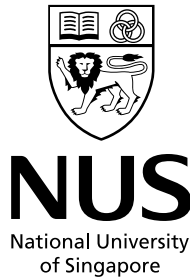


Automated Combination of Probabilistic Graphic Models from Multiple Knowledge Sources

JIANG Changan

A Thesis presented for the degree of
Master of Science



Supervisors:

Associate Professor Leong Tze Yun

Associate Professor Poh Kim Leng

Department of Computer Science

National University of Singapore

December 2004

Dedicated to

All People who have supported me in my life and study.

My Grandmother, Madam *Chen Guirong*

Automated Combination of Probabilistic Graphic Models from Multiple Knowledge Sources

JIANG CHANGAN

Submitted for the degree of Master of Science

December 2004

Abstract

It is a frequently encountered problem that new knowledge arrives when making decisions in a dynamic world. Bayesian networks and influence diagrams, two major probabilistic graph models, are powerful representation and reasoning tools for complex decision problems. Usually, domain experts cannot afford enough time and knowledge to effectively assess and combine both qualitative and quantitative information in these models. Existing approaches can solve only one of the two tasks instead of both. Based on an extensive literature survey, we propose a four-step algorithm to integrate multiple probabilistic graphic models, which can effectively update existing models with newly acquired models. In this algorithm, the qualitative part of model integration is performed first, followed by quantitative combination. We illustrate our method with a comprehensive example in a real domain. We also identify some factors that may influence the complexity of the integrated model. Accordingly, we present three heuristic methods of target variable ordering generation. Such methods show their feasibility through our experiments and are good in different situations. Furthermore, we discuss influence diagram combination and present a utility-based method to combine probability distributions. Finally, we provide some comments based on our experiments results.

Keywords:

Probabilistic graphic model, Bayesian network, Influence diagram, Qualitative combination, Quantitative combination

Declaration

The work in this thesis is based on research carried out at the Medical Computing Lab, School of Computing, NUS, Singapore. No part of this thesis has been submitted elsewhere for any other degree or qualification and it all my own work unless referenced to the contrary in the text.

Copyright © 2004 by JIANG CHANGAN.

“The copyright of this thesis rests with the author. No quotations from it should be published without the author’s prior written consent and information derived from it should be acknowledged”.

Acknowledgements

The thesis is the summary of the work during my study for Master degree in National University of Singapore. The effort of writing it cannot be separated from support from other people. I would like to express my gratitude to these people for their kindness.

Associate Professor *Leong Tze Yun* and Associate Professor *Poh Kim Leng*, my two nice supervisors, for their constructive suggestions and patience to show me research direction. As advisors, they demonstrate knowledge and encouragement that I need as a research student. They go through my work seriously word by word and provide necessary research training to us. As the director of Medical Computing Lab, Prof. Leong provides me a conducive environment to work on my research. The weekly research activities in the Biomedical Decision Engineering (BiDE) group enriched my knowledge on different research topics. I would like to express my sincere gratitude to them for their continuous guidance, insightful ideas, constant encouragement and rigorous research styles that underlie the accomplishment of this thesis. Their enthusiasm and kindness will forever be remembered.

Professor *Peter Haddawy* from Asian Institute of Technology (Thailand) and Professor *Marek Druzdzal* from University of Pittsburgh (USA), for their valuable advice and comments on my research work, during their visiting to Medical Computing Lab.

Zeng Yifeng, Li Guoliang, Rohit Joshi and *Han Bin*, four creative members in our BiDE research group, for taking their time to discuss with me, and kindly help review my thesis.

Other members in BiDE group. Their friendships make my study and life in the National University of Singapore (NUS) fruitful and enjoyable.

Jiang Liubin and *Li Zhao*, for all the care, concern, and encouragement given by them while I worked and wrote this thesis.

NUS, for the grant of research scholarship and the Department of Computer Science for the use of its facilities, without any of which I would not be able to carry out the research reported in the thesis.

Last but maybe the most important, *my parents*, for all support and sacrifice that they have given me to pursue my interest in research. Without them all these work is impossible.

Contents

| | |
|---|------------|
| Abstract | iii |
| Declaration | iv |
| Acknowledgements | v |
| 1 Introduction | 1 |
| 1.1 Background | 1 |
| 1.1.1 Bayesian Networks | 2 |
| 1.1.2 Influence Diagram | 4 |
| 1.1.3 Knowledge Sources of Probabilistic Graphic Models | 5 |
| 1.1.3.1 Experts | 6 |
| 1.1.3.2 Literature | 7 |
| 1.1.3.3 Data Set | 7 |
| 1.1.3.4 Knowledge Base | 7 |
| 1.2 Motivations | 8 |
| 1.3 Objectives | 12 |
| 1.4 Research Approach | 13 |
| 1.5 Application Domains | 14 |
| 1.6 Organization of Thesis | 14 |
| 2 Related Concepts and Technologies | 16 |
| 2.1 Structure Combination | 16 |
| 2.1.1 Multi-entity Bayesian Networks | 16 |
| 2.1.2 Multiply Sectioned Bayesian Networks | 17 |

| | | |
|----------|---|-----------|
| 2.1.3 | Topology Fusion of Bayesian Networks | 19 |
| 2.1.4 | Graphical Representation of Consensus Belief | 20 |
| 2.2 | Probability Distribution Combination | 21 |
| 2.2.1 | Behavior Approaches | 21 |
| 2.2.2 | Weighted Approaches | 22 |
| 2.2.3 | Bayesian Combination Methods | 23 |
| 2.2.4 | Interval Combination | 24 |
| 3 | Problem Analysis | 25 |
| 3.1 | Problem Formulation | 25 |
| 3.2 | Precondition of Probabilistic Graphic Combination | 26 |
| 3.2.1 | Variable Consistency | 27 |
| 3.2.2 | Model Consistency | 28 |
| 3.3 | Challenges | 28 |
| 4 | Probabilistic Graphic Model Combination | 32 |
| 4.1 | Structure Combination of Bayesian Networks | 32 |
| 4.1.1 | Re-organize Bayesian Networks | 33 |
| 4.1.2 | Adjust Variable Ordering to Maintain DAG | 35 |
| 4.1.2.1 | Order Value Computation for Variables | 35 |
| 4.1.2.2 | Two Types of Variable Ordering | 37 |
| 4.1.2.3 | Arc Reversal to Adjust Variable Ordering | 40 |
| 4.1.3 | Intermediate Bayesian Networks | 44 |
| 4.2 | Quantitative Combination of Bayesian Networks | 45 |
| 4.2.1 | CPT Computation in Arc Reversal | 46 |
| 4.2.2 | CPT Combination | 48 |
| 4.2.2.1 | Average or Weighted Combination | 48 |
| 4.2.2.2 | Interval Bayesian Networks | 51 |
| 4.3 | Heuristic Methods for Target Variable Ordering Generation | 52 |
| 4.3.1 | Target Ordering based on Original Order Values | 53 |
| 4.3.2 | Target Ordering based on Number of Parents and Network Size | 56 |
| 4.3.3 | Target Ordering based on Edge Matrix | 57 |

| | | |
|----------|---|------------|
| 4.4 | Extension to Influence Diagram Combination | 60 |
| 4.4.1 | Three Types of Nodes in Influence Diagram | 61 |
| 4.4.2 | Four Types of Arcs in Influence Diagram | 62 |
| 4.4.3 | Qualitative Combination with Constraints | 64 |
| 4.4.4 | Quantitative Combination | 65 |
| 4.4.4.1 | Utility based Parameter Combination | 67 |
| 4.5 | Implementation | 69 |
| 4.6 | Complexity Analysis | 70 |
| 5 | Case Study based Evaluation | 73 |
| 5.1 | Experimental Results on Bayesian Network Combination | 73 |
| 5.1.1 | Introduction to Heart Disease Models | 73 |
| 5.1.2 | Experimental Setting and Measurement Criteria | 75 |
| 5.1.3 | Comparison of Three Target Orderings Generation Methods | 76 |
| 5.1.4 | Comparison of Different Size Bayesian Network Combination | 85 |
| 5.2 | Experimental Results on Utility based Parameter Combination | 86 |
| 5.2.1 | Experiment Setting | 87 |
| 5.2.2 | Comparison of Weights of All Sources under 3 Methods | 91 |
| 5.2.3 | Comparison of Arithmetic Combined Probability Distribution | 94 |
| 5.2.4 | Comparison of Geometric Combined Probability Distribution | 96 |
| 5.2.5 | Comparison of Two Approaches of Combination | 96 |
| 5.2.6 | Result of Adding one more Knowledge Source | 97 |
| 6 | Conclusion and Future Work | 101 |
| 6.1 | Summary | 101 |
| 6.1.1 | Advantages | 103 |
| 6.1.2 | Limitation | 103 |
| 6.1.3 | Discussion | 104 |
| 6.2 | Future Work | 104 |
| | Bibliography | 106 |
| | Appendix | 113 |

| | |
|--|------------|
| A Glossary | 113 |
| B List of Notation | 114 |
| C Experimental Data | 115 |
| C.1 The Heart Disease Bayesian Network Models | 115 |
| C.2 Probability Distributions from Different Knowledge Sources | 128 |

List of Figures

| | | |
|------|---|----|
| 1.1 | An example Bayesian network | 2 |
| 1.2 | An example influence diagram | 4 |
| 1.3 | Knowledge combination from different sources | 6 |
| 1.4 | An example of knowledge combination in medical domain | 9 |
| 2.1 | An example multiply sectioned Bayesian networks | 18 |
| 2.2 | The cluster tree for computing e-message | 19 |
| 3.1 | Probabilistic graphic models combination | 26 |
| 3.2 | Two simple BNs to be combined | 27 |
| 3.3 | Improper Bayesian network modeling can result in problems | 28 |
| 3.4 | Direct conflict in DAG combination | 30 |
| 3.5 | Indirect conflict in DAG combination | 30 |
| 3.6 | CPT disagreement in two models | 31 |
| 4.1 | Example of three candidate Bayesian networks | 39 |
| 4.2 | Example of ordering hierarchy of nodes in a BN | 40 |
| 4.3 | General structure of arc reversal | 42 |
| 4.4 | Reconstruction resulting of candidate BN using arc reversal | 44 |
| 4.5 | Example of virtual nodes | 45 |
| 4.6 | Example of virtual arcs | 45 |
| 4.7 | Example of arc reversal | 46 |
| 4.8 | Intermediate Bayesian network 1 | 48 |
| 4.9 | Intermediate Bayesian network 2 | 49 |
| 4.10 | Intermediate Bayesian network 3 | 49 |

| | | |
|------|---|-----|
| 4.11 | Resulting Bayesian network with weighted combination | 50 |
| 4.12 | Example of resulting Interval Bayesian network | 52 |
| 4.13 | Example of resulting Bayesian networks according to order value based target variable ordering | 56 |
| 4.14 | Example of resulting Bayesian network according to num of parents and network size | 57 |
| 4.15 | Target variable ordering in the resulting edge matrix | 60 |
| 4.16 | Resulting BN according to edge matrix based target ordering | 60 |
| 4.17 | Various types of nodes in influence diagram | 61 |
| 4.18 | Four types of arcs | 63 |
| 4.19 | Utility based parameter combination method | 67 |
| 4.20 | System overview | 70 |
| 4.21 | Utility-based weighted parameter combination | 71 |
| 5.1 | Comparison of probability distributions from 5 knowledge sources | 92 |
| 5.2 | Comparison of expected utilities from probability distributions from 5 knowledge sources | 92 |
| 5.3 | Comparison of weights of all experts in three methods | 93 |
| 5.4 | Comparison of arithmetically combined opinions with three kinds of weights | 95 |
| 5.5 | Geometric combinationed expert opinions with three kinds of weights | 96 |
| 5.6 | Comparison of two combination approaches | 97 |
| 5.7 | Weights of the 6 experts | 97 |
| 5.8 | The 6 expert opinions | 98 |
| 5.9 | Combination result of the 6 expert opinions | 99 |
| 5.10 | 5 experts vs 6 experts.. | 100 |
| C.1 | Three 5-node candidate Bayesian networks | 116 |
| C.2 | Resulting Bayesian networks with 3 methods in combination of three 5-node CBN | 117 |
| C.3 | Resulting BN with a random target variable ordering in combination of three 5-node CBN | 118 |

| | | |
|------|---|-----|
| C.4 | Three 6-node candidate Bayesian networks | 118 |
| C.5 | Resulting Bayesian networks with 3 methods in combination of three 6-node CBN | 119 |
| C.6 | Resulting BN with a random target variable ordering in combination of three 6-node CBN | 120 |
| C.7 | Three 7-node candidate Bayesian networks | 120 |
| C.8 | Resulting Bayesian networks with 3 methods in combining of three 7-node CBN | 123 |
| C.9 | Resulting Bayesian networks with 3 methods in combining three 8- node CBN | 124 |
| C.10 | Three 8-node candidate Bayesian networks | 125 |
| C.11 | Three 10-node candidate Bayesian networks | 126 |
| C.12 | Three 12-node candidate Bayesian networks | 127 |

List of Tables

| | | |
|------|---|----|
| 1.1 | Possible cases in merging BNs | 12 |
| 4.1 | An example of order value in Baysian networks | 38 |
| 4.2 | Order values in candidate Bayesian networks and target ordering . . . | 42 |
| 4.3 | An example for order value based target variable ordering generation | 55 |
| 4.4 | Example of target ordering based on number of parents & size of networks | 57 |
| 4.5 | Example of edge matrices of candidate Bayesian networks | 58 |
| 4.6 | Resulting edge matrix according to edge matrix based target ordering algorithm | 58 |
| 5.1 | Ten Non-genetic factors | 74 |
| 5.2 | Variable ordering in combining three 5-node BNs with method 1 and method 2 | 77 |
| 5.3 | Variable ordering in 5-node candidate BNs with method 3 | 78 |
| 5.4 | Combination using 3 methods in three 5-node BN combination | 81 |
| 5.5 | Variable ordering in 6-node candidate Bayesian networks | 82 |
| 5.6 | Variable ordering in 6-node candidate BNs with method 3 | 83 |
| 5.7 | Comparison of 3 methods in three6-node BN combination | 84 |
| 5.8 | Comparison of different size Bayesian networks combination | 86 |
| 5.9 | Some factors influence the decision | 88 |
| 5.10 | The model of body seperation surgery | 88 |
| 5.11 | Opinion of knowledge source 1 | 89 |
| 5.12 | Opinion of knowledge source 2 | 89 |
| 5.13 | Opinion of knowledge source 3 | 89 |

| | | |
|------|---|-----|
| 5.14 | Opinion of knowledge source 4 | 90 |
| 5.15 | Opinion of knowledge source 5 | 90 |
| 5.16 | Expected utilities corresponding to probability distributions from 5 knowledge sources | 91 |
| 5.17 | Comparison of weights to 5 experts using different methods | 93 |
| 5.18 | Comparison of arithmetically combined value of expert opinions . . . | 95 |
| C.1 | Variable ordering in 7-node candidate Bayesian networks | 121 |
| C.2 | Variable ordering in 7-node candidate BNs with method 3 | 122 |
| C.3 | Variable ordering in 8-node candidate Bayesian networks | 124 |

Chapter 1

Introduction

1.1 Background

Many practical problems may include a large number of interrelated uncertainties. Probabilistic graphic modeling techniques are widely used in various areas as tools of abstracting uncertainties in the real world.

Over the past two decades, a large number of Artificial Intelligence (AI) researchers have been making their efforts on methods of learning parameters and structure from data. Graphic modeling roots in statistics, incorporating many other techniques as well, to exploit conditional independence properties of modeling, display, and computation.

Probabilistic graphical models are an intersection of probability theory and graph theory. They are graphs in which nodes represent random variables and the absence of arcs represents conditional independence assumptions.

Definition 1.1 *Probabilistic Graphic Model (PGM).* A probabilistic graphic model is a special knowledge base, which consists of 1) A set of variables; 2) Structural dependence between variables; 3) Component probabilities to the model.

According to the difference on arc's direction, such graphic models can be divided into three main groups: undirected graphs, directed graphs and mix graphs. Undirected models found their applications in the physics and vision communities, while directed models became more popular in AI and statistics communities. Directed

edges represent probabilistic influences or causal mechanisms while undirected links represent associations or correlations. There are also models that consist of both directed and undirected arcs, and they are called chain graphs.

Bayesian networks and influence diagrams are two major probabilistic graphic tools for knowledge representation and reasoning.

1.1.1 Bayesian Networks

Bayesian networks, also called *belief networks*, *Bayesian belief networks*, *causal probabilistic networks*, or *causal networks* [Pearl, 1988] are directed acyclic graphs (DAG) in which nodes represent random variables and arcs represent direct probabilistic dependences among them.

Formally, a Bayesian Network (BN), $B = (G, \theta)$ over X_1, \dots, X_n is a BN structure G , where each node X_i is associated with a *Conditional Probability Table* (CPT) $P_B(X_i | \text{Parents}(X_i))$, which specifies a distribution over X_1, \dots, X_n via the *Chain Rule* for Bayesian networks:

$$P_B(X_1, \dots, X_n) = \prod P_B(X_i | \text{Parents}(X_i)) \quad (1.1)$$

As we can see from the above definition, conditional independencies can be readily identified from the graph and are used to drastically reduce the complexity of inference.

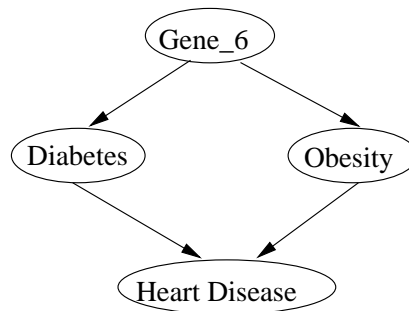


Figure 1.1: An example Bayesian network

Figure 1.1 captures a simple example of a BN. It illustrates that this compact representation can effectively reveal dependency and conditional independence re-

relationships among variables. The strengths of the links are quantified by the nodes conditional probability tables in the nodes. Bayes theorem is used to resolve uncertainties in the network. They were first described by Judea Pearl in his book [Pearl, 1988].

The network models two disorders: *Diabetes* and *Obesity*, their common cause, *Gene_6*, their common effect *Heart Disease*. Each node consists of two states indicating the presence or the absence of a given finding. Arcs denote direct probabilistic relationships between pairs of nodes. Therefore, the arc between *Gene_6* and *Obesity* represents the fact that the presence of *Gene_6* in one's body influences the likelihood of being fat. Relations like this are quantified numerically by means of conditional probability distributions.

The joint probability distribution of the example model is represented by the following equation:

$$Pr(G, D, F, H) = Pr(G) \cdot Pr(D|G) \cdot Pr(F|G, D) \cdot Pr(H|G, D, F) \quad (1.2)$$

where G stands for *Gene_6*, D stands for *Diabetes*, and F for *Fatness*. If we take into account conditional independence relationships among the modeled variables, we can rewrite Equation (1.2) as follows:

$$Pr(G, D, F, H) = Pr(G) \cdot Pr(D|G) \cdot Pr(F|G) \cdot Pr(H|D, F) \quad (1.3)$$

The third term of the right hand part of Equation (1.3) was simplified because D and F are conditionally independent given G . The fourth term was simplified because H is conditionally independent of G given its parents D and F .

The assumptions of conditional independence allow us to represent the joint probability distribution more compactly. If a network consists of m binary nodes, then the full joint probability distribution would require $O(2^m)$ space to represent, but the factored form would require $O(m2^n)$ space to represent, where n is the maximum number of parents of a node. Variables in a BN can be either discrete or continuous. The most commonly used probability distribution in BNs is the

Gaussian distribution.

1.1.2 Influence Diagram

Influence diagrams [Howard and E, 1984] are based on a graphical modeling language that can represent decision situations. An influence diagram is a way of describing the dependencies among variables and decisions. It can be used to visualize the probabilistic dependencies in a decision model and to specify the states of information for which independencies can be assumed to exist.

An influence diagram consists of a directed acyclic graph over chance nodes, decision nodes and utility nodes with the following structural properties:

- There is a directed path comprising all decision nodes;
- The utility nodes have no children.

For the quantitative specification, it is required that:

- The decision nodes and the chance nodes consist of a finite set of mutually exclusive states;
- Each chance node A is attached a conditional probability table $P(A|pa(A))$, where $pa(A)$ denotes all the parent nodes of node A ;
- Each utility node U is attached a real-valued function over $pa(U)$.

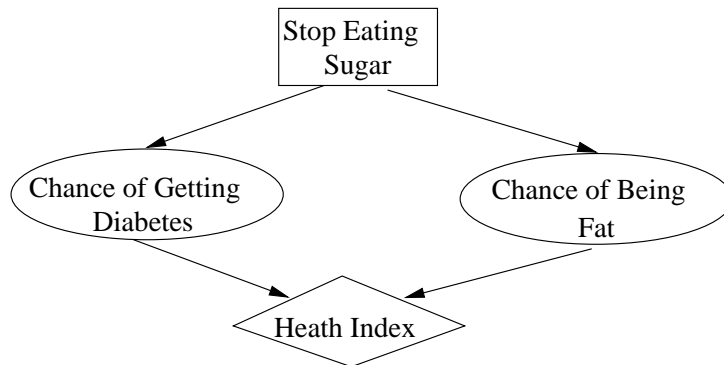


Figure 1.2: An example influence diagram

In an influence diagram, different decision elements show up as different shapes: rectangles represent decisions, ovals represent chance events, and diamonds represent the final consequence or payoff node.

A simple example of an influence diagram is shown in Figure 1.2 . The graph is interpreted as follows: *Chance of Getting Diabetes* and *Chance of Being Fat* are chance nodes, *Stop Eating Sugar* is a decision node, and *Health Index* is a value node. The outcome of variable *Chance of Being Fat* is conditioned on the decision on *Stop Eating Sugar* actually taken. The objective is to maximize the expected value of *Health Index*, which is conditioned on both *Chance of Getting Diabetes* and *Chance of Being Fat*.

Influence diagrams are mathematically precise and they have been used for more than twenty years as an aid for formulation of decision analysis problems. The major advantage of the influence diagram is an unambiguous and compact representation of probabilistic and informational dependencies. Influence diagrams capture the structure of a decision problem in a compact manner. Introducing new factors does not contribute to visual exponential growth of information.

In an influence diagram, each additional factor to be considered requires only a node and an arc. Hence influence diagrams can facilitate model construction for a sophisticated decision problem, or the communication of the overall model structure to other people.

A straightforward method to solve an influence diagram is to convert the influence diagram into a corresponding decision tree, and to solve that tree. The most common solution algorithm to influence diagrams can be found in [Shachter, 1984].

1.1.3 Knowledge Sources of Probabilistic Graphic Models

Probabilistic graphic models can be applied in a number of practical domains, for example, medical diagnosis, planning, natural language processing, etc. These models can be constructed from different *knowledge sources* in most application domains. The knowledge sources can be expert opinions, literature, data sets or knowledge bases. Probabilistic models can be obtained from one type of knowledge sources, or a combination of different types of knowledge sources.

Definition 1.2 *Knowledge Source.* From the perspective of artificial intelligence, the source of knowledge usually refers to a knowledge base created from data, knowledge base, literature or domain experts.

In this thesis, knowledge source means created probabilistic graph models created from data, knowledge base, literature or domain experts.

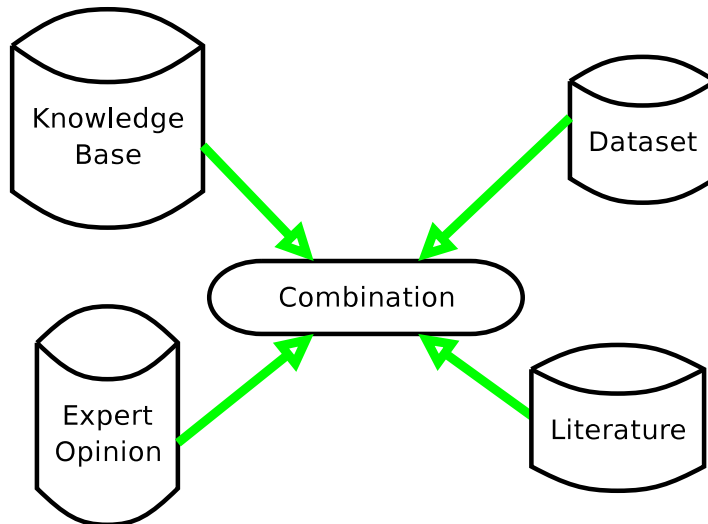


Figure 1.3: Knowledge combination from different sources

1.1.3.1 Experts

Direct manual construction of probabilistic graphic models by domain expert(s) is a quick method of acquiring probabilistic graphic models. Domain experts are good at the relationship among different variables and the conditional probabilities are assessed based on experts' knowledge. However, it is not easy in the case of large networks as not all domain experts are well versed in probability theory and the concept of conditional independence. Another challenge [Kahneman et al., 1988] in direct elicitation of domain expert opinion is the possible biases in subjective opinions from domain experts. Some researchers [Morgan and Henrion, 1992, Wang and Druzdzel, 2000] presented various techniques, such as the use of lotteries, to address these problems.

In spite of the above challenges, domain expert opinions are valuable especially when data is absent or sparse.

1.1.3.2 Literature

Materials from literature are records of domain research, experiment results or findings. Therefore, a lot of related domain glossaries together with probabilistic information are available in literature.

To derive probabilistic graphic models from literature, the challenge is to find how related knowledge is encoded in the literature so that useful information can be abstracted for model construction. Such a task sometimes needs additional domain knowledge [Lau and Leong, 1999, Korver and Lucas, 1993].

Another challenge may prohibit direct use of information from literature. Some reported findings in the literature are derived based on different data sets, or under different experimental settings [Druzdzel et al., 1999], and hence are difficult to be combined or used together.

1.1.3.3 Data Set

Data usually contains highly valuable information. Large data collections are available in some data-rich application domains. To learn probabilistic graphic models from data sets, the challenges include missing data, small data sets and selection biases, etc.

There are essentially two approaches to learning the graphical structures from data [Heckerman et al., 1994]. The first is based on constraint-based search [Pearl and Verma, 1991, Spirtes et al., 1993] and the second on Bayesian search for graphs with the highest posterior probability given the data [Cooper and Herskovits, 1992]. Once the graphical structure has been established, assessing the required probabilities is quite straightforward and amounts to studying subsets of the data that satisfy various conditions.

1.1.3.4 Knowledge Base

Knowledge base is a store of knowledge over a certain domain, which may include some factual and heuristic knowledge (for example, some rules), represented in machine-processable form [Leong, 1991]. Knowledge bases are widely used in expert systems, being able to provide better support for reasoning than databases.

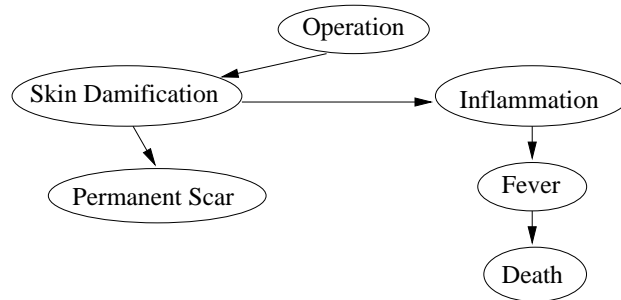
1.2 Motivations

Bayesian networks and influence diagrams are good probabilistic graphical modeling language for representing and reasoning with decision problems. Real world problems usually involve a large amount of variables, and the complex relationships among variables. We may derive multiple decision models that are heterogeneous in structure, or with different parameters, even from the same data sets or experts from the same domain.

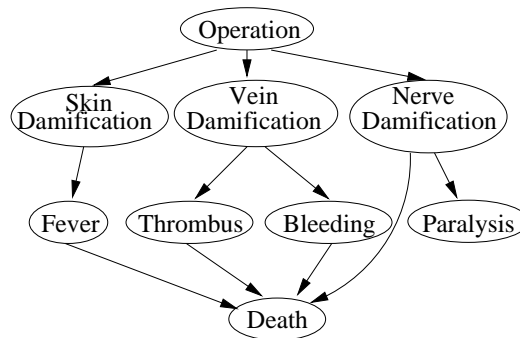
In medicine, for some complex medical decision problems, usually more than one experts are invited to provide their opinions, based on existing data or literature. These expert opinions, data or literature represent different knowledge sources. These knowledge sources may provide knowledge for the same issues. It is also quite often that different contributors are likely to have different views based on their expertise; therefore, different sets of factors (i.e. variables) will be considered.

Consider the following example: we assume that a surgeon *Jack* plans to do a head operation on his patient *Rose*. However, *Jack* is not confident of his knowledge on nerve damnification and skin damnification. In order to make a sound decision, *Jack* needs to acquire additional knowledge related to possible nerve damnification and skin damnification in a head operation. Therefore, he seeks help from dermatology literature and neurology data set.

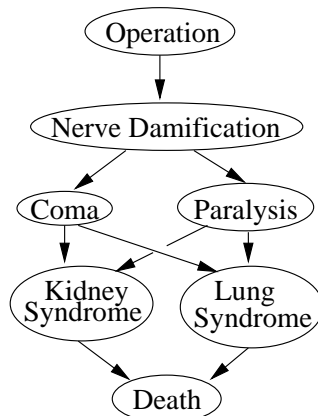
This example case on a forthcoming head operation is shown in Figure 4.1. Three Bayesian networks are modeled from dermatology literature, a surgeon's domain expertise (i.e., *Jack*) and neurology data set respectively. The variables *operation* and *death* exist in all of the three networks. The first network and the second network have another two common variables—*skin damnification* and *fever*. The second network and the third network contain another two common variables—*nerve damnification* and *paralysis*. Although there are some common variables between any two networks, the structures are different. For example, there is a direct arc from skin damnification to fever in the second network, while there is no direct arc in the first network. In the second network, there is no link from variable *paralysis* to variable *death*, while there is a route from *paralysis* to *death* through *lung syndrome*. This example is a simplified version of real medical problems. In fact, real medical



(a) From Dermatology Literature



(b) From Surgery domain expert



(c) From Neurology data set

Figure 1.4: An example of knowledge combination in medical domain

problems usually involve a large number of variables, complex relationships among the variables, and numerous parameters.

In combining different models to make a decision, one usually does not have enough ability and time to draw a reasonable conclusion and correctly integrating these models. Our research aims to develop an effective approach to combine knowledge from different sources in decision modeling.

In a rapidly changing world, different new fragments of knowledge or models may arrive when there is already an existing model. The problem of model integration is challenging. The different models to be integrated can differ in structure, or in parameters, even if they are obtained from the same data or experts from the same domain. This is due to the following reasons:

(1) The sources of different models can be different [Druzdzel and van der Gaag, 2000].

(2) Models may be constructed with different graphic modeling techniques [Heckerman et al., 1994, Heckerman, 1999]. They can be learned from data or elicited from domain experts.

A unified model is always needed for the final decision or global view of a certain problem. Our research aims to provide a solution to combine different graphic models that are either learn from data or elicited from domain experts. The sources of different models can be distinct, or the same. Integration of the various models may involve combinations in both probability distributions and structure.

Specifically, the motivations of our research include:

1. *Diversity and decentralized information sources.* Nowadays, the information explosion is accelerating, the knowledge arises from various background or sources might be different.
2. *Combine opinions from specialists who are from different subset of the whole domain.* It is easy to understand that nobody is an omni-faceted expert. Each individual can only have limited part of knowledge over the world, or over a certain domain. Different contributors are likely to have different views on their domain of expertise. As a result, when we need to have a global overview

of certain domain, it is necessary to combine knowledge from various sources.

3. *Laborious and time-consuming process.* With the emergence of a large amount of information, it is laborious and time-consuming to manually combine all the knowledge or complex models. Furthermore, the combination of models from various sources requires substantial probabilistic reasoning techniques, which is not familiar by everyone.
4. *Combine correctly.* Combination of probabilistic graphic models is not an easy task. Since there are two kinds of representation of probabilistic graphic model: qualitative representation (i.e. structure) and quantitative representation (i.e. parameters), the combination methods can also be distinguished according to the *order* of qualitative combination and quantitative combination. Qualitative combination can make aggregated estimate over consensus model's structure, while quantitative combination can provide parameters of aggregated model. Two main challenges arise in the research on combining models: how to preserve the conditional independence, the probability distributions, and avoid cyclic arcs. The first challenge concerns the structural aspects of the combination task. The second challenge concerns the parameter combination. Table 1.1 summarizes 8 task categories when combining Bayesian networks, in which *Category 1* is the easiest one and *Category 8* denote the most challenging situation. In combining influence diagrams, the types of nodes and arcs will also be considered.
5. *Combine effectively.* Beyond correct combination, optimization is another aim. For example, we hope to minimize speed of combination and the number of arcs in the aggregated graph. Although there are many research efforts on solving the model combination problems, most of them only discuss either probability distribution combination or qualitative combination only, but not both tasks at the same time. Furthermore, the existing methods cannot be easily scaled up, which means their methods can only combine two models at one time. Therefore, we are interested in developing approaches that do not have to restrict the number of BNs to be combined.

| | Numeric Parameters | Structures | Number of Nodes |
|------------|--------------------|------------|-----------------|
| Category 1 | Same | Same | Same |
| Category 2 | Same | Same | Different |
| Category 3 | Same | Different | Same |
| Category 4 | Different | Same | Same |
| Category 5 | Same | Different | Different |
| Category 6 | Different | Different | Same |
| Category 7 | Different | Same | Different |
| Category 8 | Different | Different | Different |

Table 1.1: Possible cases in merging BNs

Nevertheless, our aim is not at how to combine the raw information, but knowledge, which we mean different probabilistic graphic models from various information sources.

1.3 Objectives

To fill the gap among different graphical model combination techniques, we propose a consistent and scalable way to integrate partially or completely overlapping but possibly heterogeneous models from different information sources. The objectives of our research include:

- To propose a generic framework for combining partially or completely overlapping graphic model from different sources. Our basic goal is to accomplish both qualitative and quantitative combination of graphic models.
- To combine more than two models at one time.
- To deliver robust theoretical support for each step in our methods.
- To build a graphic model combination system. The system architecture should provide user-interactive execution environment while the detailed combination part is transparent to users.

In summary, we will propose a generic method that can effectively combine different graphic models. We also aim to develop methods to generate the resulting graphical

models. Finally, we need to ensure the correctness and scalability of our combination method.

1.4 Research Approach

To reach these objectives, first we study some related existing techniques, and briefly analyse the advantages and limitations of these techniques.

We restrict our attention to the same type of input probabilistic networks; in other words, the input probabilistic networks are either Bayesian networks or influence diagrams, but not both.

In combining Bayesian networks, we explore Joint Probability Distribution (JPD) factorization in Bayesian network, the ordering of variables, the Conditional Probability Table (CPT) encoded in models, the requirement of direction of edges, etc.

In combining influence diagrams, we extend our consideration to various types of nodes and various types of arcs in influence diagram, as well as related restrictions in the procedure of influence diagram aggregation.

Different from traditional approaches that emphasize the use of CPT to model conditional independence; we focus on using CPT to model unconditional independence among variables. In this way, we can get homogenous structures of each candidate graphic models (i.e. those graphic models to be combined), and effectively add virtual arcs among independent nodes into intermediate networks when necessary.

With identical structure of each candidate graphic models, our next step is to combine probabilities encoded in graphic models. We believe that point probability is not the only format to be encoded when the candidate graphic models are Bayesian networks, because the main usage of BN is to provide reference or get clear relationship among variables for complex problems. So we provide the user with another choice, i.e. adopting Interval Bayesian Networks (IBN) [Ha and Hadaway, 1996] (i.e., the CPTs are no longer in the format of point probability, instead in the format of interval probability distribution), as the resulting BN type after combination.

