KNOWLEDGE DISCOVERY
WITH BAYESIAN NETWORKS

BY

LI GUOLIANG

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

AT

DEPARTMENT OF COMPUTER SCIENCE

SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF SINGAPORE

COMPUTING 1, LAW LINK, SINGAPORE 117590

JANUARY, 2009

Acknowledgement

I owe a great debt to many people who assisted me in my graduate education. I would like to take this opportunity to cordially thank:

Associate Professor Tze-Yun Leong, my thesis supervisor, in School of Computing, National University of Singapore, for her guidance, patience, encouragement, and support throughout my years of graduate training. Especially when I wavered amongst different topics, her encouragement and support were very important to me. I would not have made it through the training without her patience and belief in me.

Associate Professor Louxin Zhang in Department of Mathematics, National University of Singapore, for his detailed and constructive discussions in Bioinformatics problems. His expertise in phylogenetics has enlightened me the application of Bayesian analysis in ancestral state reconstruction accuracy.

Members and alumni of the Medical Computing Lab and the Biomedical Decision Engineering (Bide) group: Associate Professor Kim-Leng Poh, Dr Han Bin, Rohit Joshi, Chen Qiong Yu, Yin Hong Li, Zhu Ai Ling, Zeng Yi Feng, Wong Swee Seong, Lin Li, Ong Chen Hui, Dinh Thien Anh, Vu Xuan Linh, Dinh Truong Huy Nguyen, Sreeram Ramachandran, for their caring advice, insightful comments and suggestions.

Mr. Guo Wen Yuan for his broad discussion of philosophical issues and his recommendation of the book "*Philosophical theories of probability*" by Donald Gillies. This book was very helpful in enlightening me the different philosophical

perspectives of probability.

Dr Chew-Kiat Heng for his kindness to share the heart disease data with me.

Dr Qiu Wen Jie for sharing his biological domain knowledge in Actin cytoskeleton genes of yeast with me.

Dr. Qiu Long for taking his precious time to proofread my thesis.

Singapore-MIT Alliance (SMA) classmates: Zhao Qin, Yu Bei, Qiu Long, Qiu Qiang, Edward Sim, Ou Han Yan and Yu Xiao Xue. The discussion with them is broad and insightful for my research.

Finally, I owe a great debt to my family: my parents, my sisters, my daughter Wei Hang, and especially to my wife Wang Hui Qin for their love and support.

Table of Contents

Summary

Causal knowledge is essential for comprehension, diagnosis, prediction, and control in many complex situations. Identification of causal knowledge is an important research topic with a long history and many challenging issues. The majority of existing approaches to causal knowledge discovery are based on statistical randomized experiments and inductive learning from observational data.

This thesis proposes a three-step iterative framework for causal knowledge discovery with Bayesian networks under a manipulation criterion. Its goal is to exploit available resources, including observational data, interventional data, topological domain knowledge, and interventional experiments, to discover new causal knowledge, and minimize the number of interventional experiments required to validate the causal knowledge. The main challenges are in automatically generating new hypotheses of causal knowledge, systematically incorporating domain knowledge for hypothesis refinement, and effectively selecting hypotheses for verification.

Direct causal influence relationships between variables are regarded as hypotheses and are modeled as edges of causal Bayesian networks. The statistical significance of the hypotheses of the direct causal influence relationships between variables can be estimated from data with Bayesian network structure learning. We propose variable grouping as a new method for hypothesis generation; this method partitions the variables with similar conditional probabilities into groups to support learning of the Bayesian network structures simultaneously.

Domain knowledge is specified as topological constraints in Bayesian network structure learning for hypothesis refinement. We propose two canonical formats to model topological domain knowledge. The effects of different topological constraints are examined experimentally.

The hypotheses of the direct causal relationships between variables from data can be verified with interventional experiments. The situation with multiple data instances collected in each intervention step is first considered. We propose node-based interventions to establish the causal ordering of variables and edge-based interventions to examine the direct causal relationships between variables, propose non-symmetrical entropy from the available data as a selection measure to rank the hypotheses for verification, and propose structure entropy as a criterion to stop the active learning process.

The proposed methods build on and extend various well-established algorithms for the respective tasks. The different tasks are integrated in a systematic way to support cost-effective causal knowledge discovery. Promising results are shown in a set of synthetic and benchmark Bayesian networks with practical implications. In particular, we illustrate the effectiveness of the proposed methods in a class of problems where: i) variable grouping groups the similar variables together and generates relevant hypotheses; ii) hypothesis refinement with topological domain knowledge improves the relevance of the generated hypotheses; and iii) non-symmetrical entropy from the data reduces the computational cost and leads to minimal interventional experiments to validate causal knowledge. The proposed

framework is applicable to many domains for causal knowledge discovery, such as in reverse engineering tasks.

Keywords: Causal knowledge, Bayesian networks, knowledge discovery, hypothesis generation, hypothesis refinement, hypothesis verification, observational data, interventional data, non-symmetrical entropy, active learning

List of Tables

List of Figures

Glossary of Terms

AODE: Aggregating One-Dependence Estimators 21
BD metric: Bayesian Dirichlet metric 37
BDe metric: Bayesian metric with Dirichlet priors and equivalence 37
BIC: Bayesian Information Criterion 36
BN: Bayesian network 4
CAD: coronary artery disease 53
Causal knowledge: the cause-and-effect relationship between different events 5
CBMI: current body-mass index 54
cDNA: complementary Deoxyribonucleic acid 18
CPD: conditional probability distribution 27
DAG: directed acyclic graph 26
DIC: Dynamic Itemset Counting 23
EM: expectation-maximization algorithm 31
HMM: Hidden Markov Model 23
IC algorithm: Inductive Causation algorithm 43
KDD: Knowledge discovery in database 2
MAR: Missing-At-Random 32
MCMC: Markov Chain Monte Carlo 34
MDL: Minimum description Length 37
MH: Metropolis-Hastings 40
MI: mutual information 186
ML: maximum likelihood 36
NP: non-deterministic polynomial time 34
PC algorithm: A Bayesian network structure learning algorithm named after its authors P. Spirtes, C. Glymour 34
QMR-DT: Quick Medical Reference (Decision-Theoretic) Network 30
SGS algorithm: a Bayesian network structure leaning algorihtm named after its authors P. Spirtes, C. Glymour, R. Scheines 34
SNP: single nucleotide polymorphism 101
SVM: Support vector machine 21

$\mathbf{X} = \{X_1, ..., X_n\}$: A finite set of random variables

X_i, X_j: Specific random variables

$Val(X_i)$: A finite set of values discrete variable X_i can take

X, Y, Z, W: Different variables

x, y, z, w: Specific values of variables X, Y, Z, W

x_1, x_2: Different values of variable X

$\hat{x}$, $\hat{z}$: Specific values variables X and Z are manipulated to
A, B: Different variables

a_1, a_2: Different values of variable A

$do(A = a_1)$: Manipulating variable A to a specific value a_1

D: A data set
N: The number of data instances in a data set
n: The number of variables in a domain
m: The number of groups in a domain for variable grouping
K: Background knowledge or domain knowledge
G: A Bayesian network

G_0: An initial Bayesian network structure

$E(G)$: The set of edges in Bayesian network G

$Pa(X_i)$: The parents of variable X_i in Bayesian network G

$p(X_1,..., X_n)$: Joint probability distribution of a domain with variables $\mathrm{X} = \{X_1,..., X_n\}$

$p(X_i \mid Pa(X_i))$: The conditional probability X_i given its parents $Pa(X_i)$

$p(Y \mid X = x_1)$: The conditional probability of variable Y given that variable X is observed with value x_1

$p(Y \mid do(X = x_1))$: The conditional probability of variable Y given that variable X is manipulated to value x_1

$p(B \mid do(A = a_1))$: The conditional probability of variable B when variable A is manipulated to value a_1

$p(D)$: The probability of data D

$X_i \rightarrow X_j$: An edge from X_i to X_j

$X_i \leftarrow X_j$: An edge from X_j to X_i

$X_i \perp X_j$: No edge between X_i and X_j

$H(X_i \leftrightarrow X_j)$: Edge entropy between X_i and X_j

$H_S(A,B)$: Symmetrical edge entropy between variables A and B

$H_{NS}(A \rightarrow B)$: Non-symmetrical edge entropy between variables A and B

$H_S(A)$: Symmetrical node entropy of variable A

$H_{NS}(A)$: Non-symmetrical node entropy of variable A

$H_S(G)$: Structure entropy of Bayesian network G

$N(0,1)$: Normal distribution with 0 mean and unit standard deviation

Chapter 1 Introduction

["... Knowledge Discovery is the most desirable end-product of computing. Finding new phenomena or enhancing our knowledge about them has a greater long-range value than optimizing production processes or inventories, and is second only to task that preserve our world and our environment. It is not surprising that it is also one of the most difficult computing challenges to do well. ..."] – Gio Wiederhold (1996) [170]

Knowledge is used in every scenario of our life for comprehension, diagnosis, prediction and control. Causal knowledge is important for dealing with complex problems and representing knowledge more logically, and especially useful in manipulating current systems for expected effects or re-engineering current systems to create new systems. Discovering new causal knowledge from observations is a sustaining and continuing effort of human beings. Generally, knowledge discovery involves several steps such as data (or observation) analysis and hypothesis generation. Usually, these steps are studied separately in the literature and the connections among them are harder to identify. A unified framework that would integrate these steps and facilitate knowledge discovery is needed.

My research is about knowledge discovery with observational data, interventional data, domain knowledge and interventional experiments. A three-step framework for causal knowledge discovery with Bayesian networks is proposed. The steps include: *hypothesis generation*, *hypothesis refinement*, and *hypothesis verification*. In this framework, hypotheses are the direct causal influence relationships between variables

and are modeled as edges of Bayesian networks. Observational data and interventional data are used to generate hypotheses (selecting the possible causal relationships between variables with statistical significance), domain knowledge is used to refine the generated hypotheses, and interventional experiments are suggested to verify the top-ranked hypotheses for knowledge discovery.

The application of this framework is shown on problems in biomedical domains. The experiments show that for this class of problems, the framework and its algorithms can make use of all available resources and facilitate the knowledge discovery process: sound hypotheses can be generated from data with Bayesian network structure learning, domain knowledge can improve the validity of hypotheses generated from data, and non-symmetrical entropy can minimize the number of interventional experiments to verify the hypotheses in a domain.

1.1 Background and Motivation

With advanced information technology, we are using more sensors and electronic recording devices in various fields, collecting and storing more data in databases. With these accumulated data, people are able to utilize them to unearth patterns in the domain, which can be used as new knowledge after verification. This process is known as *knowledge discovery in databases*.

There are different definitions for knowledge discovery in database. According to the widely-cited definition by Fayyad, Piatetsky-Shapiro and Smyth [54]: "*knowledge discovery in database (KDD) is the nontrivial process of identifying valid, potentially*

useful, and ultimately understandable patterns in data". This definition is well-known for its emphasis on the properties of new knowledge discovered from data.

Research in Computer Science, Statistics, Database and other disciplines has led to various techniques for knowledge discovery. Classification, regression, clustering and association rule mining are four representative tasks in knowledge discovery and the discovered knowledge is represented in different patterns based on the tasks. Patterns in classification and regression reflect the relationships between one target variable and all other variables[1]. Patterns in clustering reflect the similarities among some part of the data to distinguish them from other parts of the data. Association rule mining is used to identify items frequently occurring together in different scenarios. In practice, the majority of these tasks are often applied to correlational relationship discovery from observational data.

