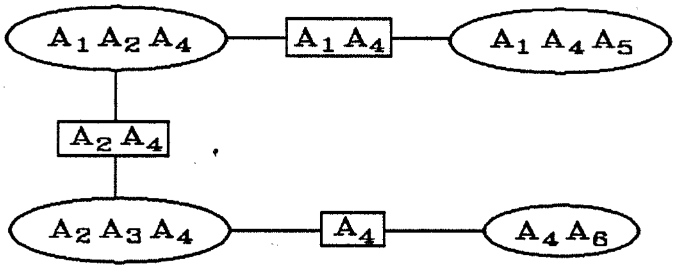


Figure A.39: A junction tree with nodes (ovals) labeled with cliques and links labeled with sepsets of cliques (squares).

Appendix B

Reference for Theory of Probabilistic Logic (TPL)

B.1 Axioms of TPL

The content of this section is taken from Aleliunas [1988] with minor changes to simplify the presentation.

Axiom 1 (TPL axioms) Axioms about the domain and range of each f in $\mathbf{F}$.

AX1 The set of probabilities, $\mathbf{P}$, is a partially ordered set. The ordering relation is '$\leq$'.

AX2 The set of sentences, $\mathbf{L}$, is a free boolean algebra with operations &, $\vee$, *and*, and it is equipped with the usual equivalence relation '$\equiv$'. The generators of the algebra are a countable set of *primitive propositions.* Every sentence in $\mathbf{L}$ is either a primitive proposition or a finite combination of them.

AX3 If $P \equiv X$ and $Q \equiv Y$, then $f(P|Q) = f(X|Y)$.

Axioms that hold for all f in $\mathbf{F}$, and for any P, Q, R in $\mathbf{L}$.

AX4 If Q is absurd (i.e., $Q \equiv R\&\overline{R}$), then $f(P|Q) = f(P|P)$.

AX5 $f(P\&Q|Q) = f(P|Q) \leq f(Q|Q)$.

AX6 For any other g in $\mathbf{F}$, $f(P|P) = g(P|P) = 1$.

AX7 There is a monotone non-increasing total function, i, from $\mathbf{P}$ into $\mathbf{P}$ such that $f(\overline{P}|R) = i(f(P|R))$.

AX8 There is an order-preserving total function, h, from $\mathbf{P} \times \mathbf{P}$ into $\mathbf{P}$ such that $f(P\&Q|R) = h(f(P|Q\&R), f(Q|R))$. Moreover, if $f(P\&Q|R) = 0$, then $f(P|Q\&R)$ $= 0$ or $f(Q|R) = 0$, where $0 = f(\overline{R}|R)$ is defined as a function of f and R.

AX9 If $f(P|R) \leq f(P|\overline{R})$ then $f(P|R) \leq f(P|R \vee \overline{R}) \leq f(P|\overline{R})$.

Axioms about the richness of the set $\mathbf{F}$.

Let $\mathbf{1} = P \vee \overline{P}$. For any distinct primitive propositions A, B and C in $\mathbf{L}$, and for any arbitrary probabilities a, b and c in $\mathbf{P}$, there is a probability assignment f in $\mathbf{F}$ (not necessarily the same one in each case) for which

AX10 $f(A|\mathbf{1}) = a$, $f(B|A) = b$, and $f(C|A\&B) = C$.

AX11 $f(A|B) = f(A|\overline{B}) = a$ and $f(B|A) = f(B|\overline{A}) = b$.

AX12 $f(A|\mathbf{1}) = a$, and $f(A\&B|\mathbf{1}) = b$, whenever $b \leq a$.

B.2 Probability Algebra Theorem

The content of this section is taken from Aleliunas [1986].

Theorem 15 (Probability algebra theorem) *Any probability algebra (under TPL axioms) defined on the partially ordered set (poset)* $\mathbf{P}$ *satisfies the following conditions:*

T1 $\mathbf{P}$ *is a partially ordered semigroup with an order preserving operation* ‘$*$’.

T2 $0 * x = 0$ *and* $1 * x = x$ *for any* x *in* $\mathbf{P}$.