To effectively demonstrate that our methods are correct and reasonable, in this thesis we also provide theoretical proofs and use case studies to evaluate our approaches.

In addition, we design and develop a software architecture of probabilistic graphical model combination (the PGMC system), which is based on SMILE (Structure Modeling, Inference and Learning Engine) C++ API under Windows environment and GeNIe (Graphical Network Interface), developed by University of Pittsburgh ¹

1.5 Application Domains

The problem of probabilistic model integration from various sources is prevalent, and can be applied in various domains, such as medical diagnosis, stocks, business, air traffic control, military operation, and so on. Therefore, the research of knowledge combination in this thesis should be a general system that supports a wide spectrum of decision problems.

1.6 Organization of Thesis

We now give a brief description of the content of this thesis.

Chapter 1 mainly gives an introduction on the motivation and objectives of our research work, and the structure of the whole thesis.

In Chapter 2, we provide a global overview of existing approaches of probabilistic model combination and probability distribution combination.

In Chapter 3, we provide a formal problem formulation for probabilistic graphic model combination and discuss the existing challenges.

In Chapter 4, we present our four-step approach to effectively combine of probabilistic graphic models. This approach involves a series of key techniques including *arc reversal*, *variable ordering*, etc. We further analyze some special properties of influence diagrams, which are different from Bayesian networks. Based on the attributes of influence diagrams, we get special precondition for influence diagram

¹More information about the GeNIe and SMILE can refer to <http://www.sis.pitt.edu/~genie/>.

combination. In addition, we present a utility-based method of parameter combination for influence diagrams.

In Chapter 5, we examine our case studies on over 30 heart disease models on Bayesian network combination, and a body separation operation case from medical domain on utility-based parameter combination in influence diagrams.

In Chapter 6, we conclude our work and our findings. We discuss the advantages and limitations of our approaches. We also postulate to some further study directions based on our research work in this thesis.

Chapter 2

Related Concepts and Technologies

A probabilistic graphic model consists of the qualitative part (i.e., structure) and the quantitative part (i.e., parameter). This chapter briefly surveys four major approaches to structure combination of probabilistic graphic models and four major approaches to parameter combination of probability distributions. This survey also makes some analysis on their advantages and limitations.

2.1 Structure Combination

Multiple probabilistic graphic models that represent information or knowledge from multiple sources can happen under different situations. It can be the design of a distributed system, or a team, which is initially unaware of other team member's opinions or existence. It can also be some fusion of local networks into global networks. Bayesian network combination is a problem that has been tried to solve from more than 10 years ago [Matzkevich and Abramson, 1992]. The simplest way to deal with multiple probabilistic graphic models is to stick to one network and discard all others. Different methods will result in different answers to the combination of probabilistic graphic models.

2.1.1 Multi-entity Bayesian Networks

Before we introduce *Multi-entity Bayesian Networks* (MEBN) [Laskey et al., 2001], we need to mention BN fragments, which are the basic units in MEBN. The network

fragment consists of a set of related variables together with knowledge about the probabilistic relationships among the variables.

Multi-entity Bayesian network is a collection of BN fragments that satisfy consistency criteria such that the collection specifies a probability distribution over attributes of and relationships among a collection of interrelated entities. A MEBN implicitly encodes a probability distribution over an unbounded number of hypotheses.

The main idea of MEBN is that the active selection of related knowledge base. For any given problem, only a finite subset of these hypotheses will be relevant. To reason about specified target hypotheses given evidence about a particular situation, an ordinary finite Bayesian network, called a *situation-specific network* (SSN) [Laskey and Levitt, 2002], is constructed from an MEBN knowledge base. The SSN construction process is initiated when clusters of reports trigger firing of a *suggestor*. Trigger suggestors are rules that use to given situation to decide which hypotheses need to be represented. SSN is ordinary finite BN constructed from an MEBN knowledge base, to reason about specific target hypothesis, with a particular evidence.

Therefore, it is MEBN's advantage that it can pull from the entire knowledge base on a certain target hypothesis, which allows a faster response to widely dispersed, but related events.

Furthermore, MEBN logic extends ordinary Bayesian networks to provide first-order expressive power, and extends first-order predicate calculus (FOPC) to provide a means of specifying probability distributions over interpretations of first-order theories.

However, MEBN has its own limitations. MEBN has to get a set of pre-defined first order logic in order to quick search related knowledge base, which is not fit for solving unexpected uncertain problems.

2.1.2 Multiply Sectioned Bayesian Networks

The formal statement for *Multiply Sectioned Bayesian Networks* (MSBN) [Xiang et al., 1993, Xiang, 1995] is as follows. A MSBN M is a triplet (V, G, P) . V is

the union domain from all agents. G is the structure, i.e. hypertree MSDAG. P is the Joint Probability Distribution over G . $P(X|pa(x))$ is assigned to exactly one occurrence of x and uniform potential to all other occurrences.

MSBN are presented to solve the problem of multi-agent probabilistic reasoning without an exposition of its single-agent counterpart and build intelligent decision support systems offered by multi-agent. Therefore, MSBN are a set of subnets. Each subnets can be transformed into a junction tree to allow efficient inference in each sub-domain.

An example of MSBN is shown in Figure 2.1, containing two Bayesian networks G_1 and G_2 as subnets (see Figure 2.1.(a)). The local graphs after moralization are shown in Figure 2.1.(b). From the local graphs, every agent in a MSBN system needs to compile its subnet into a junction tree representation for effective local inference, as shown in Figure 2.1.(c). As no cluster in either junction tree contains the d-sepset $\{f, g, h\}$, to fix this problem, a link $\{f, h\}$ is added to each of the local graphs in Figure 2.1.(b). The resulting junction trees are show at Figure 2.1.(d).

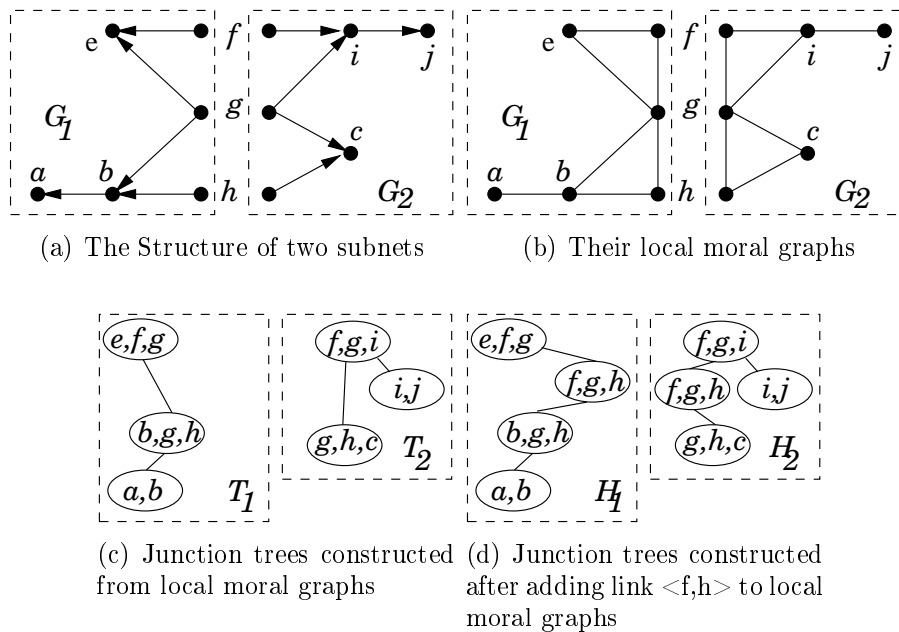


Figure 2.1: An example multiply sectioned Bayesian networks

Information channels called *linkage* between junction trees are created to allow propagation of evidence during attention shift. Figure 2.2 shows the graphical structure for computing the e-message, i.e. the cluster tree L .

son, 1993].

The second limitation for such works is that no parameter combination for Bayesian networks has been discussed.

The third limitation of these works is that only two models can be combined at one time. Besides the shortcoming of unscalability, the resulting model can also be influenced by the order of combination, if there are more than two models to be combined.

2.1.4 Graphical Representation of Consensus Belief

Different from the work in topological fusion of Bayesian networks, the work of graphical representation of consensus belief extends well-known results from the aggregation of joint distributions to the case of graphical model combination [Pennock and Wellman, 1999].

This piece of work focuses on how to combine multiple experts' opinions since in many situations, more than one expert will be consulted. So if each one of the k consulted experts holds a subjective belief expressed in the form of joint probability distribution P_i , then a consensus joint probability distribution P is any function of P_i ,

$$P \equiv f(P_1, \dots, P_k)$$

where P itself is a legal joint probability distribution and f is the *aggregation* or *combination* function. Pennock and Wellman [Pennock and Wellman, 1998] have devised several procedures to build consensus Markov networks and consensus Bayesian networks that are consistent with the logarithmic opinion pool

$$P(x_j) = m \star \prod_{i=1}^n (P_i(x_j))^{w_i} \quad (2.1)$$

2.2 Probability Distribution Combination

We can classify combination methods [Genest and Zidek, 1986, Winkler and Clemen, 1992, Rantilla and Budescu, 1999] into behavioral approaches and mathematical approaches [Rantilla and Budescu, 1999, Downs et al., 1997]. Mathematical combination, which focuses more in computational side, uses certain properties to assign equal weights or different weights to the experts. For decades, a series of researchers have been working hard on mathematic approaches of combination [Schmittlein et al., 1990], and therefore quite a few methods are presented. Here we classified them into three categories: Weighted combination, Bayesian combination and fuzzy arithmetic combination.

2.2.1 Behavior Approaches

Behavioral approach, also called psychological scaling [Cooke, 1991], is obtained through a facilitated discussion among the experts to some agreeable common values with perhaps a confidence interval or outer quartile values. Some approaches such as face-to-face group meetings, interaction by computer or sharing of information [Rebecca, 1995] in other ways. Experts can formally discuss their assessments to related events or variables, or informally talk about related issues. So the focus of behavioral approach can be different: sometimes on reaching agreement simply by discussion, sometimes on promoting communication of experts or information sharing among experts [Winkler, 1968].

The disadvantages of behavioral approach are also analyzed. For example, some experts might have the desire to dominate the discussion so the importance of information are decided upon some most active experts, instead of being scientifically decided, where some really important information can be neglected and new ideas can be discouraged. Hogarth [Hogarth, 1977] presents a way to prevent the dysfunction, which utilize additional analyst or experts who have good experiences to facilitate the order of discussion of experts.

Another famous but old approach to help make multi-experts decision-making is the Delphi method, which requires indirect iteration [Dalkey, 1969, Turoff and

Linstone, 2002]. Despite the different variations of experts, experts make individual judgment first and then exchange opinions anonymously. Each expert can revise the probabilities and such process can be repeated. It is ideal that all experts make consensus after a few round but unfortunately this seldom happen. After a number of rounds, all experts final probabilities still need seek for help from mathematic combination.

Note that the literature review in this part is not complete because the approaches are not deterministic and such research on the behavioral approach are beyond scope in this work. More literature on the behavioral approaches can be found in some behavioral psychology publications [Poulton, 1994].

2.2.2 Weighted Approaches

French [French, 1985] and Genest and Zidek [Genest and Zidek, 1986] provide summaries over a variety of methods of weighted combining probabilities, which are also called Axiomatic approach [Morris, 1983]. Given E experts with the i^{th} expert providing a vector of n probability values, $p_{1i}, p_{2i}, \dots, p_{ni}$, for sample space outcomes A_1, A_2, \dots, A_n , the E expert opinions can be combined using weight factors w_1, w_2, \dots, w_E , that sum up to one, using one of the following methods.

- **Weighted arithmetic average.** The weighted arithmetic mean for outcome j can be computed as $\sum_{i=1}^E w_i p_{ji}$

The weighted arithmetic means are then normalized using their total to get the 1-norm probability for an outcome for each outcome as $\frac{M_1(j)}{\sum_{k=1}^n M_1(k)}$.

- **Weighted geometric average.** The weighted geometric mean for outcome j can be computed as $\prod_{i=1}^E (p_{ji})^{w_i}$

The weighted geometric means are then normalized using their total to obtain the 0-norm probability for an outcome for each outcome as $\frac{M_0(j)}{\sum_{k=1}^n M_0(k)}$.

- **Generalized weighted average.** The generalized weighted average for outcome j can be computed as $(\sum_{i=1}^E w_i p_{ji})^{1/r}$.

The generalized weighted for averages are then normalized using their total to obtain the r -norm probability for an outcome which each outcome as $\frac{M_r(j)}{\sum_{k=1}^n M_r(k)}$.

where when $r = 1$, it is actually the weighted arithmetic average method and when $r = -1$, it is the weighted harmonic average formula.

2.2.3 Bayesian Combination Methods

The special character of Bayesian approaches is that it needs evidence to update the prior probabilities. So when we use the Bayesian combination method [Morris, 1977], we regard expert opinions as 'observations', then use Bayes Theorem to update the decision maker's prior distribution on the basis of these observations [French, 1990]. Many Bayesian models [Clemen and Winkler, 1999, Cooke, 1991, Ayyub, 2001, Morris, 1977] have been proposed in the past decades. We selectively review three methods of combination expert opinions in probability forms. We use $p = \{p_1, \dots, p_n\}$ to represent expert i 's expressed opinion in probability form that event θ occurs (i.e. $\theta = 1$). According to the posterior odds of the occurrence θ , $q^* = p^*/1 - p^*$.

- **Independence Model.** This model reflect the situation that each expert give independent opinions to the problem of assessing p^* . In this way more experts means confidence.
- **Genest and Schervish's model.** Genest and Schevish [Genest and Zidek, 1986] proposed a model-based on the assumption that the decision maker can only evaluate certain aspects of the marginal distribution of expert i 's probability p_i . The advantage and difference of this model over the previous independence model is that it permits mis-calibration of the p_i s .
- **Normal Method.** This model is from French [French, 1990] and Lindley [Lindley, 1985] and Clemen and Winkler adopt this model to study meteorological forecasts [Clemen and Winkler, 1999, Winkler and Clemen, 1986]. This method has the notable advantage of capturing the dependence among the experts' probabilities through the multivariate-normal likelihood functions.

All the above methods obey the Bayesian paradigm but have differences among them. As we can see different methods are suitable for different situations. Currently there is no one best is the best method, that fits all kinds of problems.

2.2.4 Interval Combination

Precise values are sometimes difficult to get, or sometimes unnecessary. Interval probability of an event can be specified as an interval of possible values rather than only as a precise one.

Dempster-Shafer approach [Dempster, 1968, Shafer, 1976] in uncertainty reasoning systems use a probability interval to estimate the need for more evidence. The probability interval represents the difference between the probability given the current evidence, and the maximum probability that could be achieved given more evidence. The size of the probability interval gives a good indication of the need for more evidence before making a decision. If the interval is large, then more evidence is probably required. If the interval is small, one can be fairly confident in making a decision.

The interval probabilities combination can borrow some ideas from the Dempster-Shafter theory. Alternatively, interval probabilities combination can be very simple and intuitive, just providing the interval from minimum probability to be combined to the maximum probability to be combined.

Chapter 3

Problem Analysis

Before we introduce our method of probabilistic graphic model combination, we firstly elaborate the problem that we intend to solve and discuss the challenges.

3.1 Problem Formulation

We assume a finite number of probabilistic graphic models M_1, \dots, M_m . $M_i = (V_i, \vec{E}_i)$ where $i = 1, 2, \dots, m$, and $\vec{E} = (a, b)$ denote directed edges between every pair of nodes a and b within one probabilistic graphic model. The direction of edge is from a to b , which we denote $\langle a, b \rangle$. These m probabilistic graphic models can satisfy $\emptyset \subseteq \bigcap_{i=1}^m V_i$ and $\emptyset \subseteq \bigcap_{i=1}^m \vec{E}_i$ where \emptyset denotes an empty set. These available probabilistic graphic models to be combined are termed as *candidate probabilistic graphic models* (if the models are BN models, they are *candidate Bayesian networks*).

To combine the k probabilistic graphic models, we aim to get a single *resulting probabilistic graphic model* (in the case of BN model combination, it is *resulting Bayesian network*) $M_{result} = (V_{result}, \vec{E}_{result})$, where $|M_{result}| = 1$, M_{result} have to remain to be a DAG, $V_{result} = \bigcup_{i=1}^m V_i$ and $\phi \subseteq \bigcap_{i=1}^m \vec{E}_i$.

Note that it is possible that $E_{result} = \bigcup_{i=1}^m \vec{E}_i$ is true only in some special cases.

The models to be combined can be described as in a 1) separate relationship 2) partial overlapping relationship; and 3) completely overlapping relationship.

Definition 3.1 *Separate relationship.* Any two of the k probabilistic graphic models do not have any common node.

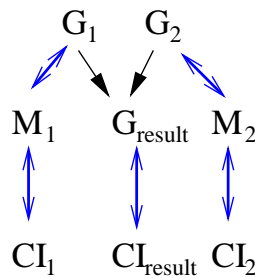


Figure 3.1: Probabilistic graphic models combination

Definition 3.2 *Partial overlapping relationship.* Any two of the k probabilistic graphic models have at least one common node.

Definition 3.3 *Complete overlapping relationship.* Every edge from one of the k probabilistic graphic models has an equal mapping from every edge of another model among the k probabilistic graphic models.

Figure 3.1 shows an example case of combination of two probabilistic graphic models. M_1 and M_2 are the two candidate models to be combined. G_1 and G_2 are graphs that correspond to M_1 and M_2 respectively. As M_1 implies G_1 , and G_1 encodes the conditional independence CI_1 in M_1 , M_1 is not only a valid probabilistic graphic model, but also a perfect map of the underlying dependency. Therefore, the problem that we are facing is to get the resulting G_{result} where the underlying CI_{result} (i.e., the conditional independence relationships in the resulting model) breaks the least conditional independency from CI_1 and CI_2 . In other words, we need to try to get a minimum $CI_{result}/\{CI_1 \cup CI_2\}$ as possible as we can.

3.2 Precondition of Probabilistic Graphic Combination

Before the probabilistic graphic model combination can be conducted, two preconditions have to be satisfied, in order to guarantee that the model combination make sense. Normally such process needs the knowledge engineer or domain expert's inspection.

3.2.1 Variable Consistency

The first precondition for probabilistic graphic model combination is variable consistency, which make identical constraints over overlapping variables.

1. Variables with the same name model the same world entity. Since the models are from different sources, they might not be identical in all variables. We need the domain expert's help to examine whether each variable with the same name is representing the same physical thing. Figure 3.2 are an example of two partially overlapping Bayesian networks to be combined, in which variables A and C are overlapping nodes. For the sake of correct combination, the A in Figure 3.2 (a) and the A in Figure 3.2 (b) have to denote same thing, for example, *Happy*; while the two C also have to represent same meaning, for example, *Laugh*.
2. Variables with the same name have to be over the same domain. In other words, those variables with same name must own same number of possible values, and their values must be the same. For example, In Figure 3.2, possible outcomes of variable A from Bayesian network 1 must be identical with the possible outcome of variable A in Figure 3.2 (b), either binary, 3 possible state etc. However, it is not enough for variable A in both models to be binary only, the states of A in each model must be same too. For example, each A in Figure 3.2 is a binary variable, and the two states are '*yes*' and '*no*' at the same time; or the two states can be '*in China*' and '*in Singapore*' at the same time.

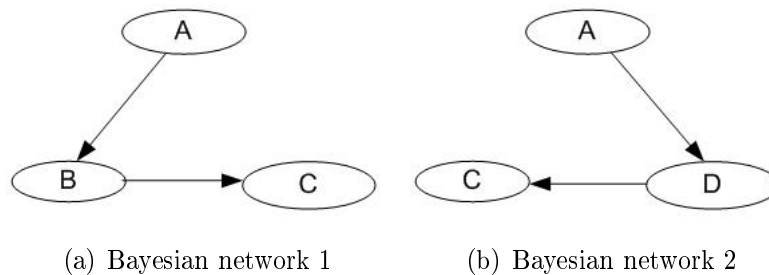


Figure 3.2: Two simple BNs to be combined

3.2.2 Model Consistency

Model consistency among those candidate probabilistic graphic models is another prerequisite. The possible model inconsistency problems may occur due to improper knowledge modeling, or model learning from or evaluate based on different population set of data. Figure 3.3 shows an example of possible incorrect knowledge modeling step. Assume there are two models on a certain internal disease D and its most important symptom S . In the modeling stage, the residents focused on obtaining numerical parameters for their network. However, the second resident model the knowledge based on data collected from patients in hospital ($H = h$), while the first resident model the knowledge from the data from a general population data set (not only those people who are hospitalized). Interested reader can find detailed explanation on this problem from [Druzdzal and Diez, 2003].

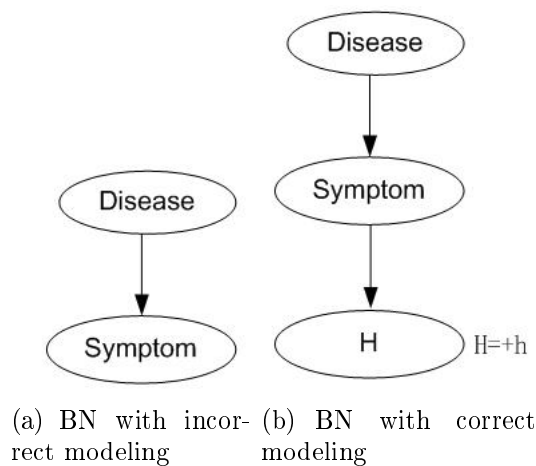


Figure 3.3: Improper Bayesian network modeling can result in problems

3.3 Challenges

Each probabilistic graphic model consists of two parts: the qualitative part and the quantitative part. As influence diagram is a special case of Bayesian network, in the description below, we use Bayesian network to illustrate the challenges.

The *structure* of a Bayesian network is a graphical, qualitative illustration of the interactions among the set of variables that it models. The structure of the

directed graph can mimic the causal structure of the modeled domain, although this is not necessary. When the structure is causal, it gives a useful, modular insight into the interactions among the variables and allows for prediction of effects of external manipulation.

A Bayesian network also represents the *quantitative* relationships among the modeled variables. Numerically, it represents the joint probability distribution among them. This distribution is described efficiently, exploring probabilistic independences among the modeled variables. Each node is described by a probability distribution conditional on its direct predecessors. Nodes with no predecessors are described by prior probability distributions.

There are four major challenges in this task:

The **first** challenge is in qualitative combination. How to avoid cycles after combination of multiple Bayesian networks? Figure 3.4 and Figure 3.5 present two example of possible situations which may incur cycle in combination. We call these two situations *Direct Conflict* cases and *Indirect Conflict* cases, respectively.

For m candidate Bayesian networks B_1, B_2, \dots, B_m . $B_j = (V_j, \vec{E}_j) B_j \subseteq \bigcup_1^m B_i$. $\exists v_x, v_y$, satisfy

$v_x \in V_j$ and $v_y \in V_j$, i.e., both are nodes in B_j .

if

(i) $\langle v_x, v_y \rangle \in \vec{E}_j$, i.e., there is a directed edge from v_x to v_y in B_j .

(ii) $\langle v_x, v_y \rangle \notin \{\bigcup_{i=1}^m \vec{E}_i \setminus \vec{E}_j\}$, i.e., the directed edge from v_x to v_y does not exist in other candidate Bayesian networks.

(iii) $\langle v_y, v_x \rangle \in \{\bigcup_{i=1}^m \vec{E}_i \setminus \vec{E}_j\}$, i.e., the directed edge from v_y to v_x exist in at least one of other candidate Bayesian networks.

if all of (i), (ii), (iii) are satisfied in candidate Bayesian networks, this situation is called *direct conflict*, in other words, the direction of edge between two nodes are different, as shown in Figure 3.4.

(iv) $\langle v_x, v_y \rangle \notin \vec{E}_j$, i.e., there is NOT a directed edge from v_x to v_y in B_j .

(v) $\{\langle v_x, v_{x+1} \rangle, \dots, \langle v_{x+c}, v_{y-d} \rangle \dots, \langle v_{y-1}, v_y \rangle\} \subseteq \vec{E}_j$, i.e., there exist an link whose length is not less than 2 from v_x to v_y in B_j .

(vi) $\{\langle v_y, v_{y+1} \rangle, \dots, \langle v_{y+h}, v_{x-i} \rangle \dots, \langle v_{x-1}, v_x \rangle\} \subseteq \{\bigcup_{i=1}^m \vec{E}_i \setminus \vec{E}_j\}$, i.e., there

exist links whose length is not less than 2 from v_y to v_x in at least one of Candidate Bayesian networks except B_j .

if all of $(iv), (v), (vi)$ are satisfied, or all of $(iv), (v), (iii)$ are satisfied in the Candidate Bayesian networks, this situation is called *indirect conflict* situation. Figure 3.5 shows an example of indirect conflict.

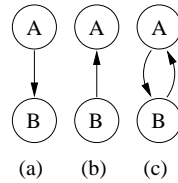


Figure 3.4: Direct conflict in DAG combination

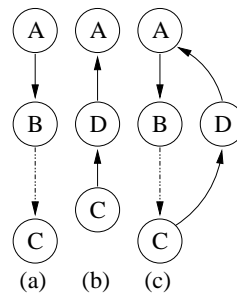


Figure 3.5: Indirect conflict in DAG combination

The **second** challenge is how to keep the minimal dependence relationship between any two nodes in the resulting Bayesian networks, because independence relationships may change after combination. If we add one arc into a Bayesian network, it is possible that some independent relationship among variables will be broken [Geiger et al., 1989].

Definition 3.4 Dependency map. A Directed Acyclic Graph G is said to be a dependency map (D -map) of a dependency model M if

$$D(X, Y|Z)_G \Rightarrow D(X, Y|Z)_M$$

That is, all Conditional Independent statements derived from G hold in M .

There is a number of dependency relationship in D-map . The difficult point is how to avoid introducing extra dependency relationship between arcs in the resulting Bayesian network.

Definition 3.5 Maximal Dependency map. *A Directed Acyclic Graph is said to be a maximal D-map of a dependency model M if it is a D-map of M, but it is not a D-map of M when adding any extra link into it.*

The **third** challenge is about the quantitative computation. Different candidate Bayesian networks may have different structures, which means the internal Conditional Probability Table (CPT) may be very different, not only in numbers, but also in the size of CPT. For example, the value of $P(A)$ in Model 1 is different from the value of $P(A)$ in Model 2. The CPT over node B in Model 1 is not only different from CPT over node B of Model 2 in numbers, but also in CPT size. as shown in Figure 3.6.

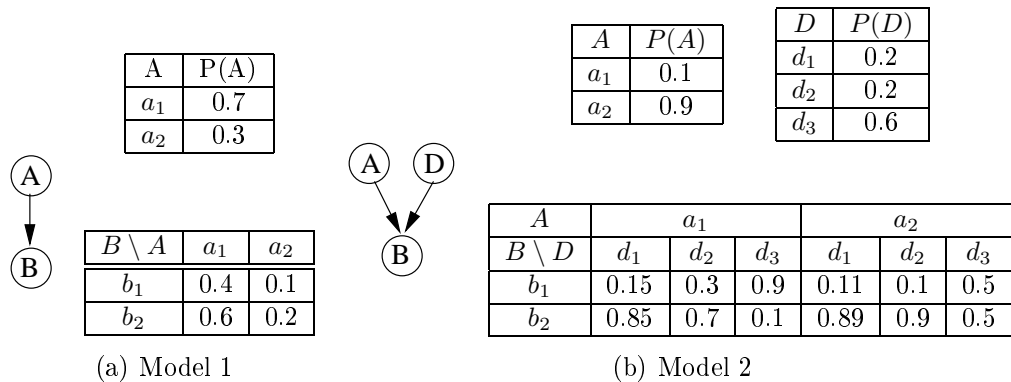


Figure 3.6: CPT disagreement in two models

The **fourth** challenge in model integration is how to integrate more than two models at the same time. In some large Bayesian network model learning problems, knowledge engineering sometimes learn some small part of Bayesian networks and then combine them into a global Bayesian network. We can imagine a possible case that there are many small Bayesian networks to be combined. If all these models can be combined at a time, some manual work or time can be saved. Unfortunately, existing methods can only combine two models at a time. This is also a problem that we attempt to solve.

Chapter 4

Probabilistic Graphic Model Combination

4.1 Structure Combination of Bayesian Networks

In this section, we will introduce the qualitative part of our approach of Bayesian network model combination. Section 4.1.1 firstly explains why Bayesian networks can be re-organized, followed by the demonstration on how to adjust all the input Bayesian networks into isomorphic topology in Section 4.1.2, in which definitions related to target variable ordering are provided. Section 4.1.3 introduces the concept of intermediate Bayesian networks to store the altered models in order to retain the original structure and parameter of candidate Bayesian networks.

Our proposed approach for integration of multiple Bayesian networks consists of four steps, as follows.

1. Reorganize original Bayesian networks;
2. Adjust variable ordering and edge direction;
3. Save amended models as Intermediate Bayesian networks; and
4. Combine CPT .

4.1.1 Re-organize Bayesian Networks

In order to demonstrate that Bayesian networks can be re-organized, our first step is to introduce the inherent *Joint Probability Distribution* (JPD) factorization properties of Bayesian networks, which theoretically supports the feasibility of re-organization of Bayesian networks.

Joint probability distribution is the probability distribution over all unknown quantities in which a probability is assigned to each possible combination of values. Let X and Y be two disjoint subsets of variables such that $p(y) > 0$. Then, the conditional probability distribution (CPD) of X given $Y = y$ is given by

$$p(X = x|Y = y) = p(x|y) = \frac{p(x, y)}{p(y)} \quad (4.1)$$

Equation 4.1 implies that the JPD of X and Y can be written as

$$p(x, y) = p(y)p(x|y) \quad (4.2)$$

One particular case of Equation 4.2 is obtained when X is a single variable and Y is a subset of variables. In this case, Equation 4.2 becomes

$$p(x_i|x_1, \dots, x_k) = \frac{p(x_i, x_1, \dots, x_k)}{p(x_1, \dots, x_k)} \quad (4.3)$$

$$= \frac{p(x_i, x_1, \dots, x_k)}{\sum_{x_i} p(x_i, x_1, \dots, x_k)} \quad (4.4)$$

which is the CPD of the i^{th} variable, X_i , given a subset of variables $\{X_1, \dots, X_k\}$. The sum in the dominator of Equation 4.4 is taken over all possible values of X_i .

Building probabilistic models relates to a specification of the joint probability distribution. Unfortunately, a direct specification of a JPD requires a huge number of parameters. For example, for k binary variables, the most general JPD involves 2^k parameters. However, in practice many subsets of variables are independent or conditionally independent. Hence, imposing some global or special cases of independence assumptions can decrease the number of parameters needed.

With the powerful combination of probability and graph theories, it is possible to

represent joint probability distributions over a set of variables in a Bayesian network in the following way:

$$P(x_1, x_2, \dots, x_n) = \prod_{i=1}^n P(x_i | \pi) \quad (4.5)$$

where x_i represents the instantiation of the variable X_i and π_i represents the instantiation of the parents of X_i .

Any JPD of a set of random variables can be defined in terms of a set of smaller Conditional Probability Distribution (CPD)s.

Example Consider a case of four variables $\{X_1, \dots, X_4\}$ partitioned as $Y_1 = \{X_1\}$, $Y_2 = \{X_2\}$, $Y_3 = \{X_3\}$, $Y_4 = \{X_4\}$. We can get the following equivalent chain rule factorizations of the JPD:

$$p(x_1, \dots, x_4) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2)p(x_4|x_1, x_2, x_3) \quad (4.6)$$

and

$$p(x_1, \dots, x_4) = p(x_1|x_2, x_3, x_4)p(x_2|x_3, x_4)p(x_3|x_4)p(x_4) \quad (4.7)$$

This implies that the JPD can be expressed as a product of four CPDs. Note that chain rule factorizations are NOT unique because one can apply the chain rule to different partitions of $\{X_1, \dots, X_4\}$ and obtain different chain rule factorizations. For example the following are two different but equivalent chain rule factorizations associated with the same JPD, obtained from different partitions of $\{X_1, \dots, X_4\}$. For example:

The partition $Y_1 = \{X_1\}$, $Y_2 = \{X_2, X_3\}$, and $Y_3 = \{X_4\}$ gives

$$p(x_1, \dots, x_4) = p(x_1)p(x_2, x_3|x_1)p(x_4|x_1, x_2, x_3) \quad (4.8)$$

The partition $Y_1 = \{X_1, X_4\}$ and $Y_2 = \{X_2, X_3\}$ gives

$$p(x_1, \dots, x_4) = p(x_1, x_4)p(x_2, x_3|x_1, x_4) \quad (4.9)$$

Therefore, without change of JPD of the whole model, a Bayesian network model can be re-organized into different Bayesian networks with different structure.

4.1.2 Adjust Variable Ordering to Maintain DAG

In the previous section, we explain that the variable ordering in a Bayesian network can be altered without modification of JPD of the original models. In this section, we will discuss some definitions related to generation of target variable ordering for the resulting Bayesian networks.

The basic relationship among nodes is parent-children pairs, where one node can have more than one children, and one node may have more than one parent nodes. Therefore, ancestral ordering is an intuitive description as defined below [Castillo et al., 1997].

Definition 4.1 Ancestral Ordering. *Let $G = (V, E)$ be a Directed Acyclic Graph. Then an ancestral ordering of the vertices in V exists if and only if G is a DAG. A more natural description of Ancestral Ordering is that every node comes after its parents.*

4.1.2.1 Order Value Computation for Variables

Definition 4.2 Order Value. *For every node $v \in V$ in DAG D (Note that both Bayesian networks and influence diagrams are DAGs), their $Ordervalue(v)$ is defined with the value assigned by the longest path from a rooted node to them.*

Thus a recursive function is defined, $Ordervalue(v) = 0$ iff $Parent(v) = \phi$, $Ordervalue(v) = \max(Ordervalue(Parent(v))) + 1$ when $Parent(v) \neq \phi$. The detailed algorithms about assigning order value to each node in one Bayesian networks are shown in Algorithm 4.1 and Algorithm 4.2, which have been implemented in our system. The aim of Algorithm 4.1 is to get all the root nodes and assign order value to these root nodes while Algorithm 4.2 is to assign order value to those non-root nodes.

In this way, if every arc in a DAG is always from a node with lower order value to another node that with higher order value, no cycles can exist. Therefore, the

DAG structure can be maintained.

Algorithm 4.1 AssignOrder

Require: $\mathcal{B} = (V, E)$

```

1:  $k = |V|$ ;
2: for  $i = 1$  to  $k$  do
3:   if  $\text{Parent}(v_i) = \phi$  then
4:      $\text{RootNodes.push}(v_i)$ 
5:   end if
6: end for
7: for  $i = 1$  to  $|\text{RootNodes}|$  do
8:    $\text{AssignOrderFromRoot}(\text{RootNodes}[i])$ 
9: end for

```

Algorithm 4.2 AssignOrderFromNode(RootNode)

Require: $\mathcal{B} = (V, E)$

Ensure: OrderValue is defined on all descendant nodes of RootNode

```

1: let  $\text{ChildNodes} = \{v | v \text{ is a child of RootNode}\}$ 
2: for  $j = 1$  to  $|\text{ChildNodes}|$  do
3:   if  $\text{OrderValue}(\text{ChildNodes}[j]) < \text{OrderValue}(\text{RootNodes}[i]) + 1$  then
4:      $\text{OrderValue}(\text{ChildNodes}[j]) = \text{OrderValue}(\text{RootNodes}[i]) + 1$ 
5:      $\text{AssignOrderFromNode}(\text{ChildNodes}[j])$ 
6:   end if
7: end for

```

Definition 4.3 Variable Ordering. λ is the sequence of $\text{OrderValue}(v)$ for all node $v \in V$ in DAG D .