Besides the patterns mentioned above, an important pattern in many domains is causal relationships between variables – the entire set of direct influence[2] relationships between variables in a domain. Causal relationship is an indispensable part of our life and causal knowledge is essential to dealing with complex situations and summarizing results more logically [143]. **Causal knowledge** is the superset of the causal relationship between variables. It is crucial for the manipulation of the system to achieve the expected effects and crucial for the re-engineering process to

[1] The target variables in classification are categorical variables and the target variables in regression are continuous variables.

[2] In this thesis, the "influence" means the "causal influence". If variable A influences variable B, it means that variable A is a cause of variable B. Refer to the definition of causal knowledge in Section 1.1.1 for details.

create new systems from the existing systems, such as in Engineering, Biology and Economics. A critical problem in the re-engineering process is to predict the behavior (or property) of the new system before re-engineering. Such prediction cannot be done merely with the correlation relationships between variables from observational data. We need to know which properties of the system will remain unchanged after re-engineering and how other properties will change. Causal knowledge can model these properties as the structural invariance and the manipulation invariance of the system, and tell us how the properties change after manipulation.

The focus of this thesis is on the discovery of patterns that can be represented as **causal relationships** – direct causal influence relationships between variables in a domain. **Correlational relationships** are mainly the association between variables from observational data, and are not causal relationships in general, although such information may be used as the initial hypotheses of causal knowledge before verification with interventional experiments.

One approach to modeling causal influence relationships between variables in a domain is Bayesian networks (BNs). The goal of this work is to discover causal knowledge represented by Bayesian networks from observational data, interventional data, topological domain knowledge and interventional experiments. The main challenges are to generate the hypotheses of causal relationships from data, to refine the hypotheses with domain knowledge and to minimize the number of interventional experiments needed to verify the hypotheses. I argue that the combination of observational and interventional data can effectively and economically discover

causal relationships.

1.1.1 Causal Knowledge

Causal knowledge captures the cause-and-effect relationship between different events. The study of causal knowledge has a long history. Aristotle spoke of the doctrine of four causes, while others proposed different forms of causality afterwards [90,106,130,155,171]. In this thesis, I follow the definition from Spirtes *et al*. [155] and consider causal knowledge from a probabilistic perspective with a **manipulation criterion** (refer to [155], Section 3.7.2):

> **Definition of causal relationship** (Spirtes *et al*. [155]): *Suppose we can manipulate the variables in a domain and* A *and* B *are two variables in the domain; If 1) we manipulate variable* A *to different values* a_1 *or* a_2*, 2) measure the effects on variable* B*, and 3) observe the changes in the probability distribution of variable* B *under different values of variable* A*,*
>
> $$p(B \mid do(A = a_1)) \neq p(B \mid do(A = a_2)),$$
>
> *we say that variable* A *causally influences variable* B*, variable* A *is a (direct or indirect) cause of variable* B*, and variable* B *is an effect of variable* A*. The operator* $do()$ *is from Pearl's book "Causality" [130], and* $do(A = a_1)$ *means that variable* A *is manipulated to a specific value* a_1*, rather than observed with value* a_1 *from observational data.*

The reason I adopt this definition of causal relationship is that this definition is general and operational, and this kind of causal knowledge can be verified by

experiments with manipulation.

The main scientific method for causal knowledge discovery from data relies on randomized experiments in statistics discipline [58,125,144]. The interventional data is collected in randomized experiments to infer causal strength of the randomized variables on other variables. However, the problem of hypothesis generation is not discussed in experiment design in statistics, even though the hypothesis is most important as the *starting point* of the experiment design.

1.1.2 Causal Knowledge Discovery with Bayesian Networks

Bayesian networks are graphical models that can be used to represent causal knowledge as the probabilistic causal relationships between variables in a domain and model multiple direct causal influence relationships simultaneously. Judea Pearl [130,131] and Spirtes *et al*. [155,156] have developed a comprehensive theory for causal knowledge discovery from observational data with Bayesian networks. There are many applications of their work on causal knowledge discovery [73,145,151].

The previous work on Bayesian networks [38,87,132,156] mainly focused on hypothesis generation from data as Bayesian network structure learning problem, which is the process to infer the Bayesian network structure from data with a certain criterion to best explain the data. In this thesis, I will use Bayesian networks to model causal knowledge in a domain, to generate hypotheses of causal relationships from data, to model domain knowledge as topological constraints in Bayesian networks and to select hypotheses for verification with interventional experiments.

It is widely accepted that causal knowledge can be extracted from intervention (when intervention is possible), such as randomized experiments. It is debatable whether causal knowledge can be inferred from observational data alone with Bayesian networks. Spirtes *et al.* [155,156], Pearl [130], and Korb and Wallace [100] are examples of proponents of Bayesian networks for causal knowledge discovery, while CartWright [19,20], Humphreys and Freedman [91], and McKim and Turner [118] represent the opponents. The arguments are more on the assumptions in Bayesian networks – causal Markov assumption and faithfulness assumption, and whether these assumptions are reasonable. In this thesis, I will not discuss this controversial issue – I will take Bayesian networks as a knowledge discovery framework for granted.

1.1.3 Why Bayesian Networks?

The reasons I chose Bayesian networks as the model for knowledge discovery are:

i) Bayesian networks can be used to generate hypotheses of causal relationships from data for causal knowledge discovery, while randomized experiments do not consider hypothesis generation for causal inference in mathematical form;

ii) Bayesian networks can model multiple hypotheses of causal relationships with many target variables simultaneously, while randomized experiments and classification and regression methods only consider one target variable;

iii) Bayesian networks can model joint probability distribution in a domain with fewer parameters, by exploiting conditional independence relationships among variables;

iv) Bayesian networks can explicitly model uncertainty and address noisy and missing data;

v) It is easy to combine prior knowledge (such as causal knowledge) into the structure and parameters of Bayesian networks;

vi) Results from Bayesian network structure learning algorithms can be extended for causal knowledge discovery, especially when interventional data is considered; and

vii) Manipulation methods are available in many domains (such as Biology or Electrical Engineering) to verify the hypotheses generated from Bayesian networks.

1.1.4 Data

The data for knowledge discovery can be divided into two categories by the observation conditions: observational data and interventional data.

i) Observational data – This category of data is observed when the system of interest evolves autonomously and there is no manipulation on the system. A typical example is the system of the Sun, the planets and the stars. Currently (or even in the near future), humans can only observe the movements of the Sun, the planets and the stars and cannot manipulate the system. In Biology, we can observe the expression level of proteins without any reagents added. In Electrical Engineering, we can observe the system working without external signals added.

ii) Interventional data – This category of data is observed when some variables in the system have been manipulated to specific values and other variables evolve simultaneously by following the system's causal mechanism. In Biology, we can

manipulate the expression levels of some genes by knock-out or over-expression experiments, and observe the expression levels of other genes. In Electrical Engineering, we can cut connections in the circuit or add some external signals at some points of the system, and observe the effect on other parts of the system.

The main difference between observational data and interventional data is whether some variables in the system are under manipulation when the data is collected. A **manipulation**[3] is represented by the introduction of an exogenous variable into the current causal system as a cause of the variable to be manipulated. When there is no manipulation, the system functions as normal. When there is manipulation, the relationships between the manipulated variable and its original causes in the system will be changed – the values of the manipulated variables are determined by the manipulation while the values of other variables will be determined by the mechanism of the system. In this way, the relationship between two variables, whether causal or merely correlational, can be verified with interventional data.

Here we need to distinguish the probabilities from different types of data: $p(Y \mid X = x_1)$ from observational data and $p(Y \mid do(X = x_1))$ from interventional data. $p(Y \mid X = x_1)$ means the conditional probability distribution of variable Y given that variable X is observed with value x_1. $p(Y \mid do(X = x_1))$ means the conditional probability distribution of variable Y given that variable X is manipulated to value x_1.

Compared to interventional data, observational data can be collected economically.

[3] For more details of manipulability, refer to the book by J. Woodward, *Making things happen: a theory of causal explanation*, Oxford University Press, 2003.

In some domains, such as in Social Science or Clinical Science, only observational data can be obtained, and intervention on some variables is infeasible due to financial, legal or ethical reasons. This is why most traditional methods for knowledge discovery in database [53,86] only consider observational data, leading to some researchers developing methods to discover causal relationships with observational data [130,143,155].

1.1.5 Hypotheses

The knowledge discovered from data can be represented in different forms, such as rules, differential equations, structural equation models and more [28,81,136,172]. The interest in this thesis is the **direct causal influence relationships between variables**, which can be represented as Bayesian network structures. The process used to discover new knowledge is equivalent to learning of Bayesian network structures. Directed *edges* in the learned Bayesian networks will be regarded as **hypotheses of causal relationships** generated from data and domain knowledge.

1.1.6 Domain Knowledge

In every domain, we have certain domain knowledge, such as the number of variables and the meanings of these variables. Such domain knowledge could come from scientific laws, expert opinions, accumulated personal experience, as well as other sources [37]. From common sense, we know that domain knowledge is usually correct, since it has been verified by experiments or real applications.

In the applications of Bayesian network structure learning from data, it is not uncommon to observe that some edges in the learned Bayesian network structures are inconsistent with domain knowledge. The potential reason for the inconsistency is that the available data is inadequate or not representative of the probability distribution in the domain. To resolve this inconsistency, one should consider incorporating the available domain knowledge in the knowledge discovery process.

Representation of domain knowledge in Bayesian networks can be quantitative and qualitative. The quantitative domain knowledge is conditional probabilities or constraints on conditional probabilities, and the study on quantitative domain knowledge can be referred to [11,94,95,126]. The qualitative domain knowledge can be represented as topological constraints in Bayesian networks [38,87]. This work will provide a detailed discussion of topological constraints in Chapter 4 for refining the hypotheses generated from observational data.

1.2 The Application Domain

While the issues in knowledge discovery I have addressed are general, the applications I examined were mainly from biomedical domains. The purpose of knowledge discovery in biomedical domains is not merely to predict the values of some variables based on their correlation with other variables from observational data – the purpose is to predict the behaviors of the system after the manipulation of some variables in the system, like the responses after treatments in the medical domain or system properties after gene sequence changes in Biology.

In biomedical domains, there are sufficient observational data, interventional data, domain knowledge and possible ways of manipulation to verify the hypotheses. All these make biomedical domains an ideal area to explore the idea of combining observational and interventional data for causal knowledge discovery.

1.3 Contributions

This thesis focuses on causal knowledge discovery with Bayesian networks. The objective is to identify direct causal influence relationships between variables in a domain. The main challenges are how to effectively exploit the available resources and minimize the number of interventions for causal knowledge discovery. Utilizing the available resources will improve the relevance of the generated hypotheses, and minimizing the number of interventions will reduce the cost and resources required for causal knowledge discovery. From our best knowledge, no work has combined observational data, interventional data, domain knowledge and interventional experiments for causal knowledge discovery.

A three-step framework of knowledge discovery with Bayesian networks is proposed. The steps are:

1) Hypothesis generation from data;

2) Hypothesis refinement with topological domain knowledge; and

3) Hypothesis verification with interventional experiments.