T3 $\mathbf{P}$ *is a self-dual poset equipped with an isomorphism,* i, *from* $\mathbf{P}$ *to its dual.*

T4 $\mathbf{P}$ *is a poset bounded by* 0 *and* 1, *i.e.,* $0 \leq x \leq 1$ *for any* x *in* $\mathbf{P}$.

T5 $\mathbf{P}$ *is commutative, i.e.,* $x * y = y * x$.

T6 $\mathbf{P}$ *has non-trivial zero, i.e.,* $x \leq y$ *implies that* $x = 0$ *or* $y = 0$.

T7 **P** *is a naturally ordered semigroup, i.e.,* $x \leq y$ *implies* $\exists z(x = y * z)$.

Conversely, the above 7 conditions are also sufficient to characterize a probability algebra.

Appendix C

Examples on Finite Totally Ordered Probability Algebras

C.1 Examples of Legal FTOPAs

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
e_1	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
e_2	e_2	e_3	e_4	e_5	e_6	e_7	e_7	e_8
e_3	e_3	e_4	e_5	e_6	e_7	e_7	e_7	e_8
e_4	e_4	e_5	e_6	e_7	e_7	e_7	e_7	e_8
e_5	e_5	e_6	e_7	e_7	e_7	e_7	e_7	e_8
e_6	e_6	e_7	e_7	e_7	e_7	e_7	e_7	e_8
e_7	e_7	e_7	e_7	e_7	e_7	e_7	e_7	e_8
e_8	e_8	e_8	e_8	e_8	e_8	e_8	e_8	e_8

$q\ ^p$	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
e_1	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
e_2		e_1	e_2	e_3	e_4	e_5	$[e_7, e_6]$	e_8
e_3			e_1	e_2	e_3	e_4	$[e_7, e_5]$	e_8
e_4				e_1	e_2	e_3	$[e_7, e_4]$	e_8
e_5					e_1	e_2	$[e_7, e_3]$	e_8
e_6						e_1	$[e_7, e_2]$	e_8
e_7							$[e_7, e_1]$	e_8

Solution table p/q

$M_{8,3}$

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
e_1	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
e_2	e_2	e_2	e_3	e_4	e_5	e_6	e_7	e_8
e_3	e_3	e_3	e_3	e_4	e_5	e_6	e_7	e_8
e_4	e_4	e_4	e_4	e_4	e_5	e_6	e_7	e_8
e_5	e_5	e_5	e_5	e_5	e_5	e_6	e_7	e_8
e_6	e_6	e_6	e_6	e_6	e_6	e_6	e_7	e_8
e_7	e_7	e_7	e_7	e_7	e_7	e_7	e_7	e_8
e_8	e_8	e_8	e_8	e_8	e_8	e_8	e_8	e_8

$q\ ^p$	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
e_1	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
e_2		$[e_2, e_1]$	e_3	e_4	e_5	e_6	e_7	e_8
e_3			$[e_3, e_1]$	e_4	e_5	e_6	e_7	e_8
e_4				$[e_4, e_1]$	e_5	e_6	e_7	e_8
e_5					$[e_5, e_1]$	e_6	e_7	e_8
e_6						$[e_6, e_1]$	e_7	e_8
e_7							$[e_7, e_1]$	e_8

Solution table p/q

$M_{8,8}$

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
e_1	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
e_2	e_2	e_3	e_4	e_4	e_5	e_6	e_7	e_8
e_3	e_3	e_4	e_4	e_4	e_5	e_6	e_7	e_8
e_4	e_4	e_4	e_4	e_4	e_5	e_6	e_7	e_8
e_5	e_5	e_5	e_5	e_5	e_6	e_7	e_7	e_8
e_6	e_6	e_6	e_6	e_6	e_7	e_7	e_7	e_8
e_7	e_7	e_7	e_7	e_7	e_7	e_7	e_7	e_8
e_8	e_8	e_8	e_8	e_8	e_8	e_8	e_8	e_8

q p	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
e_1	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
e_2		e_1	e_2	$[e_4, e_3]$	e_5	e_6	e_7	e_8
e_3			e_1	$[e_4, e_2]$	e_5	e_6	e_7	e_8
e_4				$[e_4, e_1]$	e_5	e_6	e_7	e_8
e_5					$[e_4, e_1]$	e_5	$[e_7, e_6]$	e_8
e_6						$[e_4, e_1]$	$[e_7, e_5]$	e_8
e_7							$[e_7, e_1]$	e_8