Lemma Every arc in Bayesian networks are from nodes with lower ordervalue to nodes with higher ordervalue.

Proof:

Given k Bayesian networks B_1, \dots, B_k , $\lambda_1, \dots, \lambda_k$ are the variable ordering of B_1, \dots, B_k respectively.

We assume there exists one arc $\langle s_1, s_2 \rangle$ in B_i , $B_i \in \{B_1, \dots, B_k\}$, in which

$$\text{ordervalue}(s_1) \geq \text{ordervalue}(s_2) \quad (4.10)$$

is satisfied.

According to the definition of ordervalue in Bayesian networks, $\langle s_1, s_2 \rangle$ denotes an arc starting from s_1 , and ends at s_2 . Therefore we can get

$$ordervalue(s_2) = \max(Ordervalue(Parent(s_2)) + 1) > ordervalue(s_1) \quad (4.11)$$

which is conflict with 4.10. Therefore, every arc in Bayesian networks is from nodes with lower *ordervalue* to nodes with higher *ordervalue*.

□

With this lemma, we may easily guarantee that there is no cycle in the resulting Bayesian networks if all arcs in the resulting Bayesian networks are from nodes with lower ordervalue to nodes with higher ordervalue. Now we come to discuss the variable ordering in resulting Bayesian network.

Definition 4.4 Target Variable Ordering. *In probabilistic model combination problems, the variable ordering λ_{result} is the sequence of $Ordervalue(v)$ for all nodes $v \in V_{result}$ in DAG D_{result} . λ_{result} are not necessarily the same as one of any candidate models although it is allowed.*

Selection of target variable ordering for the resulting Bayesian network is a key step in BN models combination. The process of selecting target variable ordering is indeed a procedure of assigning $Ordervalue_{result}(v)$ to $v \in B_{candidate}$, i.e., each node will have only one target order value in each candidate model so that for each $B_{candidate}$, there is only one target ordering λ_{result} . We will continue to discuss the type of target ordering in Section 4.1.2.2 and how to generate it automatically in Section 4.3.

4.1.2.2 Two Types of Variable Ordering

Definition 4.5 Linear Ordering. *A graph is linearly ordered for V_s iff $\forall V_i, V_i \in \{V_s\} \cup anc(V), \exists V_j, V_j \in pa(V_i), \exists V_k, V_k \in pa(V_i), \Rightarrow (V_j = V_k) \vee (V_j \in pa(V_k)) \vee (V_k \in pa(V_j))$ where $anc(V)$ denotes all ancestor nodes of node V and $pa(V)$ denotes all the parent nodes of node V .*

| | Candidate BN_1 | Candidate BN_2 | Candidate BN_3 |
|-------------------|------------------|------------------|------------------|
| Ordervalue(A) | | 1 | 1 |
| Ordervalue(B) | 1 | 0 | 2 |
| Ordervalue(C) | 2 | | |
| Ordervalue(D) | | 1 | |
| Ordervalue(E) | 0 | | |
| Ordervalue(F) | | 0 | |
| Ordervalue(M) | | | 0 |
| Ordervalue(N) | | | 0 |

Table 4.1: An example of order value in Bayesian networks

In a Bayesian network with linear variable ordering, each node is not allowed to have more than one parents but can have more than one children. If D is a DAG that are connected and linear ordering network, $Ordervalue(v_i) = Ordervalue(v_j)$ iff $v_i = v_j$, and λ_k is a limited non-monotonic natural number sequence, i.e. $\{0,1,\dots,n\}$.

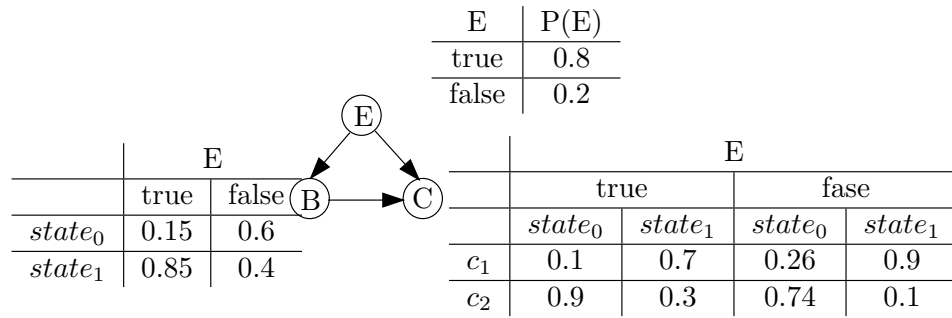
In Figure 4.1.(a) λ_{BN_1} is an example of linear variable ordering. The order value for each variable Figure 4.1 is listed in Table 4.1. In model of BN_1 , $Ordervalue(E) = 0$ because E does not have any parent nodes, while $Ordervalue(B) = 1$ because the longest path from top of the network is 1. However, λ_{BN_1} is not an example of linear ordering. In BN_2 , $D \in BN_2$, $B \in pa(D)$ and $F \in pa(D)$, but $B \neq F$.

From Figure 4.1, we can see that not every Bayesian network can satisfy the linear ordering definition. Therefore, we may relax the criteria of target ordering, from linear ordering to hierarchical variable ordering, as defined below.

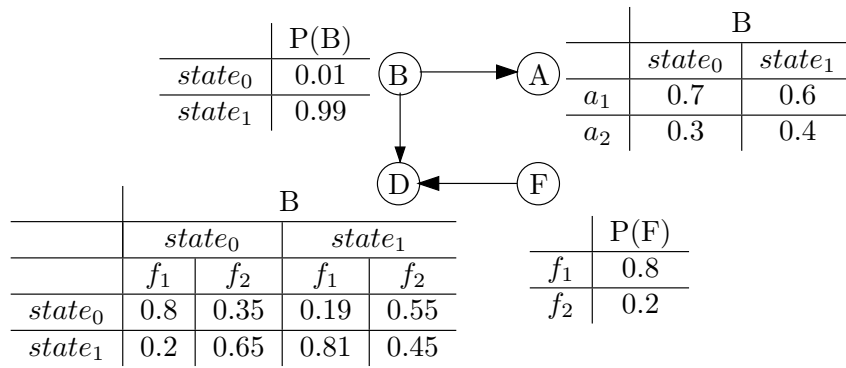
Definition 4.6 Hierarchical Variable Ordering. *A graph is hierarchically ordered for V_s if $\forall V_i, V_i \in \{V_s\} \cup anc(V)$, where $anc(V)$ denotes all the ancestor nodes of node V , $\exists V_j, V_j \in pa(V_i), \exists V_k, V_k \in pa(V_i), (V_j \in pa(V_k)) \vee (V_k \in pa(V_j))$ does not necessarily means $(V_j = V_k)$.*

In Figure 4.2, where there exist different node A, D, C where $Ordervalue(A) = Ordervalue(D) = Ordervalue(C)$ and $(A \neq D) \wedge (A \neq C) \wedge (D \neq C)$. To ensure there is not cycle in the network, the arcs are only allowed from nodes with lower *ordervalue* to nodes with higher *ordervalue*.

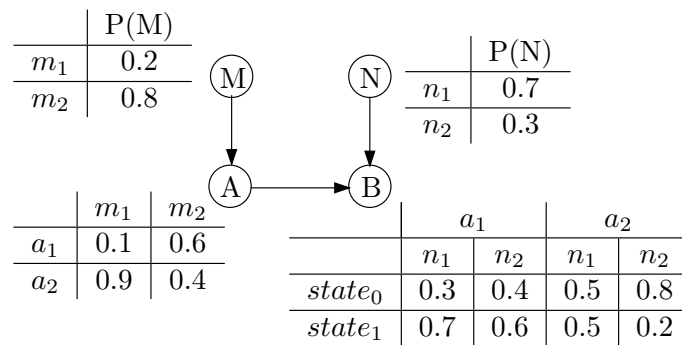
The definition of linear ordering provides a way of judging property of ordering from existing models. On the contrary, the following definition of linear target



(a) Candidate BN 1



(b) Candidate BN 2



(c) Candidate BN 3

Figure 4.1: Example of three candidate Bayesian networks

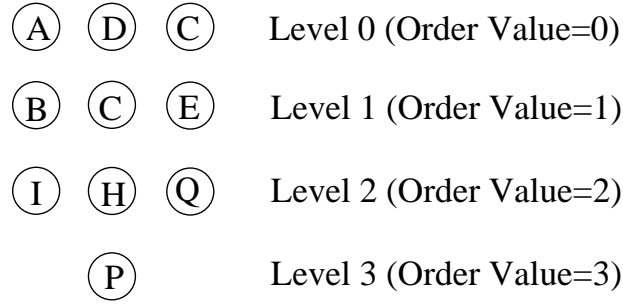


Figure 4.2: Example of ordering hierarchy of nodes in a BN

ordering offers a method of judging the ordering from the *ordervalue* of each node.

Definition 4.7 Linear Target Ordering. λ_{result} is the target ordering for more than two DAGs D_1, \dots, D_k where $k \geq 2$. If λ_{result} is a limited non-monotonic ascending sequence of natural number, then λ_{result} is a linear target ordering.

Definition 4.8 Hierarchical Target Ordering. λ_{result} is the target ordering for more than two DAGs D_1, \dots, D_k where $k \geq 2$. If λ_{result} is a limited sequence of natural number, in which $\lambda_i = \lambda_j$ when $i \neq j$, then λ_{result} is a hierarchical target ordering.

4.1.2.3 Arc Reversal to Adjust Variable Ordering

In this section, we will continue to discuss how to set the directed edges consistently with respect to the original candidate Bayesian networks. Based on the definition of *ordervalue* in Section 4.1.2.1, our next step is to explain why and when the operation of *Arc Reversal* is needed.

Arc reversal is needed here so that variable ordering in these networks are consistent with target variable ordering with preservation of JPD, but with some structural changes [Shachter, 1984]. The three candidate Bayesian networks after arc reversal can be seen in Figure 4.4.

Arc reversal is an important technique for Bayesian networks and influence diagrams. It has significance in the evaluation of Bayesian network through stochastic simulation [Fung and Chang, 1989, Kanazawa et al., 1995]. The reversal of the arc between two nodes in a Bayesian network means that the direction of arc will be reversed so that the BN can be re-constructed while the original probabilistic

distributions remain the same [Cheuk and Boutilier, 1997].

In our research, we use the arc reversal operation to reconstruct candidate Bayesian networks so that variable ordering in these networks is consistent with the target variable ordering. The definition of consistency in variable ordering is provided below.

Definition 4.9 Consistency in Variable Ordering. *Given two Bayesian networks $B_1 = (V_1, \vec{E}_1)$ and $B_2 = (V_2, \vec{E}_2)$, λ_1 and λ_2 are variable ordering in B_1 and B_2 respectively. Given any two variables $\langle V_i, V_j \rangle \in B_1$, we denote λ_1 is consistent with λ_2 iff $[\text{Ordervalue}_1(V_i) - \text{Ordervalue}_1(V_j)] \cdot [\text{Ordervalue}_2(V_i) - \text{Ordervalue}_2(V_j)] > 0$.*

Based on the circumscription of consistency in variable ordering, we may notice that the arc reversal operation is only necessary in some situations, as we conclude as following.

Lemma1 Given a pair of nodes (V_i, V_j) , if (1) the arc $\langle V_i, V_j \rangle \in B_i, i = 1, \dots, k$ where B_i is a Candidate Bayesian Network. and (2) $[\text{Ordervalue}(V_i) - \text{Ordervalue}(V_j)] \cdot [\text{Ordervalue}_{target}(V_i) - \text{Ordervalue}_{target}(V_j)] < 0$, i.e. the variable ordering λ_i in B_i is not consistent with λ_{target} . We need to adjust λ_i to λ_i'' so that $[\text{Ordervalue}(V_i) - \text{Ordervalue}(V_j)] \cdot [\text{Ordervalue}_{target}(V_i) - \text{Ordervalue}_{target}(V_j)] > 0$ is satisfied, in other words, λ_i'' is consistent with λ_{target} .

The key nodes in an arc reversal operation are the two nodes in the to-be-reversed arc, and the parent nodes of the two nodes. As we could see from Figure 4.3 (a), node A and node B is a parent-children pair, where A is the parent node and B is the children node. We denote the set of parent nodes of A as $Parent_{old}(A)$ before arc reversal. After the arc reversal, the set of parent nodes of A is denoted as $Parent_{new}(A)$. The parent nodes of node A and B before arc reversal, i.e. $Parent_{old}(A) \cup Parent_{old}(B)$, can be divided into three categories: 1) $U = Parent_{old}(A) \setminus Parent_{old}(B)$; 2) $V = Parent_{old}(A) \cap Parent_{old}(B)$; 3) $W = Parent_{old}(B) \setminus Parent_{old}(A)$.

Suppose the target arc to be reversed is the arc (A, B) , we want to get a result of another topology which B is the parent node of A . In order to make the conditional

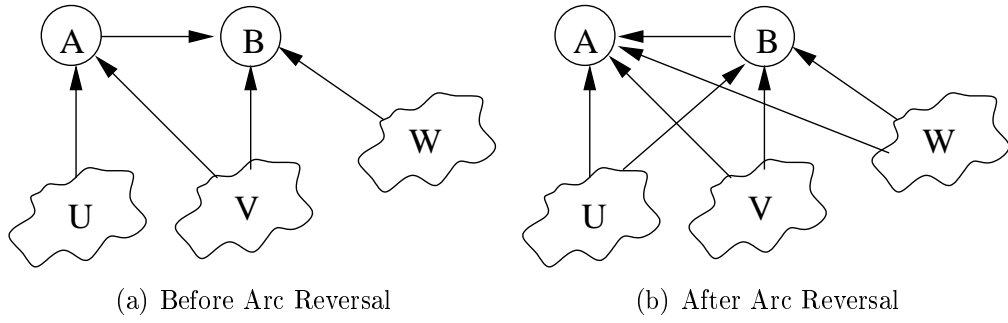


Figure 4.3: General structure of arc reversal

independence are correctly encoded, we must add parents of both A and B , so every variable in set U are parents of node B , and every variable in set W are parents of node A . At the same time, A and B will keep their original parents as well [Shachter, 1984]. The topology of the resulting network is shown at Figure 4.3 (b). The parent nodes of node A and B after arc reversal, i.e., $Parent_{new}(A) = BUUVVW = Parent_{old}(A) \cup Parent_{old}(B) \cup B$; while $Parent_{new}(B) = UUVW = Parent_{old}(A) \cup Parent(B) \setminus A$.

The Algorithm 4.3, which has been implemented in our system, describes the detailed steps of adjusting original candidate Bayesian networks into new Bayesian network so that λ_{new} is consistent with target ordering λ_{target} .

| Node | $BN_{Candidate1}$ | $BN_{Candidate2}$ | $BN_{Candidate3}$ | User Specified Target Ordering |
|-----------------|-------------------|-------------------|-------------------|--------------------------------|
| $Ordervalue(A)$ | | 1 | 1 | 1 |
| $Ordervalue(B)$ | 1 | 0 | 2 | 2 |
| $Ordervalue(C)$ | 2 | | | 3 |
| $Ordervalue(D)$ | | 1 | | 4 |
| $Ordervalue(E)$ | 0 | | | 5 |
| $Ordervalue(F)$ | | 0 | | 6 |
| $Ordervalue(M)$ | | | 0 | 7 |
| $Ordervalue(N)$ | | | 0 | 8 |

Table 4.2: Order values in candidate Bayesian networks and target ordering

Figure 4.1 shows an example of three candidate Bayesian networks to be combined. We assume there is a target ordering that specified by user, as shown in Table 4.2.

We notice that in *candidate* BN_1 , we have $ordervalue(B) > ordervalue(E)$

Algorithm 4.3 ArcReversal

Require: A number of Bayesian Networks B_1, \dots, B_k , ($k \geq 2$);
 $B_i = (V_i, E_i)$, where $i \in [1 \dots k]$;
OrderValue(v), where $v \in V$

- 1: **for** $i = 1$ to k **do**
- 2: Get NewOrder(v) for $v \in V_i$
- 3: Init a heap *ReverseArcSet*
- 4: **for** $e = \langle v_m, v_n \rangle \in E_i$ and $NewOrder(v_m) > NewOrder(v_n)$ **do**
- 5: *ReverseArcSet* $\leftarrow e$
- 6: **end for**
- 7: **end for**
- 8: **while** *ReverseArcSet* $\neq \phi$ **do**
- 9: pop $\min_{e=\langle v_m, v_n \rangle \in ReverseArcSet} e$ from *REV*
- 10: **for** $v \in Parent(v_m)$ **do**
- 11: **if** $!(v \in Parent(v_n))$ **then**
- 12: Add Arc $\langle v, v_n \rangle$ to B_i
- 13: **if** $NewOrder(v) > NewOrder(v_n)$ **then**
- 14: *ReverseArcSet* $\leftarrow \langle v, v_n \rangle$
- 15: **end if**
- 16: **end if**
- 17: **end for**
- 18: **for** $v \in Parent(v_n)$ **do**
- 19: **if** $!(v \in Parent(v_m))$ **then**
- 20: Add Arc $\langle v, v_m \rangle$ to B_i
- 21: **if** $NewOrder(v) > NewOrder(v_m)$ **then**
- 22: *ReverseArcSet* $\leftarrow \langle v, v_m \rangle$
- 23: **end if**
- 24: **end if**
- 25: **end for**
- 26: **end while**

and $ordervalue(C) > ordervalue(E)$; however, in the target ordering, there are $ordervalue_{target}(B) < ordervalue_{target}(E)$ and $ordervalue_{target}(C) < ordervalue_{target}(E)$. Therefore we need to reverse the arcs $\langle E, B \rangle, \langle E, C \rangle$ of candidate BN_1 in order to be consistent with target ordering. The same is applied to candidate BN_2 and candidate BN_3 . There are two arcs need to be reversed in candidate BN_2 : $\langle B, A \rangle, \langle F, D \rangle$; and two arcs need to be reversed in candidate BN_3 : $\langle M, A \rangle, \langle N, B \rangle$. Figure 4.4 shows the result after arc reversal in the three candidate Bayesian networks.

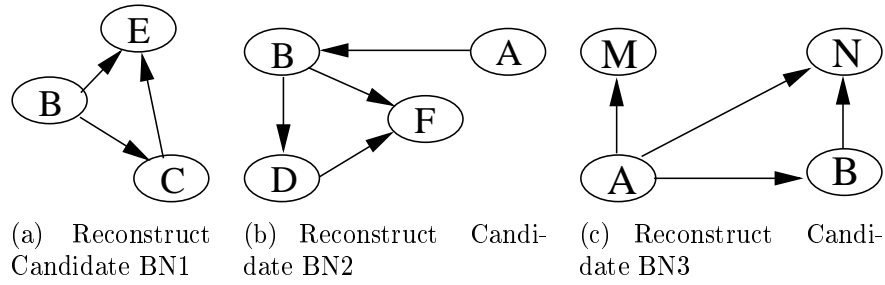


Figure 4.4: Reconstruction resulting of candidate BN using arc reversal

4.1.3 Intermediate Bayesian Networks

The first reason to adopt *Intermediate Bayesian Networks* is that we may preserve both structure and parameters of original candidate Bayesian networks. Thus all the changes or adjust acts are made on Intermediate Bayesian networks only. The definition of Intermediate Bayesian networks is as below.

Definition 4.10 Intermediate Bayesian Networks. *Given k candidate Bayesian networks to be combined, B_1, \dots, B_k . We copy both qualitative part and quantitative part of these candidate Bayesian networks, $B_{1.1}, \dots, B_{k.1}$, where (i), (ii), (iii) are all satisfied. (i) $|\{B_1, \dots, B_k\}| = |\{B_{1.1}, \dots, B_{k.1}\}|$; (ii) $(V_{1.1} = V_1) \wedge \dots \wedge (V_{k.1} = V_k) = \text{true}$; (iii) $(\vec{E}_{1.1} = \vec{E}_1) \wedge \dots \wedge (\vec{E}_{k.1} = \vec{E}_k) = \text{true}$.*

Given a specific target variable ordering, it is still not enough to derive the structure of models. For example, in Figure 4.4, although the variable ordering in every network is consistent with target ordering, their topologies are still different. $\langle A, N \rangle$ is in $BN_{Candidate3}$ after arc reversal but is NOT in $BN_{Candidate1}$.

In order to solve this problem, we propose the ideas of *virtual nodes* and *virtual arcs*, so that every candidate Bayesian network can have the same topology. An example of the *virtual node* concept is shown in Figure 4.5.

Definition 4.11 Virtual Node. *Given k Bayesian networks (B_1, \dots, B_k) to be combined. $\forall v_j \in V_j$, if $v_j \notin \bigcap_{i=1}^k V_i$, virtual node v_j is added to those networks in which $v_j \notin V_i$ without changing JPD of B_i and λ_i , so that $v_j \in \bigcap_{i=1}^k V_i$.*

With all nodes ready, we now introduce virtual arcs, as shown in Figure 4.6.

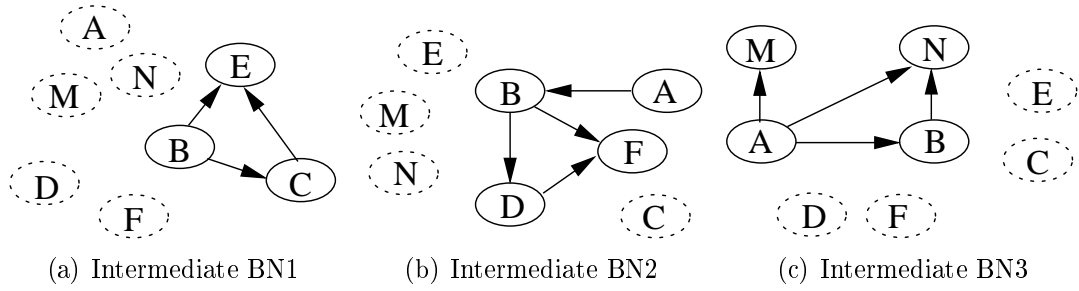


Figure 4.5: Example of virtual nodes

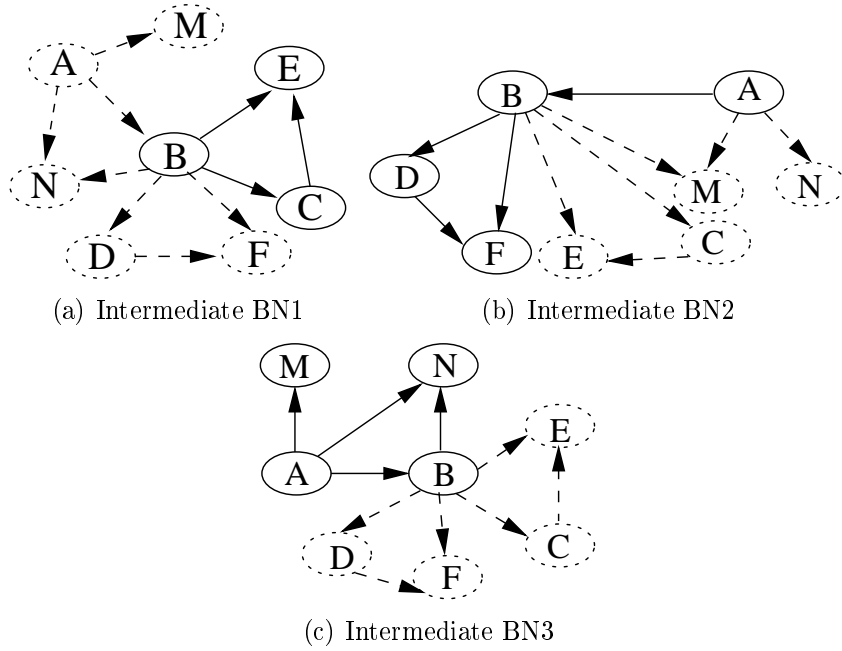


Figure 4.6: Example of virtual arcs

Definition 4.12 Virtual Arc. Given k Bayesian networks (B_1, \dots, B_k) to be combined. $\forall \vec{e}_j \in \vec{E}_j$, if $\vec{e}_j \notin \bigcap_{i=1}^k \vec{E}_i$, virtual edge \vec{e}_j is added to those networks in which $\vec{e}_j \notin \vec{E}_i$ without changing JPD of B_i and λ_i , so that $\vec{e}_j \in \bigcap_{i=1}^k \vec{E}_i$.

The computation of intermediate Bayesian networks with virtual nodes and virtual arcs will be discussed in next section.

4.2 Quantitative Combination of Bayesian Networks

From the Section 4.1, m candidate Bayesian networks to be combined, we can get m Intermediate Bayesian networks with consensus structure, thus completing the

qualitative combination procedure. In this way, the internal conditional probability tables for same node in each Intermediate Bayesian networks are of the same size. The only difference is in the probability distributions.

4.2.1 CPT Computation in Arc Reversal

The structure change after arc reversal operation may refer to Figure 4.3. The computation of arc reversal was discussed in [Shachter, 1984].

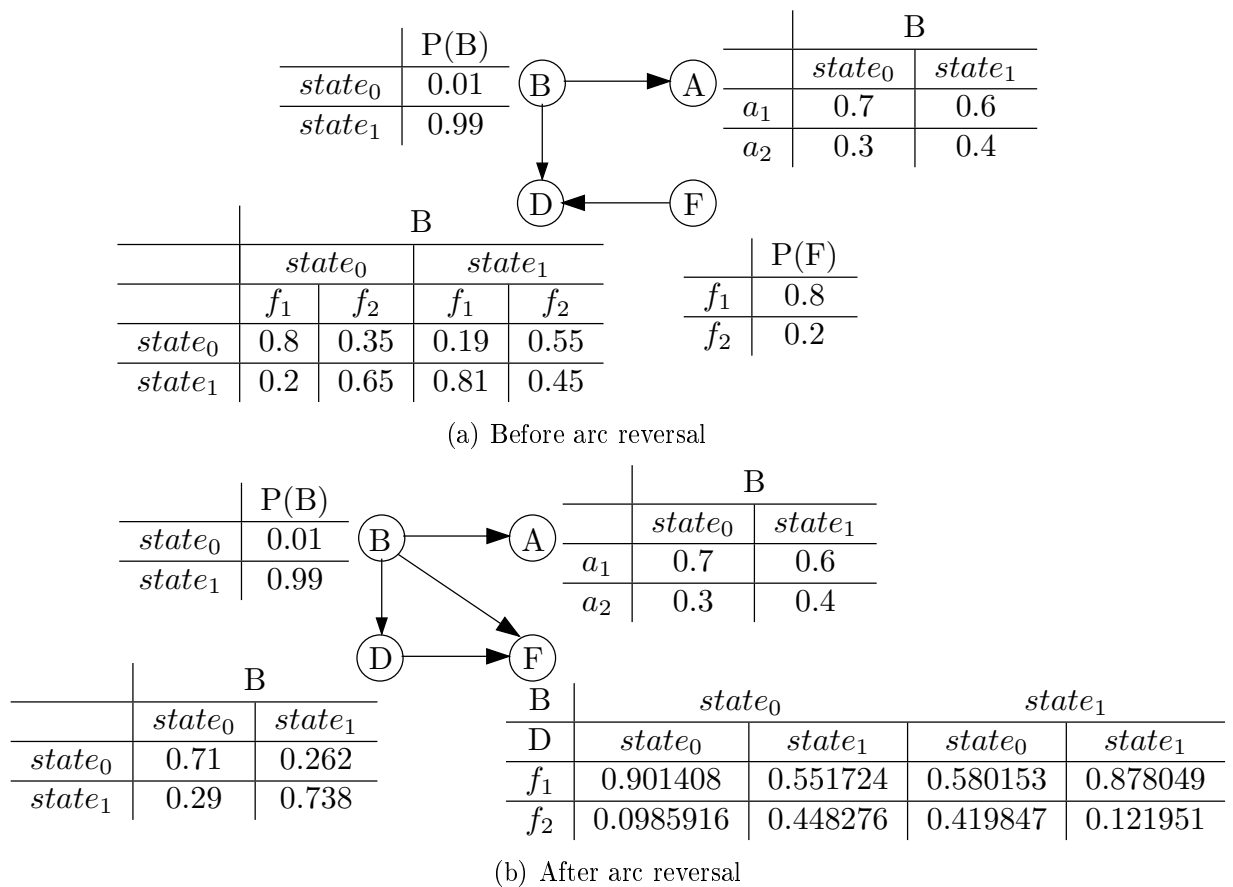


Figure 4.7: Example of arc reversal

In the following description, we denote the arc to be reversed as $\langle F, D \rangle$, i.e., before the arc reversal, the starting node is F and the ending node is D .

The computation begins from probability distributions over node D :

$$P(D|Pa_{new}D) \quad (4.12)$$

$$= P(D|\{Pa_{old}F \cup Pa_{old}D \setminus F\}) \quad (4.13)$$

$$= \sum_F P(D, F|F, B) \quad (4.14)$$

$$= \sum_F P(D, F|Pa_{new}D) \quad (4.15)$$

$$= \sum_F P(D|F, B)P(F|B) \quad (4.16)$$

$$= \sum_F P(D|F, B)P(F) \quad (4.17)$$

A computation example, which is corresponding to Figure 4.7, is as follows,

$$P(d|\neg b) = P(d|\neg b)P(f) \quad (4.18)$$

$$= P(d|\neg b, f)P(f) + P(d|\neg b, \neg f)P(\neg f) \quad (4.19)$$

$$= 0.19 * 0.8 + 0.55 * 0.2 \quad (4.20)$$

$$= 0.262 \quad (4.21)$$

After conditional probability distributions over B are obtained, the conditional probability distribution of F is now ready to be computed.

$$P(F|Pa_{new}F) \quad (4.22)$$

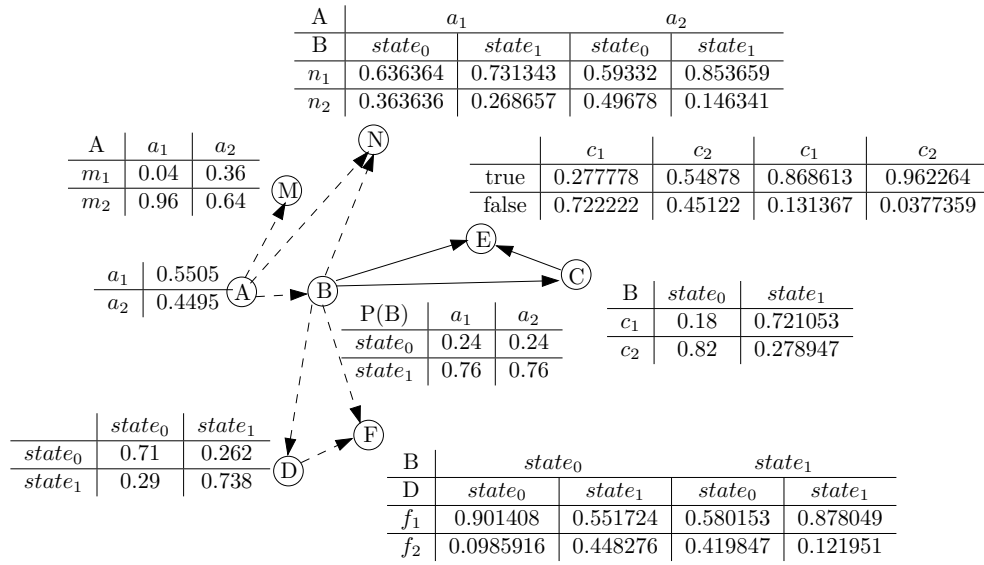
$$= P(F|\{Pa_{old}F \cup D \setminus F\}) \quad (4.23)$$

$$= \frac{P(D, F|B)}{P(D|B)} \quad (4.24)$$

$$= \frac{P(D|F, B) * P(F|B)}{P(D|B)} \quad (4.25)$$

$$= \frac{P(D|B, F)P(F)}{P(D|B)} \quad (4.26)$$

An real computation example is as follows,



(a) Intermediate BN1

Figure 4.8: Intermediate Bayesian network 1

$$P(f|\neg b, \neg d) = \frac{P(\neg d|\neg b, f)P(f)}{P(\neg d|\neg b)} \tag{4.27}$$

$$= \frac{0.81 * 0.8}{0.738} \tag{4.28}$$

$$= 0.878049 \tag{4.29}$$

4.2.2 CPT Combination

The three intermediate Bayesian networks after the qualitative combination, with detailed CPT, are shown in Figure 4.8 through 4.10 respectively.

4.2.2.1 Average or Weighted Combination

Average combination for m probability distributions p_1, \dots, p_m means that the combination result is the mean of these probability distributions.

$$p_{combined} = (\sum_{i=1}^m p_i) / m \tag{4.30}$$

The weighted combination of CPT is shown in Algorithm 4.4. *WeightBN* is

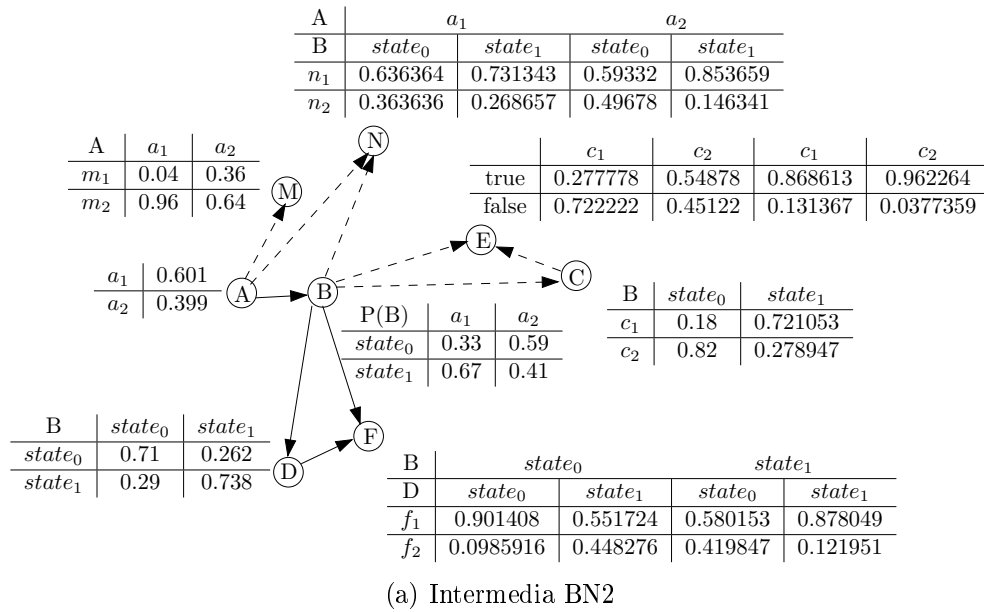


Figure 4.9: Intermediate Bayesian network 2

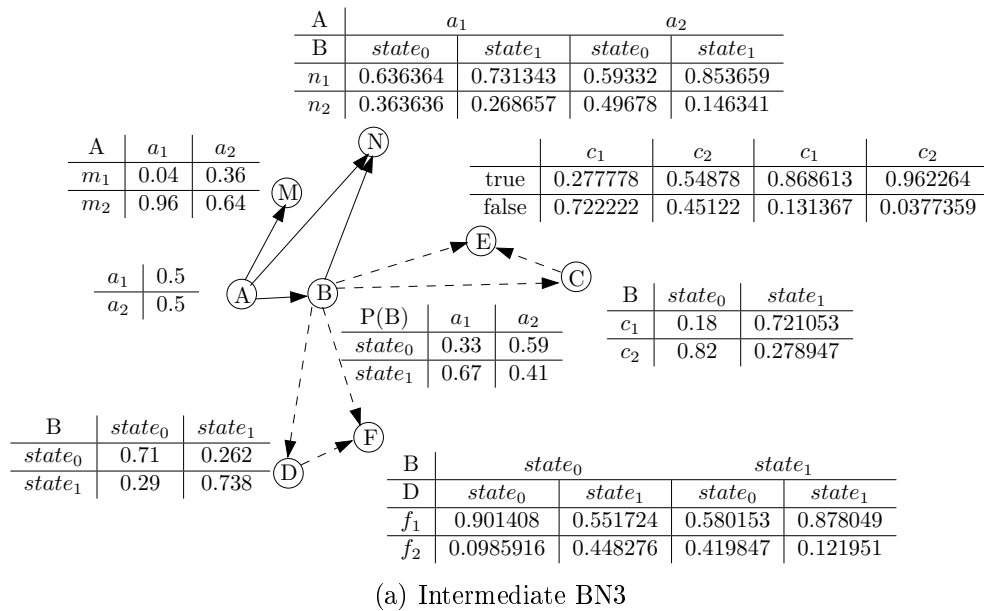


Figure 4.10: Intermediate Bayesian network 3

Algorithm 4.4 Computation of Weighted CPT Combination

Require: A number of Bayesian Networks $B_1, \dots, B_k, (k \geq 2)$;

$B_i = (V_i, E_i)$, where $i \in [1 \dots k]$;

Require: Target variable ordering λ_{result} ;

- 1: Using *WeightBN* to assign a weight value to each B_i , denote as w_i , such that $\sum w_i = 1$.
 - 2: Using algorithm *ArcReversal* to adjust each B_i to a new BN{i.e. Intermediate BN} $B_{i,1} = (V_{i,1}, E_{i,1})$ such that there is no conflict between $\lambda_{i,1}$ and λ_{result} and keep JPD consistency.
 - 3: Generate a new BN as the result BN $B_{result} = (V_{result}, E_{result})$ where $V_{result} = \bigcup_{i,1=1}^k V_{i,1}$ and $E_{result} = \bigcup_{i,1=1}^k E_{i,1}$
 - 4: **for** each variable v that $v \notin V_i$ and $v \in V_{i,1}$ **do**
 - 5: let $CPT(v) = \frac{\sum_{i,1=1}^k w_{i,1} \cdot CPT_{i,1}(v)}{\sum w_{i,1}}$ where $w_{i,1} = 0$ if $v \notin B_{i,1}$, else $w_{i,1} = w_i$.
 - 6: **end for**
-

a piece of programs that allow user to manually assign weight to each candidate Bayesian network. The weight assigned to each model can reflect the confidence of every candidate probabilistic graphic model.