The input-output model of the framework can be illustrated as

Data + domain knowledge + experiment + algorithm ➔ new knowledge

The flowchart of knowledge discovery framework is shown in Figure 1.

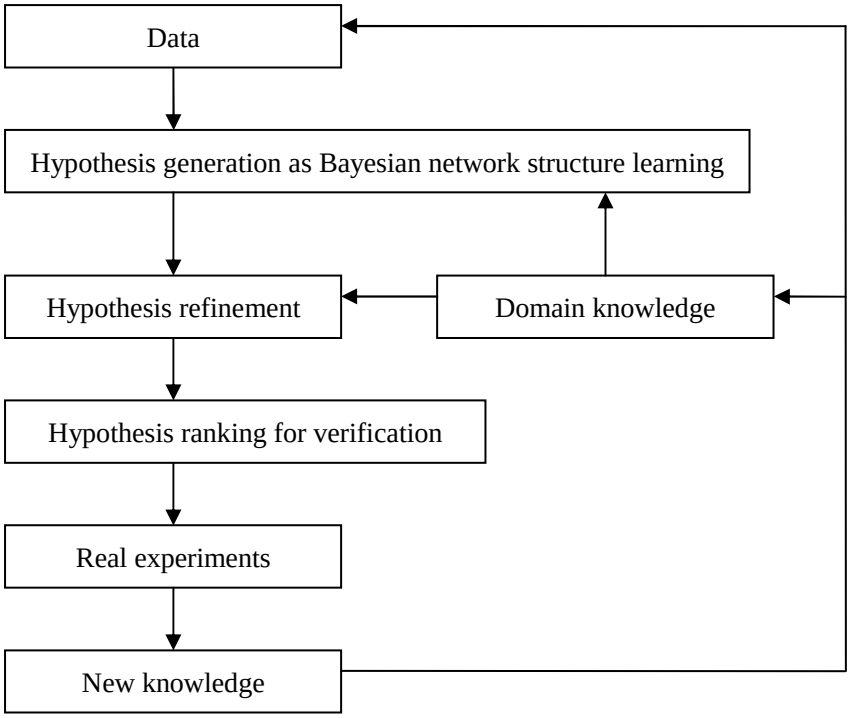


Figure 1 Diagram for the proposed knowledge discovery framework

1) Hypothesis generation from data

The **hypotheses** are the direct influence relationships between variables in a domain as edges in Bayesian networks in this thesis. Hypothesis generation in the proposed framework is equivalent to learning of Bayesian network structure from data. The probabilities of individual edges and complete Bayesian networks can be estimated from data with Bayesian network structure learning as the statistical significance of the hypotheses.

In this step, a new algorithm is proposed to learn Bayesian networks with variable grouping in a domain with similar variables. **Group variables** are introduced to represent groups of variables with similar conditional probabilities and are used to learn Bayesian networks. Variable grouping can reduce the number of variables and Bayesian network search space, which can

lead to speed up the learning process. The experiments with synthetic examples and a real microarray data show that this algorithm is capable of generating reasonable hypotheses in the domain of interest.

2) **Hypothesis refinement with topological domain knowledge**

Topological domain knowledge contains known root nodes, leaf nodes, edges, and so on, and is used in Bayesian network structure learning to resolve the possible inconsistency between the learned structure and domain knowledge.

Two canonical forms, i) the rule format and ii) the matrix format, have been proposed to represent topological domain knowledge. The rule format is general and easy to extract from domain experts, while the matrix format is easy for domain knowledge consistency checking and easy to combine in the Bayesian network learning. From our best knowledge, the matrix format of topological domain knowledge has not been discussed in other work.

Topological domain knowledge has been used in Bayesian network structure learning. However, the effects of different kinds of topological constraints have not been comprehensively studied. Experiments in this thesis show that topological constraints such as roots, leaves and distribution-indistinguishable edges are important in hypothesis refinement with Bayesian network structure learning.

The application of Bayesian network structure learning in a real heart disease domain shows the inconsistency between the learned Bayesian network and domain knowledge, which suggests the requirement of topological domain

knowledge for hypothesis refinement in real applications. With topological domain knowledge, Bayesian network structure learning can generate more justifiable hypotheses from data and the learning process can be sped up.

3) **Hypothesis verification with interventional experiments**

The generated hypotheses are not the final product of causal knowledge discovery. They have to be verified with interventional experiments to ensure their effectiveness for causal diagnosis, prediction and control.

The objective of hypothesis verification is to select the appropriate hypotheses for verification and to minimize the number of interventional experiments required. **Node-based and edge-based interventions** are proposed for hypothesis verification. In node-based interventions, some variables are manipulated to specific values and their effects on other variables are measured to evaluate the influence relationships between variables learned from the previous data. In edge-based interventions, $n-2$ variables in the domain are fixed to specific values by manipulation and one of two remaining variables is manipulated to different values to observe its effect on the last variable. To my knowledge, this thesis is the first to discuss the edge-based intervention for hypothesis verification under the Bayesian network framework.

Hypothesis verification starts with a data set collected in each active learning step. Node entropy and edge entropy from the current available data are used to rank the hypotheses for intervention to reduce the computational complexity.

A new criterion, **non-symmetrical entropy**, is first proposed to select hypotheses for verification, and a new entropy-based criterion is proposed to stop the active learning process. Non-symmetrical entropy considers the probabilities of two states between two variables (say, A and B): an edge from A to B and the state without such an edge. In contrast, symmetrical entropy considers the probabilities of three states between two variables: an edge from A to B, an edge from B to A and the state of no edge between A and B.

Since intervention is non-symmetrical in nature, non-symmetrical entropy is better than other methods to rank hypotheses for verification. Experiments show that, on average, non-symmetrical entropy minimizes the number of interventional experiments required to verify the direct causal influence between variables in interventional experiments.

The proposed framework is interactive and iterative, which involves the repeated application of specific Bayesian network structure learning algorithms and interpretation of hypotheses generated by these algorithms ([54], page 4). The reason for an iterative framework is that knowledge discovery in a domain cannot be completed in one round, and there is no closed-loop framework formalized for knowledge discovery with causal Bayesian networks, although the idea of a closed-loop framework for causal knowledge discovery is implicitly used in practice.

The structure of the framework is stable, and the details of the three components of the framework can be updated or further extended in future. The two main

components to be emphasized in the framework are: i) hypothesis refinement and ii) hypothesis verification. The general knowledge discovery process has been discussed for expert systems [74,133] and data mining [13,23] (more references in the survey [101]). However, hypothesis refinement and hypothesis verification have not been sufficiently taken into account. Little work has been done on hypothesis selection for verification with interventional experiments. The proposed framework can be a step in the right direction for hypothesis verification. More detailed comparisons between our methods and related work can be referred to Section 7.2.

The framework is implemented using MATLAB with Bayes Net Toolbox [122]. Some preliminary results of the work have been published before [107,108][4].

1.4 Structure of the Thesis

This chapter briefly summarizes the research motivations and objectives of this work. The remainder of the thesis is organized as follows:

Chapter 2 summarizes the background and related work of this thesis.

Chapter 3 discusses methods for hypothesis generation in three situations:

[4] Some of the results have appeared in the following papers. Reprinted with permission from IOS Press.
G. Li, T.-Y. Leong, A framework to learn Bayesian Networks from changing, multiple-source biomedical data, Proceedings of the 2005 AAAI Spring Symposium on Challenges to Decision Support in a Changing World Stanford University, CA, USA, 2005, pp. 66-72.
Q. Chen, G. Li, T.-Y. Leong, C.-K. Heng, Predicting Coronary Artery Disease with Medical Profile and Gene Polymorphisms Data, World Congress on Health (Medical) Informatics (MedInfo), IOS Press, Brisbane, Australia, 2007, pp. 1219-1224.
G. Li, T.-Y. Leong, Biomedical Knowledge Discovery with Topological Constraints Modeling in Bayesian Networks: A Preliminary Report, World Congress on Health (Medical) Informatics (MedInfo), IOS Press, Brisbane, Australia, 2007, pp. 560-565.

individual Bayesian networks, individual edges in Bayesian networks and Bayesian networks learned with variable grouping.

Chapter 4 discusses hypothesis refinement. Two canonical formats are proposed to represent domain knowledge as topological constraints in Bayesian networks.

Chapter 5 discusses hypothesis verification with node-based interventions and edge-based interventions. Non-symmetrical entropy criterion is proposed to select hypotheses for verification, and entropy-based criterion is proposed to stop the active learning process.

Chapter 6 demonstrates the complete process of knowledge discovery with Bayesian networks on a protein signal network as a working example.

Chapter 7 summarizes the achievements, the limitations of this study and the potential future work.

1.5 Declaration of Work

During my PhD study, I have worked on different topics, including Bayesian network structure learning, translation initiation site prediction from human cDNA sequences, and ancestral state accuracy analysis in phylogenetics. I have published four papers in the leading international journals and nine papers in the leading international conferences. The details of the selected publications are available in Appendix A.D.

Chapter 2 Background and Related Work

There are two categories of high-level tasks in knowledge discovery ([73], preface, page xi). The first category of the tasks is to predict the values of some variables from the values of other variables based on correlation information from *observational data*, such as classification and regression with *observational data*, or to summarize *observational data*, such as density estimation, clustering and association rule mining. The second category of the tasks in knowledge discovery is to predict the causal change of some variables based on causal relationships between variables from *interventional data* when other variables are manipulated to different values.

In this chapter, I first briefly summarize the methods using *observational data* for correlational knowledge discovery. Next, I discuss randomized experiments to collect *interventional data* for causal knowledge discovery. Lastly, I survey the methods for Bayesian network learning, which are the fundamentals of this thesis and can be applied to both categories of tasks in knowledge discovery.

2.1 Knowledge Discovery with Correlation Information

Knowledge discovery with correlation information is based on *observational data*. The representative tasks in this category include classification, regression, clustering,

and association rule mining with *observational data*[5]. These methods are useful and important in many applications, such as marketing [2], investment [80], fraud detection [149], manufacturing [116], and biomarker prediction [109].

2.1.1 Classification

Classification is a kind of supervised learning [81]. With the available data and the class labels, we need to find a function that maps the features to class labels as accurately as possible. The features, extracted from the data, can be discrete, continuous, or mixed. The mapping function can be expressed explicitly in some models or implicitly in the data. Some representative methods for classification are decision trees [136], Naïve Bayes [83], K nearest neighbors [4], artificial neural networks [9], and support vector machines [17], to name a few.

Decision tree methods [136] use a tree structure to classify the instances[6]. The classification process starts from the root of the tree. In the root of the tree, one feature (or some combinations) of the instance is compared to a specified function to decide which branch to follow. In the next internal node encountered, another feature will be compared to a new specified function. This comparison process will continue until the instance reaches a leaf node, where the associated class label is assigned to the instance.

Naïve Bayes [83] is a probability-based method. It assumes that the features are

[5] Usually, classification and regression can also be applied to interventional data for causal knowledge discovery.

[6] In this thesis, an "instance" is the same as a case, a sample, or an example in a data set. An instance includes the values of all the variables in a specific case.

independent of each other given the class label. The advantage of Naïve Bayes classifier is that it is easy to build and it is robust in prediction. However, the independence assumption between features given the class label is sometimes strong. Some extensions of Naïve Bayes relax the independence assumption, such as Tree-Augmented Naïve Bayes [62] and Aggregating One-Dependence Estimators (AODE) [169], to improve the classification accuracy.