Solution table p/q

One of $M_{8,4}$ with idempotent elements e_1, e_4, e_7 and e_8

C.2 Derivation of $p(fire|smoke\&alarm)$ under TPL

$$p(f|s\&a) = p(s|f\&a) * p(f|a)/p(s|a)$$

where

$$\begin{aligned} p(s|f\&a) &= p(s|f); \\ p(s|a) &= p(s\&(f \vee \overline{f})|a) \\ &= i[i[p(s|\overline{f})] * i[p(s|f) * p(f|a)/i[p(s|\overline{f}) * p(\overline{f}|a)]]]; \end{aligned}$$

and

$$p(f|a) = p(a|f) * p(f)/p(a)$$

where

$$\begin{aligned} p(a|f) &= p(a\&((f\&t) \vee (f\&\overline{t}) \vee (\overline{f}\&t) \vee (\overline{f}\&\overline{t}))|f) \\ &= i[i[p(a|f\&\overline{t}) * p(\overline{t})] * i[p(a|f\&t) * p(t)/i[p(a|f\&\overline{t}) * p(\overline{t})]]] \end{aligned}$$

and

$$p(a) = p(a\&((f\&t) \vee (f\&\overline{t}) \vee (\overline{f}\&t) \vee (\overline{f}\&\overline{t})))$$

$$= i[f_1 * f_2 * f_3 * f_4]$$

where

$$\begin{aligned} f_1 &= i[p(a|\overline{f}\&\overline{t}) * p(\overline{f}) * p(\overline{t})] \\ f_2 &= i[p(a|\overline{f}\&t) * p(\overline{f}) * p(t)/f_1] \\ f_3 &= i[p(a|f\&\overline{t}) * p(f) * p(\overline{t})/(f_1 * f_2)] \\ f_4 &= i[p(a|f\&t) * p(f) * p(t)/(f_1 * f_2 * f_3)]. \end{aligned}$$

C.3 Evaluation of Legal FTOPAs with Size 8

Figure C.40 plots the evaluation results of the following 14 posterior probabilities from smoke-alarm example by Poole and Neufeld [1988] using all the 32 legal FTOPAs of size 8. The conventional probability model (M0) is included for comparison.

$$p(s|f), p(a|f), p(l|f), p(r|f); p(s|t), p(a|t), p(l|t), p(r|t); p(f|s), p(f|a), p(f|s\&a); p(t|s), p(t|a), p(t|s\&a)$$

The following table lists model labels (L) used in Figure C.40 and their corresponding idempotent element subscripts (S). 'M0' labels the conventional probability model.

L	S	L	S	L	S	L	S
$M1$	$1,7,8$	$M9$	$1,2,5,7,8$	$M17$	$1,2,3,4,7,8$	$M25$	$1,3,5,6,7,8$
$M2$	$1,2,7,8$	$M10$	$1,2,6,7,8$	$M18$	$1,2,3,5,7,8$	$M26$	$1,4,5,6,7,8$
$M3$	$1,3,7,8$	$M11$	$1,3,4,7,8$	$M19$	$1,2,3,6,7,8$	$M27$	$1,2,3,4,5,7,8$
$M4$	$1,4,7,8$	$M12$	$1,3,5,7,8$	$M20$	$1,2,4,5,7,8$	$M28$	$1,2,3,4,6,7,8$
$M5$	$1,5,7,8$	$M13$	$1,3,6,7,8$	$M21$	$1,2,4,6,7,8$	$M29$	$1,2,3,5,6,7,8$
$M6$	$1,6,7,8$	$M14$	$1,4,5,7,8$	$M22$	$1,2,5,6,7,8$	$M30$	$1,2,4,5,6,7,8$
$M7$	$1,2,3,7,8$	$M15$	$1,4,6,7,8$	$M23$	$1,3,4,5,7,8$	$M31$	$1,3,4,5,6,7,8$
$M8$	$1,2,4,7,8$	$M16$	$1,5,6,7,8$	$M24$	$1,3,4,6,7,8$	$M32$	$1,2,3,4,5,6,7,8$