As we can see, both average combination and weighted combination of probabilities are simple and conceptually straightforward methods, nevertheless they are robust and surprisingly powerful [Ng and Abramson, 1992].

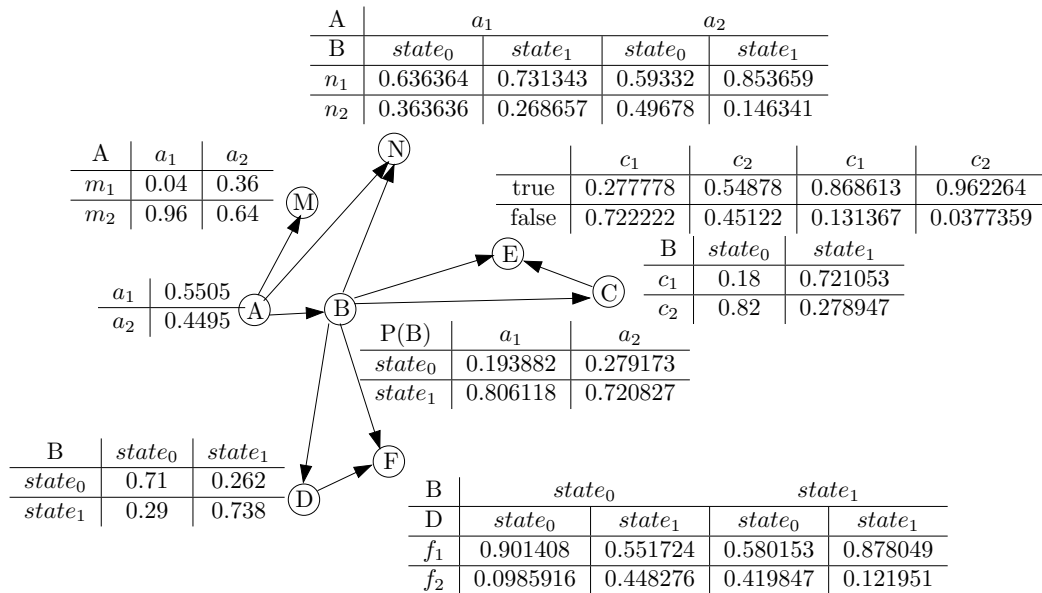


Figure 4.11: Resulting Bayesian network with weighted combination

The resulting Bayesian network from the three intermediate Bayesian networks is shown in Figure 4.11.

4.2.2.2 Interval Bayesian Networks

As the combined model can be saved as a reference for the user, it does not necessarily indeed precise probabilities. In fact, imprecise probabilities have been applied in a number of areas. In this section, we first introduce interval probabilities and then we discuss the concept of interval Bayesian networks.

For CPT combination problems, we may have two ways to approach it. The first one is to get CPT with point probabilities. Another approach is to get interval CPT, which means that CPTs are full of interval probabilities.

The concept of interval Bayesian networks is introduced by [Ha and Haddawy, 1996]. Interval Bayesian networks are a generalization of Bayesian networks where we allow the probabilities to be interval-values. An Interval Bayesian network represents a set of Bayesian networks, each of which is obtained from an intermediate Bayesian network by *instantiation*, a process that replaces each interval probability with a *consistent* point probability, where consistency means that the points must be in their corresponding intervals, and obey the axioms of probability.

For a certain event i , if we use $p(x_i)$ to denote the probability of event i , then we let $p_*(x_i)$ and $p^*(x_i)$ denote the *lower bound* and *upper bound*, for $p(x_i)$.

We now give a geometric interpretation of these probabilities and their bounds. The Cartesian product of the intervals $[p_*(x_i), p^*(x_i)], i = 1, \dots, n$, gives us a hyper-rectangle in the n -dimensional space R^n . The i th dimension of this space represents the value of $p(x_i)$. Within this box, only points where coordinates add up to 1 are probability distributions. Thus we have a convex set of probability distribution in R^n restricted by in equalities

$$p_*(x_i) \leq p(x_i) \leq p^*(x_i), i = 1, \dots, n \quad (4.31)$$

and

$$\sum_{i=1}^n p(x_i) = 1 \quad (4.32)$$

Algorithm 4.5 describe how to combine CPTs from intermediate Bayesian networks into a resulting interval Bayesian network.

Algorithm 4.5 Computation of Interval CPT Combination**Require:** A number of Bayesian Networks B_1, \dots, B_k , ($k \geq 2$); $B_i = (V_i, E_i)$, where $i \in [1 \dots k]$;**Require:** Target variable ordering λ_{result} ;

- 1: Using *WeightBN* to assign a weight values to each B_i , denote as w_i , such that $\sum w_i = 1$.
- 2: Using algorithm *ArcReversal* to adjust each B_i to a new BN{i.e. Intermediate BN} $B_{i.1} = (V_{i.1}, E_{i.1})$ such that there is no conflict between $\lambda_{i.1}$ and λ_{result} and keep JPD consistency.
- 3: Generate a new BN as the result BN $B_{result} = (V_{result}, E_{result})$ where $V_{result} = \bigcup_{i.1=1}^k V_{i.1}$ and $E_{result} = \bigcup_{i.1=1}^k E_{i.1}$
- 4: **for** each variable v that $v \notin V_i$ and $v \in V_{i.1}$ **do**
- 5: let $CPT(v) = (\min(CPT_{i.1}(v)), \max(CPT_{i.1}(v)))$.
- 6: **end for**

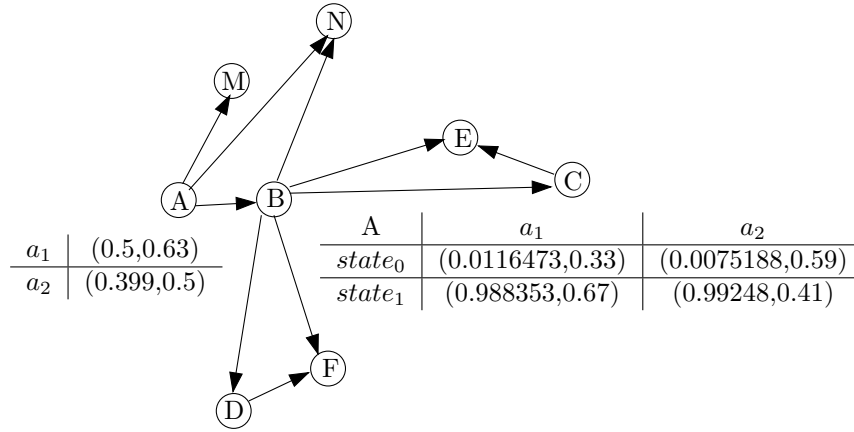


Figure 4.12: Example of resulting Interval Bayesian network

The resulting Bayesian network from the three intermediate Bayesian networks can be seen in Figure 4.11, where we only denote CPTs of node A and node B.

4.3 Heuristic Methods for Target Variable Ordering Generation

Different target variable ordering will result in different resulting models with different structure and parameters. In our research, we notice that there are some factors, which will influence the generation of target variable ordering:

- The original order values of each variable in the original candidate Bayesian

networks;

- The number of parents of each variable;
- The size of each candidate Bayesian network.

Based on the above factors, we present three heuristic methods for target variable ordering generation, which we will introduce in this section, along with an example of Bayesian network combination based on the three candidate Bayesian networks in Figure 4.1. Such heuristic methods aim to reduce the complexity of the resulting model and save the time of combination.

4.3.1 Target Ordering based on Original Order Values

Algorithm 4.6 Order value based Target Variable Ordering Generation in BNs Combination

Require: B_1, \dots, B_k , $k \geq 2$ and $B_i = (V, E)$, $i = 1 \dots k$

```

1: for  $i = 1$  to  $k$  do
2:    $\forall v$  in  $B_i$ , Store  $OrderValue(v)$ ;
3: end for
4:  $V_{result} = \cup_{i=1}^k V_i$  {push all nodes from candidate BNs into  $V_{result}$ };
5:  $NodesNum = |V_{result}|$ ;
6: initiate an array
7:  $AllNodes[NodesNum] = [NodeID, OrderValue, NewOrderValue]$ ;
8: for  $i = 1$  to  $NodesNum$  do
9:   for  $j = 1$  to  $k$  do
10:     $v = AllNode(i)$ ;
11:     $Sum_v = \sum_{j=1}^k OrderValue(v_j)$  ;{sum the node's ordervalue in all candidate BNs}
12:     $NodeAppear[i]++$ ;{count how many models that this node exist}
13:   end for
14: end for
15:  $AverageOrder[NodesNum] = \frac{Sum_v}{NodeAppear}$ ;
16: end for
17: sort  $AverageOrder[NodesNum]$  according to averagedordervalue and nodeID
   {for two nodes with same average ordervalue, sort according to nodeID }
18: Assign NewOrdervalue to each Node after sorting

```

Algorithm 4.6 describes a method of automated target variable ordering generation—*Order value based Target Variable Ordering Generation* method. In this algorithm,

we firstly compute the average value for each node from all candidate Bayesian networks. The target *ordervalue* of each node is assigned according to its average value of original *ordervalue* from all candidate Bayesian networks.

The lower average value that a node can get, the lower target *ordervalue* it will be assigned. In order to prove the correctness of this point, the following proof is provided.

Proof:

Given k Bayesian networks B_1, \dots, B_k , $k \geq 2$, $B_i = (V_i, \vec{E}_i)$, $a, b \in \bigcup_{i=1}^k B_i$, and we denote $a_i (i = 1, \dots, k)$ as node a in $B_i (i = 1, \dots, k)$. $b_i (i = 1, \dots, k)$ as node b appear in $B_i (i = 1, \dots, k)$. We assume that a_i appears in the same number n of Bayesian networks as b_i .

$$(1) \text{ When } \sum_{i=1}^k \text{Ordervalue}(a) = \sum_{i=1}^k \text{Ordervalue}(b), \\ \text{AverageOrdervalue}(a) = \frac{\sum_{i=1}^k \text{Ordervalue}(a)}{n} = \frac{\sum_{i=1}^k \text{Ordervalue}(b)}{n} = \text{AverageOrdervalue}(b).$$

Therefore only according to *Ordervalue*, we cannot decide the order between $\text{Ordervalue}_{\text{target}}(a)$ and $\text{Ordervalue}_{\text{target}}(b)$. (In Algorithm 4.6 we will continue to check the *nodeID* of a and b , if $\text{nodeID}(a) \geq \text{nodeID}(b)$, then we let $\text{Ordervalue}_{\text{target}}(a) \geq \text{Ordervalue}_{\text{target}}(b)$, and vice versa.

$$(2) \text{ When } \sum_{i=1}^k \text{Ordervalue}(a) \geq \sum_{i=1}^k \text{Ordervalue}(b), \\ \text{AverageOrdervalue}(a) = \frac{\sum_{i=1}^k \text{Ordervalue}(a)}{n} \geq \frac{\sum_{i=1}^k \text{Ordervalue}(b)}{n} = \text{AverageOrdervalue}(b).$$

if $\forall a_i, b_i$, $\text{Ordervalue}(a_i) \geq \text{Ordervalue}(b_i)$, no doubt we will let $\text{Ordervalue}_{\text{target}}(a) \geq \text{Ordervalue}_{\text{target}}(b)$.

But what will happen if $\exists a_i, b_i$,

$$[\text{Ordervalue}(a_i) \geq \text{Ordervalue}(b_i)] \wedge [\text{Ordervalue}(a_i) \leq \text{Ordervalue}(b_i)] ?$$

We assume for in $k - 1$ Bayesian networks, $\text{Ordervalue}(a_i) \leq \text{Ordervalue}(b_i)$, and only in 1 Bayesian networks, $\text{Ordervalue}(a_i) \geq \text{Ordervalue}(b_i)$, as listed below:

$$\text{Ordervalue}(a_1) \leq \text{Ordervalue}(b_1) \quad (4.33)$$

$$\text{Ordervalue}(a_2) \leq \text{Ordervalue}(b_2) \quad (4.34)$$

$$\dots \quad (4.35)$$

$$\text{Ordervalue}(a_{k-1}) \leq \text{Ordervalue}(b_{k-1}) \quad (4.36)$$

$$\text{Ordervalue}(a_k) \geq \text{Ordervalue}(b_k) \quad (4.37)$$

Given the inequations above, we let $|b_1 - a_1| = \Delta_1$, $|b_2 - a_2| = \Delta_2$, ..., $|b_{k-1} - a_{k-1}| = \Delta_{k-1}$, $|a_k - a_k| = \Delta_k$. In order to satisfy all the above inequations, we must have $\Delta_k \geq \Delta_1 + \Delta_2 + \dots + \Delta_k$. Suppose we let $\text{Ordervalue}_{\text{target}}(a) \leq \text{Ordervalue}_{\text{target}}(b)$, then at least Δ_k arcs need to be added in adjusting B_k to intermediate Bayesian networks $B_{k,1}$. This violates our principle of adding least number of arcs in the combination procedure. So we have to let $\text{Ordervalue}_{\text{target}}(a) \geq \text{Ordervalue}_{\text{target}}(b)$.

□

According to Algorithm 4.6, the generated target variable ordering of the three candidate Bayesian networks in Figure 4.1 is shown in Table 4.3. Note that $\text{TargetOrder}(A)$ is equal to $\text{TargetOrder}(D)$. In this situation, we can compare their node IDs stored in the network, or seek help from domain experts.

| Node | <i>CandidateBN</i> ₁ | <i>CandidateBN</i> ₂ | <i>CandidateBN</i> ₃ | <i>Average</i> | <i>Target Order</i> |
|----------------------|---------------------------------|---------------------------------|---------------------------------|----------------|---------------------|
| <i>Ordervalue(A)</i> | | 1 | 1 | 1 | 2* |
| <i>Ordervalue(B)</i> | 1 | 0 | 2 | $\frac{2}{3}$ | 1 |
| <i>Ordervalue(C)</i> | 2 | | | 2 | 3 |
| <i>Ordervalue(D)</i> | | 1 | | 1 | 2* |
| <i>Ordervalue(E)</i> | 0 | | | 0 | 0 |
| <i>Ordervalue(F)</i> | | 0 | | | |
| <i>Ordervalue(M)</i> | | | 0 | 0 | 0 |
| <i>Ordervalue(N)</i> | | | 0 | 0 | 0 |

Table 4.3: An example for order value based target variable ordering generation

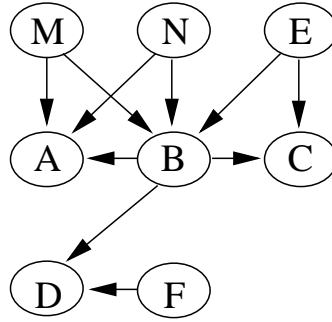


Figure 4.13: Example of resulting Bayesian networks according to order value based target variable ordering

4.3.2 Target Ordering based on Number of Parents and Network Size

Algorithm 4.7 Target Variable Ordering Generation based on Number of Parents and Network Size

Require: B_1, \dots, B_k , $k \geq 2$ and $B_i = (V, E)$, $i = 1 \dots k$

- 1: **for** $i = 1$ to k **do**
 - 2: $\forall v$ in B_i ,
 Store $NumParents_i(v)$;
 Store $NetSize_i(v)$; { $NetSize_i(v)$ denotes the size of network that node v is in };
 $tempValue_i[v] = NumParents_i(NodeNum) * NetSize_i(NodeNum)$;
 - 3: **end for**
 - 4: $V_{result} = \cup_{i=1}^k V_i$ { push all nodes from candidate BNs into V_{result} };
 - 5: $NodesNum = |V_{result}|$;
 - 6: initiate an array
 - 7: $AllNodes[NodeNum] = [NodeID, OrderValue, NewOrderValue]$;
 - 8: **for** $i = 1$ to $NodeNum$ **do**
 - 9: $TargetOrder[NodeNum] = \sum tempValue_i[NodeNum]$;
 - 10: **end for**
 - 11: sort $NodeNum$ according to $TargetOrder[NodeNum]$ and $nodeID$ { for two nodes with same $TargetOrder[NodeNum]$, sort according to $nodeID$ }
 - 12: Assign $NewOrderValue$ to each Node according to the position of each node after sorting
-

The key idea in Algorithm 4.7 is to generate target variable ordering according to linear computation over 1) the number of parent nodes of each variable in candidate Bayesian networks and 2) the number of nodes for each candidate Bayesian network.

We use the node A as an example to illustrate the computation procedure. The

weighted sum of node A is $1 * 4 + 1 * 4 = 8$, combining the value of number of parents in each candidate Bayesian network that node A appears and the size of each candidate Bayesian network that node A appears.

| Node | <i>CandidateBN</i> ₁ | <i>CandidateBN</i> ₂ | <i>CandidateBN</i> ₃ | <i>Weighted sum</i> | <i>Target Order</i> |
|----------------------|---------------------------------|---------------------------------|---------------------------------|---------------------|---------------------|
| <i>Num_Parent(A)</i> | | 1 | 1 | 8 | 2* |
| <i>Num_Parent(B)</i> | 1 | 0 | 2 | 11 | 3 |
| <i>Num_Parent(C)</i> | 2 | | | 6 | 1 |
| <i>Num_Parent(D)</i> | | 2 | | 8 | 2* |
| <i>Num_Parent(E)</i> | 0 | | | 0 | 0 |
| <i>Num_Parent(F)</i> | | 0 | | 0 | 0 |
| <i>Num_Parent(M)</i> | | | 0 | 0 | 0 |
| <i>Num_Parent(N)</i> | | | 0 | 0 | 0 |
| Num of Nodes | 3 | 4 | 4 | | |

Table 4.4: Example of target ordering based on number of parents & size of networks

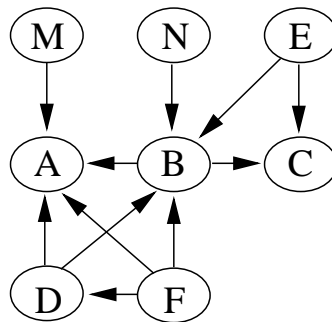


Figure 4.14: Example of resulting Bayesian network according to num of parents and network size

4.3.3 Target Ordering based on Edge Matrix

In this method, we consider the relative difference in order value between every pair of nodes in the candidate Bayesian networks. We construct the edge matrix, by storing the difference in order value for each variable in candidate Bayesian networks. Thus there are k edge matrices if there are k candidate Bayesian networks. To get the target variable ordering, we need to get a final edge matrix after computation over these k candidate Bayesian networks. According to the final edge matrix, we may get the relative difference in order value of between each pair of nodes in the resulting Bayesian networks.

| <i>Ending Node</i> \ <i>Starting Node</i> | <i>E</i> | <i>B</i> | <i>C</i> |
|---|----------|----------|----------|
| <i>E</i> | 0 | -1 | -2 |
| <i>B</i> | 1 | 0 | -1 |
| <i>C</i> | 2 | 1 | 0 |

(a) Candidate BN_1

| <i>Ending Node</i> \ <i>Starting Node</i> | <i>A</i> | <i>B</i> | <i>D</i> | <i>F</i> |
|---|----------|----------|----------|----------|
| <i>A</i> | 0 | 1 | 0 | 1 |
| <i>B</i> | -1 | 0 | -1 | 0 |
| <i>D</i> | 0 | 1 | 0 | 1 |
| <i>F</i> | -1 | 0 | -1 | 0 |

(b) Candidate BN_2

| <i>Ending Node</i> \ <i>Starting Node</i> | <i>A</i> | <i>B</i> | <i>M</i> | <i>N</i> |
|---|----------|----------|----------|----------|
| <i>A</i> | 0 | -1 | 1 | 1 |
| <i>B</i> | 1 | 0 | 2 | 1 |
| <i>M</i> | -1 | -2 | 0 | 1 |
| <i>N</i> | -1 | -1 | -1 | 0 |

(c) Candidate BN_3

Table 4.5: Example of edge matrices of candidate Bayesian networks

| <i>Ending Node</i> \ <i>Starting Node</i> | <i>A</i> | <i>B</i> | <i>C</i> | <i>D</i> | <i>E</i> | <i>F</i> | <i>M</i> | <i>N</i> |
|---|----------|----------|----------|----------|----------|----------|----------|----------|
| <i>A</i> | 0 | 0* | 0 | 0 | 0 | 1 | 1 | 1 |
| <i>B</i> | | 0 | -1 | -1 | 1 | 0 | 2 | 1 |
| <i>C</i> | | | 0 | 0 | 0 | 1 | 0 | 0 |
| <i>D</i> | | | | 0 | 0 | 0 | 0 | 0 |
| <i>E</i> | | | | | 0 | 0 | 0 | 0 |
| <i>F</i> | | | | | | 0 | 0 | 0 |
| <i>M</i> | | | | | | | 0 | 1 |
| <i>N</i> | | | | | | | | 0 |

Table 4.6: Resulting edge matrix according to edge matrix based target ordering algorithm

Table 4.5 shows the edge matrices from the three candidate Bayesian networks from Figure 4.1. Table 4.6 is the resulting edge matrix with simple addition from the three edge matrix from Figure 4.5.

The relative difference in Table 4.6 can also be shown in the following format:

$$Ordervalue_{target}(A) - Ordervalue_{target}(F) = 1 \quad (4.38)$$

$$Ordervalue_{target}(A) - Ordervalue_{target}(M) = 1 \quad (4.39)$$

$$Ordervalue_{target}(A) - Ordervalue_{target}(N) = 1 \quad (4.40)$$

$$Ordervalue_{target}(B) - Ordervalue_{target}(C) = -1 \quad (4.41)$$

$$Ordervalue_{target}(B) - Ordervalue_{target}(D) = -1 \quad (4.42)$$

$$Ordervalue_{target}(B) - Ordervalue_{target}(E) = 1 \quad (4.43)$$

$$Ordervalue_{target}(B) - Ordervalue_{target}(M) = 2 \quad (4.44)$$

$$Ordervalue_{target}(B) - Ordervalue_{target}(N) = 1 \quad (4.45)$$

$$Ordervalue_{target}(C) - Ordervalue_{target}(F) = 1 \quad (4.46)$$

$$Ordervalue_{target}(M) - Ordervalue_{target}(N) = 1 \quad (4.47)$$

In the resulting edge matrix, we may get a consistent list of relative order value difference between each pair of nodes in the resulting Bayesian network, so that we can sort the variables according to the resulting edge matrix. The target ordering generated from the resulting edge matrix are not guaranteed to be linear. In fact, in most situations, it is a hierarchical variable ordering. The target variable ordering corresponding to the resulting edge matrix can be seen in Figure 4.15. According to this target variable ordering, in *Candidate BN₃*, arcs need to be reversed include $\langle A, B \rangle$, $\langle N, B \rangle$. In fact, there are 3 arcs that are reversed in the process: $\langle A, B \rangle$, $\langle M, B \rangle$, $\langle N, B \rangle$. The resulting Bayesian network is shown in Figure 4.16.

$level_1:$ (N)
 $level_2:$ (F) (M)
 $level_3:$ (A) (E)
 $level_4:$ (B)
 $level_5:$ (C) (D)

Figure 4.15: Target variable ordering in the resulting edge matrix

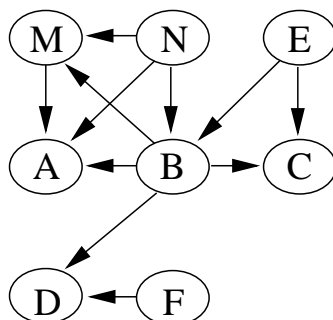


Figure 4.16: Resulting BN according to edge matrix based target ordering

4.4 Extension to Influence Diagram Combination

It might be difficult to make decisions based on different models. When we want to extend our objective from just knowledge representation to decision analysis, we extend our research issues from Bayesian networks to influence diagrams, in which *decision nodes* and *utility nodes* are added, and those nodes corresponding to BN variables are now called *chance nodes*. A prevailing and intuitive way is weighted combination. However, it is a big problem in how to decide the weight of significance for each model from different sources. We propose a new approach that determines the weights based on Expected Utility (EU). In this section, we first give a formal definition of an influence diagram, and then develop the notations needed to explain the influence diagram combination algorithm.

An influence diagram [Howard and E, 1984] is a network consisting of a directed graph $G = (N, A)$ and associated node sets and functions. It contains three types of nodes in the set N , partitioned into sets V , C and D . There is at most one value node $v \in V$, drawn as a rounded rectangle, which represents the objective to be maximized in expectation. There are zero or more chance nodes in the set C , shown

as circles, representing random variables (or uncertain quantities). Finally, there may be zero more decision nodes in the set D , drawn as squares, choices available to the decision maker. The arcs A in the graph have different meanings, based on the target. Arcs into utility and chance nodes are conditional and represent probabilistic dependence. They do not imply causality or time precedence. Arcs into decision nodes are informational and imply time precedence. Any uncertainties or decisions at the tails of such arcs have been resolved before the decision at the head of the arc to be made.

4.4.1 Three Types of Nodes in Influence Diagram

After an influence diagram is defined, a number of concepts and objects will be introduced in this section.

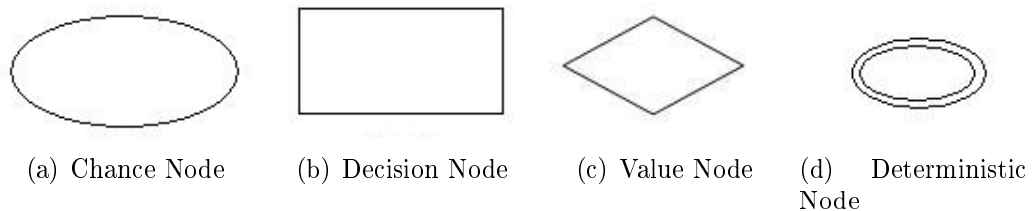


Figure 4.17: Various types of nodes in influence diagram

- *Chance Node*

A chance node contains a set of *mutually exclusive* and *collectively exhaustive* outcomes. Mutually exclusive means that only one of the outcomes can happen. Collectively exhaustive means that no other possibilities exist; one of the specified outcomes has to occur. In an investment decision, the outcomes can be either succeed or fail, but not both.

Chance node can be classified into two types of nodes: deterministic node and non-deterministic node.

A deterministic node is a special chance node that represents a variable whose value is a deterministic function of its parents. A deterministic function is an algebraic expression and differs from probabilistic function in that if the values of the

parents are known, the value of the deterministic function is known with certainty. A deterministic node is denoted by a double-oval.

- *Decision Node*

Decision nodes are denoted by square or rectangle nodes. A decision node is one that is within the control of the decision maker. A decision variable is represented by a square/rectangular node in an influence diagram. In each decision node, we store a list of possible alternatives associated with the decision variable in which only one of the alternatives can be chosen by decision maker. For example, in the capital investment decision, the decision maker can either invest or not, but not both.

Arcs into decision nodes indicate time precedence, which the information at the source of the arc is available at the time the decision is made. It is not possible to reverse an arc into or out of a decision node.

- *Value Node*

A *value node* is a node designated by the decision maker to be quantity whose certain equivalent is to be optimized (maximized). Every influence diagram representing a valid decision model must have a value node. It indicates the set of nodes whose outcomes have direct impact on the preference of the decision maker.

A value node contains a table that represents the utility of all possible outcomes. The parents of the value node indicate the events and decisions that affect the utility directly. The parents of the value node depend on the utility model used for the analysis. Only one value node is allowed in a standard influence diagram. It is a sink node, which means it must not have any out going arc; only incoming arcs are allowed.

4.4.2 Four Types of Arcs in Influence Diagram

Similar to Bayesian networks, influence diagrams do not allow *cycles*. In Bayesian networks, arcs always represent influence or causal relationships. Different from Bayesian networks, arcs in influence diagrams can represent either *relevance* or *sequence*. The context of the arcs indicates the meaning.

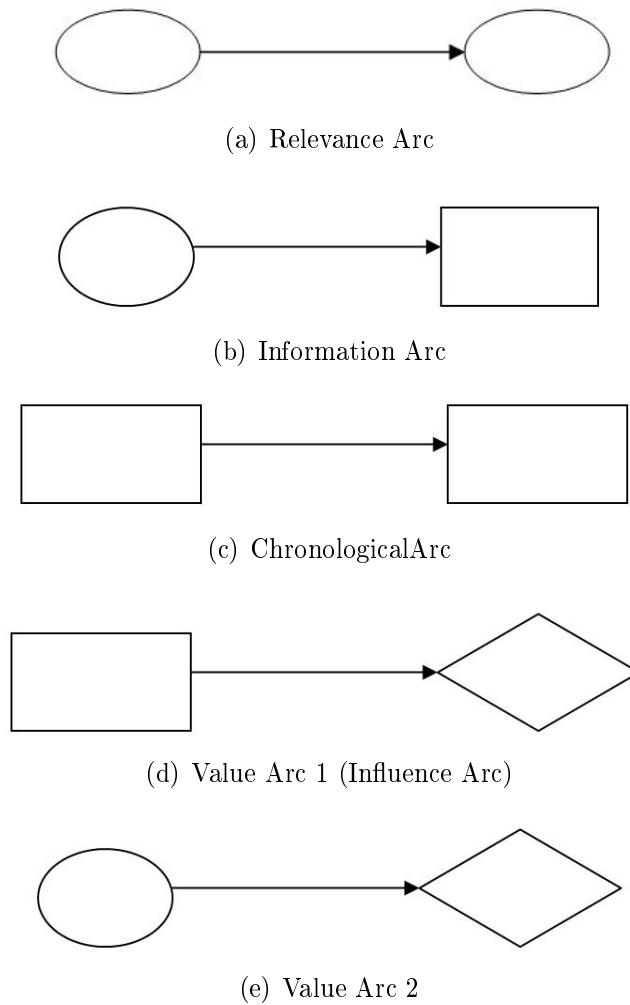


Figure 4.18: Four types of arcs

- *Relevance Arc*

A *relevance arc* is an arc from one chance node to another chance node. It indicates that there may be dependence, indicating that predecessor is relevant for assessing the chances associated with the uncertain event.

- *Information Arc*

An *information arc* (into a decision node) indicates that the information must be available at the time of the decision. Arcs that point to decision nodes represent information available at the time that the decision is made knowing the outcome of predecessor node.

Arc from a chance node into a decision node indicates that information about the outcome of the chance node is known exactly when the decision is being carried out; in other words, from decision maker's point of view, all uncertainty associated with a chance event is resolved and the outcome known when the decision is made.

Informational arcs cannot be reversed. An information arc indicates that the information must be available at the time of the decision.

- *Chronologic Arc*

An arc from one decision node to another decision node indicates the *chronological* order in which the decisions are being carried out.

- *Value Arc*

A *value arc* is an arc from an uncertain or decision node into the value node. It indicates that outcome or decision affects the value directly. An arc from a decision node to a chance node are called *influence arc*, indicating that the choice taken by the decision maker will affect (or influence the likelihood for the outcome of the uncertain variable). Influence arc denotes that the decision is relevant for assessing the chances associated with the event in the chance node.

For example, the chance that a person will become a manager depends to some extent on the choice of his university education.

The fundamental idea of decision theory is that *an agent is rational if and only if it chooses the action that yields the highest expected utility, averaged over all the possible outcomes of the action*. This is called the principle of Maximum Expected Utility (MEU) [Russell and Norvig, 2002].

4.4.3 Qualitative Combination with Constraints

After introducing the components of an influence diagram, we now talk about the creation of influence diagram. We can draw the influence diagram with variables ordered according to the time when they will be observed. One of the advantages of the influence diagram, however, is that it can be drawn with the variable ordered

such that probabilities can be assessed most easily. This order of conditioning is called the *assessment ordering*.

As influence diagram is a modeling tool for decision analysis, such orderings cannot be changed without assistance of the decision analyst or domain experts. Therefore, only ordering between a pair of chance node can be adjusted when there is an arc between this pair of chance node, i.e., arc reversal is only possible between two chance nodes in an influence diagram, which shares a common information state and have no other directed path between them.

The structure change is same as we described in Section 4.1, the two chance nodes must inherit each other's conditional predecessors before reversal of an arc between them. Bayes Theorem is invoked, and to determine the probability distribution after arc reversal.

Algorithm 4.8 Target Variable Ordering Generation in influence diagram

Require: A number of influence diagrams $ID_1, \dots, ID_k, (k \geq 2)$; $ID_i = (V_i, E_i)$, where $i \in [1 \dots k]$;

Require: Target variable ordering λ_{result} ;

- 1: Using *WeightBN* to assign a weight values to each ID_i , denote as w_i , such that $\sum w_i = 1$.
 - 2: Find the arcs need to be reversed, push them into *ArcList()*;
 - 3: **for** $i = 1$ to $|ArcList()|$ **do**
 - 4: **if** $ArcList(i).startNode.Nodetype = chanceNode$ **then**
 - 5: **if** $ArcList(i).endNode.Nodetype = chanceNode$ **then**
 - 6: Using algorithm *ArcReversal* to reverse $ArcList(i)$;
 - 7: **end if**
 - 8: **end if**
 - 9: **end for**
-

4.4.4 Quantitative Combination

Decision theory introduces a measure of preference, known as utility. Utility theory was proposed by Bernoulli [Bernoulli, 1738] to describe how people think and behave (e.g., when they make the St. Petersburg Paradox choice), but many of the subsequent versions (including von Neumann & Morgenstern's) were initially proposed as normative, optimally rational models that might or might not describe actual behavior. Utility is a function mapping the attributes of the possible out-

comes of a decision process to the set of real numbers. Utility is determined up to a linear transformation, i.e., a decision maker's preference over different decision alternatives is invariant to multiplying the utility by a non-negative number and adding a constant. This implies that utility has neither a meaningful zero point, nor a meaningful scale.