K nearest neighbor [4] is a method based on the intuition that, if the values of the features in different instances are similar (or the same), the instances should be in the same class. The training process is simple: just keep the training data set. The mapping function from the features to the class labels is implicitly expressed with the training instances. However, the prediction with K nearest neighbor method is time-consuming – It searches the similar instances throughout the training data set for each new instance to make a prediction.

Artificial neural network [9,84] is a method inspired by a biological neural system which consists of many neurons. The neurons in artificial neural network are inter-connected and work together to realize a mapping function. The links between neurons can be trained with data to strengthen the particular patterns. The representative training method for artificial neural networks is Back-propagation [84]. A neural network can approximate any functions with any accuracy when the number of neurons, connection functions, and the weights of the connections are properly selected.

Support vector machines (SVMs) [17,164] map data from the original

low-dimension space into a high-dimension space and learn a hyperplane which separates the learning examples into their different classes. The hyperplane in the high-dimension space is selected based on the maximal margin between two classes. With kernel methods, the real mapping from the original dimension to the higher dimension can be achieved implicitly. SVMs are among the best methods for classification. However, they are sensitive to noises, since the noises may change the margin, the position of the hyperplane and then the classification accuracy.

2.1.2 Regression

Regression [141] has been extensively studied in statistics. It examines the relationship between a dependent variable (or response variable) and independent variables (or explanatory variables). The representative methods are linear regression and logistic regression. Different from Bayesian network structure learning (refer to Section 2.3 for details), where there is no specific target variable, a target variable is pre-specified in regression models. The purpose of regression analysis is to learn the relationship between the target variable and all the other variables. In contrast, the purpose of Bayesian network structure learning is to identify all possible direct causal influence relationships between variables in a domain.

2.1.3 Clustering

Clustering is a common unsupervised descriptive task where a finite set of categories or clusters are identified to describe the data [53,55,92,159]. It is a very helpful

method for discovering new and interesting patterns in the underlying data. The patterns in clustering are some kinds of similarities within a subset of the data to distinguish them from the rest. After clustering, the instances in each cluster are similar to each other with respect to some similarity measure, and dissimilar to the instances in other clusters. Two categories of clustering methods are commonly used: partitional clustering and hierarchical clustering [92]. Detailed surveys on clustering methods can be found in [7,75,92,93,96,176].

2.1.4 Association Rule Mining

Association rule mining was originally proposed to identify items frequently co-occurring in commercial transactions. The co-occurrence of the items indicates that consumers tend to buy these items together. Such information is important for marketing and has applications in other domains, such as analysis of dependence between genes in Biology. Representative methods for association rule mining are Apriori [3] and Dynamic Itemset Counting (DIC) [14].

2.1.5 Time-series Analysis

Time-series data can be modeled with a Markov process or its variants [12,137]. In a Markov process, the future state of the system is only dependent on the current state and independent of the past states. The discrete time-series data can be modeled with hidden Markov models (HMM) [137]. The continuous time-series data can be modeled with time-series regression models or state-space models [12].

A special issue in time-series analysis is Granger causality, which is widely used in econometrics. Ordinarily, regressions from observational data reflect "mere" correlations, but Clive Granger [76] argued that an interpretation of a set of tests can reveal something about causality: If a variable X at time t_1 can predict another variable Y at time t_2 (t_1 is before t_2 in time) well by regression, then variable X is a cause of variable Y.

2.1.6 Disadvantages of Correlation-based Knowledge Discovery

Correlation-based knowledge discovery from observational data, including Granger causality, only measure correlational dependencies between variables. Correlation-based knowledge discovery can predict the values of some variables from the observational values of other variables when there is no change in the mechanism of the system. When some variables are manipulated to specific values, however, correlation-based knowledge discovery cannot predict the change of other variables. For example, if two variables X and Y are the effects of a common cause, but with a different lag, one variable may predict another variable well based on correlation and Granger causality may be established between them. However, manipulating either one of X and Y would not change the value of the other. Since the change of some variables with other variables under manipulation is important for control, causal prediction and system re-engineering, causal knowledge discovery is needed with manipulation criterion.

2.2 Causal Knowledge Discovery with Randomized Experiments

Causal knowledge discovery appeared from the very beginning of human history when our ancestors started to explore the nature. In ancient time, human inferred causal knowledge from their experiences and manipulations implicitly. The modern methods for causal inference started with Statistics in scientific research. Randomized experiments [58,125,143] are the established method to collect interventional data for causal knowledge discovery. The objective of a typical **randomized experiment** is to test whether one variable will affect another variable causally. The first variable will be manipulated to different values to examine its effects on the second variable. The values of the first variable are randomly assigned – the manipulation of the first variable does not depend on any other variables in the domain. In this case, the change of the second variable is just due to the manipulation of the first variable, not by other factors. The collected interventional data is analyzed with regression or other methods. There are a number of applications of randomized experiments for causal knowledge discovery [120,144].

Neyman [125] introduced the potential outcome notation for causal knowledge inference in the context of randomized experiments, and proved that the difference of the observed sample mean between different manipulations was the unbiased estimator of the average causal effect over all the tested subjects [143]. Fisher [57] recognized that, without randomization, an experiment has little value irrespective of the subsequent treatment ([139], page 45).

However, randomized experiments just deal with how to efficiently and effectively test the statistical significance of the hypothesis. The randomized experiment methods do not deal with hypothesis generation explicitly with mathematical models. And the hypothesis in the randomized experiments is constrained to one target variable. Alternatively, Bayesian network method can generate new hypotheses and model causal relationships between many variables.

2.3 Bayesian Network Learning

Bayesian network learning can be used in knowledge discovery from both observational and interventional data. This section starts by introducing the basics of Bayesian networks and follows by giving the reasons to learn Bayesian networks from data. The later sub-sections give a survey of parameter learning and structure learning in Bayesian networks, respectively. The last sub-sections cover the related work on causal knowledge discovery with Bayesian networks and active learning.

2.3.1 Basics of Bayesian Networks

Bayesian networks [131] offer a graphical representation of probabilistic relationships between a set of random variables. Given a finite set $\mathbf{X} = \{X_1,...,X_n\}$ of discrete random variables where each variable X_i may take values from a finite set, denoted by $Val(X_i)$. A Bayesian network is an annotated **directed acyclic graph** (DAG) $G = \{V, E\}$ that encodes a joint probability distribution over $\mathbf{X}$. The nodes[7] of G correspond to random variables $X_1,...,X_n$. The edges of G represent direct

[7] We will use "node" and "variable" interchangeably in this thesis if there is no ambiguity.

causal influences between variables. If there is a directed edge from variable X_i to variable X_j, variable X_i will be a parent of variable X_j, and variable X_j will be a child of variable X_i. Each node is associated with a conditional probability distribution (CPD) $p(X_i \mid Pa(X_i))$, where $Pa(X_i)$ denotes the parents of X_i in G. The pair (G, CPD) encodes the joint probability distribution $p(X_1,...,X_n)$ given Bayesian network G. A unique joint probability distribution over X from G is factorized as:

$$p(X_1,...,X_n) = \prod_i p(X_i \mid Pa(X_i))$$

Figure 2 shows an example of a Bayesian network: the **Cancer network** from Cooper and Yoo [39], which is hypothetically about a medical domain with 5 variables[8].

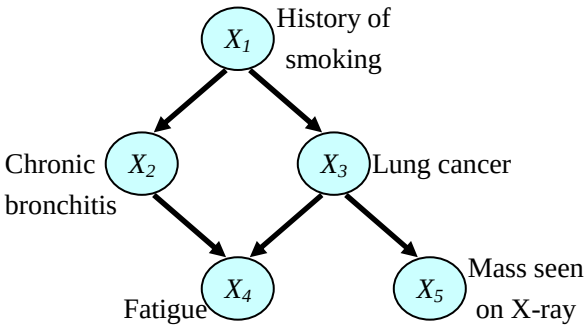


Figure 2 A simple example of a Bayesian network

Causal Bayesian networks

A **causal Bayesian network** [130] of a domain is similar to the general Bayesian network. The difference is in the interpretation of edges in the Bayesian networks. In a general Bayesian network, the edges between variables can be explained as correlations or associations. In a causal Bayesian network, the edges represent causal

[8] Re-printed with permission from Elsevier.

relationships (Refer to Section 1.1.1 for causal knowledge):

When we manipulate the parent variable of an edge by fixing its state to different values, we can observe the change in the probability distribution of the child variable; however, when we manipulate the child variable, the probability of the parent will not change.

This corresponds to the causality with agency: manipulating causes can change effects but not vice versa [135,171].

Moreover, when one variable is manipulated, the causal influence relationships between other variables will not change, i.e., the conditional probability of the child variable given its parents will remain the same if the child variable is not the manipulated variable. This is a modularity property of the causal system: the manipulation on one part of the system will not change the mechanism of other parts of the system.

2.3.1.1 Qualitative Part and Quantitative Part in Bayesian Networks

A Bayesian network has two main components: i) qualitative part and ii) quantitative part. The qualitative part of a Bayesian network encodes the causal influence relationships between the variables and the conditional independence statements in Bayesian network structure. Based on the causal Markov assumption, variable X_i is independent of all its non-descendants given its parents $Pa(X_i)$ in Bayesian network G. For example, in Figure 2, variables X_5 and X_1 are conditionally independent

given variable X_3:

$$p(X_5 \mid X_3, X_1) = p(X_5 \mid X_3).$$

The quantitative part of a Bayesian network represents the strength of direct causal influences between variables. Each variable associates with a set of conditional probability distributions with respect to each configuration of its parents $Pa(X_i)$, regardless of other variables.

2.3.2 Bayesian Network Construction from Domain Knowledge

There are several ways to construct Bayesian networks. One way is to construct Bayesian networks completely from domain knowledge. This is generally achieved in three main steps [46] that:

1) Determine the number of variables and the meaning of these variables in the domain of interest;

2) Determine whether there exist direct causal influence relationships between the variables in the domain; and

3) Determine the conditional probability distributions given the structure of the Bayesian network from the first two steps.

To construct a Bayesian network from domain knowledge, we assume that: 1) all variables are known in advance – the variables in the Bayesian network are determined; 2) domain knowledge can readily assert the causal relationships (typically correspond to the assertions of conditional dependencies [86]) between variables – the

edges in the Bayesian network can be determined by domain knowledge; and 3) the values of conditional probabilities can be estimated from domain knowledge. Quite a few Bayesian networks have been constructed in this way, *e.g.*, QMR-DT [150]. Various methods have been proposed to facilitate the process to construct Bayesian networks with causal domain knowledge [46,89,124].

2.3.3 Reasons to Learn Bayesian Networks from Data

Although there are examples of successful Bayesian networks built from domain knowledge, this approach may be limited by available sources of domain knowledge. The limitations of expert-based knowledge acquisition process are: i) The process is tedious and arduous for an expert; ii) The probabilities are hard to elicit; and iii) When several experts are involved, it is difficult to assure a consistent network structure and probability estimates.

Alternatively, accompanied with the improvement in electronic devices, more data are available in science or application areas. We can utilize the available data for causal knowledge discovery in the domain of interest.