In each sub-graph, the vertical scale represents the probability *range* with $0 = e_8$ and $1 = e_1$. And horizontally arranged are the probabilities with the same order above. The predictive probabilities with identical condition are shown in the first 8 positions in 2 groups. The diagnostic probabilities with a fixed hypothesis are shown in the last 6 positions in 2 groups. As is analyzed in Section 2.5, none of the models through 1 to 32 gives satisfactory results.

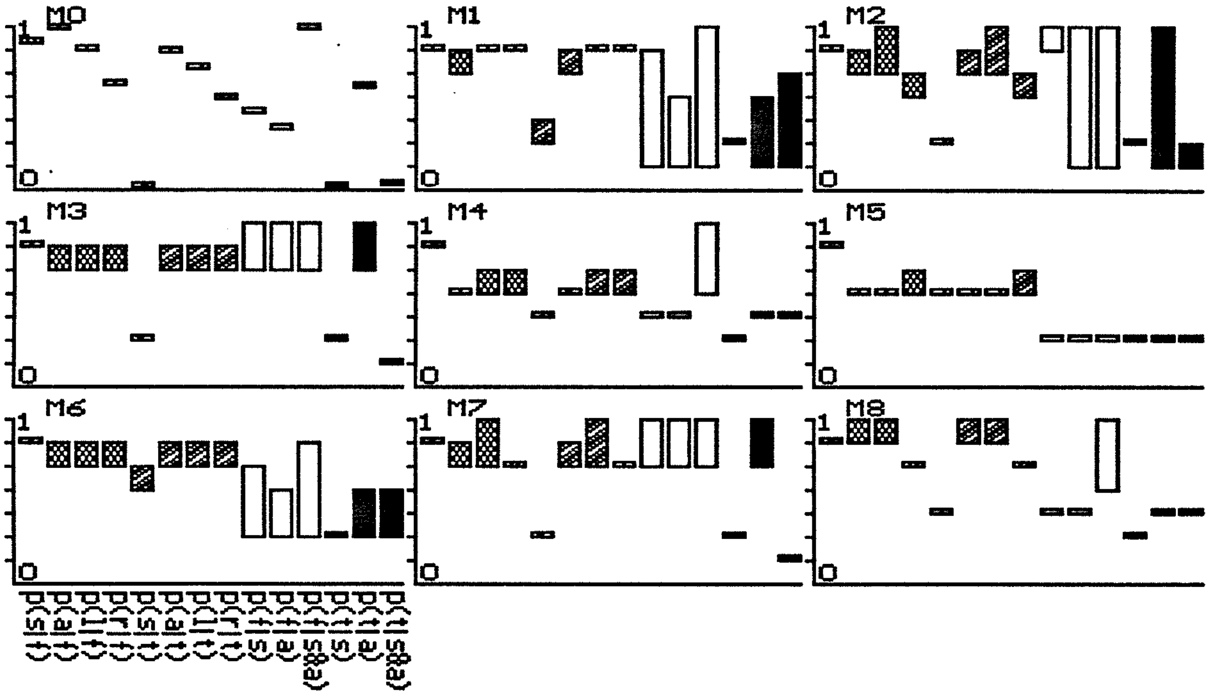


Figure C.40: Evaluation result of smoke-alarm example by 32 legal FTOPAs of size 8
f: fire; t: tampering; a: alarm; s: smoke; l: leaving; r: report.