In fact, many social scientists outside of psychology believe that utility theory provides a good description of behavior [Edwards, 1992]. Utility is by assumption subjective: various decision makers facing the same choice may choose differently because of their different preference structure and different utility functions. A utility function for any decision problem needs to be obtained from a decision maker. The process of obtaining a utility function from a decision maker is known as utility elicitation.

Here are some useful definitions.

- *Certain Equivalent*: The amount of a commodity under conditions of certainty, which is deemed by the decision maker to be equivalent in value to a given lottery.
- *Lottery*: A set of uncertain outcomes paired with their respective probabilities.
- *Utility*: A subjective measure of value.
- *Utility Function*. We define the quantity $u_i (i = 1, \dots, k)$ as the utility of outcome A_i and the function that returns the values u_i given A_i as a utility function, i.e. $u(A_i) = u_i$.
- *Expected Utility*. The quantities $\sum_{i=1}^k p_i u(A_i)$ and $\sum_{i=1}^k q_i u(A_i)$ are known as the Expected Utilities for lotteries L_1 and L_2 respectively.
- *Maximum Expected Utility*. When there are more than 2 lotteries, the decision maker must prefer the one with Maximum Expected Utility, i.e.

$$\arg \max_1 \sum_{i=1}^k p_i^j u(A_i)$$

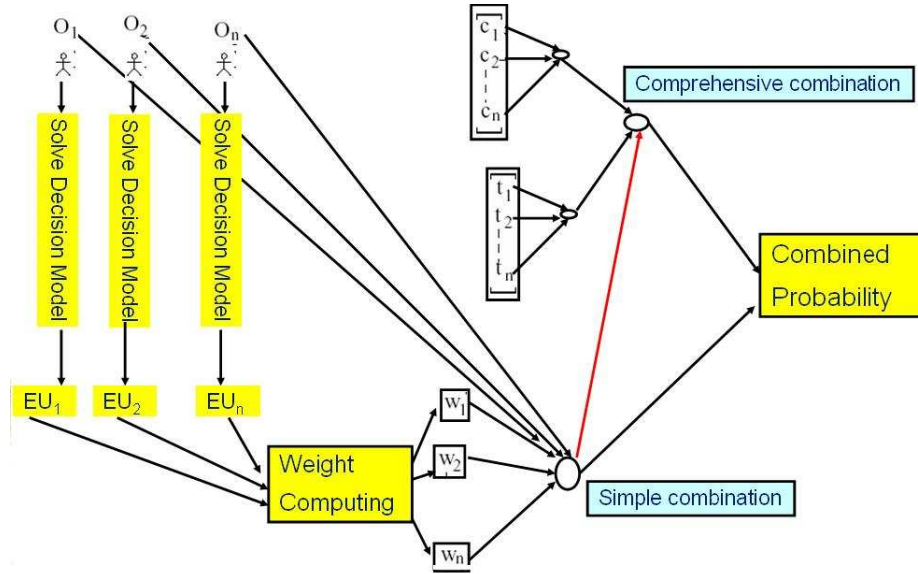


Figure 4.19: Utility based parameter combination method

4.4.4.1 Utility based Parameter Combination

From the series of definitions above, we can see that the values of expected utility and maximum expected utility depend on p_i . The basic idea of our method of utility-based weight computation is shown in Figure 4.19. Firstly, at each time we focus on only one knowledge source, solving the decision model and get the expected utility of each knowledge source, then we get the weight of each knowledge source according to their expected utility. The combination of parameters are weighted combination.

Given two different knowledge sources S_1 and S_2 , their parameter set $O_1 = (p_{11}, \dots, p_{1i})$ and $O_2 = (p_{21}, \dots, p_{2i})$. If p_{1k} is always larger than p_{2i} , the expected utility according to O_1 will be larger than the expected utility based on O_2 . So we interestingly find that the dependency between O_i and expected utility can reveal the attitude of each knowledge source.

Definition 4.13 Local Dominance. *Given more than one knowledge sources $S = \{S_1, \dots, S_m\}$, they provide their “opinions” O_i in the same quantity. Knowledge source S_A and S_B represent any two knowledge sources from the knowledge sources set S . Knowledge source S_A holds the opinion $O_A = \{o_{A1}, \dots, o_{Am}\}$ and knowledge source S_B holds the opinion $O_B = \{o_{B1}, \dots, o_{Bm}\}$. If o_{Ai} ($1 \leq i \leq m$) is larger*

than o_{B_i} ($1 \leq i \leq m$), we say that o_{A_i} dominate o_{B_i} and knowledge source S_A has the local dominance over knowledge source S_B in event (i.e., decision variable) i . It also means that knowledge source S_A is more optimistic than knowledge source S_B in variable i .

Definition 4.14 Global Dominance. Given more than one knowledge sources $S = \{S_1, \dots, S_m\}$ and their “opinions” O_i in the same quantity. Knowledge source S_A and knowledge source S_B represent any two knowledge sources from the knowledge sources set S . Knowledge source S_A holds the opinion $O_A = \{o_{A_1}, \dots, o_{A_m}\}$ and knowledge source S_B holds the opinion $O_B = \{o_{B_1}, \dots, o_{B_m}\}$ and the expected utilities from O_A , and O_B are EU_A and EU_B . If EU_A is larger than EU_B , we say that EU_A dominate EU_B and knowledge source S_A has the global dominance over knowledge source S_B based on their opinions O_A and O_B . It also means that knowledge source S_A is more optimistic than knowledge source S_B in general degree.

Local dominance can reveal the attitude of knowledge sources toward each decision variables while global dominance denotes the overall attitude of knowledge source.

The method of combination based our utility-based weight is described in Algorithm 4.9.

According to step 4 and step 5 of Algorithm 4.9, we can know that $S_{EU_{max}}$ is the most optimistic one while $S_{EU_{min}}$ is the most pessimistic one. After identification of the property of each knowledge source, next step is to decide the weights of each knowledge source. Because the extremeness of $S_{EU_{max}}$ and $S_{EU_{min}}$, in step 9 through step 13, we pick up the two knowledge sources in the two end of the ordered knowledge sources set and give less weight to both of these two knowledge sources.

Our utility-based weight method has the advantage of representing the possible lower bound and upper bound of possibility of all decision variables. In addition, it can help the decision maker to realize the optimistic or pessimistic property of each knowledge source.

Algorithm 4.9 Utility-based Weight Computation for each Knowledge source

Require: A number of knowledge sources $\mathcal{KS} = (S_1, \dots, S_m)$

Require: A number of probabilistic opinions from knowledge sources $\mathcal{O} = (O_1, \dots, O_m)$ where $O_k = (p_1, \dots, p_n), k = 1, \dots, m$ $\{O_k$ is the probability distribution from the knowledge source toward the probability that 's event happen.}

Require: A probabilistic decision model \mathcal{M}

```

1: for  $i = 1$  to  $m$  do
2:   Solve the decision model  $M$  with  $O_k$ ;
3:   Save  $EU_k$ ;
4:    $EU_{max} = \max(EU_k)$ ;
5:    $EU_{min} = \min(EU_k)$ ;
6:    $EU_{sum} = EU_{sum} + EU_k$ ;
7: end for
8: Sort  $\mathcal{KS}$  by  $EU_k$  in descending order;
9:  $EU_{extrem} = (EU_{max} + EU_{min})/2$ ;
10:  $EU_{inbetween} = EU_{sum} - EU_{max} - EU_{min}$ ;
11: for  $i = 1$  to  $k$  do
12:   if  $(i = 1)$  OR  $(i = k)$  then
13:      $W_i = \frac{EU_{extrem}}{EU_{extrem} + EU_{inbetween}} \times \frac{1}{2}$ ;
14:   else
15:      $W_i = \frac{EU_i}{EU_{extrem} + EU_{inbetween}}$ ;
16:   end if
17: end for

```

4.5 Implementation

Our research includes both general probabilistic graphic model combination and some extensive work in utility-based weighted parameter combination. The general probabilistic graphic model combination methods are implemented in our Probabilistic Graphic Model Combination (PGMC) system, which is developed using C++, based on SMILE¹ and SMILE.NET, which is the .NET implementation of SMILE. As a nice graphical interface for defining belief networks with discrete variables, GeNIe² is the tool that allow user to view or manually manipulate the probabilistic graphic model(s).

As Figure 4.20 presents, the PGMC system allows more than one probabilistic graphic model as inputs, and the output of the system is a resulting model. A

¹Standard Modeling, Inference and Learning Engine (SMILE) is a platform independent library of C++ classes for reasoning in graphical probabilistic models, such as Bayesian networks and influence diagrams.

²More information about the GeNIe and SMILE can refer to <http://www.sis.pitt.edu/~genie/>.

standard probabilistic graphic model can be saved in different format. The ideal formats of input probabilistic graphic models are those can be opened and visualized GeNIe, including *.dsl, *.xdsl.

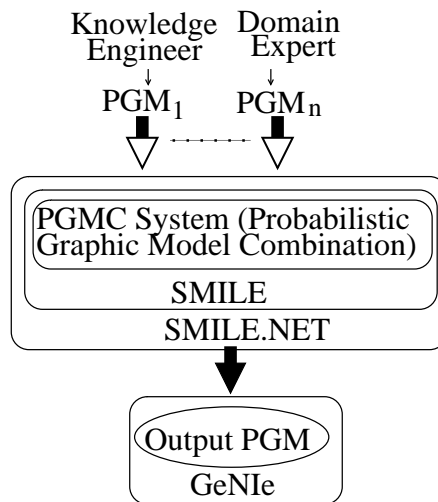


Figure 4.20: System overview

The implementation of utility-based weighted parameter combination is shown in Figure 4.21, which we call it as CoExperts Framework. This framework works together with some other previous research work together. In Figure 4.21, we assume that the decision maker is a clinical doctor, each knowledge source is an domain expert or specialist. The PROBE framework [Lau and Leong, 1999] is a existing framework for probabilities elicitation from experts and model solving. The DynaMoL [Leong and Cao, 1998] is a dynamic decision modeling language that can solve the resulting decision model from CoExperts Framework.

4.6 Complexity Analysis

This section analyzes the computational cost for general probabilistic model combination and the utility-based influence diagram combination.

As we discussed in section 4.4.3, the combination of influence diagrams has more constraints in structure than Bayesian network. Therefore, we only analyze the qualitative combination of Bayesian networks because influence diagram combination is much simpler than Bayesian network combination. The four steps of qual-

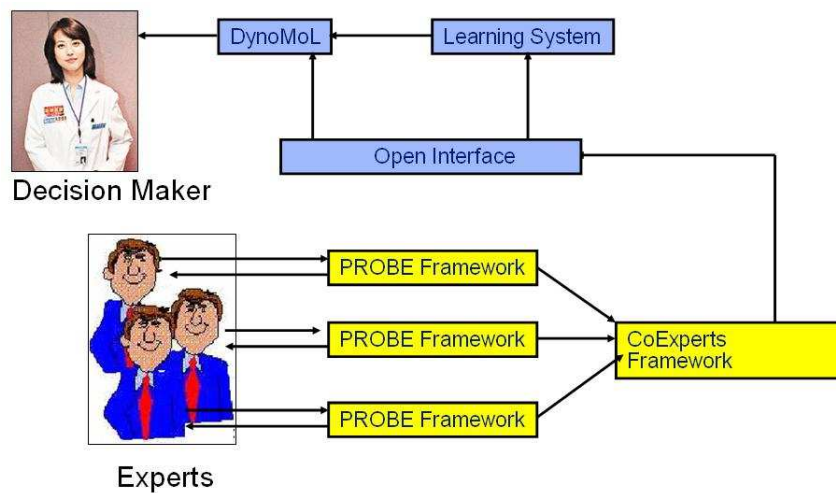


Figure 4.21: Utility-based weighted parameter combination

itative combination of Bayesian networks require three times of traverse operation over each candidate Bayesian network. We use the breadth-first search to traverse each Bayesian network, where the computation complexity is $O(|v| + |e|)$. As our target problem is to combine limited number of probabilistic graphic models, not any amount of models, even we make the traverse operation over every candidate Bayesian network, the complexity increase linearly as the number of candidate models increase.

The problem of getting the optimal variable ordering generation has been investigated and is found to be a NP-hard problem [Matzkevich and Abramson, 1993]. In our research, we use three heuristic methods to get the target variable ordering. The costs in these three heuristic methods only lie in the computation of traversing each candidate models.

Another special operation in Bayesian network combination is the arc reversal step. Under the situation of Bayesian network combination within limited number of variables, for each arc $\langle a, b \rangle$ in a BN model with m nodes, $\max(\text{NumParents}(a) + \text{NumParents}(b)) = m - 2$. Therefore, after arc reversal, the arcs to be added are at most $m - 2$. In the worst case, the CPT size of node a can increase from 2^0 to 2^{m-1} if all of the $m - 2$ nodes plus node b become parents of node a after arc reversal. The complexity of arc reversal operation is $O(2^{m-1})$.

In our utility-based weighted parameter combination method, the computation

cost comes from the procedure of getting the weight for each knowledge source. In order to get the weight of each knowledge source (for example, k knowledge sources), we solve the same model k times, with different parameters. As the model structure is fixed, we may assume that the overhead of solving the model one time is $O(S)$. Therefore the computation cost of our method is $O(kS)$.

Chapter 5

Case Study based Evaluation

5.1 Experimental Results on Bayesian Network Combination

This section introduces the experiments designed to show the feasibility and the performance of the four-step probabilistic graphic model combination algorithm, together with the three heuristic target variable ordering generation methods described in Chapter 4.

5.1.1 Introduction to Heart Disease Models

In our case study, a number of Bayesian network models are used in our experiments, which are from the *Heart Disease Project* in Medical Computing Lab in National University of Singapore. The purpose of this project is to understand the interactions of physical and genetic factors concerning a person's risk of developing Coronary Artery Disease (CAD).

These heart disease models are learnt using Probabilistic Networks Library (PNL) software ¹ (Of July 2004 release) from the HEART dataset. The HEART dataset has been used in [Tham et al., 2003] previously, to study the prediction of Coronary Artery Disease (CAD) using neural networks. The number of human subject profiles

¹PNL is a full function, free, open source, graphical models library released under a BSD style license, which is part of the collection of Intel's Open Source Libraries.

has since grown from 704 to 2,949 in total. Of these subjects 1,462, or 49.6%, were diagnosed to have coronary artery disease at the time of data collection, and the other subjects were healthy. Each medical profile of a subject consists of the CAD status of the subject (0 for healthy and 1 for diseased), 10 non-genetic risk factors, and 30 candidate gene markers, or gene polymorphisms. There are no missing values in this data set.

The 10 non-genetic, or physical risk factors, and their possible outcomes are shown in Table 5.1.

| | |
|---|-----------------|
| 1. SM-smoker | (a) 1-none |
| | (b) 2-ex-smoker |
| | (c) 3-smoker |
| 2. HY-hypertension | (a) 1 - no |
| | (b) 2 - yes |
| 3. DM - diabetic | (a) 1 - no |
| | (b) 2 - yes |
| 4. FHY - family history of hypertension (up to grandparent) | (a) 1 - no |
| | (b) 2 - yes |
| 5. FDM - family history of diabetic | (a) 1 - no |
| | (b) 2 - yes |
| 6. FCAD - family history of cardiac disease | (a) 1 - no |
| | (b) 2 - yes |
| 7. RACE | (a) 1 - Chinese |
| | (b) 2 - Malay |
| | (c) 3 - Indian |
| | (d) 4 - others |
| 8. SEX | (a) 1 - male |
| | (b) 2 - female |
| 9. AGE | |
| 10. CBMI - body mass index | |

Table 5.1: Ten Non-genetic factors

The age and BMI are discretized according to intervals as following,

- age: 0 ~ 10, 11 ~ 20, 21 ~ 30 and so on, the last group is 81 ~ 90.
- BMI: less than 15, 15 ~ 20, 20 ~ 25, 25 ~ 30, 30 ~ 35, 35 ~ 40, > 40

In addition to these physical attributes, 30 genetic attributes are also recorded. The names of the genetic attributes are G1 to G30 (without G20, G24). Each

attribute denotes a candidate gene marker that may affect the patient's chance of developing CAD. The possible outcomes of these gene markers denote the genotypes, or polymorphisms at that particular gene, often a substitution of a carbon base (e.g. T substituting C) at certain positions of the gene. In this data set, each gene marker has 3 possible polymorphisms.

We use these models in our case study because all of these models are learnt from the same data set (i.e. Heart Disease dataset). Therefore, the BN learning results from different approaches or different BN software can satisfy the requirements of our graphic model combination approach very well: 1) each variable with the same name among different models denotes same meaning; 2) the states of variables with the same name is identical in different networks; 3) the structure of these models are different.

We have over 30 Bayesian networks. The first experimental set contains three candidate Bayesian networks, each of which is with 5 node, three candidate Bayesian networks of 6-nodes which are in Figure C.4, three Bayesian networks of 7 nodes, 8 nodes, 10 nodes, 12 nodes that are in Figure C.7, Figure C.10, Figure C.11 and Figure C.12 respectively.

5.1.2 Experimental Setting and Measurement Criteria

The experiments were run on a PC with Windows 2000, AMD Athlon Processor 1.2G, 320M RAM.

The measurement criteria in our case study include:

- Number of arcs added in the resulting Bayesian network.
- Running time of combination, excluding the time of target variable ordering generation. The measurement unit of running time is counted in second(s).
- Number of arcs that are reversed in the combination procedure.

The candidate Bayesian networks are generated randomly.

For simplicity, we use Method 1, Method 2, and Method 3 to represent the proposed target variable ordering generation methods.

- Method 1: target ordering based on original target order value;
- Method 2: target ordering based on number of parents and network size;
- Method 3: target ordering based on edge matrix.

5.1.3 Comparison of Three Target Orderings Generation Methods

As we described in chapter 3, provided the same candidate Bayesian networks to be combined, the resulting Bayesian networks may be different according to different target ordering of variables in the resulting Bayesian networks. Here we would like to use a number of experiments to investigate how different target ordering of variables in resulting Bayesian networks influence the combination result. In the following tables, $TO = Target Ordering$, $CBN = Candidate Bayesian Network$.

Our first experiment is the combination over five candidate Bayesian networks, among which each Bayesian network is in the size of 5 variables, as shown in Figure C.1.

| | $CBN_{5.1}$ | $CBN_{5.2}$ | $CBN_{5.3}$ | $TO_{method1}$ |
|---------------|-------------|-------------|-------------|----------------|
| $Order(CAD)$ | 1 | 2 | 0 | 2 |
| $Order(Race)$ | 2 | 1 | 3 | 3 |
| $Order(CBMI)$ | 0 | | | 0 |
| $Order(SEX)$ | | 0 | 1 | 1 |
| $Order(HY)$ | | 3 | | 3 |
| $Order(SM)$ | | | 2 | 2 |
| $Order(G7)$ | | | 0 | 0 |
| $Order(G13)$ | 0 | | | 0 |
| $Order(G26)$ | 0 | | | 0 |
| $Order(G30)$ | | 2 | | 3 |

(a) Variable ordering generation with method 1

| | $CBN_{5.1}$ | $CBN_{5.2}$ | $CBN_{5.3}$ | $TO_{method2}$ |
|-------------------|-------------|-------------|-------------|----------------|
| $NumParent(CAD)$ | 1 | 2 | 0 | 3 |
| $NumParent(Race)$ | 2 | 1 | 3 | 4 |
| $NumParent(CBMI)$ | 0 | | | 0 |
| $NumParent(SEX)$ | | 0 | 1 | 1 |
| $NumParent(HY)$ | | 3 | | 2 |
| $NumParent(SM)$ | | | 2 | 2 |
| $NumParent(G7)$ | | | 0 | 0 |
| $NumParent(G13)$ | 0 | | | 0 |
| $NumParent(G26)$ | 0 | | | 0 |
| $NumParent(G30)$ | | 2 | | 1 |

(b) Variable ordering generation with method 2

Table 5.2: Variable ordering in combining three 5-node BNs with method 1 and method 2

| <i>End \ Start</i> | <i>CAD</i> | <i>Race</i> | <i>CBMI</i> | <i>G13</i> | <i>G26</i> | <i>End \ Start</i> | <i>CAD</i> | <i>Race</i> | <i>SEX</i> | <i>HY</i> | <i>G30</i> |
|--------------------|------------|-------------|-------------|------------|------------|--------------------|------------|-------------|------------|-----------|------------|
| <i>CAD</i> | 0 | 1 | -1 | -1 | -1 | <i>CAD</i> | 0 | -1 | -2 | 1 | 0 |
| <i>Race</i> | | 0 | -2 | -2 | -2 | <i>Race</i> | | 0 | -1 | 2 | -1 |
| <i>CBMI</i> | | | 0 | 0 | 0 | <i>SEX</i> | | | 0 | 3 | 2 |
| <i>G13</i> | | | | 0 | 0 | <i>HY</i> | | | | 0 | -1 |
| <i>G26</i> | | | | | 0 | <i>G30</i> | | | | | 0 |

(a) Edge matrix of $CBN_{5,1}$ (b) Edge matrix of $CBN_{5,2}$

| <i>End \ Start</i> | <i>CAD</i> | <i>Race</i> | <i>SEX</i> | <i>SM</i> | <i>G7</i> |
|--------------------|------------|-------------|------------|-----------|-----------|
| <i>CAD</i> | 0 | 3 | 1 | 2 | 0 |
| <i>Race</i> | | 0 | -2 | -1 | -3 |
| <i>SEX</i> | | | 0 | 1 | -1 |
| <i>SM</i> | | | | 0 | -2 |
| <i>G7</i> | | | | | 0 |

(c) Edge matrix of $CBN_{5,3}$

| <i>End \ Start</i> | <i>CAD</i> | <i>Race</i> | <i>CBMI</i> | <i>SEX</i> | <i>HY</i> | <i>SM</i> | <i>G7</i> | <i>G13</i> | <i>G26</i> | <i>G30</i> |
|--------------------|------------|-------------|-------------|------------|-----------|-----------|-----------|------------|------------|------------|
| <i>CAD</i> | 0 | 3 | -1 | -1 | 1 | 2 | 0 | -1 | -1 | 0 |
| <i>Race</i> | | 0 | -2 | -3 | 2 | -1 | -3 | -2 | -2 | -1 |
| <i>CBMI</i> | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| <i>SEX</i> | | | | 0 | 3 | 1 | -1 | 0 | 0 | 2 |
| <i>HY</i> | | | | | 0 | 0 | 0 | 0 | 0 | -1 |
| <i>SM</i> | | | | | | 0 | -2 | 0 | 0 | 0 |
| <i>G7</i> | | | | | | | 0 | 0 | 0 | 0 |
| <i>G13</i> | | | | | | | | 0 | 0 | 0 |
| <i>G26</i> | | | | | | | | | 0 | 0 |
| <i>G30</i> | | | | | | | | | | 0 |

(d) Edge matrix of resulting Bayesian network

| | <i>CAD</i> | <i>Race</i> | <i>CBMI</i> | <i>SEX</i> | <i>HY</i> | <i>SM</i> | <i>G7</i> | <i>G13</i> | <i>G26</i> | <i>G30</i> |
|-----------------------------|------------|-------------|-------------|------------|-----------|-----------|-----------|------------|------------|------------|
| <i>TO_{method3}</i> | 2 | 4 | 1 | 1 | 5 | 3 | 0 | 1 | 1 | 0 |

(e) Target variable ordering with method 3

Table 5.3: Variable ordering in 5-node candidate BNs with method 3

$$\text{Ordervalue}_{\text{target}}(\text{Race}) - \text{Ordervalue}_{\text{target}}(\text{CAD}) = 3 \quad (5.1)$$

$$\text{Ordervalue}_{\text{target}}(\text{CBMI}) - \text{Ordervalue}_{\text{target}}(\text{CAD}) = -1 \quad (5.2)$$

$$\text{Ordervalue}_{\text{target}}(\text{SEX}) - \text{Ordervalue}_{\text{target}}(\text{CAD}) = -1 \quad (5.3)$$

$$\text{Ordervalue}_{\text{target}}(\text{HY}) - \text{Ordervalue}_{\text{target}}(\text{CAD}) = 1 \quad (5.4)$$

$$\text{Ordervalue}_{\text{target}}(\text{SM}) - \text{Ordervalue}_{\text{target}}(\text{CAD}) = 2 \quad (5.5)$$

$$\text{Ordervalue}_{\text{target}}(\text{G13}) - \text{Ordervalue}_{\text{target}}(\text{CAD}) = -1 \quad (5.6)$$

$$\text{Ordervalue}_{\text{target}}(\text{G26}) - \text{Ordervalue}_{\text{target}}(\text{CAD}) = -1 \quad (5.7)$$

$$\text{Ordervalue}_{\text{target}}(\text{CBMI}) - \text{Ordervalue}_{\text{target}}(\text{RACE}) = -2 \quad (5.8)$$

$$\text{Ordervalue}_{\text{target}}(\text{SEX}) - \text{Ordervalue}_{\text{target}}(\text{RACE}) = -3 \quad (5.9)$$

$$\text{Ordervalue}_{\text{target}}(\text{HY}) - \text{Ordervalue}_{\text{target}}(\text{RACE}) = 2 \quad (5.10)$$

$$\text{Ordervalue}_{\text{target}}(\text{SM}) - \text{Ordervalue}_{\text{target}}(\text{RACE}) = -1 \quad (5.11)$$

$$\text{Ordervalue}_{\text{target}}(\text{G7}) - \text{Ordervalue}_{\text{target}}(\text{RACE}) = -3 \quad (5.12)$$

$$\text{Ordervalue}_{\text{target}}(\text{G13}) - \text{Ordervalue}_{\text{target}}(\text{RACE}) = -2 \quad (5.13)$$

$$\text{Ordervalue}_{\text{target}}(\text{G26}) - \text{Ordervalue}_{\text{target}}(\text{RACE}) = -2 \quad (5.14)$$

$$\text{Ordervalue}_{\text{target}}(\text{G30}) - \text{Ordervalue}_{\text{target}}(\text{RACE}) = -1 \quad (5.15)$$

$$\text{Ordervalue}_{\text{target}}(\text{HY}) - \text{Ordervalue}_{\text{target}}(\text{SEX}) = 3 \quad (5.16)$$

$$\text{Ordervalue}_{\text{target}}(\text{SM}) - \text{Ordervalue}_{\text{target}}(\text{SEX}) = 1 \quad (5.17)$$

$$\text{Ordervalue}_{\text{target}}(\text{G7}) - \text{Ordervalue}_{\text{target}}(\text{SEX}) = -1 \quad (5.18)$$

$$\text{Ordervalue}_{\text{target}}(\text{G30}) - \text{Ordervalue}_{\text{target}}(\text{SEX}) = 2 \quad (5.19)$$

$$\text{Ordervalue}_{\text{target}}(\text{G30}) - \text{Ordervalue}_{\text{target}}(\text{HY}) = -1 \quad (5.20)$$

$$\text{Ordervalue}_{\text{target}}(\text{G7}) - \text{Ordervalue}_{\text{target}}(\text{SM}) = -2 \quad (5.21)$$

Using method 1, only one arc $\langle Race, CAD \rangle$ need to be reversed in $CBN_{5.2}$ and only one arc $\langle CAD, SEX \rangle$ need to be reversed in $CBN_{5.3}$. The combination time is 0.10 seconds.

Using method 2, arcs need to be reversed in $CBN_{5.2}$ include $\{\langle RACE, CAD \rangle, \langle RACE, G30 \rangle, \langle CAD, HY \rangle\}$, in which in fact $\{\langle CAD, G30 \rangle, \langle HY, G30 \rangle\}$ are reversed too. Arcs need to be reversed in $CBN_{5.3}$ include $\{\langle CAD, SEX \rangle, \langle CAD, SM \rangle\}$. The combination time is 0.711 seconds.

The combination time of three 5-node Candidate Bayesian networks is 0.651 seconds with method 3. In $CBN_{5.2}$, two arcs are required to be reversed: $\{\langle RACE, CAD \rangle, \langle RACE, G30 \rangle\}$, which actually incur another two arcs to be reversed: $\{\langle SEX, G30 \rangle, \langle CAD, G30 \rangle\}$. In $CBN_{5.3}$ there are one arc to be reversed: $\langle CAD, SEX \rangle$.

We also use a random target variable ordering to test how the resulting model can be. The random target variable orderig is $\{HY = 0, G30 = 0, RACE = 1, SM = 2, CAD = 2, CBMI = 3, SEX = 4, G26 = 5, G13 = 6, G7 = 7\}$. Under such a target ordering, in $CBN_{5.1}$, there are 7 arcs to be reversed: $\{\langle CBMI, CAD \rangle, \langle G13, RACE \rangle, \langle G26, CAD \rangle, \langle CBMI, RACE \rangle, \langle G26, RACE \rangle, \langle CAD, RACE \rangle, \langle RACE, HY \rangle\}$; in $CBN_{5.2}$, there are 5 arcs to be reversed: $\{\langle SEX, SM \rangle, \langle SM, RACE \rangle, \langle G7, RACE \rangle, \langle CAD, RACE \rangle, \langle G7, SM \rangle\}$. It looks a bit strange that the combination time under this random target variable ordering is only 0.5 seconds. It may because only arcs in $CBN_{5.1}$ and $CBN_{5.2}$ are reversed. In $CBN_{5.1}$ and $CBN_{5.2}$, the variable sets of $\{G26, CBMI, RACE, CAD\}$ and $\{SEX, RACE, CAD, HY\}$ are almost fully connected already. Therefore the arc reversal operation will need less computation. However, Method 2 and Method 3 both require arc reversal operated on $CBN_{5.3}$.

In the degree of broken of original conditional independence, any of our heuristic method have better performance than the random target variable ordering.

As we can see from Table 5.7, the performance of method 2 is worst. Size of the network is considered in method 2, however, the three candidate Bayesian networks are with equal size (each with 5 nodes), which should be the reason of why method 2 perform worst. Therefore, we cannot draw the conclusion that method 2 is the worst

| | Num_ArcReversed | Num_ArcAdded | Combination Time |
|------------------------|-----------------|--------------|------------------|
| Method 1 | 2 | -3 | 0.10 |
| Method 2 | 7 | 0 | 0.711 |
| Method 3 | 5 | -1 | 0.651 |
| Random Target Ordering | 12 | 4 | 0.5 |

Table 5.4: Combination using 3 methods in three 5-node BN combination

method yet. In addition, we will try another experiment on some BN combination over candidate BNs with different network size.

Our second experiment is the combination based on three candidate Bayesian networks, of which each network is of 6-node size.

Using method 1 over combination of 6-node Bayesian networks, there is one arc ($\langle CBMI, CAD \rangle$), three arcs ($\langle RACE, CAD \rangle$, $\langle RACE, G30 \rangle$, $\langle SEX, CAD \rangle$) and two arcs ($\langle CBMI, SEX \rangle$, $\langle SM, RACE \rangle$) need to be reversed in $CBN_{6.1}$, $CBN_{6.2}$ and $CBN_{6.3}$ respectively. The combination time is 1.953 seconds.

| | $CBN_{6.1}$ | $CBN_{6.2}$ | $CBN_{6.3}$ | $TO_{method1}$ |
|--------------------|-------------|-------------|-------------|----------------|
| $ordervalue(CAD)$ | 1 | 2 | 0 | 1 |
| $ordervalue(Race)$ | 2 | 1 | 5 | 4 |
| $ordervalue(CBMI)$ | 0 | | 2 | 3 |
| $ordervalue(SEX)$ | | 0 | 3 | 2 |
| $ordervalue(HY)$ | | 3 | | 5 |
| $ordervalue(SM)$ | | | 4 | 6 |
| $ordervalue(Age)$ | | | 1 | 1 |
| $ordervalue(G13)$ | 0 | | | 0 |
| $ordervalue(G14)$ | 1 | | | 1 |
| $ordervalue(G26)$ | 0 | 0 | | 0 |
| $ordervalue(G30)$ | | 2 | | 3 |

(a) Variable ordering generation with method 1

| | $CBN_{6.1}$ | $CBN_{6.2}$ | $CBN_{6.3}$ | $TO_{method2}$ |
|-------------------|-------------|-------------|-------------|----------------|
| $NumParent(CAD)$ | 2 | 3 | 0 | 4 |
| $NumParent(Race)$ | 5 | 2 | 4 | 5 |
| $NumParent(CBMI)$ | 0 | | 1 | 1 |
| $NumParent(SEX)$ | | 0 | 3 | 3 |
| $NumParent(HY)$ | | 2 | | 2 |
| $NumParent(SM)$ | | | 2 | 2 |
| $NumParent(Age)$ | | | 1 | 1 |
| $NumParent(G13)$ | 0 | | | 0 |
| $NumParent(G14)$ | 1 | | | 1 |
| $NumParent(G26)$ | 0 | 0 | | 0 |
| $NumParent(G30)$ | | 1 | | 1 |

(b) Variable ordering generation with method 2

Table 5.5: Variable ordering in 6-node candidate Bayesian networks

| <i>End \ Start</i> | <i>CAD</i> | <i>Race</i> | <i>CBMI</i> | <i>G13</i> | <i>G14</i> | <i>G26</i> |
|--------------------|------------|-------------|-------------|------------|------------|------------|
| <i>CAD</i> | 0 | 1 | -1 | -1 | 0 | -1 |
| <i>Race</i> | | 0 | -2 | -2 | -1 | -2 |
| <i>CBMI</i> | | | 0 | 0 | 1 | 0 |
| <i>G13</i> | | | | 0 | 1 | 0 |
| <i>G14</i> | | | | | 0 | -1 |
| <i>G26</i> | | | | | | 0 |

| <i>End \ Start</i> | <i>CAD</i> | <i>Race</i> | <i>SEX</i> | <i>HY</i> | <i>G26</i> | <i>G30</i> |
|--------------------|------------|-------------|------------|-----------|------------|------------|
| <i>CAD</i> | 0 | -1 | -2 | 1 | -2 | 0 |
| <i>Race</i> | | 0 | -1 | 2 | -1 | 1 |
| <i>SEX</i> | | | 0 | 2 | 0 | 2 |
| <i>HY</i> | | | | 0 | -3 | -1 |
| <i>G26</i> | | | | | 0 | 2 |
| <i>G30</i> | | | | | | 0 |

(a) Edge matrix of $CBN_{6,1}$ (b) Edge matrix of $CBN_{6,2}$

| <i>End \ Start</i> | <i>CAD</i> | <i>Race</i> | <i>SEX</i> | <i>SM</i> | <i>AGE</i> | <i>CBMI</i> |
|--------------------|------------|-------------|------------|-----------|------------|-------------|
| <i>CAD</i> | 0 | 5 | 3 | 4 | 1 | 2 |
| <i>Race</i> | | 0 | -2 | -1 | -4 | -3 |
| <i>SEX</i> | | | 0 | 1 | -2 | -1 |
| <i>SM</i> | | | | 0 | -3 | -2 |
| <i>AGE</i> | | | | | 0 | 1 |
| <i>CBMI</i> | | | | | | 0 |

(c) Edge matrix of $CBN_{6,3}$

| <i>End \ Start</i> | <i>CAD</i> | <i>Race</i> | <i>CBMI</i> | <i>SEX</i> | <i>HY</i> | <i>SM</i> | <i>AGE</i> | <i>G13</i> | <i>G14</i> | <i>G26</i> | <i>G30</i> |
|--------------------|------------|-------------|-------------|------------|-----------|-----------|------------|------------|------------|------------|------------|
| <i>CAD</i> | 0 | 5 | 1 | 1 | 1 | 4 | 1 | -1 | 0 | -3 | 0 |
| <i>Race</i> | | 0 | -5 | -3 | 2 | -1 | -4 | -2 | -1 | -3 | 1 |
| <i>CBMI</i> | | | 0 | 1 | 0 | 2 | -1 | 0 | 1 | 0 | 0 |
| <i>SEX</i> | | | | 0 | 2 | 1 | -2 | 0 | 0 | 0 | 2 |
| <i>HY</i> | | | | | 0 | 0 | 0 | 0 | 0 | -3 | -1 |
| <i>SM</i> | | | | | | 0 | -3 | 0 | 0 | 0 | 0 |
| <i>AGE</i> | | | | | | | 0 | 0 | 0 | 0 | 0 |
| <i>G13</i> | | | | | | | | 0 | 1 | 0 | 0 |
| <i>G14</i> | | | | | | | | | 0 | -1 | 0 |
| <i>G26</i> | | | | | | | | | | 0 | 2 |
| <i>G30</i> | | | | | | | | | | | 0 |

(d) Edge matrix of resulting Bayesian network

| | <i>CAD</i> | <i>Race</i> | <i>CBMI</i> | <i>SEX</i> | <i>HY</i> | <i>SM</i> | <i>AGE</i> | <i>G13</i> | <i>G14</i> | <i>G26</i> | <i>G30</i> |
|-----------------------------|------------|-------------|-------------|------------|-----------|-----------|------------|------------|------------|------------|------------|
| <i>TO_{method3}</i> | 3 | 9 | 5 | 4 | 11 | 7 | 6 | 2 | 8 | 1 | 10 |

(e) Target variable ordering with method 3

Table 5.6: Variable ordering in 6-node candidate BNs with method 3

Using method 2 over combination of 6-node Bayesian networks, there are four arcs ($\langle RACE, CAD \rangle, \langle RACE, G30 \rangle, \langle SEX, HY \rangle, \langle CAD, HY \rangle$) to be reversed in $CBN_{6,2}$, which led to another three arcs ($\langle CAD, G30 \rangle, \langle SEX, G30 \rangle, \langle HY, G30 \rangle$) in $CBN_{6,2}$ to be reversed too. In $CBN_{6,3}$, there are four arcs ($\langle CAD, AGE \rangle, \langle CAD, SEX \rangle, \langle CAD, SM \rangle, \langle SEX, SM \rangle$) are reversed. The combination time with method 3 is 5.999 seconds.