2.3.4 Categories of Bayesian Network Learning Problems

The problem of learning Bayesian networks has been extensively studied in the literature [6,10,15,24,34,38,60,61,65,71,87,103,153]. The Bayesian network learning problems can be divided into different categories according to two criteria: 1) whether the Bayesian network structure is known; and 2) whether the data set is complete.

Table 1 shows four different categories of problems and corresponding methods respectively from this division.

	Complete data	**Incomplete data**
Known structure	Statistical parametric estimation (closed-form equations)	Parametric optimization (EM, gradient descent …)
Unknown structure	Discrete optimization over structure (discrete search)	Combined (Structural EM, mixture models …)

Table 1 Categories of Bayesian network learning problems

If the structure is known beforehand, the problem is usually referred to as the *parameter learning problem*. The objective of the parameter learning problem in Bayesian networks is to optimize the parameters in a given structure with respect to the likelihood of the data. When the data is complete, the parameter learning problem is a statistical parametric estimation problem and closed-form solutions are available. When the data is incomplete, the parameter learning problem does not have a closed-form solution. In this case, the expectation-maximization algorithm (EM) [42] and gradient descent algorithm can be used to estimate the parameters.

When the structure is unknown, the Bayesian network learning problem becomes a *structure learning problem*. The objective of the structure learning problem is to find a structure in the Bayesian network structure space that optimizes some measure of the structure quality. Since the parameters in Bayesian networks are dependent on the structure, the structure learning problem needs to learn the structure and the parameters simultaneously. This problem is more difficult than the parameter learning problem, especially when the data set is incomplete. In this thesis, we focus on the problem of Bayesian network learning with unknown structure and complete data.

2.3.5 Parameter Learning in Bayesian Networks

2.3.5.1 Complete Data

There are assumptions for closed-form solutions in parameter learning with complete data [15,153]. The first assumption is that there are no missing values in the data set D, which can be called a **complete data**. The second assumption is that parameter vectors are mutually independent. Under these two assumptions, the parameters can be updated independently. The third assumption is that the probability distribution of the problem is from the exponential family. With the exponential family assumption, the prior probability and the posterior probability are in the same form. With the three assumptions, the probabilities can be updated with a closed-form.

2.3.5.2 Incomplete Data

Learning parameters of Bayesian networks from incomplete data is typically done under the Missing-At-Random (MAR) assumption [142], which states that the pattern of missingness is not dependent on the missing values and it may only depend on the values of the observed variables.

When the data is incomplete, the parameters are not independent anymore, and no closed-form solution for parameter learning exists. Approximate solutions have been proposed, such as gradient method [8,157], the EM method [103] and Monte Carlo methods such as Gibbs sampling [71]. Gradient method and EM method are more efficient than Monte Carlo methods, but they tend to converge to a local maximum.

Monte Carlo methods can yield accurate results, but they are intractable and converge slowly. For more details on parameter learning in Bayesian networks, please refer to [6,44,126,127,160].

2.3.6 Structure Learning in Bayesian Networks

The objective of Bayesian network structure learning is to find a Bayesian network structure that best describes the observed data. This problem is more difficult than parameter learning, because the number of possible structures (DAGs) to search is super-exponential in the number of variables in the domain. Robinson [140] derived a recursive function to determine the number of possible DAGs with n variables:

$$f(n) = \sum_{i=1}^{n} (-1)^{i+1} C_i^n 2^{i(n-i)} f(n-i)$$

The numbers of possible DAGs with 1 to 10 variables are calculated from the formula and shown in Table 2. We can see that the number of Bayesian network structures increases very fast with the number of variables in the domain.

Number of variables in DAG	Number of possible DAGs
1	1
2	3
3	25
4	543
5	29,281
6	3,781,503
7	1.1×10^9
8	7.8×10^{11}
9	1.2×10^{15}
10	4.2×10^{18}

Table 2 Number of DAGs

Since the number of DAGs is super-exponential in the number of variables, it is impossible to enumerate all possible structures and score them, even with a small

number of variables n in a Bayesian network. The Bayesian network structure learning problem has been proven to be NP-complete [30]. Heuristic-based methods have been proposed to find a local maximum in the structure space.

Two approaches for Bayesian network structure learning exist. The first approach is the **score-and-search-based approach** [32,38,87]. This approach starts from an initial structure (generated randomly or from domain knowledge), and moves to the neighbors of the current structure with the best score in the structure space deterministically or stochastically, until a local maximum of the optimization criterion is reached. The learning process can re-start several times with different initial structures to improve the final result. The representative methods of the score-and-search-based approach are K2 algorithm [38], Greedy search, Markov Chain Monte Carlo (MCMC), and Structural EM [60].

The second approach is the **constraint-based approach** [132,155]. This approach starts to test the statistical significance of the pairs of variables conditioning on other variables to induce the conditional independence between the pairs of variables. The pairs of variables that pass some threshold of the statistical significance are deemed as directly connected in the Bayesian networks. The complete Bayesian network structure is constructed from the induced conditional independence and dependence information of variables. The representative methods of the constraint-based approach are SGS algorithm and PC algorithm [155].

To discuss the Bayesian network structure learning methods further, Markov equivalence and model selection criteria need to be introduced first.

2.3.6.1 Markov Equivalence

If two DAGs encode the same conditional independencies, they are said to be Markov equivalent. Bayesian networks are Markov equivalent if and only if they have the same skeleton and the same v-structures [165], where **v-structure** is a graphical relationship of any three variables such that there are edges from variable X to variable Z and from variable Y to Z but no adjacency between X and Y. All DAGs with the same conditional independencies can form a Markov equivalent class [131]. Such a class can be represented by a complete partially directed acyclic graph (CPDAG) called an essential graph or pattern. The directed edges in this CPDAG mean that these edges must be oriented in a certain direction in all the DAGs of the same equivalence class, and the undirected edges mean that these edges can be in either direction subject to the acyclic constraint in Bayesian network.

In Bayesian network structure learning, it is unlikely to distinguish the structures in a Markov equivalent class with observational data. The model selection criteria will give the same score to the set of equivalent structures. In this case, we cannot hope to recover the "true" generating structure with the observational data only. The best solution to be expected is a structure within the same Markov equivalent class.

To distinguish different Bayesian networks within the same Markov equivalent class, we need domain knowledge to justify the direction of the edges or we need interventional data to learn the direction of the edges (refer to Chapter 5 and also [121,161]).

2.3.6.2 Model Selection Criteria

The score-and-search-based approach to Bayesian network structure learning is based on a scoring function that estimates how well a given Bayesian network G matches the data D. The best Bayesian network is the one that maximizes a scoring function given the data D.

An ad-hoc scoring function is based on the maximum likelihood (ML) principle: selecting the structure which generates the data D with the highest probability. One disadvantage of ML principle is that the models with more parameters[9] can predict the data well, but may lead to overfitting problem. Therefore, a penalty of the model complexity is needed in the scoring function.

Two scoring functions with complexity penalty are: **Bayesian Information Criterion** (BIC) and **Bayesian score**. The Bayesian Information Criterion (BIC) [147] is defined as

$$\log p(D \mid \hat{\theta}_G, G) - \frac{d}{2} \log N$$

where D is the data, G is the Bayesian network to be evaluated, $\hat{\theta}_G$ is the maximum likelihood (ML) estimate of the parameters in Bayesian network G with data D, d is the number of parameters in Bayesian network G, and N is the number of instances in the data. The BIC criterion has several properties. First, it does not depend on the prior, so we do not need to specify the prior to score the structure. Second, it is quite intuitive. Namely, it contains a term $\log p(D \mid \hat{\theta}_G, G)$ measuring how well the parameterized model predicts the data and a term $d/2 * \log N$ that

[9] The number of parameters in a model is used to measure the complexity of the model.

punishes the complexity of the model. Third, it is exactly minus the Minimum Description Length (MDL) criterion [86]. BIC is often used in practice. However, it has a drawback that it tends to choose models that are too simple due to the heavy penalty on the complexity of the model.

The Bayesian score for measuring the quality of Bayesian network G is its posterior probability given the data:

$$p(G \mid D) = p(D,G) / p(D)$$

where the marginal probability $p(D)$ of the data D is a normalization constant which does not depend on Bayesian network G. Since $p(D)$ is a constant relative to G and will not affect the ordering of the different models, the relative posterior probability $p(D,G) = p(G) * p(D \mid G)$ is often used for model selection. This criterion has two components: the prior of the structure and the marginal likelihood of the data given the structure. The prior can be specified by experts or just set uniformly to all possible structures. The marginal likelihood can be calculated by integrating the parameters of the model. The Bayesian score for Bayesian network learning is originally discussed by Cooper and Herskovits [38] as BD metric and further developed by Heckerman *et al.* [87] as BDe metric. Compared to BIC, the Bayesian score is a more accurate criterion, since it considers the prior information. However, it needs more computation. In comparison, BIC can be derived as a large sample approximation to the marginal likelihood. In practice, the sample size does not need to be very large for the approximation to be good.

Besides the criteria mentioned above, some other criteria have also been proposed

for Bayesian network structure selection, such as cross-validation criterion [5] and Minimum Message Length [167]. The details are deferred to the above references.

2.3.6.3 Score-and-search-based Approach

The score-and-search-based approach relies on the model selection criterion and a search method. Any of the model selection criteria mentioned above can be used for the former. In the following sections, the focus will be on the latter. As to the different combinations of search methods and model-selection criteria, Checkering [31] showed that greedy search with random restarts can produce better structures when the computational time is fixed.

Exhaustive Search

The brute-force approach to structure learning is to enumerate all possible DAGs, score each one, and select the one with the best score. Since the number of the possible DAGs is super-exponential in the number of variables, it is infeasible to enumerate all possible DAGs when the number of variables is greater than 5. However, this provides a "gold standard" to gauge other algorithms. And, one can evaluate any reasonably-sized set of hypotheses in this way (*e.g.*, the nearest neighbors of one Bayesian network structure).

K2 Algorithm

If we know a total causal ordering of variables, finding the best structure amounts to picking the best set of parents for each variable independently. K2 algorithm [38] adopts this idea and applies a greedy method to search the parents of variables from

the set of variables before the variable on question in the ordering. The algorithm starts by assigning each variable without parents. It then incrementally adds a parent to the current variable which mostly increases the score of the resulting structure. When any addition of a single parent cannot increase the score, it stops adding parents to the variable. Since an ordering of the variables is known beforehand, the search space under this constraint is much smaller than the entire structure space, and there is no need to check cycles in the learning process.

If the ordering of the variables is unknown, we can search over orderings. The space of orderings is much smaller and more regular than the space of the structures, and has a smoother posterior "landscape". As a result, the search over ordering is more efficient than the search over DAGs [65].

Greedy Search

If we do not know the ordering of the variables, we can treat the structure learning problem as an optimization problem over a discrete space of Bayesian networks. The intuitive way is greedy search. Greedy search starts at an initial structure in the structure space as the current structure, considers all the nearest neighbors of the current structure, and moves to the neighbor that has the highest score; if no neighbors have a higher score than the current structure, the algorithm stops.