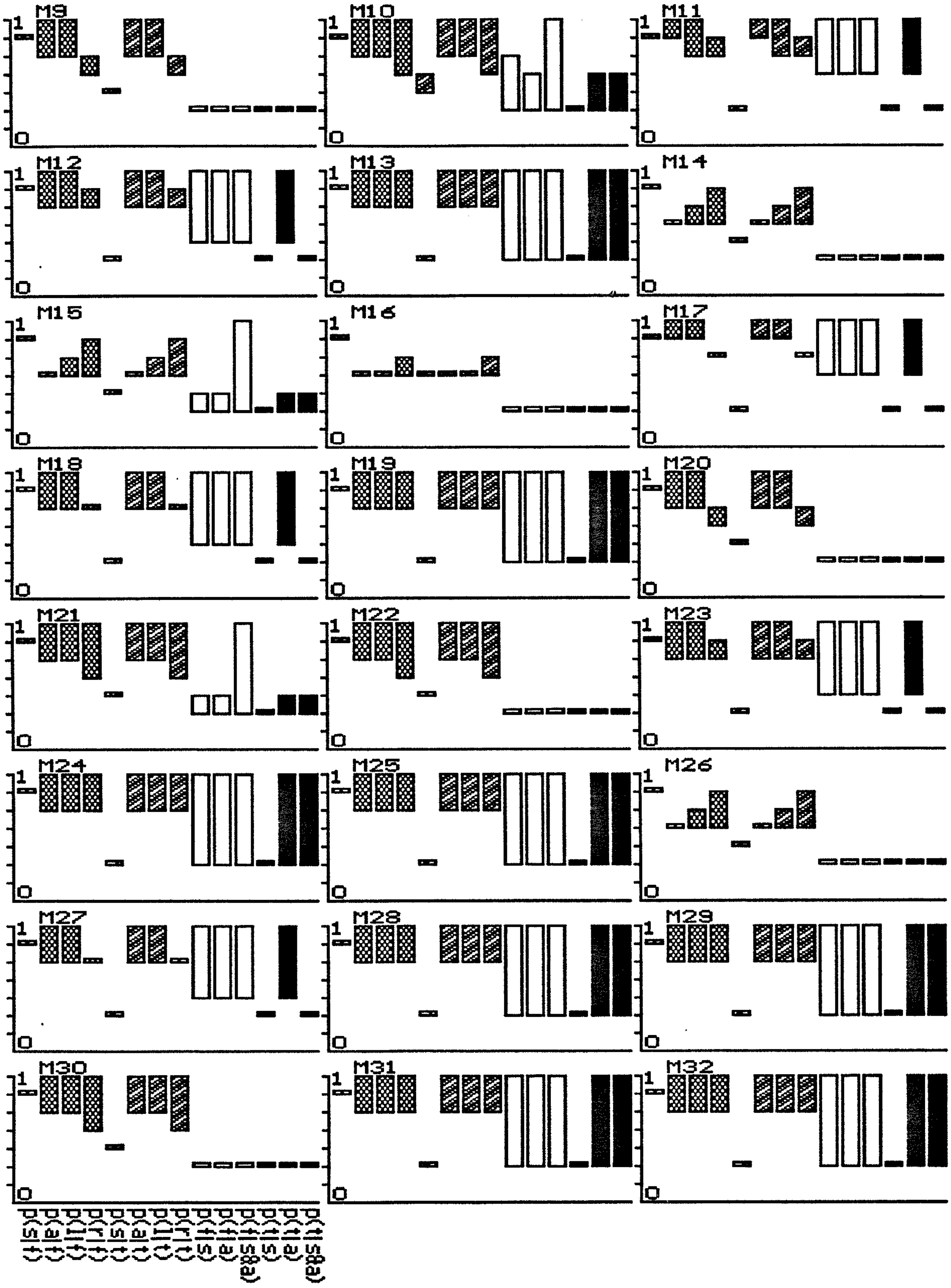


Figure C.40 (cont)
f: fire; t: tampering; a: alarm; s: smoke; l: leaving; r: report.

Appendix D

Proofs for corollary and propositions

Proof of Corollary 2

Proof:

To prove $p/q = r$, it suffices to show $r * q = p$. Below the 7 cases are shown in the order of their appearance in the Corollary.

(Case 1) Equivalent to $e_k * e_1 = e_k$ which is in turn equivalent to Cond8 of proposition 1.