The combination of method 3 over three 6-node Bayesian networks has incur one arc ($\langle CBMI, CAD \rangle$), two arcs ($\langle RACE, CAD \rangle, \langle SEX, CAD \rangle$) and three arcs ($\langle AGE, CBMI \rangle, \langle AGE, SEX \rangle, \langle CBMI, SEX \rangle$) to be reversed during the combination. The combination time is 1.813 second.

| | Num_ArcReversed | Num_ArcAdded | Combination Time (Sec) |
|------------------------|-----------------|--------------|------------------------|
| Method 1 | 6 | 5 | 1.953 |
| Method 2 | 10 | 5 | 5.999 |
| Method 3 | 6 | 3 | 1.813 |
| Random Target Ordering | 16 | 8 | 3.84 |

Table 5.7: Comparison of 3 methods in three6-node BN combination

In additiona, we use a random target variable ordering to test how the resulting model can be. The random target variable orderig is $\{HY = 0, G30 = 1, RACE = 4, SM = 2, CAD = 5, CBMI = 7, SEX = 3, G26 = 0, G13 = 0, Age = 1, G14 = 1\}$. Under such a target ordering, in $CBN_{6,1}$, there are 3 arcs to be reversed: $\{\langle CBMI, CAD \rangle, \langle CBMI, RACE \rangle, \langle CAD, RACE \rangle\}$; in $CBN_{6,2}$, there are 5 arcs to be reversed: $\{\langle SEX, HY \rangle, \langle RACE, G30 \rangle, \langle RACE, HY \rangle, \langle SEX, HY \rangle, \langle CAD, HY \rangle\}$; in $CBN_{6,3}$, there are 8 arcs to be reversed: $\{\langle CAD, AGE \rangle, \langle CBMI, SEX \rangle, \langle SEX, SM \rangle, \langle CAD, RACE \rangle, \langle CAD, SEX \rangle, \langle CBMI, RACE \rangle, \langle CAD, SM \rangle, \langle CBMI, CAD \rangle\}$. The combination time under this random target variable ordering is 3.84 seconds. In the degree of broken of original conditional independence, any of our heuristic method have better performance than the random target variable ordering.

5.1.4 Comparison of Different Size Bayesian Network Combination

In this section, we mainly describe some experiments based on combination over candidate Bayesian networks in different size. Three groups of candidate Bayesian networks are compared in their combination. We use $|var|$ to denote the number of variables and use $|arc|$ to indicate the number of arcs in a Bayesian network.

The first group consists of three different size Bayesian networks, in which the first candidate Bayesian network is of five variables and six arcs, the second candidate Bayesian network is of eight variables and thirteen arcs, and the third candidate Bayesian network is of twelve variables and seventeen arcs.

The second group consists of three size Bayesian networks, in which each Bayesian network is of seven variables but with different number of arcs. The first candidate Bayesian network is with ten arcs. The second candidate Bayesian network is with nine arcs and the third candidate Bayesian network is with eleven arcs. Using method 1 over combination of 7-node Bayesian networks, there is three arcs ($\langle RACE, CAD \rangle$, $\langle RACE, G30 \rangle$, $\langle SEX, CAD \rangle$) and two arcs ($\langle CAD, AGE \rangle$, $\langle SM, RACE \rangle$) need to be reversed in $CBN_{7.2}$ and $CBN_{7.3}$ respectively. The combination time is 162.754 seconds. Using method 2 over combination of 7-node Bayesian networks, there are two arcs ($\langle RACE, CAD \rangle$, $\langle RACE, G30 \rangle$) are required to be reversed in $CBN_{7.2}$, which incur another two arcs ($\langle CAD, G30 \rangle$, $\langle SEX, G30 \rangle$) in $CBN_{7.2}$ to be reversed too. In $CBN_{7.3}$, there are four arcs ($\langle CAD, AGE \rangle$, $\langle CAD, SEX \rangle$, $\langle CAD, SM \rangle$, $\langle SEX, SM \rangle$) are reversed. The combination time with method 3 is 350.143 seconds. Using method 3, $order_{7.1}(CBMI) - order_{7.1}(CAD) = 2$ and $order_{7.3}(CBMI) - order_{7.3}(CAD) = -2$, therefore we randomly let $TO_{method3}(CAD) > TO_{method3}(CBMI)$. There are one arc ($\langle CBMI, CAD \rangle$), two arcs ($\langle RACE, CAD \rangle$, $\langle SEX, CAD \rangle$) and three arcs ($\langle AGE, CBMI \rangle$, $\langle AGE, SEX \rangle$, $\langle AGE, RACE \rangle$) are reversed in $CBN_{7.1}$, $CBN_{7.2}$ and $CBN_{7.3}$ respectively. The combination time is only 15.222 seconds! Method 1 and method 2 do not consider number of parents of each node, it is possible that arc reversal acts are operated at some nodes that have many parent

nodes. This may be the explanation of the large amount of combination time.

The third group consists of three different size Bayesian networks, in which the first candidate Bayesian network is of five variables and six arcs, the second candidate Bayesian network is of seven variables and nine arcs, and the third candidate Bayesian network is of ten variables and seventeen arcs.

The experimental results show that each of our heuristic target variable generation method can result in less arcs in the resulting model. Method 2 can provide better performance than Method 1 towards different size Bayesian network combination, and sometimes better than Method 3.

| | CBN_1 | | CBN_2 | | CBN_3 | | ResultBN | | Combination Time (sec) |
|------------------------|-------|-----|-------|-----|-------|-----|----------|-----|------------------------|
| | Var | Arc | Var | Arc | Var | Arc | Var | Arc | |
| Method 1 | 5 | 6 | 8 | 13 | 12 | 16 | 12 | 17 | 11.998 |
| Method 2 | 5 | 6 | 8 | 13 | 12 | 16 | 12 | 14 | 3.53 |
| Method 3 | 5 | 6 | 8 | 13 | 12 | 16 | 12 | 16 | 11.73 |
| Random Target Ordering | 5 | 6 | 8 | 13 | 12 | 16 | 12 | 21 | 12.933 |
| Method 1 | 7 | 10 | 7 | 9 | 7 | 11 | 14 | 31 | 162.754 |
| Method 2 | 7 | 10 | 7 | 9 | 7 | 11 | 14 | 38 | 350.143 |
| Method 3 | 7 | 10 | 7 | 9 | 7 | 11 | 14 | 29 | 15.222 |
| Random Target Ordering | 7 | 10 | 7 | 9 | 7 | 11 | 14 | 39 | 353.101 |
| Method 1 | 5 | 6 | 7 | 9 | 10 | 18 | 12 | 22 | 15.781 |
| Method 2 | 5 | 6 | 7 | 9 | 10 | 18 | 12 | 19 | 3.64 |
| Method 3 | 5 | 6 | 7 | 9 | 10 | 18 | 12 | 20 | 13.392 |
| Random Target Ordering | 5 | 6 | 7 | 9 | 10 | 18 | 12 | 25 | 16.939 |

Table 5.8: Comparison of different size Bayesian networks combination

5.2 Experimental Results on Utility based Parameter Combination

This section is devoted to the description of experiment settings, experiment result and analysis based on the utility-based combination of probability distributions from multiple influence diagrams.

5.2.1 Experiment Setting

Patients: Twins with conjoined body part from breast to navel: Tom and Smith.

Situation: Tom and Smith have two complete set of organs but now Smith suffered from Cancer A while Tom did not, the operation to separate them is to secure the chance of Tom's survival against diffusion of cancer cell from Smith's body to Tom's. However, whether they share veins, arteries or nerves can only be known in the operation, so we need probability distributions on possible probabilities to predict Tom's chance of survive the surgery.

InformationSources: Five Sources E_1, \dots, E_5 .

DecisionMaker: Doctor Alex who will be the primary Surgeon of this operation if the operation will be implemented.

Alternatives: Do the surgery operation to separate their body so as each person has his own body.

Objective: Maximize their chance of survival of .

Risk attitude: Risk neutral.

Known: The twins decide the importance of Tom's survival is 3 times more than Smith's. They are 20 year old. Without operation, they are expected to live another 4 years and 0 year, respectively. If they can survive the operation, Tom and Smith can survive for additional 50 years and 1 year respectively.

Experiment1: Getting utility based weights of each knowledge source.

Experiment2: Solve the decision model based on arithmetic combination of probability distributions with equal weight of each knowledge source and with utility-based weight of each knowledge source respectively, make comparisons.

Experiment3: Solve the decision model based on geometric combination of probability distributions with equal weight of each knowledge source and with utility-based weight of each knowledge source respectively, make comparisons.

| Factors | Variables | Possible Values |
|----------------------|-----------|-----------------|
| Do they share nerves | SN | {Yes, No} |
| Do they share artery | SA | {Yes, No} |
| Do they share vein | SV | {Yes, No} |

Table 5.9: Some factors influence the decision

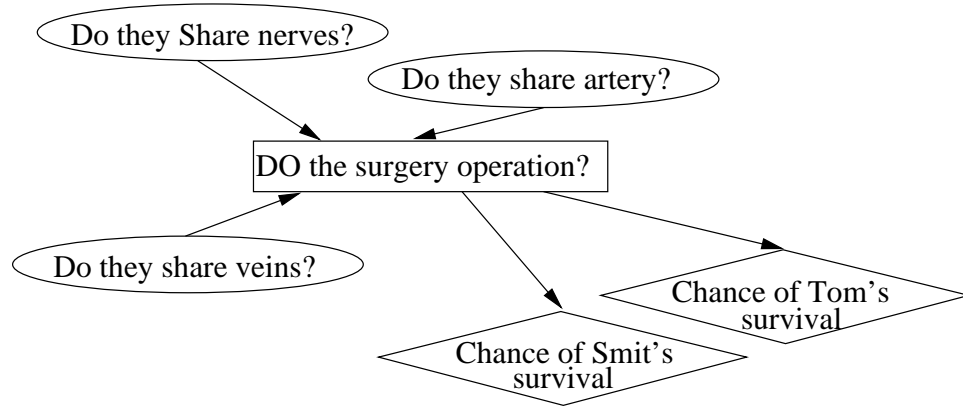


Table 5.10: The model of body separation surgery

Experiment4: Adding a new knowledge source (knowledge source 6) into the knowledge sources set. Make comparisons.

Organ separation operation no doubt will have quite a number of factors to be considered. For simplicity, we only select 3 factors (whether the twins share nerves, whether the twins share artery, whether the twins share vein) that have influence on the decision which is listed in Table 5.9. The model is shown in Figure 5.10.

Also we assume we get corresponding probabilities from knowledge source 1 on three variables on the three factors which displayed in Table 5.11, and probabilities from the opinion of knowledge source 2 (Table 5.12), knowledge source 3 (Table 5.13), knowledge source 4 (Table 5.14) and knowledge source 5 (Table 5.15).

When solving the decision model based on each knowledge source, we get the 5 expected utility corresponding to each of the 5 knowledge sources, as shown in Table 5.16.

| ShareNerve | ShareArtery | ShareVein | Survival_Tom | Survival_Smith |
|-----------------|-----------------|-----------------|--------------|----------------|
| $P(SN = Y)=0.7$ | $P(SA = Y)=0.6$ | $P(SV = Y)=0.9$ | | |
| Y | Y | Y | 0.1 | 0.1 |
| Y | Y | N | 0.2 | 0.2 |
| Y | N | Y | 0.35 | 0.35 |
| Y | N | N | 0.4 | 0.4 |
| N | Y | Y | 0.15 | 0.15 |
| N | Y | N | 0.18 | 0.18 |
| N | N | Y | 0.6 | 0.6 |
| N | N | N | 0.9 | 0.9 |

Table 5.11: Opinion of knowledge source 1

| ShareNerve | ShareArtery | ShareVein | Survival_Tom | Survival_Smith |
|-----------------|-----------------|-----------------|--------------|----------------|
| $P(SN = Y)=0.7$ | $P(SA = Y)=0.6$ | $P(SV = Y)=0.9$ | | |
| Y | Y | Y | 0.1 | 0.09 |
| Y | Y | N | 0.21 | 0.2 |
| Y | N | Y | 0.4 | 0.35 |
| Y | N | N | 0.5 | 0.45 |
| N | Y | Y | 0.2 | 0.15 |
| N | Y | N | 0.18 | 0.1 |
| N | N | Y | 0.65 | 0.6 |
| N | N | N | 0.95 | 0.9 |

Table 5.12: Opinion of knowledge source 2

| ShareNerve | ShareArtery | ShareVein | Survival_Tom | Survival_Smith |
|-----------------|-----------------|-----------------|--------------|----------------|
| $P(SN = Y)=0.7$ | $P(SA = Y)=0.6$ | $P(SV = Y)=0.9$ | | |
| Y | Y | Y | 0.3 | 0.25 |
| Y | Y | N | 0.3 | 0.3 |
| Y | N | Y | 0.45 | 0.4 |
| Y | N | N | 0.55 | 0.5 |
| N | Y | Y | 0.29 | 0.26 |
| N | Y | N | 0.25 | 0.2 |
| N | N | Y | 0.6 | 0.6 |
| N | N | N | 1 | 0.95 |

Table 5.13: Opinion of knowledge source 3

| ShareNerve | ShareArtery | ShareVein | Survival_Tom | Survival_Smith |
|-----------------|-----------------|------------------|--------------|----------------|
| $P(SN = Y)=0.8$ | $P(SA = Y)=0.6$ | $P(SV = Y)=0.95$ | | |
| Y | Y | Y | 0.09 | 0.06 |
| Y | Y | N | 0.2 | 0.15 |
| Y | N | Y | 0.3 | 0.23 |
| Y | N | N | 0.4 | 0.35 |
| N | Y | Y | 0.29 | 0.26 |
| N | Y | N | 0.1 | 0.1 |
| N | N | Y | 0.55 | 0.5 |
| N | N | N | 0.85 | 0.8 |

Table 5.14: Opinion of knowledge source 4

| ShareNerve | ShareArtery | ShareVein | Survival_Tom | Survival_Smith |
|-----------------|-----------------|------------------|--------------|----------------|
| $P(SN = Y)=0.8$ | $P(SA = Y)=0.7$ | $P(SV = Y)=0.85$ | | |
| Y | Y | Y | 0.15 | 0.15 |
| Y | Y | N | 0.25 | 0.25 |
| Y | N | Y | 0.4 | 0.35 |
| Y | N | N | 0.5 | 0.45 |
| N | Y | Y | 0.36 | 0.3 |
| N | Y | N | 0.3 | 0.25 |
| N | N | Y | 0.6 | 0.5 |
| N | N | N | 0.89 | 0.89 |

Table 5.15: Opinion of knowledge source 5

| | <i>Expert</i> ₁ | <i>Expert</i> ₂ | <i>Expert</i> ₃ | <i>Expert</i> ₄ | <i>Expert</i> ₅ |
|----------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|
| Expect Utility | 12.68574 | 14.17792 | 19.49782 | 11.73 | 13.86482 |
| $P(SN = Y)$ | 0.7 | 0.7 | 0.7 | 0.8 | 0.8 |
| $P(SA = Y)$ | 0.6 | 0.6 | 0.6 | 0.6 | 0.7 |
| $P(SV = Y)$ | 0.9 | 0.9 | 0.9 | 0.95 | 0.85 |
| θ_{1t} | 0.1 | 0.1 | 0.3 | 0.09 | 0.15 |
| θ_{1s} | 0.1 | 0.09 | 0.25 | 0.06 | 0.15 |
| θ_{2t} | 0.2 | 0.21 | 0.3 | 0.2 | 0.25 |
| θ_{2s} | 0.2 | 0.2 | 0.3 | 0.15 | 0.25 |
| θ_{3t} | 0.35 | 0.4 | 0.45 | 0.3 | 0.4 |
| θ_{3s} | 0.35 | 0.35 | 0.4 | 0.23 | 0.35 |
| θ_{4t} | 0.4 | 0.5 | 0.55 | 0.4 | 0.5 |
| θ_{4s} | 0.4 | 0.45 | 0.5 | 0.35 | 0.45 |
| θ_{5t} | 0.15 | 0.2 | 0.29 | 0.29 | 0.36 |
| θ_{5s} | 0.15 | 0.15 | 0.26 | 0.26 | 0.3 |
| θ_{6t} | 0.18 | 0.18 | 0.25 | 0.1 | 0.3 |
| θ_{6s} | 0.18 | 0.1 | 0.2 | 0.1 | 0.25 |
| θ_{7t} | 0.6 | 0.65 | 0.6 | 0.55 | 0.6 |
| θ_{7s} | 0.6 | 0.6 | 0.6 | 0.5 | 0.5 |
| θ_{8t} | 0.9 | 0.95 | 1 | 0.85 | 0.89 |
| θ_{8s} | 0.9 | 0.9 | 0.95 | 0.8 | 0.89 |

Table 5.16: Expected utilities corresponding to probability distributions from 5 knowledge sources

5.2.2 Comparison of Weights of All Sources under 3 Methods

From the information provided from the experiment settings, we can first compare the probability distributions from 5 knowledge sources. From figure 5.1, we can see that the line representing probability distributions from knowledge sources 3 is located above most other lines which corresponding to other knowledge sources, although it has some cross points with the line corresponding to probability distributions from knowledge sources 2. As we defined in Section 4.4.4.1, the quantity of probability on decision variables means the local dominance so we can say that Probability distributions from knowledge sources 3 locally dominate most other knowledge sources. Contrary to knowledge source 3, knowledge source 4 is locally dominated by most other knowledge sources.

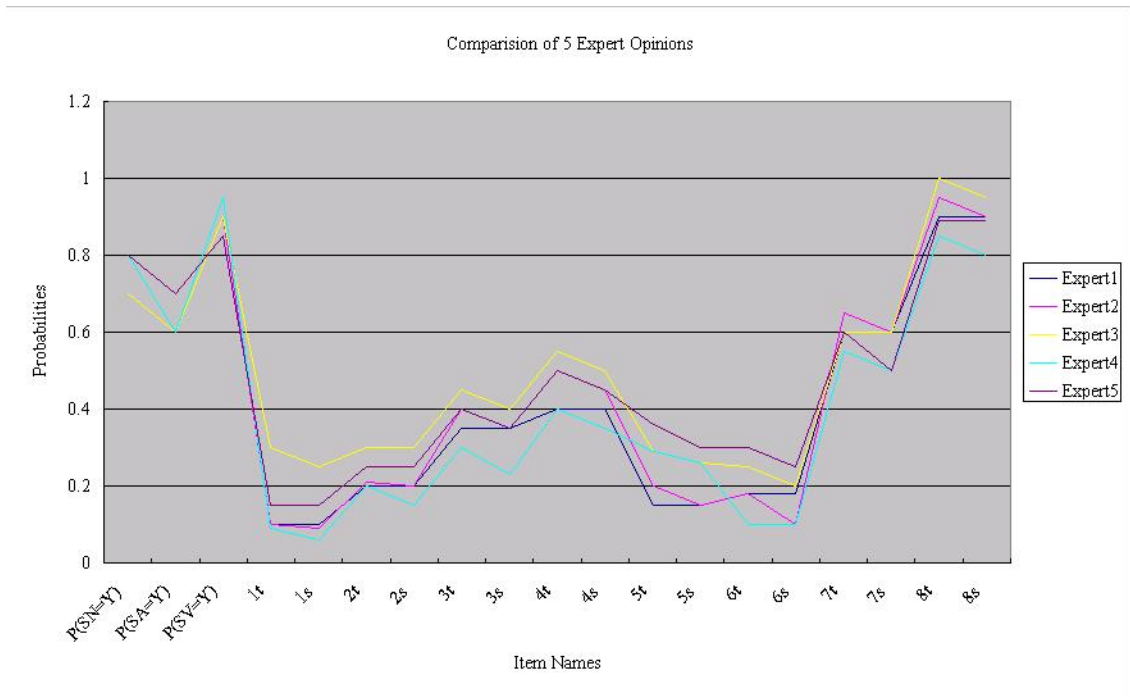


Figure 5.1: Comparison of probability distributions from 5 knowledge sources

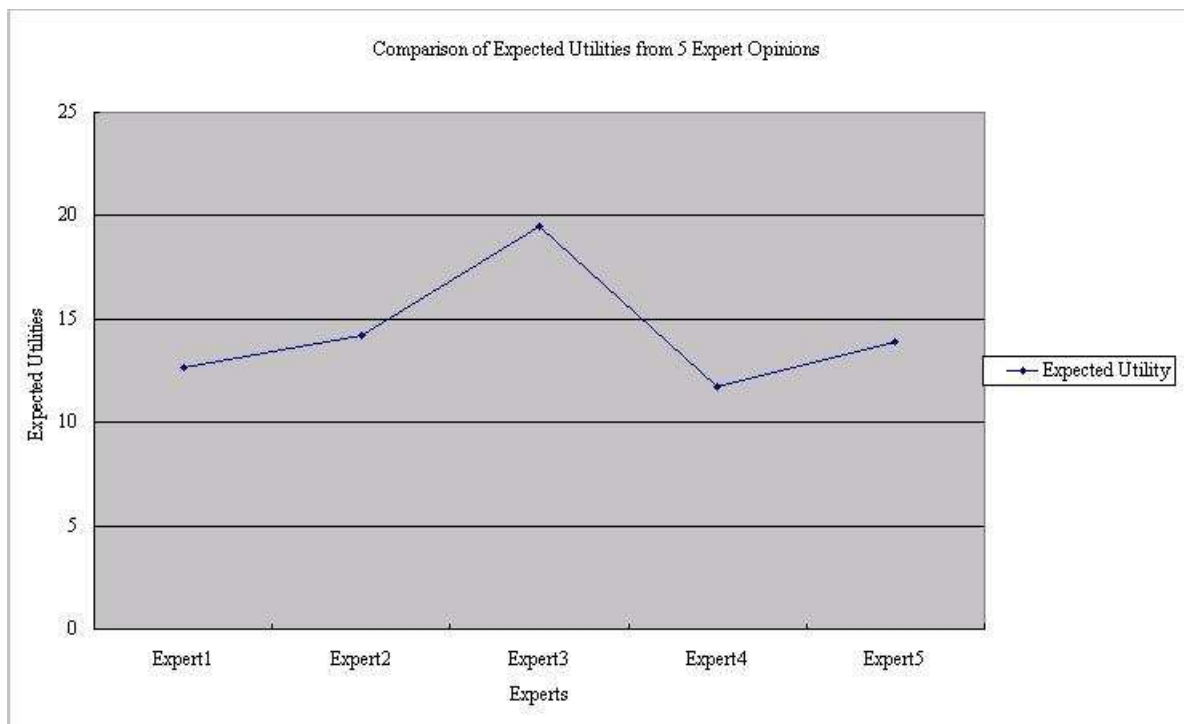


Figure 5.2: Comparison of expected utilities from probability distributions from 5 knowledge sources

Now we continue to check whether Expert3 has the global dominance over other experts through checking expected utilities of each experts, as shown in Figure 5.2. As we expected, Expert3 got the highest expected utility (19.49782) and Expert4 got the lowest expected utility (11.73).

| | $Expert_1$ | $Expert_2$ | $Expert_3$ | $Expert_4$ | $Expert_5$ |
|-------------------------------|------------|------------|------------|------------|------------|
| Expected Utility | 12.68574 | 14.17792 | 19.49782 | 11.73 | 13.86482 |
| $Weight_{average}$ | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |
| $Weight_{NeuralAttitude}$ | 0.228 | 0.254 | 0.135 | 0.135 | 0.246 |
| $Weight_{OptimisticAttitude}$ | 0.176298 | 0.1970968 | 0.270968 | 0.163016 | 0.192684 |

Table 5.17: Comparison of weights to 5 experts using different methods

After the calculation of expected utilities of all experts, we use three methods to get three different kinds of weights for the five experts.

The first method assigns equal weight (0.2) to each expert, and then summarizes the product of weight and each expert opinion.

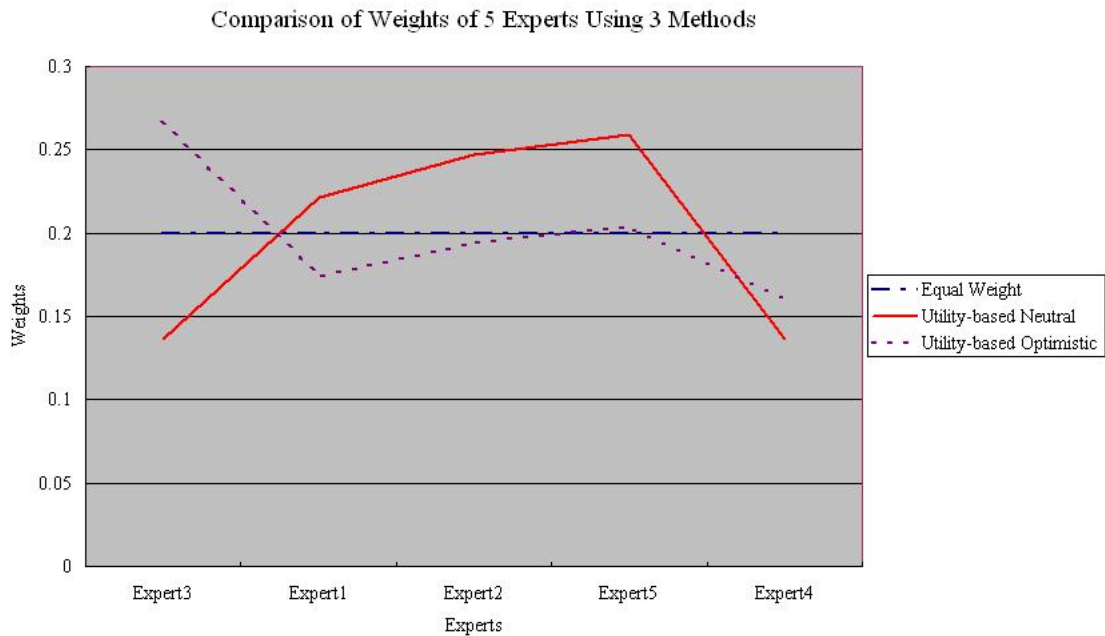


Figure 5.3: Comparison of weights of all experts in three methods

In addition, we provide a 'Utility based Weight with Optimistic Attitude' method to compute weights for all experts. This method assigns weight to each expert pro-

portional to their corresponding utilities, so Expert3 get the most weight (0.270968) while Expert4 is allocated the least weight (0.163016). Because this method is prejudicial on expert who holding optimal assessment, we name it as 'Utility based Weight with Optimistic Attitude' method.

The third method is exact the method we defined in Algorithm 4.9, which assigns less weights to both extreme high and extreme low experts.

We compare the weights of the 5 experts using three methods in Figure 5.3 where we already sort experts in descending order according to their expected utilities.

5.2.3 Comparison of Arithmetic Combined Probability Distribution

After make comparison of different methods of computing weights, in this section we will further compare combined expert opinions using arithmetic combination method with the three different weights from Section 5.2.2.

The value of combined expert opinions are listed in Table 5.18. Because all values are probabilities and the difference of combined value from different weights are very small, so we use logarithmic scale to plot Figure 5.4, from which we can see the line denoting the combined opinions with utility-based weights with optimistic attitude stay at the most above location.

| | $Weight_{average}$ | $Weight_{NeuralAttitude}$ | $Weight_{OptimisticAttitude}$ |
|---------------|--------------------|---------------------------|-------------------------------|
| $P(SN = Y)$ | 0.74 | 0.7383 | 0.73556995 |
| $P(SA = Y)$ | 0.62 | 0.6248 | 0.619268389 |
| $P(SV = Y)$ | 0.9 | 0.89435 | 0.898516586 |
| θ_{1t} | 0.148 | 0.13805 | 0.162197542 |
| θ_{1s} | 0.13 | 0.12471 | 0.141788347 |
| θ_{2t} | 0.232 | 0.22844 | 0.238701298 |
| θ_{2s} | 0.22 | 0.21915 | 0.228580166 |
| θ_{3t} | 0.38 | 0.38185 | 0.388431923 |
| θ_{3s} | 0.336 | 0.34055 | 0.343986503 |
| θ_{4t} | 0.47 | 0.47045 | 0.479617031 |
| θ_{4s} | 0.43 | 0.43185 | 0.438431923 |
| θ_{5t} | 0.258 | 0.25258 | 0.261073012 |
| θ_{5s} | 0.224 | 0.2169 | 0.226640728 |
| θ_{6t} | 0.202 | 0.20841 | 0.209048545 |
| θ_{6s} | 0.166 | 0.16894 | 0.170103163 |
| θ_{7t} | 0.6 | 0.60595 | 0.601700977 |
| θ_{7s} | 0.56 | 0.5617 | 0.56443005 |
| θ_{8t} | 0.918 | 0.91697 | 0.92687089 |
| θ_{8s} | 0.888 | 0.89077 | 0.895319976 |

Table 5.18: Comparison of arithmetically combined value of expert opinions

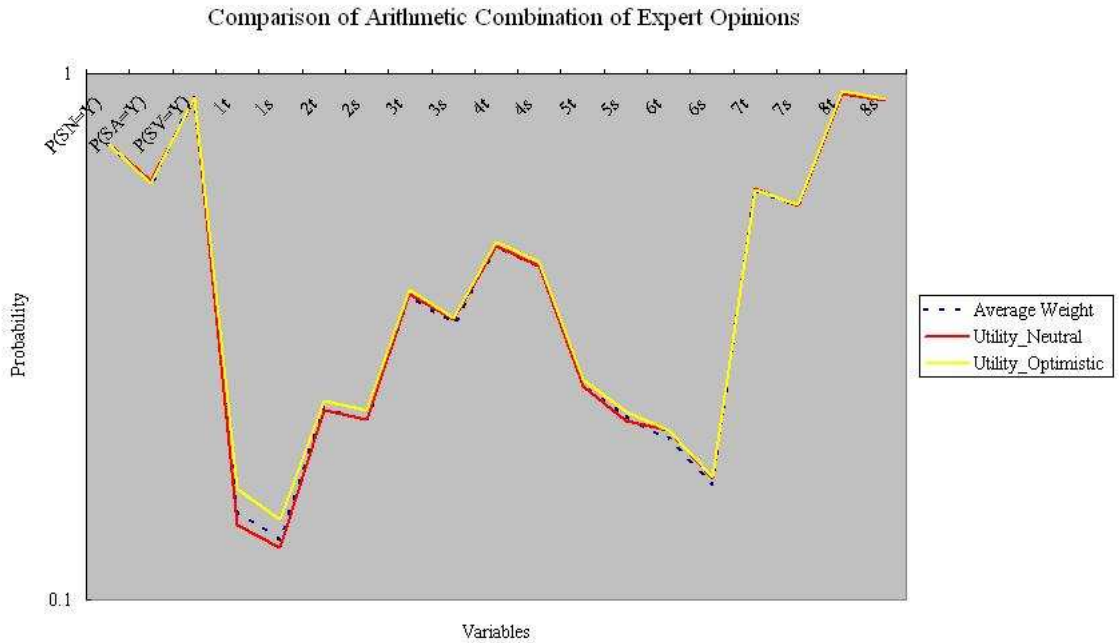


Figure 5.4: Comparison of arithmetically combined opinions with three kinds of weights

5.2.4 Comparison of Geometric Combined Probability Distribution

Now we use geometric combination approach to combine expert opinions with the three kinds of weights get from Section 5.2.2. In this section we discard the method of 'utility-based weight with optimal attitude' of determining weight and only keep our reasonable weight calculation method from Co-Experts framework to make comparison with combined opinions based on equal weights and the original 5 expert opinions. In the result of geometric combination(see Table 5.5), we can see the difference is more distinctive among most lines, but the difference between combined opinions based on equal weight and utility-based combined opinions are still very fine.

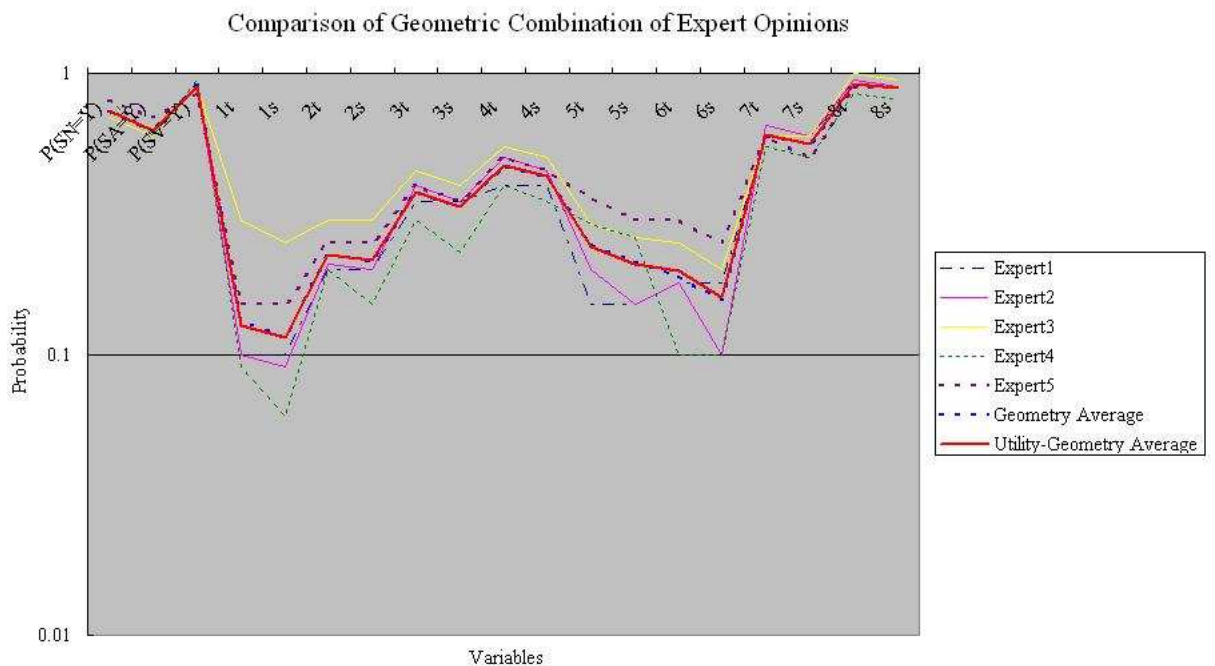


Figure 5.5: Geometric combinationed expert opinions with three kinds of weights

5.2.5 Comparison of Two Approaches of Combination

In this part, we compare the four kinds of combined expert opinions, which adopting two kinds of methods of determining the weights and two approach of combination. We can see that the geometric combination values are generally less than arithmetic

combination values. Nevertheless, within arithmetic combination, the difference in combined opinions from the two methods of determining weights is more apparent.