When greedy search stops, it always reaches a local maximum. The local maximum reached is essentially dependent on the initial structure. If a good initial structure is chosen, we can reach a good structure in a short time. If a bad initial

structure is chosen, we will reach a reasonably good structure only after a very long time, or cannot reach a reasonably good one at all. Although we know the initial structure is essential, we do not have enough domain knowledge to justify which initial structure is good. Instead of choosing one good initial structure, the alternative way is to restart greedy search with different initial structures and choose the one with the best resultant local maximum.

Markov Chain Monte Carlo

Markov Chain Monte Carlo (MCMC) method is a powerful stochastic simulation method used in many areas. Madigan and York [115] first applied MCMC algorithm Metropolis-Hastings (MH) for Bayesian network structure learning. The motivation behind this approach is to obtain samples from a (posterior) probability distribution of Bayesian network structures given the data D, rather than learning a particular Bayesian network that maximizes a certain criterion.

With an initial structure G_0, MCMC learning paradigm will transfer stochastically to G_1, one of G_0's neighbors, and calculate the posterior given G_1. The standard proposal distribution is to assign equal probabilities to all the nearest neighbors of one structure. Then the approach will transfer from G_1 to G_2, one of G_1's neighbors, and calculate the posterior given G_2. The process will continue until the required number of repetitions is reached. The convergence of the MCMC method to the target probability distribution of $P(G|D)$ is guaranteed under the conditions of irreducibility[10] and infinite samples[11]. MCMC methods can yield accurate results

[10] Irreducibility means that any possible structure can be reached from any initial structure.

[11] Infinite samples mean the process should run a long time to get enough samples.

theoretically, but it converges very slowly, especially when there are extreme conditional probabilities.

2.3.6.4 Constraint-based Approach

The constraint-based approach views the structure learning problem differently from the score-and-search-based approach. Since a Bayesian network structure encodes the dependencies and independencies between variables in a domain, this approach tries to discover the dependencies between variables from the data, and then uses these dependencies (and independencies) to infer the structure.

The dependency relationships are measured using a conditional independence (CI) test. In order to use the CI results for Bayesian network structure reconstruction, several assumptions are needed. The assumptions are: causal sufficiency assumption, causal Markov assumption, and faithfulness assumption [155].

Causal sufficiency assumption: There are no common unobserved (also known as hidden or latent) variables in the domain that are parents of one or more observed variables of the domain.

Causal Markov assumption: Given a Bayesian network G, any variable is independent of all its non-descendants in G given its parents.

Faithfulness assumption: A Bayesian network G and a probability distribution P generated by G are faithful to each other if and only if every conditional independence relationship valid in P is entailed by the causal Markov assumption on G.

With these assumptions, one can ascertain the existence of edges between variables and the directions of the edges in certain cases. The output of constraint-based approach will be a CPDAG to represent the entire Markov equivalent class.

The SGS Algorithm

The SGS algorithm, named after Spirtes, Glymour, and Scheines [154], tests the dependency of any two variables X and Y given every subset of other variables in a Bayesian network. If X and Y are conditional independent given any subset of other variables, there will be no edge between X and Y. Otherwise, there will be an edge between X and Y. After testing all the pairs of variables, an undirected graph will be determined.

With the undirected graph, SGS algorithm determines the directionality of these edges by the v-structure within triples of variables. If i) X is adjacent to Z ($X-Z$), ii) Y is adjacent to Z ($Y-Z$), iii) X and Y are not adjacent to each other, and iv) X and Y are conditional dependent given any subset of variables in a Bayesian network with Z but without X and Y, then the directionalities of the edges $X-Z$ and $Y-Z$ are $X \to Z$ and $Y \to Z$, respectively. After the directions of the edges in the v-structure are determined, the directions will be propagated to other edges while maintaining acyclicity of the Bayesian networks.

Assigning directions to edges depends on the true structure of the underlying Bayesian network. As we mentioned above, the SGS algorithm - and any other constraint-based algorithms - cannot necessarily assign directions to every edge. For

example for a Bayesian network with three variables, $X \rightarrow Y \rightarrow Z$, the direction of either edge cannot be determined by any set of independence statements, because two other networks with the same undirected structure, namely $X \leftarrow Y \leftarrow Z$ and $X \leftarrow Y \rightarrow Z$, belong to the same Markov equivalent class and encode the same conditional independence statements.

The IC Algorithm

Similar to SGS algorithm, IC algorithm (Inductive Causation) was proposed by Pearl and Verma [132]. While SGS algorithm starts from a complete undirected graph and then removes edges between any two variables if they are independent given a subset of the remaining variables, IC algorithm starts from an empty graph, and adds edges between any two variables if they are dependent given all subsets of the remaining variables. After this step, IC algorithm will build an undirected independence graph, and the remaining steps are the same as those in SGS algorithm.

The PC Algorithm

Since SGS algorithm requires to test the dependency of any pair of variables given all possible subsets of remaining variables, the time complexity is exponential in the number of variables. This makes it impractical for domains with many variables.

The PC algorithm [156] makes the learning more efficient by reducing the number of conditional independence tests. Since conditional independence of variables X and Y is implied by the subsets of variables linked to them, conditional independence of variables X and Y can be tested given a subset of the

variables linked to them. If variables X and Y are independent given a subset of the variables linked to X and Y, the edge between variables X and Y can be removed and there is no need to test conditional independence of X and Y conditioning on other subsets of variables.

2.3.7 Causal Knowledge Discovery with Bayesian Networks

Generally, the Bayesian networks learned from observational data are interpreted as dependency models, and the structure represent the probabilistic conditional independence. Many people have tried to interpret Bayesian networks causally. Spirtes *et al.* [155] and Pearl [130] developed theories to represent and discover causal knowledge with Bayesian networks from observational data. Spirtes *et al.* [155] supposed that the learning results from SGS algorithm and PC algorithm can be interpreted causally. V-structure is mainly used to determine the direction of edges in Bayesian networks.

Pearl [129] proposed the following three rules to make it possible to infer the probabilities under manipulation from the observational data with graphical models:

Rule 1 Insertion/deletion of observations

$$p(y \mid \hat{x}, z, w) = p(y \mid \hat{x}, w), if\ (Y \perp Z \mid X, W)_{G_{\overline{X}}}$$

Rule 2 Action/observation exchange

$$p(y \mid \hat{x}, \hat{z}, w) = p(y \mid \hat{x}, z, w), if\ (Y \perp Z \mid X, W)_{G_{\overline{X}\underline{Z}}}$$

Rule 3 Insertion/deletion of actions

$$p(y \mid \hat{x}, \hat{z}, w) = p(y \mid \hat{x}, w), if\ (Y \perp Z \mid X, W)_{G_{\overline{X}\,\overline{Z(W)}}}$$

Where $\hat{x}$ means that variable X is manipulated to a specific value x, $\hat{z}$ means that variable Z is manipulated to a specific value z, $G_{\overline{X}}$ means the original graph G with all edges pointing to X removed, $G_{\overline{X}\underline{Z}}$ means the original graph G with all edges pointing to X removed and all edges out of Z removed, $Z(W)$ is the set of variables in Z that are not ancestors of any variables in W in $G_{\overline{X}}$, and $G_{\overline{X}\overline{Z(W)}}$ means the original graph with all edges pointing to X removed and all edges pointing to $Z(W)$ removed.

The first rule states that, if variables Y and Z are independent given X and W in the mutilated graph $G_{\overline{X}}$, the probability of Y given the observed variables Z, W and the manipulated variable X is the same as the probability of Y given the observed variable W and the manipulated variable X. In this rule, one observed variable can be added or deleted from the probability expression if the condition is satisfied.

The second rule states that, if variables Y and Z are independent given X and W in the mutilated graph $G_{\overline{X}\underline{Z}}$, the probability of Y given the observed variable W and the manipulated variables X and Z is the same as the probability of Y given the observed variables Z, W and the manipulated variable X. In this rule, one observed variable can be changed to a manipulated variable in the probability expression if the condition is satisfied.

The third rule states that, if variables Y and Z are independent given X and W in the mutilated graph $G_{\overline{X}\overline{Z(W)}}$, the probability of Y given the observed variable W and the manipulated variables X and Z is the same as the probability of Y

given the observed variable W and the manipulated variable X. In this rule, one manipulated variable can be added or deleted from the probability expression if the condition is satisfied.

With these three rules, we can estimate the interventional effects from observational data when the Bayesian network structure is known. This is very important for the domains where we cannot conduct interventional experiments.

However, Spirtes *et al.* [155] and Pearl [130] only considered the observational data. Since interventional data can provide concrete causal information, they should be incorporated into the knowledge discovery process.

Cooper and Yoo [39] first examined the assumptions that would allow one to combine observational and interventional data in the knowledge discovery process with Bayesian networks. With their assumptions, the parameters in Bayesian networks can be updated with both observational and interventional data in a closed form.

2.3.8 Active Learning of Bayesian Networks with Interventional Data

Traditionally, the methods for knowledge discovery assume that a data set is available before learning, and the data set will not change in the learning process. Alternatively, **active learning** is a method for knowledge discovery that assumes active collection of new data during the learning process. The collection of new data can be guided with the existing data to reduce the total data collected. This idea has been studied for a long time with a standard framework [35,113,138].

Recently, Tong and Koller [160,161] and Murphy [121] applied active learning framework to learn Bayesian network structure. In their work, they assume a small data set is available first, and the probabilities of edges in the Bayesian network are estimated from this data set. The expected posterior loss of different interventions is estimated and the intervention with the maximal expected posterior loss is selected for the new data collection step. The new data is then combined with the existing data for the next round of active learning. The process is repeated until some stopping criterion is satisfied.

Specifically, Tong and Koller [161] considered three possible conditions between two variables X_i and X_j: 1) there is an edge from X_i to X_j, $X_i \rightarrow X_j$; 2) there is an edge from X_j to X_i, $X_i \leftarrow X_j$; and 3) there is no edge between X_i and X_j, $X_i \perp X_j$. The edge probabilities are $p(X_i \rightarrow X_j \mid D,K)$, $p(X_i \leftarrow X_j \mid D,K)$ and $p(X_i \perp X_j \mid D,K)$, where D is the available data, and K is the background knowledge. In the following discussions, D and K will be omitted for brevity. The uncertainties of the edges are measured with edge entropy

$$\begin{aligned} H(X_i \leftrightarrow X_j) = & -p(X_i \rightarrow X_j)\log p(X_i \rightarrow X_j) \\ & - p(X_i \leftarrow X_j)\log p(X_i \leftarrow X_j) \\ & - p(X_i \perp X_j)\log p(X_i \perp X_j) \end{aligned}$$

To reduce the uncertainties in the Bayesian network, a variable is selected for an interventional experiment based on the expected posterior entropy loss, and one new instance is collected. The new instance is incorporated in the original data set to update the probabilities of edges in the Bayesian network.

However, they only considered the situation when one data instance is collected

at each intervention step. The situation when a data set is collected in each intervention step is not considered, since it is not feasible to calculate the expected posterior loss in reasonable time.

2.3.9 Applications of Causal Knowledge Discovery with Bayesian Networks

Bayesian networks have been used for causal knowledge discovery in many different domains. In Cognitive Science, Bayesian networks were used to model causal learning in human behaviors [29,77,78,146]. The application domains were modeled with Bayesian networks and the causal strengths were estimated with Bayesian network learning. In Biology, Bayesian networks were used to model the interaction relationships between different molecules, for example proteins as described in Sachs *et al.* [145].