(Case 2) Equivalent to $e_n * e_j = e_n$ which is in turn equivalent to Cond4 of proposition 1.

(Case 3) Equivalent to $e_x * e_y = e_k$ where $e_x = e_{k-j+i_l}$ and $e_{i_{l+1}-2} \leq e_y \leq e_{i_{l+1}}$. First, show this is the second case of Theorem 1. Clearly, $i_l < y < i_{l+1} - 1 \leq i_{l+1}$. Also $x = k - j + i_l > i_l$, and $k - j + i_l \leq i_{l+1} - 1 - (i_l + 1) + i_l = i_{l+1} - 2 \leq i_{l+1}$.

Second, show $e_{min(x+j-i_l,i_{l+1})} = e_k$. This is true because $x - j - i_l = k < i_{l+1}$.

(Case 4) Equivalent to $e_x * e_k = e_k$ where $x \leq= i_l$ and $i_l < k < i_{l+1}$. It is true by the third case of Theorem 1.

(Case 5) Equivalent to $e_x * e_y = e_{i_{l+1}}$ where $e_{i_{l+1}} \leq e_x \leq e_{i_{l+1}+i_l-j}$ and $e_{i_{l+1}-1} \leq e_y \leq e_{i_l+1}$. First, show this is the second case of Theorem 1, that is, $e_x, e_y \in [e_{i_{l+1}}, e_{i_l+1}]$. This is obviously true for e_y and the lower bound of e_x. For the upper bound, from $i_{l+1} > j$, one has $i_{l+1} - j \geq 1$, and therefore $e_{i_{l+1}+i_l-j} \leq e_{i_l+1}$.

Second, show $e_{min(x+y-i_l,i_{l+1})} = e_{i_{l+1}}$ which is equivalent to $x + y - i_l \geq i_{l+1}$. Since the lower bound is $i_l + 1$ for x, and i_{l+1} for y, this is certainly true.

(Case 6) Equivalent to $e_x * e_{i_l} = e_{i_l}$ where $e_{i_l} \leq e_x \leq e_1$ which is true by Lemma 3.

(Case 7) Covered by the third case of Theorem 1.

□

Proof of Proposition 5

Proof:

A solution table (see appendix C.1) can be viewed as a different layout of a product table. An entry in the column labeled 'q' is one factor. A table entry in the same line as the first factor is another factor. The entry in the line labeled 'p' and in the same column as the second factor is the product.

Since product operation '*' is well defined, all the probabilities will appear in each line of the table entries (possibly several of them appear together in a range). Therefore, there are $n(n-1)$ table entries. There are $0.5n(n+1)-1$ distinct solution pairs. Their difference is just the amount of ambiguity by definition.

$$A = [n(n-1)] - [0.5n(n+1)-1] = (n-1)(n-2)/2$$

□

Proof of Proposition 6

Proof:

(O_d) From Corollary 2, all the incidences of $e_x/e_y = e_x$ to be counted are covered by the case 7 of the corollary. Among $\{e_2, \ldots, e_n\}$, there are $k-2$ idempotent elements which determine $k-3$ intervals. For each interval, the case 7 dictates $i_{m+1} - i_m$ columns in the solution table with $i_m - 1$ entries in each column to be counted.

(O_m) From Theorem 1, $e_x * e_y < min(e_x, e_y)$ happens only in the second case. Since idempotent elements do not follow the relation by Lemma 3, only $i_m < x, y < i_{m+1}$ needs to be considered, and there are $k-2$ such intervals ((i_{k-1}, i_k) does not contribute to O_m). For each interval, if x is fixed, y goes through $i_{m+1} - i_m - 1$ values. Since only distinct product pair is to be counted, each interval contributes $\sum_{j=1}^{i_{m+1}-i_m-1} j$.

Now it suffices to show $min(x+y-i_m, i_{m+1}) > x, y$. Trivially, $i_{m+1} > x, y$. Also $x + (y - i_m) \geq x + 1 > x$. Similarly $x + y - i_m > y$.