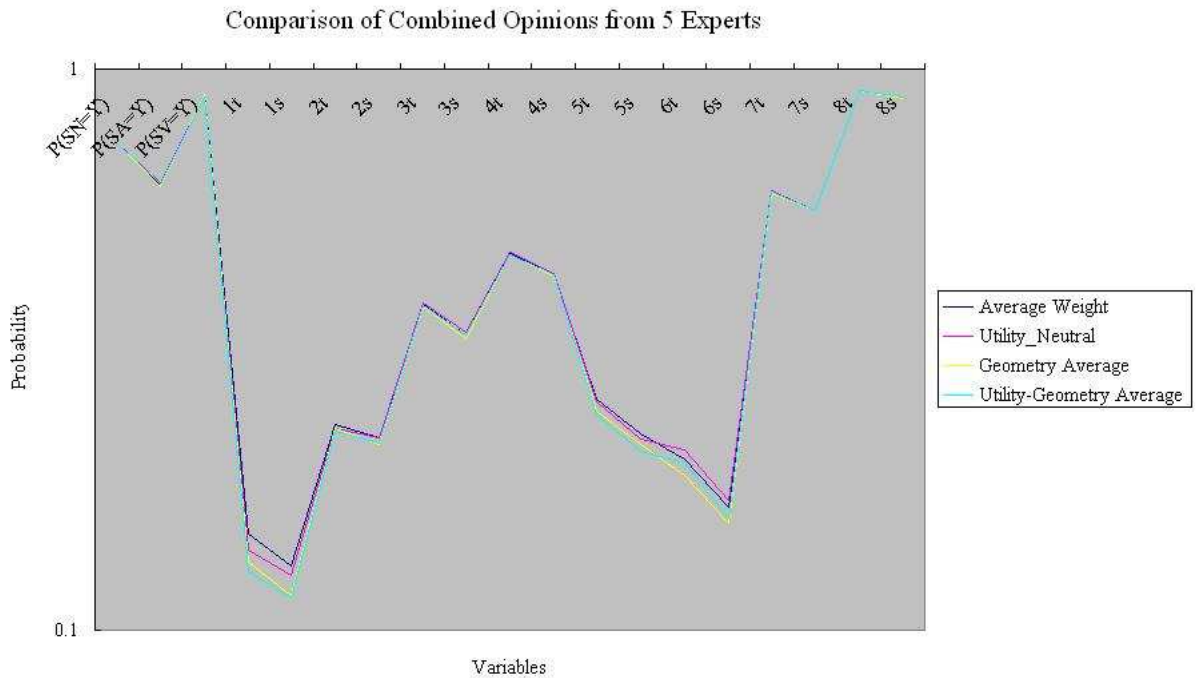


Figure 5.6: Comparison of two combination approaches

5.2.6 Result of Adding one more Knowledge Source

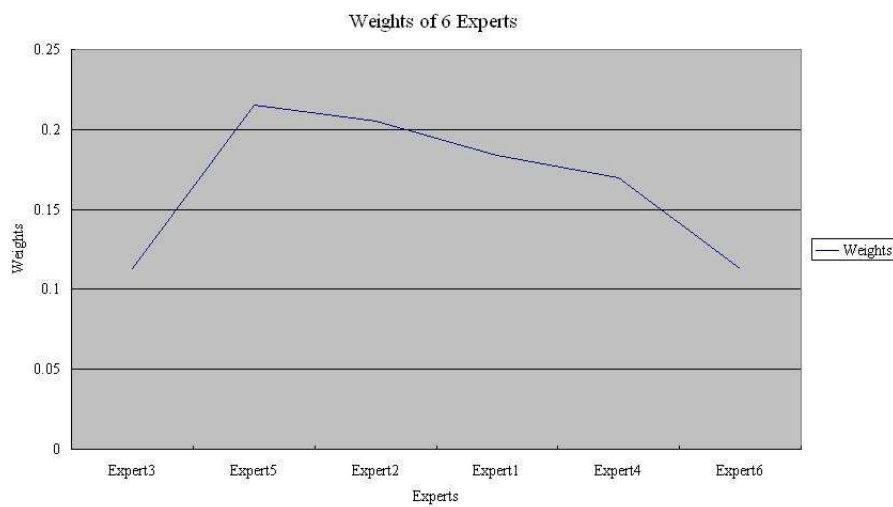


Figure 5.7: Weights of the 6 experts

We conduct one more experiment to test the effect of combination after adding new expert. The new expert to be added (Expert6) happened to have the same expected utility as Expert4 although their opinions on each decision variables are not same(as shown in Figure 5.8). Therefore, we can randomly select either Expert4 or Expert6 as the expert with lowest expected utility. The corresponding weight of the 6 experts can be seen in Figure 5.7.

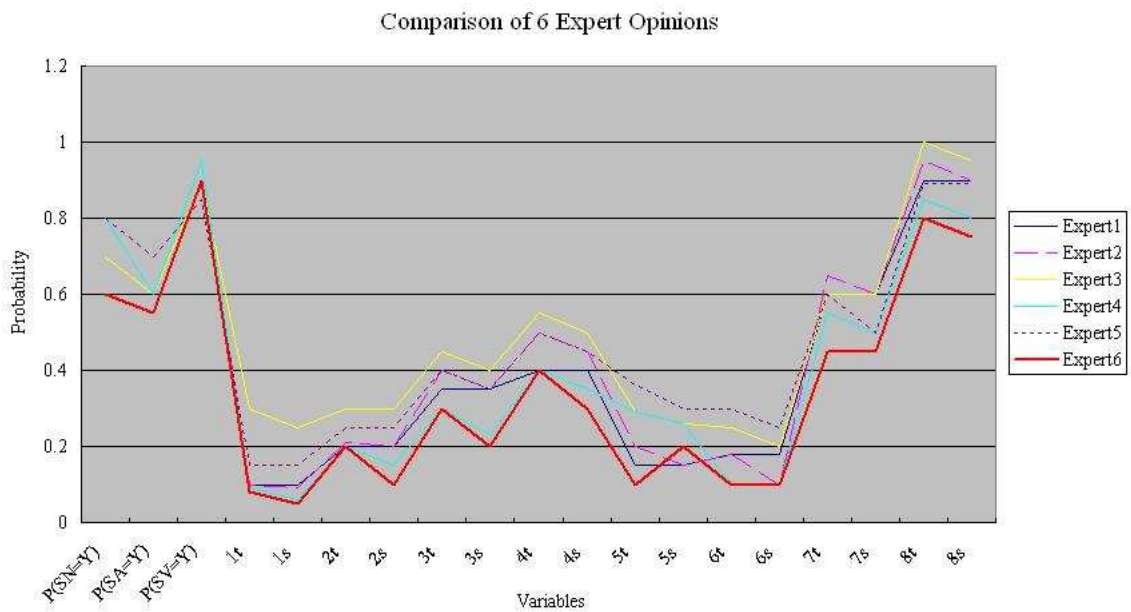


Figure 5.8: The 6 expert opinions

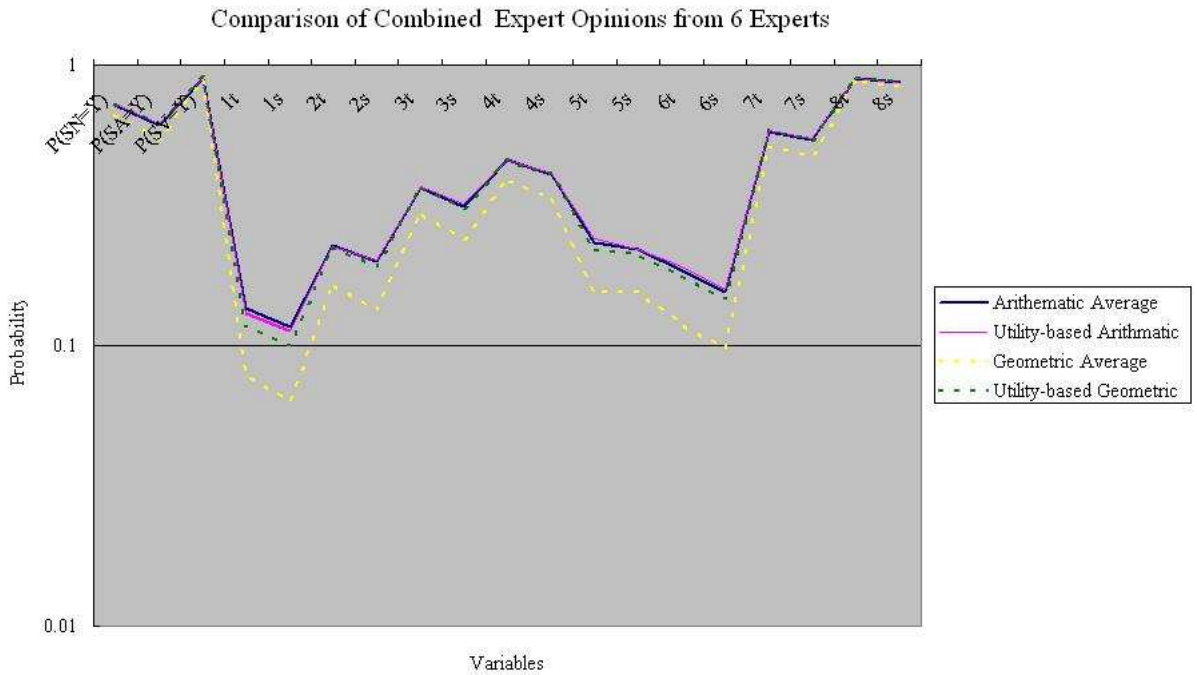


Figure 5.9: Combination result of the 6 expert opinions

Figure 5.9 shows the combined expert opinions. Interestingly we find that the line of geometric combination with equal weights locates at the lowest position among the four lines and the arithmetic combination with equal weights locates at the most upper position. Using our utility-based weight method, either geometric combination or arithmetic combination performs very well because they are between the upper and lower lines.

Finally we make comparison of combined opinions based on 5 experts and combined opinions based on 6 experts. We can see that geometric combination based on equal weights decreases a lot. The possible reason for this phenomenon might be the weight of Expert6 (0.2) is higher than the weight of Expert6 in our utility-based weight method (0.113).

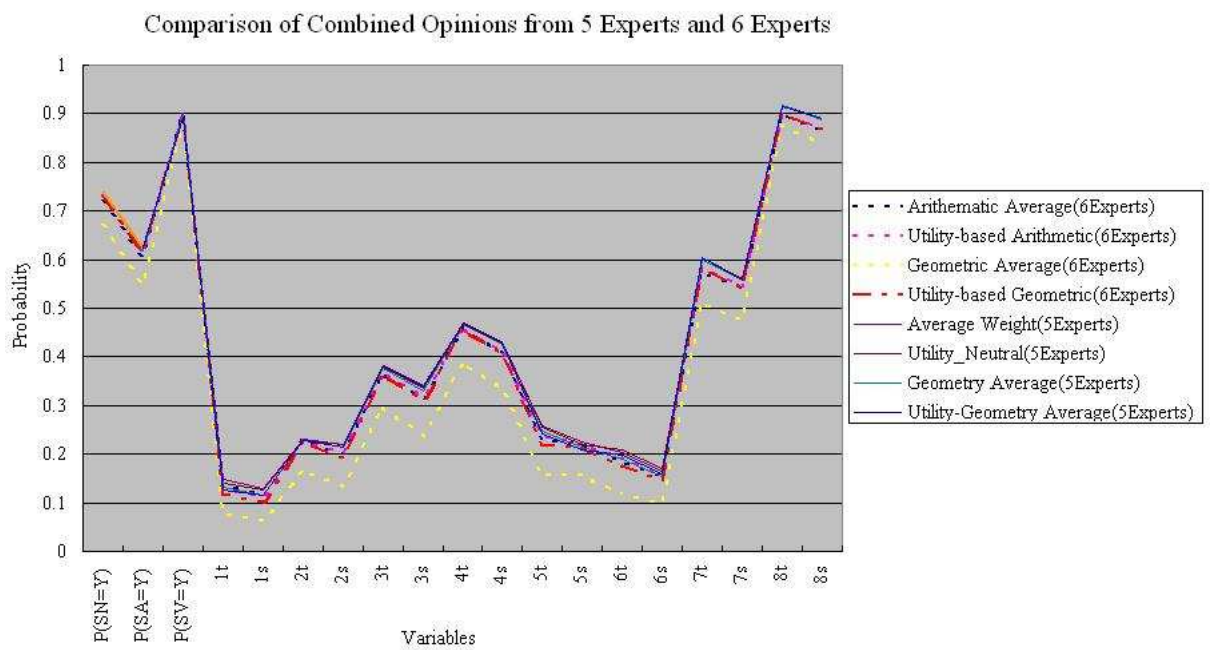


Figure 5.10: 5 experts vs 6 experts..

Chapter 6

Conclusion and Future Work

6.1 Summary

Bayesian networks and influence diagrams are powerful graphic representation tools for uncertainty management and decision making, respectively. They are especially helpful when the problems have a high degree of condition independence, when compact representation of extremely large models are needed, when communication of the probabilistic relationships is important, or when the analysis requires extensive Bayesian updating.

Research over graphical representation of single knowledge source receives many applications of modeling uncertainty. In this thesis, we mainly examine the problem of combining several graphical models at one time, to form a single resulting model mathematically. We make two intuitively reasonable assumptions: 1) Variables with the same name have to denote the same meaning. 2) Variables with the same name have to possess the same number of possible values, and their values must be the same.

The first part of our research focus on *multiple Bayesian networks combination* problem. We separate the task into two subtasks: qualitative combination and quantitative combination.

Qualitative combination of Bayesian networks is the first task. We theoretically demonstrate that Bayesian networks can be reconstructed because JPD is factorizable with different partition of variables. A basic idea of our method is to get a target

variable ordering for resulting Bayesian networks so that the direction of arcs in the resulting Bayesian networks are only allowed when it is from nodes with lower order value to nodes with higher order value. With target variable ordering, we introduce the arc reversal operation, which can adjust the order value of variables within one probabilistic model. Our method also aims to reach consensus topology for each input Bayesian network, so that the quantitative combination step can be clear and easy. Therefore, we present the concept of Intermediate Bayesian networks, so that all modification steps, including arc reversal, filling of virtual nodes and virtual arcs, are operated over Intermediate Bayesian networks and the structure and parameters of original Bayesian networks can be preserved.

The target variable ordering can be specified by user, for example, domain experts. However, we propose three heuristic methods of target variable ordering generation, which is very helpful when domain experts are absent. The three methods are not guarantee to be optimal solution as it is a NP-hard problem.

In the quantitative combination of Bayesian networks, we discussed computation in steps of reaching consensus model, including arc reversal, virtual nodes and virtual arcs. A key step in quantitative combination is CPT combination. Traditional CPTs are filled with point probability distributions. We argue that the CPT in resulting Bayesian network after combination can be filled with either point probability distribution or interval probability.

The second part of our research focuses on the *multiple influence diagrams combination* problem. Influence diagrams are different from Bayesian networks in types of arcs, types of nodes, and some other constraints in both the structure and the parameters. Upon our investigation on these special properties of influence diagrams, we conclude that only arcs between chance nodes are allowed to be reversed.

In the quantitative combination of influence diagrams, we propose a utility-based probability distribution combination method, to avoid extremeness of different knowledge sources, because utility is a criterion that can reflect risk attitude of the decision maker.

6.1.1 Advantages

1. Model Combination. Our main contribution in this work is to extend existing research of combining probability distributions to the case of aggregating probabilistic graphic models.
2. No cycle in resulting models. The resulting model after combination with our methods is established according to a certain target variable ordering. All the arcs are only permitted when they are from nodes with low *ordervalue* to nodes with high *ordervalue*. In this way, no cycle will be generated in the procedure of combination.
3. Completeness. Previous research work only focuses on either structure combination of Bayesian networks, or probability distribution. Solution to accomplish both tasks at one time is not available yet. Our research covers both structure combination and parameter combination for two types of probabilistic graph models, i.e. Bayesian networks and influence diagrams.
4. Scalability. Our methods are able to guarantee that more than two probabilistic models of the same type (i.e., either multiple Bayesian networks or multiple influence diagrams) can be combined at the same time.
5. Robust property. The result of combination will not be influenced by the order of combination.

6.1.2 Limitation

One of the disadvantages of our approaches results from the inherent property of the arc reversal operation. Arc reversal often significantly increases the number of parents of the nodes that the two nodes in the arc are involved. Since the size of CPT may increase exponentially with the number of parents, the resulting CPTs can become very large and require a prohibitive amount of computation to construct.

6.1.3 Discussion

1. Target ordering is important in probabilistic graphic model combination. In Chapter 5, we tested our three proposed target ordering generation methods over some heart disease models. The experimental results show that the target ordering in probabilistic model combination can result in different resulting probabilistic graphic models, with different structure or different parameters.
2. According to the experiment results, we also find that the three proposed target ordering generation methods are good in different situations. The computation of Method 1 is intuitive and simple but the performance are always not as good as the other two heuristic methods. Method 2 (target ordering based on number of parents and network size) has good performance in combination over candidate models with different network size. When the candidate probabilistic graphic models are of the same or nearly the same size, both Method 1 (target ordering based on original target order value) and Method 3 (target ordering based on edge matrix) have good performance. Method 3 has better performance over the other two methods especially in combining larger or more complex candidate probabilistic graphic models. But any of the three proposed heuristic target variable ordering generation methods is better than some random target variable ordering, according to the degree of breaking conditional independence of the original candidate Bayesian networks.

6.2 Future Work

Our future work, agenda include the following:

1. Our current work requires examination of input models to guarantee that there are no incompatible variables or cycles. Improvements can be made so that the system can reject incompatible incoming information, or sound alarm when incoming information is incompatible.
2. Although we proposed three heuristic methods of automatical target variable ordering generation, the methods can be still improved.

3. Our methods are applied in general probabilistic graphic models. Future research may examine causal discovery and structuring in certain domains and develop domain specified probabilistic graph models combination method.
4. In our research, we only consider static probabilistic networks. In other words, all of the candidate Bayesian networks to be combined are static probabilistic graph models. Extension to *temporal* probabilistic models should be interesting. The work that we have done in this thesis can provide meaningful support for the combination of temporal models.
5. The resulting models in both Bayesian network combination and influence diagram combination can be with either point probability distributions or interval probabilities. Reasoning based on Interval Bayesian networks or Interval influence diagrams can be quite challenging, and very interesting as well.

Bibliography

- [Ayyub, 2001] Ayyub, B. M. (2001). *Elicitation of Expert Opinions for Uncertainty and Risks*. CRC Press.
- [Bernoulli, 1738] Bernoulli, D. (1738). Exposition of a new theory on the measurement of risk. *Econometrica*, 22:23–36.
- [Castillo et al., 1997] Castillo, E., Jose, M., Gutierrez, Ali, S., and Hadi (1997). *Expert Systems and Probabilistic Network Models*. Springer,Verlag, New York.
- [Cheuk and Boutilier, 1997] Cheuk, A. Y. W. and Boutilier, C. (1997). Structured arc reversal and simulation of dynamic probabilistic networks. In *Proceeding of Conference on Uncertainty of Artificial Intelligence 1997*.
- [Clemen and Winkler, 1999] Clemen, R. T. and Winkler, R. (1999). Combining probability distributions from experts in risk analysis. *Risk Analysis*, 19:187–203.
- [Cooke, 1991] Cooke, R. M. (1991). *Experts in uncertainty : opinion and subjective probability in science*. New York : Oxford University Press.
- [Cooper and Herskovits, 1992] Cooper, G. F. and Herskovits, E. (1992). A bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347.
- [Dalkey, 1969] Dalkey, N. C. (1969). An experimental study of group opinion. Technical Report Rand RM-5888-PR, Rand Corporation, Santa Monica, California.
- [Dempster, 1968] Dempster, A. P. (1968). A generalization of bayesian inference. *Journal of the Royal Statistical Society*, B(30):205–247.

- [Downs et al., 1997] Downs, S., Friedman, C., Marasigan, F., and Gartner, G. (1997). A decision analytic method for scoring performance on computer-based patient simulations. In *In Proceeding of American Medical Informatics Association Annual Fall Symposium*, pages 67–71.
- [Druzdzel and Diez, 2003] Druzdzel, M. J. and Diez, F. J. (2003). Combining knowledge from different sources in probabilistic models. *Journal of Machine Learning Research*, 4:295–316.
- [Druzdzel et al., 1999] Druzdzel, M. J., Onisko, A., Schwartz, D., Dowling, J. N., and Wasyluk, H. (1999). Knowledge engineering for very large decision -analytic medical models. In *Proceedings of the 1999 Annual Meeting of the American Medical Informatics Association (AMIA 99)*, page 1049, Washington D.C.
- [Druzdzel and van der Gaag, 2000] Druzdzel, M. J. and van der Gaag, L. C. (2000). Building probabilistic networks: ‘where do the numbers come from?’ *IEEE Transactions on Knowledge and Data Engineering*, 12(4):481–486.
- [Edwards, 1992] Edwards, W. (1992). *Utility Theories: Measurements and applications*. Boston: Kluwer.
- [French, 1985] French, S. (1985). Group consensus probability distributions: A critical survey. *Bayesian Statistics*, 2:183–197.
- [French, 1990] French, S. (1990). Updating of belief in the light of someone else’s opinion. *Journal Royal Statistic Society Series*, 142:43–48.
- [Fung and Chang, 1989] Fung, R. and Chang, K. (1989). Weighting and integrating evidence for stochastic simulatin in bayesian networks. In *Proceeding of Conference on Uncertainty of Artificial Intelligence 1989*, pages 209–219. Windsor.
- [Geiger et al., 1989] Geiger, D., Verma, T., and Pearl, J. (1989). D-separation: from theorems to algorithms. In *Uncertainty in Artificial Intelligence*.
- [Genest and Zidek, 1986] Genest, C. and Zidek, J. V. (1986). Combining probability distributions: A critique and an annotated bibliography. *Statistical Science*, 1:114–135.

- [Ha and Haddawy, 1996] Ha, V. and Haddawy, P. (1996). Theoretical foundations for abstraction-based probabilistic planning. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*, pages 291–298.
- [Heckerma, 1999] Heckerman, D. H. (1999). *A Tutorial on Learning With Bayesian Networks*, pages 301 – 354. MIT Press Cambridge, MA, USA.
- [Heckerman et al., 1994] Heckerman, D., Geiger, D., and Chickering, D. M. (1994). Learning bayesian networks: The combination of knowledge and statistical data. In *KDD Workshop*, pages 85–96.
- [Hogarth, 1977] Hogarth, R. M. (1977). *Methods for aggregating opinions*, pages 231–255. Decision Making and Change in Human Affairs. Dordrecht, Reidel, Netherlands.
- [Howard and E, 1984] Howard, R. A. and E, M. J. (1984). *Influence Diagrams*, volume 2 of *Readings on the Principles and Applications of Decision Analysis*, chapter The Principles and Applications of Decision Analysis, pages 719–762. Strategic Decisions Group, Menlo Park, CA.
- [Kahneman et al., 1988] Kahneman, D., Slovic, P., and Tversky, A. (1988). *Judgment under Uncertainty: Heuristics and Biases*. Cambridge University Press, Cambridge.
- [Kanazawa et al., 1995] Kanazawa, K., Koller, D., and Russell, S. (1995). Stochastic simulation algorithms for dynamic probabilistic networks. In *IJCAI-95*, pages 346–351. Montreal.
- [Korver and Lucas, 1993] Korver, M. and Lucas, P. J. F. (1993). Converting a rule-based expert system into a belief network. *Medical Informatics*, 18:219–241.
- [Laskey and Levitt, 2002] Laskey, K. and Levitt, T. S. (2002). Multisource fusion for probabilistic detection and opportunistic assessment of terrorist threats. *Aerosense*.
- [Laskey et al., 2001] Laskey, K. B., Mahoney, S. M., and Wright, E. (2001). Hypothesis management in situation-specific network construction. In Jack S. Breese,

- D. K., editor, *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, pages 301–309, University of Washington, Seattle, Washington, USA. Morgan Kaufmann.
- [Lau and Leong, 1999] Lau, A. H. and Leong, T. Y. (1999). Probes: A framework for probabilities elicitation from experts. In *Proceedings of the American Medical Informatics Association Annual Fall Symposium*, pages 301–305.
- [Leong, 1991] Leong, T. Y. (1991). Representation requirements for supporting decision model formulation. In *Uncertainty in Artificial Intelligence: Proceedings of the Seventh Conference*, pages 212–219.
- [Leong and Cao, 1998] Leong, T. Y. and Cao, C. (1998). Modeling medical decisions in dynamol: A new general framework of dynamic decision analysis. In *Proceedings of the 9th World Congress on Medical Informatics (MEDINFO98)*, pages 483–487.
- [Lindley, 1985] Lindley, D. V. (1985). *Reconciliation of discrete probability distributions*, pages 375–390. Amsterdam: North-Holland.
- [Matzkevich and Abramson, 1992] Matzkevich, I. and Abramson, B. (1992). The topological fusion of bayes nets. In Dubois, D. and Wellman, M. P., editors, *Proceedings of the 8th Annual Conference on Uncertainty in Artificial Intelligence*, pages 152–158. Morgan Kaufmann.
- [Matzkevich and Abramson, 1993] Matzkevich, I. and Abramson, B. (1993). Some complexity considerations in the combination of belief networks. In *Proceeding of Uncertainty of Artificial Intelligence 1993*, pages 152–158.
- [Morgan and Henrion, 1992] Morgan, M. G. and Henrion, M. (1992). *Uncertainty: A Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis*. New York: Cambridge University Press.
- [Morris, 1977] Morris, P. A. (1977). Combining expert judgements: A bayesian approach. *Management Scienc*, 23(9):679–693.
- [Morris, 1983] Morris, P. A. (1983). An axiomatic approach to expert resolution. *Management Scienc*, 29(1):24–32.

- [Ng and Abramson, 1992] Ng, K. C. and Abramson, B. (1992). Consensus diagnosis: A simulation study. *IEEE Transactions on Systems, Man, and Cybernetics*, 22:916–928.
- [Pearl, 1988] Pearl, J. (1988). *Probabilistic Reasoning in Intelligence Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Los Altos, CA.
- [Pearl and Verma, 1991] Pearl, J. and Verma, T. S. (1991). A theory of inferred causation. In Allen, J., Fikes, R., and Sandewall, E., editors, *KR'91: Principles of Knowledge Representation and Reasoning*, pages 441–452, San Mateo, California. Morgan Kauffman Publishers.
- [Pennock and Wellman, 1998] Pennock, D. M. and Wellman, M. P. (1998). A market framework for pooling opinions. *Decision Analysis*.
- [Pennock and Wellman, 1999] Pennock, D. M. and Wellman, M. P. (1999). Graphical representations of consensus belief. In *Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 531–540, Stockholm.
- [Poulton, 1994] Poulton, E. C. (1994). *Behavioral Decision Theory: A New Approach*. New York: McGraw-Hill Inc.
- [Rantilla and Budescu, 1999] Rantilla, A. K. and Budescu, D. V. (1999). Aggregation of expert opinions. In *Thirty-Second Annual Hawaii International Conference on System Sciences*, volume 1, page 1058.
- [Rebecca, 1995] Rebecca, H. A. (1995). Improving group judgment accuracy: Information sharing and determining the best member. *Organizational Behavior and Human Decision Processes*, 62(2):190–197.
- [Russell and Norvig, 2002] Russell, S. and Norvig, P. (2002). *Artificial Intelligence: A Modern Approach*. Prentice Hall.
- [Sagrado and Moral, 2003] Sagrado, J. D. and Moral, S. (2003). Qualitative combination of bayesian networks. *International Journal of Intelligent Systems*, 18(2):237 – 249.

- [Schmittlein et al., 1990] Schmittlein, D. C., Kim, J., and Morrison, D. G. (1990). Combining forecasts: Operational adjustments to theoretically optimal rules. *Management Science*, 36:1044–1056.
- [Shachter, 1984] Shachter, R. D. (1984). Evaluating influence diagrams. *Operations Research*, 34(6):871–872.
- [Shafer, 1976] Shafer, G. (1976). *A Mathematical Theory of Evidence*. Princeton University Press.
- [Spirtes et al., 1993] Spirtes, P., Glymour, C., and Scheines, R. (1993). *Causation, Prediction, and Search*. Springer Verlag, New York.
- [Tham et al., 2003] Tham, C. K., Heng, C. K., and Chin, W. C. (2003). Predicting risk of coronary artery disease from dna microarray-based genotyping using neural networks and other statistical analysis tool. *Journal of Bioinformatics and Computational Biology*, 1(3).
- [Turoff and Linstone, 2002] Turoff, M. and Linstone, H. (2002). *The Delphi Method: Techniques and Applications*. Addison Wesley, Reading, Mass.
- [Wang and Druzdzel, 2000] Wang, H. and Druzdzel, M. J. (2000). User interface tools for navigation in conditional probability tables and elicitation of probabilities in bayesian networks. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufman Publishers, San Francisco, CA.
- [Winkler, 1968] Winkler, R. L. (1968). The consensus of subjective probability distributions. *Management Science*, 15:61–75.
- [Winkler and Clemen, 1986] Winkler, R. L. and Clemen, R. T. (1986). Expert resolution. *Management Science*, 32:298–303.
- [Winkler and Clemen, 1992] Winkler, R. L. and Clemen, R. T. (1992). The combination of forecasts. *Journal Royal Statistic Society Series*, 146:150–157.

-
- [Xiang, 1995] Xiang, Y. (1995). Distributed scheduling of multiagent communication. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, pages 390–398, San Francisco, CA.
- [Xiang et al., 1993] Xiang, Y., Poole, D., and Beddoes, M. P. (1993). Multiply sectioned bayesian networks and junction forests for large knowledge based systems. *Computational Intelligence*, 9(2):171–220.

Appendix A

Glossary

| | |
|-------|--|
| BN | Bayesian network |
| CI | Conditional independence |
| CAD | Coronary Artery Disease |
| CBN | Candidate Bayesian network |
| CPD | Conditional probability distribution |
| DAG | Directed acyclic graph |
| EU | Expected utility |
| G | Graph |
| GeNIe | Graphical Network Interface |
| IBN | Interval Bayesian network |
| ID | Influence diagram |
| JPD | Joint probability distribution |
| M | Model |
| MEU | Maximum expected utility |
| MEBN | Multi-entity Bayesian network |
| MSBN | Multiply sectioned Bayesian networks |
| PGM | Probabilistic graphical model |
| PGMC | Probabilistic graphical model combination |
| PNL | Probabilistic Networks Library |
| SMILE | Standard Modeling, Inference and Learning Engine |
| TO | Target Ordering |

Appendix B

List of Notation

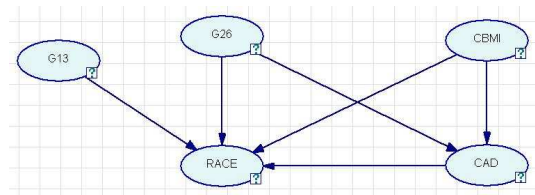
| | | |
|-----------------|-------|--------------------------------|
| $\text{anc}(A)$ | | ancestor nodes of node A |
| $\text{pa}(A)$ | | parent nodes of node A |
| $X \setminus Y$ | | X minus Y (set difference) |
| \cap | | Set intersection |
| \cup | | Set union |

Appendix C

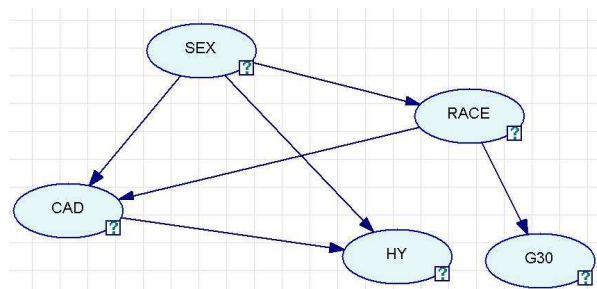
Experimental Data

C.1 The Heart Disease Bayesian Network Models

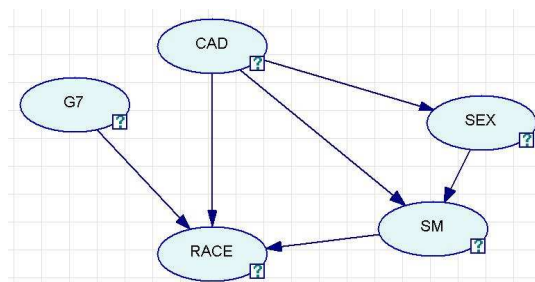
To evaluate different methods proposed in our project, we use over 30 probabilistic models that are learned from some *Heart Disease* data sets [Tham et al., 2003], which include the medical profiles of 2900 human subjects. Of these, almost half of them were healthy at the time of data collection. For each human subject, there are values for 41 attributes, including both genotype (i.e., genetic attributes with respect to the gene concerned) and phenotype (i.e., non genetic or environmental attributes). Since all outcome models are learned from the same data set, the BN learning results from different approaches or different BN software can satisfy the requirement of our graphical model combination approach very well: 1) each variable with same name among different models denotes the same meaning; 2) the structure of these models are quite different.