Due to the cost of intervention and data collection, causal knowledge discovery done purely from data is currently not applicable to domains with many variables if the relationships between variables are probabilistic. In the efforts mentioned above, most experiments work with 3-7 variables. We test the similar cases in the later chapters. When domain knowledge and some assumptions are applied, causal knowledge discovery can be applied to domains with more variables.

Chapter 3 Hypothesis Generation in Knowledge Discovery with Bayesian Networks

- Learning Bayesian Networks from Observational Data

This chapter will discuss hypothesis generation – the first step of causal knowledge discovery with Bayesian networks. We first introduce two kinds of hypotheses as parts of the Bayesian network structure learning problem: 1) whether an individual Bayesian network structure exists; and 2) whether an individual edge exists in a Bayesian network. The hypothesis space of these kinds of hypotheses exists when the variables in a domain are given. The statistical significance of these hypotheses will be evaluated with probabilities using Bayesian network learning from observational data. Selecting the significant hypotheses from the corresponding hypothesis space is our **hypothesis generation** step. We propose a new method to extend the hypothesis space for Bayesian network structure learning that is based on the idea of variable grouping. **Variable grouping** partitions the variables with similar conditional probability distributions into one group. A Bayesian network is learned with the group variables alone. Variable grouping can narrow the search space and may help to speed up the learning process.

3.1 Hypothesis Generation with Bayesian Network Structure Learning

The first kind of hypotheses, whether an individual Bayesian network structure exists, is important because it considers all direct causal influence relationships between variables in a domain. The second kind of hypotheses, whether an individual edge exists in a Bayesian network, is important, since edges reflect direct influence relationships between variables and they can be verified with manipulation experiments.

3.1.1 Probabilities of Individual Bayesian Network Structures

From the Bayesian perspective, the probabilities of individual Bayesian network structures can be estimated with the following formula with the given data D and background knowledge K:

$$p(G \mid D,K) = \frac{p(D \mid G) * p(G \mid K)}{\sum_{G} p(D \mid G) * p(G \mid K)}$$

Where G is the structure of a possible Bayesian network, $p(G \mid K)$ is the prior probability of G given the background knowledge, and $p(D \mid G)$ is the likelihood of the data D given G. In the formula, we need to calculate the probability of the data given the Bayesian network structure and normalize it by the sum of the probabilities of the data given all individual Bayesian networks. Since the number of Bayesian networks is exponential in the number of variables, it is time-consuming to

calculate this probability when the number of variables is greater than 5. Approximate methods are used as an alternative. For example, some Bayesian networks with high scores can be selected as the representatives of the entire structure space, or Markov Chain Monte Carlo (MCMC) method can be used to estimate the probability [43].

3.1.2 Probabilities of Individual Edges in Bayesian Networks

In practice, we are not only interested in complete Bayesian network structures, but also interested in individual edges and their probabilities: Do the edges in the learned Bayesian networks appear by chance or with some statistical significance? To examine the confidence of the edges in the learned structure, we can estimate the Bayesian probabilities of individual edges by the formula suggested by Buntine [16].

$$
\begin{aligned}
& p(A \to B \mid D, K) \\
&= \sum_{G} p(A \to B \mid G, D, K) p(G \mid D, K) \\
&= \sum_{G} p(A \to B \mid G) p(G \mid D, K) \\
&= \sum_{G: A \to B \in E(G)} p(G \mid D, K)
\end{aligned}
$$

Where $E(G)$ is the set of edges in Bayesian network G, and $A \to B$ means that there is an edge from A to B in Bayesian network G.

The first equation above is from the law of total probability. The second one is from the fact that the existence of an edge is independent of the data and domain knowledge given the Bayesian network structure. The third one is from the fact that the probability $p(A \to B \mid G)$ is 1 when Bayesian network G contains the edge $A \to B$; and 0, otherwise. In general, the edge $A \to B$ in the formula can be replaced with any other topological features in Bayesian networks to estimate the

probabilities of those features, *e.g.*, $A \leftarrow B$, or the partial ordering where A is before B.

In the formula, we need to sum up the probabilities of all Bayesian networks with the edge of interest. As mentioned before, the number of Bayesian networks is exponential in the number of variables and the edge probability estimation is time consuming. We have to resort to approximate methods to estimate the probabilities of individual edges. We adopt the bootstrap approach for this purpose.

A **bootstrap approach** [50] is a statistical method to measure the accuracy of statistical estimates and perform statistical inference by re-using the original instances. In a bootstrap approach, the original data set will be re-sampled with replacement to form a new data set with the same number of instances. A new model will be built from the new data set with the same method as that to analyze the original data. The re-sampling experiments are repeated many times, and the results from the repeated experiments show the confidence of the conclusions from the original instances. The process of the bootstrap approach is:

1) *Re-sample N instances from the original data set D with replacement, where N is the number of instances in the original data set. Denote the re-sampled data set as* D_{new}
2) *Apply the Bayesian network learning algorithm on* D_{new} *to learn a Bayesian network*
3) *Repeat Steps 1) and 2) many times*
4) *Count the number of edges appearing in all the learned Bayesian networks*

In our work, the probabilities of edges are the percentages of their occurrences in the learned Bayesian networks from the repeated experiments. If an edge appears in

the bootstrap experiments with a high percentage, it means that the strength of the direct dependency relationship between two variables is not spurious or accidental and indicates strong correlations. Friedman *et al*. [64] first applied the bootstrap approach to estimating the probabilities of edges in Bayesian networks.

Recently, Koivisto [97] proposed an exact method for estimating edge probabilities in Bayesian networks. Koivisto utilized the intuition that the order of the parents of a variable is irrelevant to the variable's probability estimation, and applied forward and backward dynamic programming and fast truncated Mobius transform to estimate all edge probabilities in $O(n2^n)$ time, where n is the number of variables in the domain. This method can be applied to domains with a moderate number of variables (around 25).

3.1.3 An Application of Hypothesis Generation to a Heart Disease Problem

To illustrate the hypothesis generation process, we analyzed a data set for coronary artery disease (CAD) study [26,27] collected at one of the local hospitals in Singapore. The data set consists of data on 2,949 human subjects: 1,462 of the subjects were diagnosed to have coronary artery disease at the time of data collection; the rest were healthy subjects at the time of recruitment. The assessment of CAD in this work was based on the presence of at least 50% narrowing in at least one of the major coronary arteries as detected by angiography. In addition to CAD, ten other patient variables were selected for our experiments. Out of these ten variables, eight are discrete

variables and two other variables, "*AGE*" and "*CBMI*", are continuous. These two continuous variables were discretized separately. The attributes are summarized in Table 3.

Variables	No. of states	Remarks
CAD	2	Healthy or diseased
AGE	continuous	
SEX	2	Male or female
RACE	3	Chinese, Indian and Malay
CBMI	continuous	Body-mass index
Smoking	3	Smoker, non-smoker and ex-smoker
Diabetic	2	healthy or diseased
Hypertension	2	healthy or diseased
FCAD	2	Family history of CAD: yes or no
FDM	2	Family history of diabetes: yes or no
FHY	2	Family history of hypertension: yes or no

Table 3 Attributes of the heart disease dataset

Several methods have been applied to this data set to evaluate the statistical significance of causal or association relationships between variables, including the learned Bayesian network, the probabilities of individual edges from the bootstrap approach, chi-square test, and mutual information[12]. The Bayesian network was learned with the greedy search method and Bayesian Information Criterion (BIC) score [147] (Refer to Section 2.3.6.2 for the definition and explanations of BIC). The best learned Bayesian network in our experiment is shown in Figure 3.

We applied the bootstrap approach to estimating the probabilities of edges in the learned Bayesian network. The learning program ran 500 times and the edges with top occurrences in the learned Bayesian networks from bootstrap approach are listed in Table 4. For example, the first row of Table 4 means that the edge from CAD to

[12] Refer to Appendix A for a brief introduction of chi-square, and mutual information.

Diabetic appeared 456 times (91.2%) in 500 bootstrap repeated experiments. From Table 4, we know that most edges in the learned Bayesian network appear with high probabilities from the bootstrap experiments.

The top chi-square values and mutual information values are shown in Table 5 and Table 6, which show that "*CAD*" is highly correlated with "*AGE*" and "*Hypertension*" – this is consistent with our common sense.

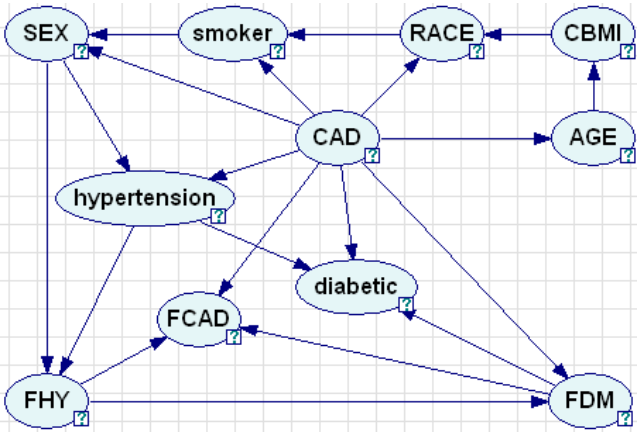


Figure 3 Bayesian network learned from the heart disease data

Order	Variable 1	Variable 2	Occurances
1	CAD	Diabetic	456 (91.2%)
2	CAD	Hypertension	438 (87.6%)
3	CAD	FDM	434 (86.8%)
4	CAD	RACE	397 (79.4%)
5	CAD	Smoker	392 (78.4%)
6	AGE	CBMI	382 (76.4%)
7	FHY	FCAD	381 (76.2%)
8	FDM	FCAD	314 (62.8%)
9	CAD	SEX	311 (62.2%)
10	SEX	Smoker	304 (60.8%)

Table 4 Top edges estimated with bootstrap approach for the learned Bayesian network

With the learned Bayesian network and the edge probabilities, we notice that some edges in the learned Bayesian network have high probabilities, which means that these edges are statistically significant. However, some edges in the best learned

structure are inconsistent with domain knowledge. For example, variable "*AGE*" is dependent on variable "*CAD*" in Figure 3. Other examples include CAD affecting Sex, Smoker affecting Sex, and Sex affecting family history, etc. Such kind of relationships contradicts our common sense: Variable "*AGE*" should not be affected by any other variables in the domain.

Order	Variable 1	Variable 2	Chi-square value
1	AGE	CAD	1331.58
2	Hypertension	CAD	1173.95
3	Hypertension	AGE	771.04
4	Diabetic	CAD	668.56
5	Hypertension	Diabetic	500.51
6	Diabetic	AGE	475.85
7	Smoker	SEX	475.43
8	Smoker	CAD	348.65
9	AGE	CBMI	333.55
10	Smoker	AGE	210.24

Table 5 Top chi-square values from the heart disease data

Order	Variable 1	Variable 2	Mutual information
1	AGE	CAD	0.267676
2	Hypertension	CAD	0.224037
3	Hypertension	AGE	0.154849
4	Diabetic	CAD	0.129434
5	Diabetic	AGE	0.098644
6	Smoker	SEX	0.096076
7	Hypertension	Diabetic	0.082606
8	Smoker	CAD	0.064145
9	AGE	CBMI	0.045032
10	Smoker	AGE	0.040836

Table 6 Top mutual information values from the heart disease data

In this situation, if we are only interested in the density estimation or correlation model, the learned structure is good enough. However, if we want to understand the causal relationships between the variables, the learned structure should be updated, and the inconsistency between the learned structure and domain knowledge should be

corrected. This leads to the second topic in this thesis – hypothesis refinement (refer to Chapter 4). The structure learned from observational data gives us a good initial approximation of the relationships between variables in the domain and a *starting point* to improve.