$(O_d + O_m)$ Rewrite O_d as

$$O_d = \sum_{m=2}^{k-2} \sum_{j=0}^{i_{m+1}-i_m-1} (i_{m+1} - i_m)$$

Thus

$$O_d + O_m = \sum_{j=1}^{i_2-2} j + \sum_{j=0}^{i_3-i_2-1} (i_2 + j - 1) + \sum_{j=0}^{i_4-i_3-1} (i_3 + j - 1) + \ldots \sum_{j=0}^{i_{k-1}-i_{k-2}-1} (i_{k-2} + j - 1)$$

Note the last addend of each sum is 1 less than the 1st addend of the next sum; and there are $i_{k-1} - 2$ addends in total. Thus,

$$O_d + O_m = \sum_{j=1}^{i_{k-1}-2} j = \sum_{j=1}^{n-3} j = (n-3)(n-2)/2$$

□

Appendix E

Reference for Statistic Evaluation of Bernoulli Trials

The content of this appendix is taken from Larsen and Marx [1981].

Any set of repeated independent trials with each trial having just 2 possible outcomes (*success* and *failure*) are called *Bernoulli trials.* The probability p associated with success is called *success probability.*

E.1 Estimation for Confidence Intervals

Definition 18 Suppose that y successes are observed in n independent Bernoulli trials. The probability interval $[p_1, p_2]$ is the $100(1-\alpha)\%$ *confidence interval* for success probability p if

$$P(p_1 < p < p_2 | Y = y) = 1 - \alpha$$

where Y is the variable for number of successes.

The above defined lower and upper confidence limits p_1 and p_2 for p are the solutions of

$$\sum_{j=y}^{n} C_n^j p_1^j (1-p_1)^{n-j} = \frac{\alpha}{2}$$

$$\sum_{j=0}^{y} C_n^j p_2^j (1-p_2)^{n-j} = \frac{\alpha}{2}$$

where

$$C_n^j = \frac{n!}{j!(n-j)!}$$

is the number of combinations taking j elements out of n.

E.2 Hypothesis Testing

Let x and y denote the numbers of successes observed in 2 independent sets of n and m Bernoulli trials, respectively. Let p_X and p_Y denote the true success probabilities associated with each set of trials. An approximate generalized likelihood ratio test at the α level of significance for hypothesis $H_0 : p_X = p_Y$ versus hypothesis $H_1 : p_X \neq p_Y$ is gotten by rejecting H_0 whenever

$$\frac{\frac{x}{n} - \frac{y}{m}}{\sqrt{\frac{(\frac{x+y}{n+m})(1-\frac{x+y}{n+m})(n+m)}{nm}}}$$

is either $\leq -z_{\alpha/2}$ or $\geq +z_{\alpha/2}$ where

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{-z_{\alpha/2}} e^{-x^2/2} dx = \alpha/2.$$

Appendix F

Glossary of PAINULIM Terms

F.1 Diseases Considered in PAINULIM

Ahcd : Anterior horn cell disease

Als : Amyotrophic Lateral Sclerosis

Cts : Carpal tunnel syndrome

Inspcd : Intrinsic cord disease

Mednn : Median nerve lesion

Mnd : Motor neuron diseases (including Ahcd and Als)