(a) Candidate BN 5.1

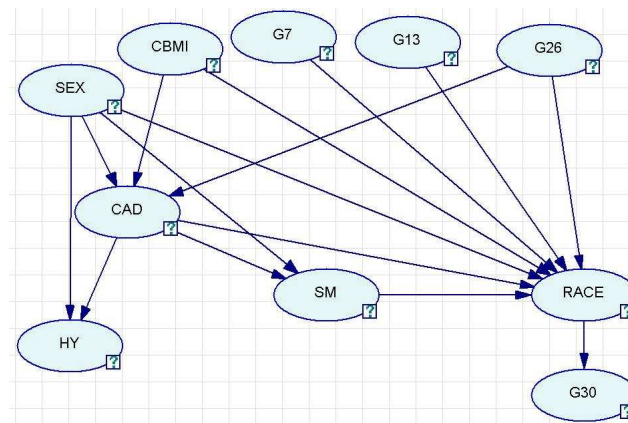


(b) Candidate BN 5.2

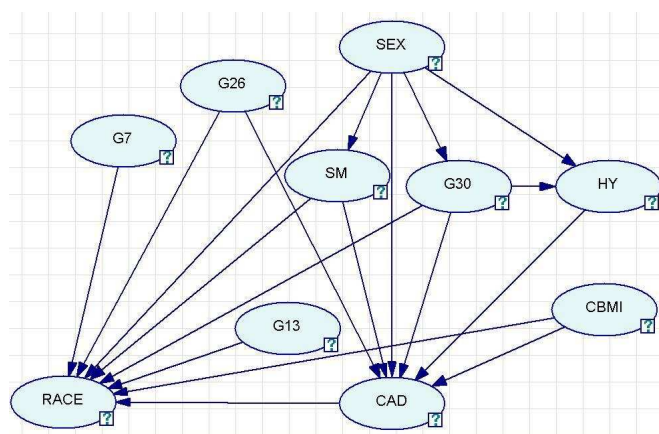


(c) Candidate BN 5.3

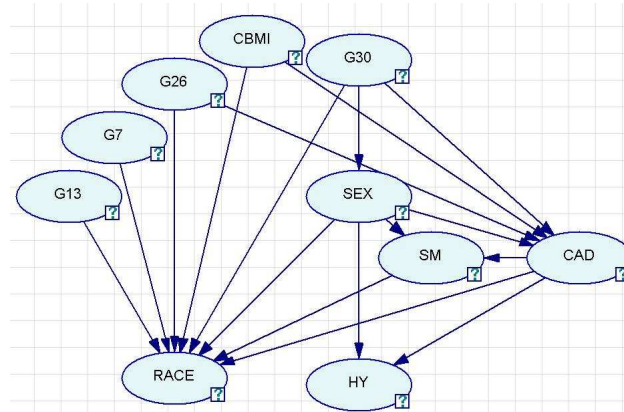
Figure C.1: Three 5-node candidate Bayesian networks



(a) With Method 1



(b) With Method 2



(c) With method 3

Figure C.2: Resulting Bayesian networks with 3 methods in combination of three 5-node CBN

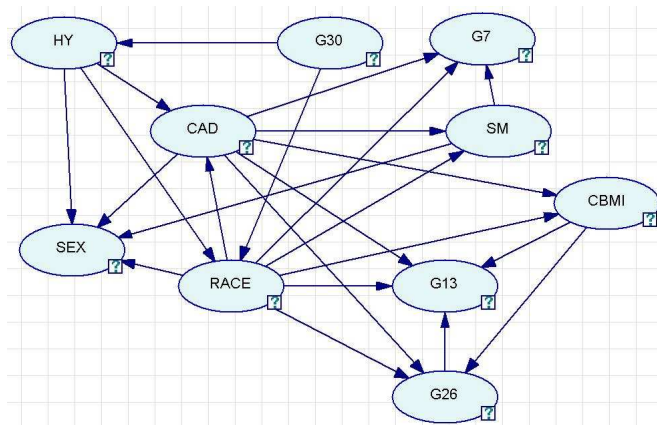
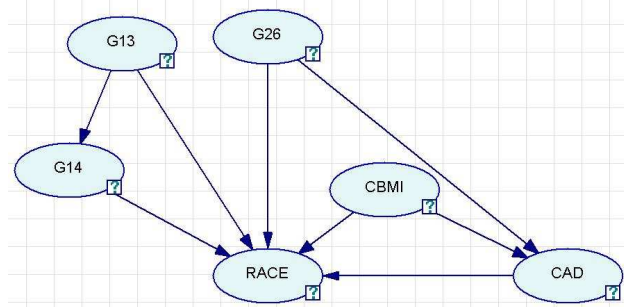
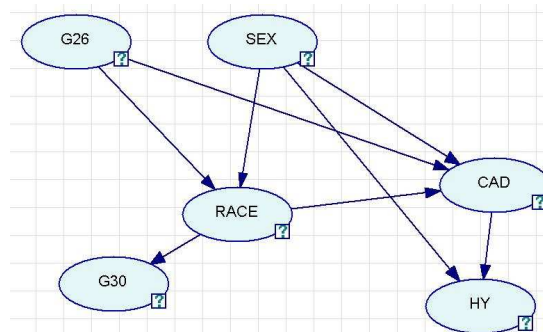


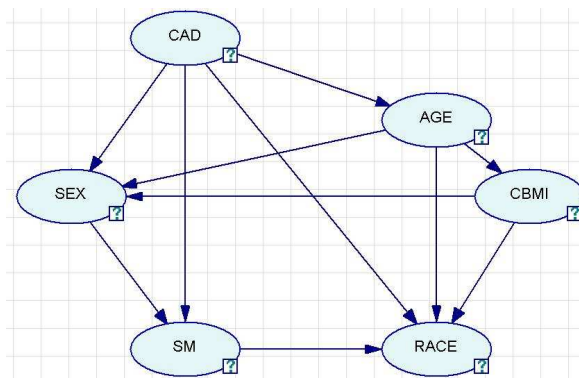
Figure C.3: Resulting BN with a random target variable ordering in combination of three 5-node CBN



(a) Candidate BN 6.1

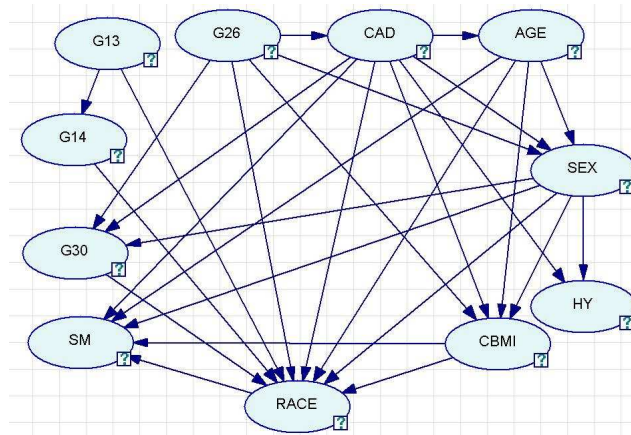


(b) Candidate BN 6.2

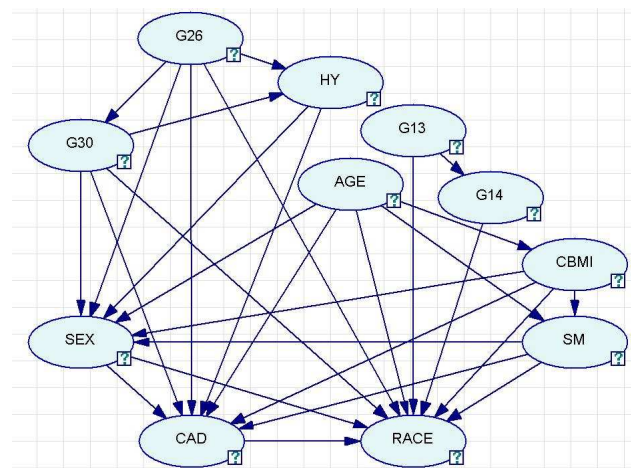


(c) Candidate BN 6.3

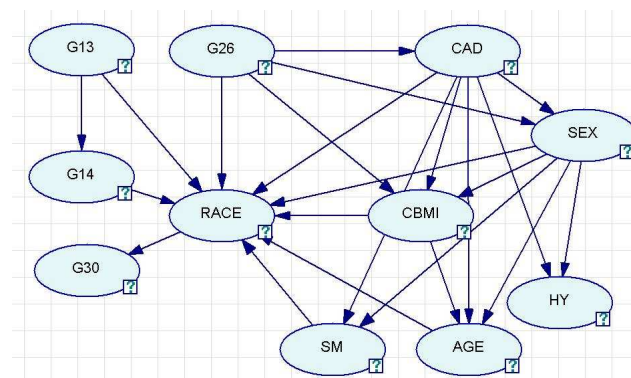
Figure C.4: Three 6-node candidate Bayesian networks



(a) With Method 1



(b) With Method 2



(c) Method 3

Figure C.5: Resulting Bayesian networks with 3 methods in combination of three 6-node CBN

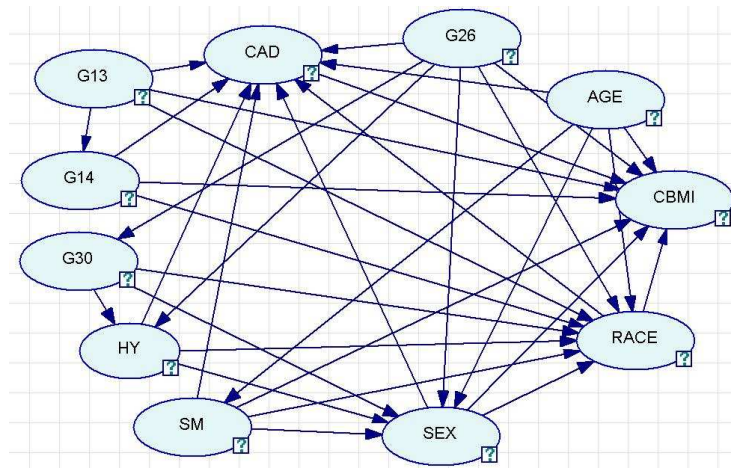
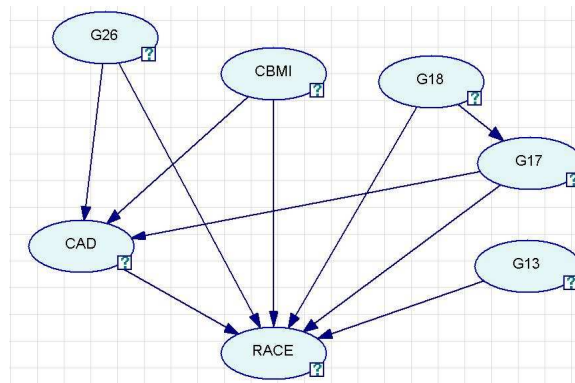
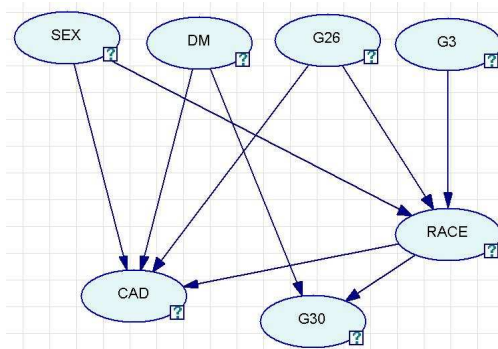


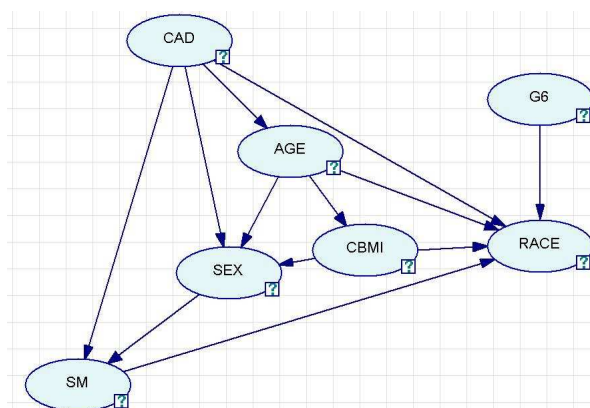
Figure C.6: Resulting BN with a random target variable ordering in combination of three 6-node CBN



(a) Candidate BN 7.1



(b) Candidate BN 7.2



(c) Candidate BN 7.3

| | $CBN_{7.1}$ | $BN_{7.2}$ | $BN_{7.3}$ | $TO_{method1}$ |
|--------------------|-------------|------------|------------|----------------|
| $ordervalue(CAD)$ | 2 | 2 | 0 | 2 |
| $ordervalue(Age)$ | | | 1 | 1 |
| $ordervalue(Race)$ | 3 | 1 | 5 | 5 |
| $ordervalue(CBMI)$ | 0 | | 2 | 1 |
| $ordervalue(SEX)$ | | 0 | 3 | 3 |
| $ordervalue(SM)$ | | | 4 | 6 |
| $ordervalue(DM)$ | | 0 | | 0 |
| $ordervalue(G3)$ | | 0 | | 0 |
| $ordervalue(G6)$ | | | 0 | 0 |
| $ordervalue(G13)$ | 0 | | | 0 |
| $ordervalue(G17)$ | 1 | | | 1 |
| $ordervalue(G18)$ | 0 | | | 0 |
| $ordervalue(G26)$ | 0 | 0 | | 0 |
| $ordervalue(G30)$ | | 2 | | 4 |

| | $CBN_{7.1}$ | $BN_{7.2}$ | $BN_{7.3}$ | $TO_{method2}$ |
|-------------------|-------------|------------|------------|----------------|
| $NumParent(CAD)$ | 3 | 4 | 0 | 4 |
| $NumParent(Age)$ | | | 1 | 1 |
| $NumParent(Race)$ | 6 | 3 | 5 | 5 |
| $NumParent(CBMI)$ | 0 | | 1 | 1 |
| $NumParent(SEX)$ | | 0 | 3 | 3 |
| $NumParent(SM)$ | | | 2 | 2 |
| $NumParent(DM)$ | | 0 | | 0 |
| $NumParent(G3)$ | | 0 | | 0 |
| $NumParent(G6)$ | | | 0 | 0 |
| $NumParent(G13)$ | 0 | | | 0 |
| $NumParent(G17)$ | 1 | | | 1 |
| $NumParent(G18)$ | 0 | | | 0 |
| $NumParent(G26)$ | 0 | 0 | | 0 |
| $NumParent(G30)$ | | 2 | | 2 |

Table C.1: Variable ordering in 7-node candidate Bayesian networks

| <i>End \ Start</i> | <i>CAD</i> | <i>RACE</i> | <i>G13</i> | <i>CBMI</i> | <i>G17</i> | <i>G18</i> | <i>G26</i> |
|--------------------|------------|-------------|------------|-------------|------------|------------|------------|
| <i>CAD</i> | 0 | 1 | -2 | -2 | -1 | -2 | -2 |
| <i>RACE</i> | | 0 | -3 | -3 | -2 | -3 | -3 |
| <i>G13</i> | | | 0 | 0 | 1 | 0 | 0 |
| <i>CBMI</i> | | | | 0 | 1 | 0 | 0 |
| <i>G17</i> | | | | | 0 | -1 | -1 |
| <i>G18</i> | | | | | | 0 | 0 |
| <i>G26</i> | | | | | | | 0 |

(a) Edge matrix of $CBN_{7.1}$

| <i>End \ Start</i> | <i>CAD</i> | <i>Race</i> | <i>SEX</i> | <i>DM</i> | <i>G3</i> | <i>G26</i> | <i>G30</i> |
|--------------------|------------|-------------|------------|-----------|-----------|------------|------------|
| <i>CAD</i> | 0 | -1 | -2 | -2 | -2 | -2 | 0 |
| <i>Race</i> | | 0 | -1 | -1 | -1 | -1 | 1 |
| <i>SEX</i> | | | 0 | 0 | 0 | 0 | 2 |
| <i>DM</i> | | | | 0 | 0 | 0 | 2 |
| <i>G3</i> | | | | | 0 | 0 | 2 |
| <i>G26</i> | | | | | | 0 | 2 |
| <i>G30</i> | | | | | | | 0 |

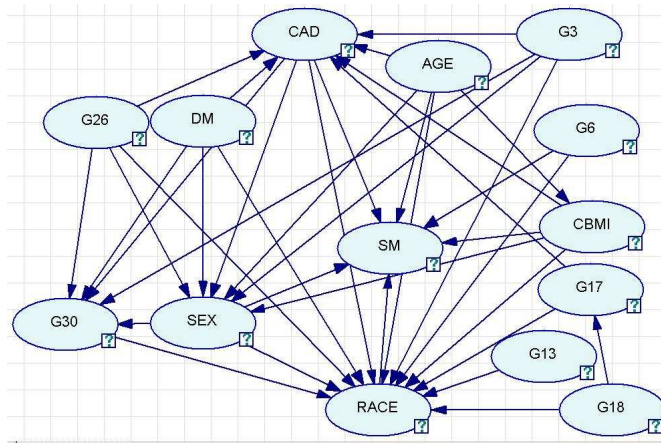
(b) Edge matrix of $CBN_{7.2}$

| <i>End \ Start</i> | <i>CAD</i> | <i>Race</i> | <i>SEX</i> | <i>SM</i> | <i>AGE</i> | <i>CBMI</i> | <i>G6</i> |
|--------------------|------------|-------------|------------|-----------|------------|-------------|-----------|
| <i>CAD</i> | 0 | 5 | 3 | 4 | 1 | 2 | 0 |
| <i>Race</i> | | 0 | -2 | -1 | -4 | -3 | -5 |
| <i>SEX</i> | | | 0 | 1 | -2 | -1 | -3 |
| <i>SM</i> | | | | 0 | -3 | -2 | -4 |
| <i>AGE</i> | | | | | 0 | 1 | -1 |
| <i>CBMI</i> | | | | | | 0 | -2 |
| <i>G6</i> | | | | | | | 0 |

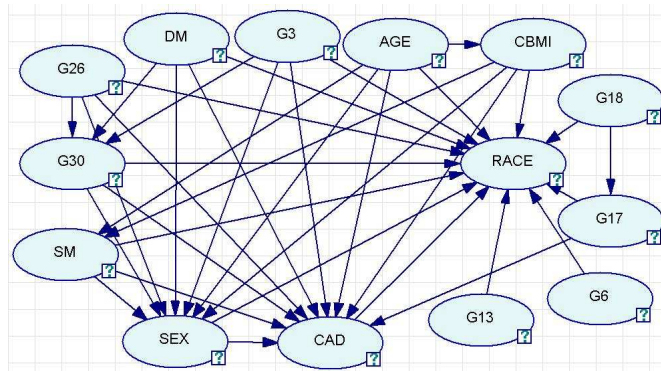
(c) Edge matrix of $CBN_{7.3}$

| <i>End \ Start</i> | <i>CAD</i> | <i>Race</i> | <i>CBMI</i> | <i>SEX</i> | <i>SM</i> | <i>DM</i> | <i>AGE</i> | <i>G3</i> | <i>G6</i> | <i>G13</i> | <i>G17</i> | <i>G18</i> | <i>G26</i> | <i>G30</i> |
|--------------------|------------|-------------|-------------|------------|-----------|-----------|------------|-----------|-----------|------------|------------|------------|------------|------------|
| <i>CAD</i> | 0 | 5 | 0* | 1 | 4 | -2 | 1 | -2 | 0 | -2 | -1 | -2 | -4 | 0 |
| <i>Race</i> | | 0 | -6 | -3 | -1 | -1 | -4 | -1 | -5 | -3 | -2 | -3 | -4 | 1 |
| <i>CBMI</i> | | | 0 | 1 | 2 | 0 | -1 | 0 | -2 | 0 | 1 | 0 | 0 | 0 |
| <i>SEX</i> | | | | 0 | 1 | 0 | -2 | 0 | -3 | 0 | 0 | 0 | 0 | 2 |
| <i>SM</i> | | | | | 0 | 0 | -3 | 0 | -4 | 0 | 0 | 0 | 0 | 0 |
| <i>DM</i> | | | | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| <i>AGE</i> | | | | | | | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 0 |
| <i>G3</i> | | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| <i>G6</i> | | | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| <i>G13</i> | | | | | | | | | | 0 | 1 | 0 | 0 | 0 |
| <i>G17</i> | | | | | | | | | | | 0 | -1 | -1 | 0 |
| <i>G18</i> | | | | | | | | | | | | 0 | 0 | 0 |
| <i>G26</i> | | | | | | | | | | | | | 0 | 2 |
| <i>G30</i> | | | | | | | | | | | | | | 0 |

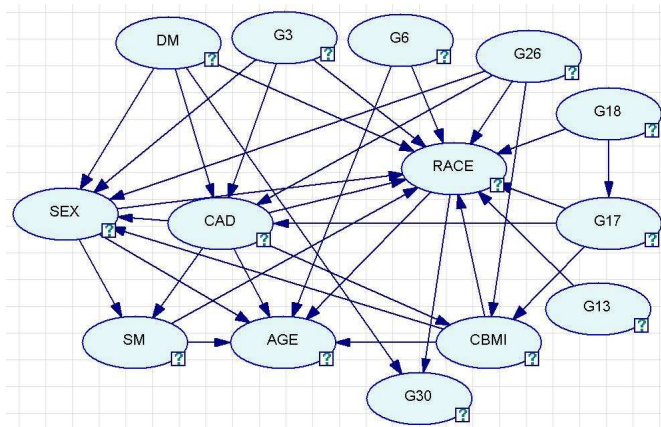
(d) Edge matrix of resulting Bayesian network



(a) With Method 1



(b) With Method 2

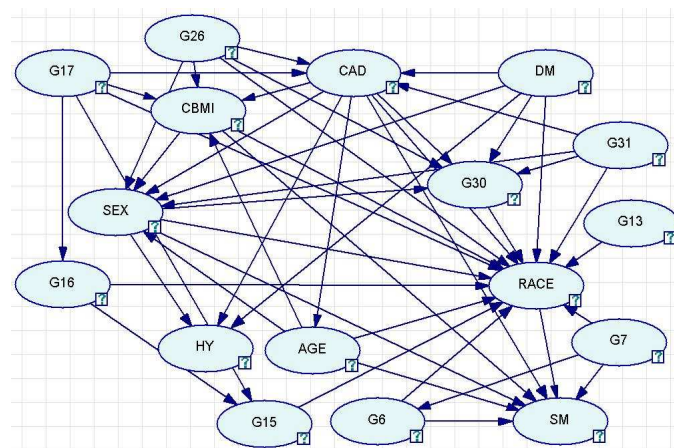


(c) With Method 3

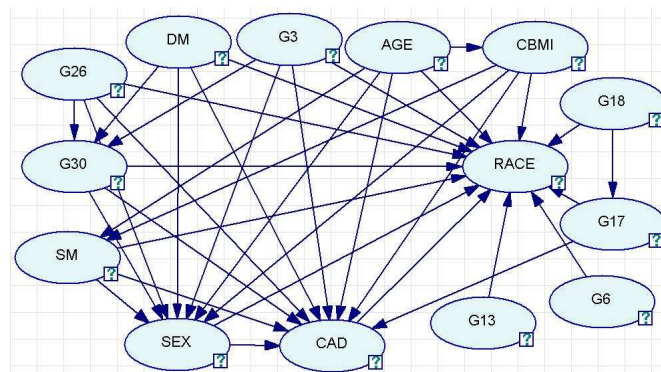
Figure C.8: Resulting Bayesian networks with 3 methods in combining of three 7-node CBN

| | $CBN_{8,1}$ | $CBN_{8,2}$ | $CBN_{8,3}$ | $TO_{method1}$ | | $CBN_{8,1}$ | $CBN_{8,2}$ | $CBN_{8,3}$ | $TO_{method2}$ |
|-------------|-------------|-------------|-------------|----------------|-------------|-------------|-------------|-------------|----------------|
| <i>CAD</i> | 1 | 2 | 0 | 1 | <i>CAD</i> | 3 | 4 | 0 | 4 |
| <i>RACE</i> | 3 | 1 | 5 | 5 | <i>RACE</i> | 7 | 3 | 6 | 5 |
| <i>CBMI</i> | 0 | | 2 | 2 | <i>CBMI</i> | 0 | | 2 | 2 |
| <i>SEX</i> | | 0 | 3 | 3 | <i>SEX</i> | | 0 | 3 | 3 |
| <i>HY</i> | | 3 | | 5 | <i>HY</i> | | 3 | | 3 |
| <i>SM</i> | | | 4 | 6 | <i>SM</i> | | | 2 | 2 |
| <i>DM</i> | | 0 | | 0 | <i>DM</i> | | 0 | | 0 |
| <i>AGE</i> | | | 1 | 2 | <i>AGE</i> | | | 1 | 1 |
| <i>G6</i> | | | 1 | 2 | <i>G6</i> | | | 1 | 1 |
| <i>G7</i> | | | 0 | 0 | <i>G7</i> | | | 0 | 0 |
| <i>G13</i> | 0 | | | 0 | <i>G13</i> | 0 | | | 0 |
| <i>G15</i> | 2 | | | 4 | <i>G15</i> | 2 | | | 2 |
| <i>G16</i> | 1 | | | 2 | <i>G16</i> | 1 | | | 1 |
| <i>G17</i> | 0 | | | 0 | <i>G17</i> | 0 | | | 0 |
| <i>G26</i> | 0 | 0 | | 0 | <i>G26</i> | 0 | 0 | | 0 |
| <i>G30</i> | | 2 | | 4 | <i>G30</i> | | 3 | | 3 |
| <i>G31</i> | | 0 | | 0 | <i>G31</i> | | 0 | | 0 |

Table C.3: Variable ordering in 8-node candidate Bayesian networks

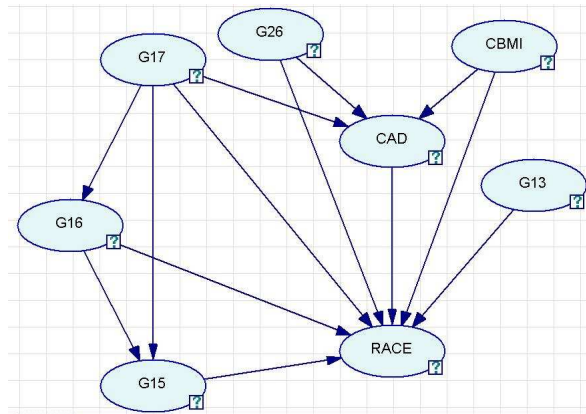


(a) With Method 1

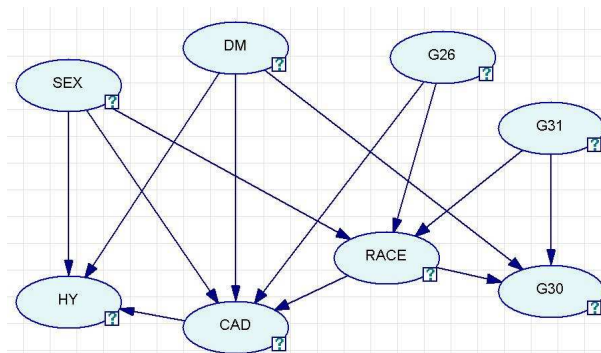


(b) With Method 2

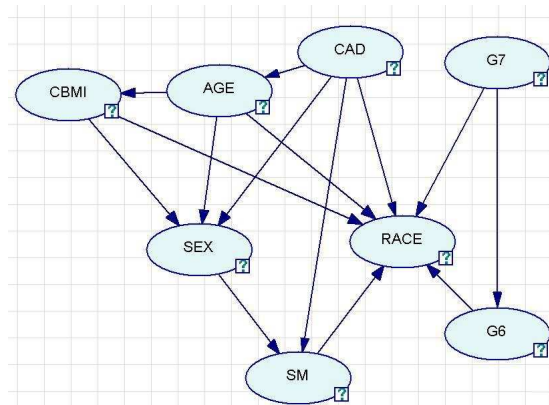
Figure C.9: Resulting Bayesian networks with 3 methods in combining three 8-node CBN



(a) Candidate BN 8.1

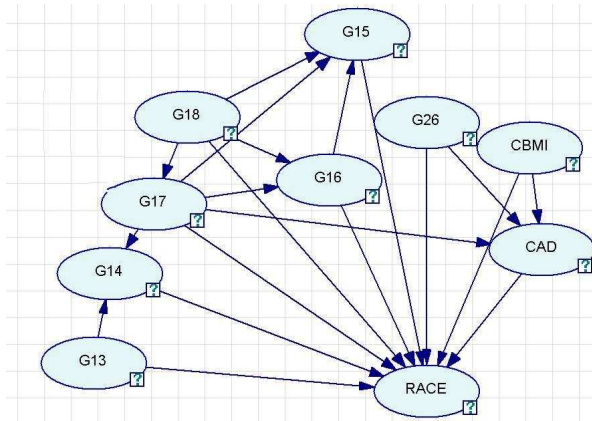


(b) Candidate BN 8.2

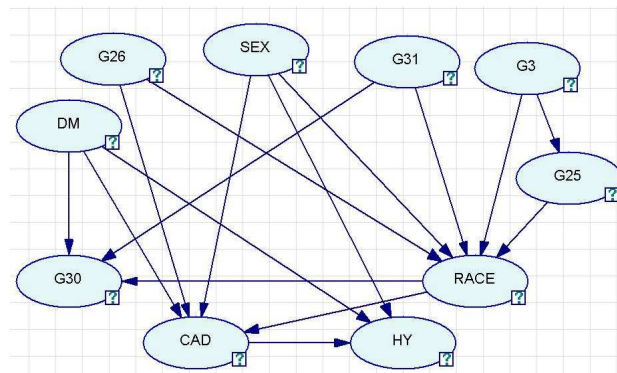


(c) Candidate BN 8.3

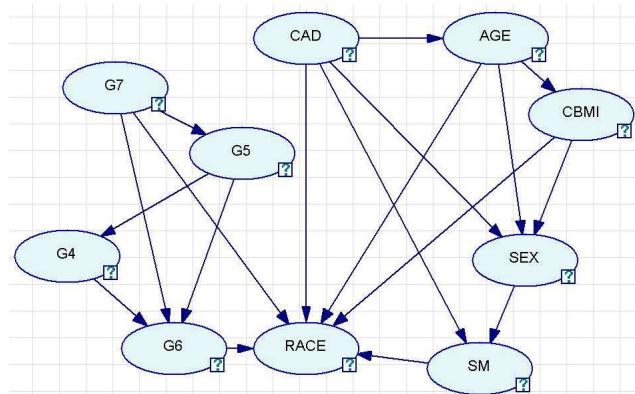
Figure C.10: Three 8-node candidate Bayesian networks



(a) Candidate BN 10.1

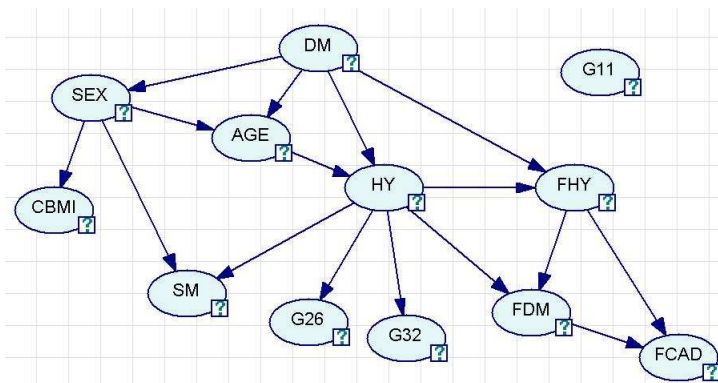


(b) Candidate BN 10.2

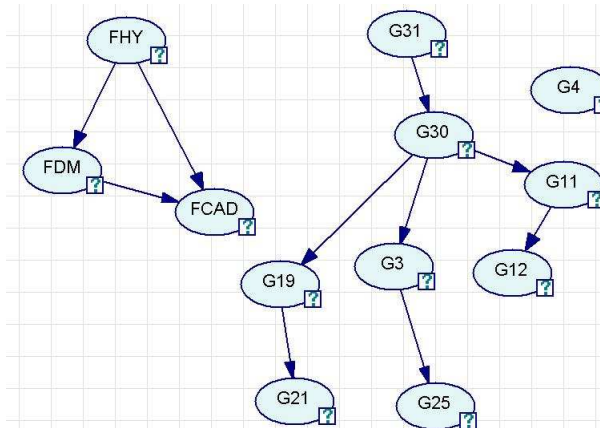


(c) Candidate BN 10.3

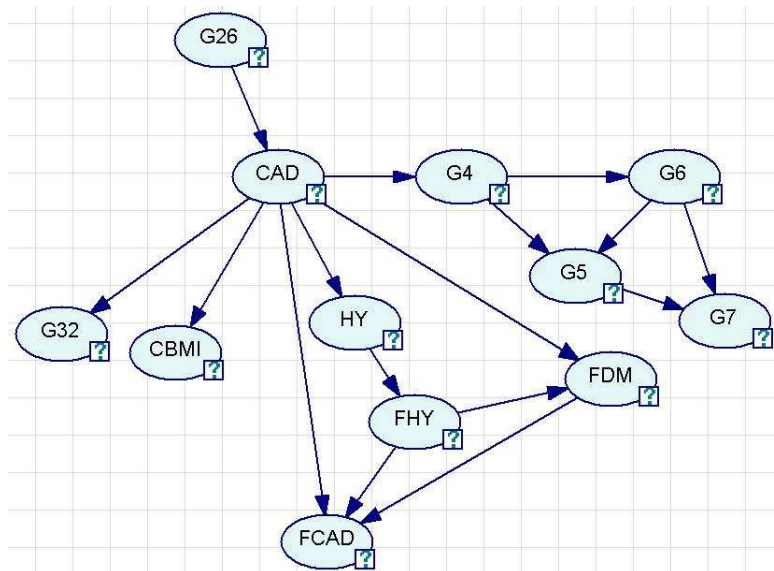
Figure C.11: Three 10-node candidate Bayesian networks



(a) Candidate BN 12.1



(b) Candidate BN 12.2



(c) Candidate BN 13.3

Figure C.12: Three 12-node candidate Bayesian networks

C.2 Probability Distributions from Different Knowledge Sources

In order to describe clearly we use

- θ_{1t} to denote the probability of survival of *Tom* under the condition of $\{ShareNerve=Y, ShareArtery=Y, ShareVein=Y\}$,
- θ_{1s} denotes the probability of survival of *Smith* under the condition of $\{ShareNerve=Y, ShareArtery=Y, ShareVein=Y\}$.
- θ_{2t} denote the probability of survival of *Tom* under the condition of $\{ShareNerve=Y, ShareArtery=Y, ShareVein=N\}$;
- θ_{2s} denote the probability of survival of *Smith* under the condition of $\{ShareNerve=Y, ShareArtery=Y, ShareVein=N\}$;
- θ_{3t} denote the probability of survival of *Tom* under the condition of $\{ShareNerve=Y, ShareArtery=N, ShareVein=Y\}$;
- θ_{3s} denote the probability of survival of *Smith* under the condition of $\{ShareNerve=Y, ShareArtery=N, ShareVein=Y\}$;
- θ_{3t} denote the probability of survival of *Tom* under the condition of $\{ShareNerve=Y, ShareArtery=N, ShareVein=Y\}$;
- θ_{3s} denote the probability of survival of *Smith* under the condition of $\{ShareNerve=Y, ShareArtery=N, ShareVein=Y\}$;
- θ_{4t} denote the probability of survival of *Tom* under the condition of $\{ShareNerve=Y, ShareArtery=N, ShareVein=N\}$;
- θ_{4s} denote the probability of survival of *Smith* under the condition of $\{ShareNerve=Y, ShareArtery=N, ShareVein=N\}$;
- θ_{5t} denote the probability of survival of *Tom* under the condition of $\{ShareNerve=N, ShareArtery=Y, ShareVein=Y\}$;

- θ_{5s} denote the probability of survival of *Smith* under the condition of $\{ShareNerve=N, ShareArtery=Y, ShareVein=Y\}$;
- θ_{6t} denote the probability of survival of *Tom* under the condition of $\{ShareNerve=N, ShareArtery=Y, ShareVein=N\}$;
- θ_{6s} denote the probability of survival of *Smith* under the condition of $\{ShareNerve=N, ShareArtery=Y, ShareVein=N\}$;
- θ_{7t} denote the probability of survival of *Tom* under the condition of $\{ShareNerve=N, ShareArtery=N, ShareVein=Y\}$;
- θ_{7s} denote the probability of survival of *Smith* under the condition of $\{ShareNerve=N, ShareArtery=N, ShareVein=Y\}$;
- θ_{8t} denote the probability of survival of *Tom* under the condition of $\{ShareNerve=N, ShareArtery=N, ShareVein=N\}$;
- θ_{8s} denote the probability of survival of *Smith* under the condition of $\{ShareNerve=N, ShareArtery=N, ShareVein=N\}$;