Pd : Parkinsons disease

Pxltk : Plexus lower trunk

Pxpcd : Plexus post cord

Pxutk : Plexus upper trunk

Radnn : Radial nerve lesion

Rc56 : C56 Root disease

Rc67 : C67 Root disease

Rc81 : C8T1 Root disease

Ulrnn : Ulnar nerve lesion

F.2 Clinical Features in PAINULIM

bcpex : Biceps reflex exaggerated

bcpls : Biceps reflex lost/diminished

lsbkhdfrm : Loss of sensation in back of hand/forearm

lsdisoc : Dissociated sensory loss

lslatfrm : Loss of sensation in lateral forearm

lslathnd : Loss of sensation in lateral hand/fingers

lsmedfrm : Loss of sensation in medial forearm

lsmedhnd : Loss of sensation in medial hand/fingers

lsuparm : Loss of sensation in upper arm

radex : Radial/Supinator reflex exaggerated

radls : Radial/Supinator reflex lost/diminished

rad_pn : Radicular pain

rigid : Rigidity

pn_frm : Pain or tingling in the forearm

pn_hnd : Pain or tingling in the hand

pn_shd : Pain or tingling in the shoulder

spstc : Spasticity

tcpex : Triceps reflex exaggerated

tcpls : Triceps reflex lost/diminished

tremor : Tremor of limbs/hands

twitch : Muscle twitch or fasciculations

wk_arm : Weakness of the arm

wk_fng_fx : Weakness of finger flexion

wk_shld : Weakness of the shoulder

wk_th_abd : Weakness of thumb Abduction

wk_th_ext : Weakness of thumb extension

wk_th_fx : Weakness of thumb flexion

wk_wst_ex : Weakness of wrist extension

F.3 EMG Features in PAINULIM

adm : Abductor digiti minimi

apb : Abductor pollicis brevis

bchrd : Brachioradialis

bcps : Biceps brachii

edc : Extensor digitorum communis

deltd : Deltoid

fasc : Fasciculation

fcu : Flexor carpi ulnaris

fdi : First dorsal interosseous

fdp23 : Flexor digitorum profundus 2,3

fdp45 : Flexor digitorum profundus 4,5

fpl : Flexor pollicis longus

latdrs : Lattismus dorsi

lvscp : Levator scapulae

other : Muscle in other limb/trunk

pmcv : Pectoralis major clavicular

pmsc : Pectoralis major sterno-clavicular

prt : Pronator teres

ps56 : Paraspinals C5,6

ps67 : Paraspinals C6,7

ps81 : Paraspinals C8,T1

rhmj : Rhomboideus major

spspin : Supraspinatus

srant : Serratus anterior

trcps : Triceps

F.4 Nerve Conduction Features in PAINULIM

mcmp : MEDIAN COMPOUND MUSCLE ACTION POTENTIAL

mf2l : MEDIAN finger2 SNAP latency

mf2a : MEDIAN finger2 SNAP amplitude

mmcb : MEDIAN motor CONDUCTION BLOCK

mmcv : MEDIAN motor CONDUCTION VELOCITY

mmdl : MEDIAN motor DISTAL LATENCY

mmfw : Median "F" response

mupl : MEDIAN to ULNAR palmar latency difference

radm : Radial motor study

rads : RADIAL sensory study

ucmp : ULNAR COMPOUND MUSCLE ACTION POTENTIAL

uf5a : ULNAR finger5 SNAP amplitude

uf5l : ULNAR finger5 SNAP latency

umcb : ULNAR motor CONDUCTION BLOCK

umcv : ULNAR motor CONDUCTION VELOCITY

umfw : Ulnar "F" response

Appendix G

Acronyms

AI : Artificial Intelligence

ALPP : Algorithm of Learning by Posterior Probability

BNS : Bayesian Network Specification (in QUALICON)

CLINICAL : clinical sect of PAINULIM

CMAP : Compound Muscle Action Potential

CP : Conditional Probability

DAG : Directed Acyclic Graph

DOCTR : DOCToR - a module in WEBWEAVR

EDITOR : a module in WEBWEAVR

EEG : ElectroEncephaloGraphy

EMG : ElectroMyoGraphy

EMGer : ElectroMyoGrapher

FE : Feature Extraction (in QUALICON)

FP : Feature Partition (in QUALICON)

FTOPA : Finite Totally Ordered Probability Algebra

MSBN : Multiply Sectioned Bayesian Network

NCV : Nerve Conduction Velocity

ND : Normal Data (in QUALICON)

NDU : Neuromuscular Diseases Unit

PAINULIM : PAINful or impaired Upper LIMb

PIE : Probabilistic Inference Engine (in QUALICON)

PP : Posterior Probability

QUALICON : QUALIty CONtrol

SNAP : Sensory Nerve Action Potential

TPL : Theory of Probabilistic Logic

TRANSNET : TRANSform NETwork - a module in WEBWEAVR

UBC : University of British Columbia

USBN : UnSectioned Bayesian Network

VGH : Vancouver General Hospital

WEBWEAVR : WEB WEAVeR