3.2 Hypothesis Generation with Variable Grouping

In Section 3.1, we discussed how the hypothesis generation process calculates probabilities of complete Bayesian networks and individual edges in Bayesian networks using Bayesian network structure learning methods. Currently, the existing algorithms only take tens of variables into consideration, which are inadequate for domains with hundreds or more variables. To solve this problem, we introduce hidden variables to represent a group of original variables and propose to learn a Bayesian network with group variables only. We conducted experiments on synthetic examples and real microarray data to analyze the approach. The results from synthetic examples show that the algorithm can work well with small data (11 instances in our small examples) and identify the expected group Bayesian network from different data sets. The expected group Bayesian network has the highest BIC score. The experiments on the real microarray data show some domain-meaningful results. We expect the algorithm to generalize well to other domains with similar assumptions.

3.2.1 Observations from Microarray Data

Microarray is a technology used in biological experiments to simultaneously measure

the expression levels of thousands of genes in the cell under different conditions. The measured gene expression levels are microarray data, in which each gene is treated as a variable and the gene expression levels from each experiment form a data instance. The data set usually has hundreds or thousands of genes, but only hundreds (or even just tens) of experiments (as instances).

One of the problems in microarray data analysis is to infer the potential regulatory relationships between genes and gene groups. Many methods have been proposed for this purpose, such as clustering methods [51,66,152] and classification methods [1,68]. Among the proposed methods, Bayesian network learning is a promising one, since the influence relationships between genes are stochastic in nature, which can be easily modeled with Bayesian networks.

However, when we want to apply the existing Bayesian network learning methods to microarray data, there are three main challenges: 1) there are many variables in the data set, 2) the sample size is small and 3) microarray data are changing from experiment to experiment. These challenges are not uncommon in the Bayesian network domain, but the third challenge has special significance. Different biological research groups perform microarray experiments for different purposes and new microarray data are emerging quickly. Since the conditions in these experiments are quite different, it is meaningless to simply combine these data sets into one large data set. Moreover, Bayesian networks learned from different data sets are not directly comparable. To maximize the interpretability of microarray data and the capability for knowledge discovery, we need to extend the Bayesian network formalism for

microarray data analysis in particular, and for the situations with the similar assumptions (see below) in general, *e.g.*, stock market with different industrial sections.

There are some assumptions in the proposed algorithm. One assumption is that some variables in the domain follow similar conditional probability distributions. The second assumption is that the variables following similar conditional probability distributions can be partitioned into one group and can be represented with group variables reliably in different conditions. The third assumption is that the influences between the variables in a group are dense and the influences between groups are sparse.

These assumptions can be interpreted from biological perspective. First, genes from the same gene complex have similar functions based on biological knowledge. These overlapping functions of genes guarantee that the defect of some genes cannot degrade the functions of an entire cell dramatically. Second, some genes act together to perform a biological function. This means that genes can be partitioned into groups according to their functions. Third, the expression levels of these genes are similar or related under different experiment conditions. Moreover, the genes in a group interact with genes in other groups, and the entire interactions between the groups are more important than the interactions between individual genes.

The assumptions can be satisfied in other domains as well. Take the data from stock market as an example. In the stock market, there are different industrial sections and stocks from the same industrial section can be categorized into one group to

represent the activity of the corresponding industry. The different industrial sections will affect each other in the industry level, not merely at the company level. For example, the construction industry grows and needs more steel, which leads to the boom of steel industry, while the boom of the steel industry will lead to the need of more electricity and coal. The growth and boom of the industry will be reflected in the stock prices of the different companies in the corresponding industries. Our algorithm could be applied to stock market data to determine the influence relationships between different industrial sections.

We are using microarray data as an example, and introduce the notion of group variables to represent the groups of the original variables and propose an algorithm to learn a Bayesian network to represent the relationships between groups. In microarray data, the values of a group variable are the expression level of the corresponding biological function performed by this group of genes, which will be learned from the expression levels of individual genes. A Bayesian network will be learned with the group variables only. We call the learned Bayesian network a **group Bayesian network**.

3.2.2 Related Work

The proposed algorithm is related to three areas of machine learning methods for microarray data analysis. The first area is clustering [51]. Clustering is one of the most commonly used methods for microarray data analysis. Clustering methods can identify genes with similar expression levels. However, clustering methods cannot

identify the dependency and possible causal relationships between genes (or groups of genes). In this sense, clustering is not sufficient for biological knowledge discovery. In our proposed algorithm, we can identify groups of genes and the dependency between groups simultaneously. The dependency between groups is a better way of hypothesis generation for gene regulatory relationship discovery.

The second related area is hidden variable discovery in Bayesian networks. The general method for hidden variable discovery uses maximal cliques [117] or semi-maximal cliques in Bayesian networks [52]. The disadvantage of the general method is that it is difficult to identify the meaning of hidden variables. In our proposed algorithm, the group variables (as hidden variables) are assumed to represent the summarized activity level of variables in the group, such as the summarized expression level of genes in individual groups of the microarray data.

The third related area is the module networks by Segal *et al.* [148]. In their work, Segal *et al*. considered groups of genes as modules whose expression levels are similar. The authors assumed that variables in the same modules have the same parents and the same conditional probability distributions. This assumption is one type of parameter tying in Bayesian networks and can narrow the structure space and the number of parameters in Bayesian network learning process. However, the authors did not introduce hidden variables in the learning process, and Bayesian networks were learned only with the original variables. As a result, the space of Bayesian network structures we need to search is still very large.

3.2.3 Learning Algorithm with Variable Grouping

Based on the above observations, we propose an algorithm to learn Bayesian network with variable grouping. The pseudo code of the algorithm is shown in Table 7. The algorithm consists of four main steps:

1) Partition the original variables into different groups;
2) Determine the values of the group variable for each group based on the individual original variables in the group;
3) Learn a Bayesian network with group variables only; and
4) Recover a potential structure of all variables from the learned group Bayesian network structure.

*1) Generate an initial partition of the original **n** variables into **m** groups as the current partition P_c*
2) Determine the values of the group variable for each group in partition P_c
3) Learn a Bayesian network with group variables from P_c, and set the BIC score from the learned Bayesian network as the current score S_c
4) For each neighbor P_i of the current partition P_c
a. Determine the values of the group variable for each group in partition P_i
b. Learn a Bayesian network with group variables from P_i, and set the BIC score from the learned Bayesian network as score S_i
5) Find the maximum of all S_i as S_{max}
6) If S_{max} is greater than S_c, set S_{max} as the current score S_c and the corresponding P_{max} as the current partition P_c, go back to Step 4)
7) If not, recover the structure with all variables from the learned group Bayesian network

Table 7 Algorithm for Bayesian network learning with variable grouping

The details of these steps will be discussed in the following sub-sections and will be illustrated on a small example in Figure 4. While the algorithm is applicable to

different variable types, our example works with Gaussian variables. Variable 1 follows a normal distribution with 0 mean and unit standard deviation – $\text{var}1 \sim N(0,1)$. Variables 2 and 3 follow the same conditional normal distribution. Their means are conditioning on the sampled value of Variable 1 – $\text{var}2 \sim N(\text{var}2; \text{var}1, 1)$[13] and $\text{var}3 \sim N(\text{var}3; \text{var}1, 1)$, and their variances are assumed to be a unit. Variable 4 follows a conditional normal distribution, and its mean is dependent on the sum of the sampled values of Variable 2 and Variable 3 – $\text{var}4 \sim N(\text{var}4; \text{var}2 + \text{var}3, 1)$.

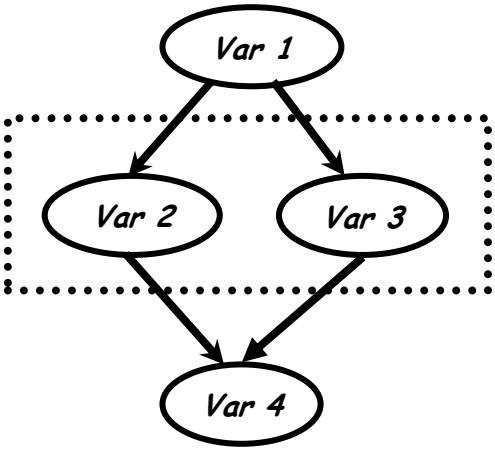


Figure 4 A simple synthetic Bayesian network for variable grouping

In this example, Variables 2 and 3 follow the same conditional probability distribution and are similar to each other. From our assumption – the variables with the similar conditional probability distribution should be grouped together, Variables 2 and 3 should be grouped together in the group Bayesian network. Since Variables 2 and 3 are dependent on Variable 1 in the original structure, the group with Variables 2 and 3 should be dependent on the group with Variable 1. Similarly, the group with Variable 4 should be dependent on the group with Variables 2 and 3.

[13] $\text{var}2 \sim N(\text{var}2; \text{var}1, 1)$ means that the values of Variable 2 are conditional on the values of Variable 1. At each sampling process, the value of Variable 1 is sampled first. Then the value of Variable 1 will be used as the mean in the distribution of Variable 2. It is similar for Variables 3 and 4.

Note that, in general, the algorithm will decide which variables to group and when to group. The grouping can be verified with the original relationships between the variables if they are available.

3.2.3.1 Partitioning of Original Variables into Different Groups

The first main step of the algorithm is to partition n original variables into m groups ($m < n$). The number of original variables n and groups m are determined by domain knowledge. This step is similar to variable clustering (not instance clustering). The aim of variable clustering is to detect the redundant variables in the data, or highly-correlated variables. The difference between variable grouping in this algorithm and the ordinary variable clustering is in the grouping criterion. The criterion of the ordinary variable clustering is just based on the similarity of the variables. The criterion in this algorithm is based on the BIC score of the learned Bayesian network with group variables, because our objective is to learn both the similarity between original variables and the dependency relationships between group variables.

The example in Figure 4 is to partition 4 variables into 3 groups. There are 6 possible ways. For example, Variables 1 and 2 can be assigned to a group $\{1,2\}$, Variable 3 can be in another group $\{3\}$, and Variable 4 can be in a third group $\{4\}$. This grouping will be expressed as $\{\{1,2\},\{3\},\{4\}\}$.

Exhaustive Search for Variable Grouping

The intuitive way to group variables is to enumerate all possible partitionings of n

variables into m groups. The number of possible configurations is the *Stirling number of the second kind* ([25], page 47), which is exponential in the number of variables n and the number of groups m. In this case, it is not feasible to do the exhaustive search for moderately large n and m, if m is not equal to 1, or m is not equal to n. This method is only implemented as a gold standard to test small cases.

Greedy Search for Grouping – Greedy Grouping

Since the grouping space is exponential in the number of variables and the number of groups, we need heuristics to speed up the grouping. Greedy search for grouping – **greedy grouping** – is adopted in this work. First, greedy grouping starts from an initial assignment of the variables to different groups. The initial assignment may be generated randomly, or from domain knowledge. For example, the initial grouping may be randomly assigned to be $\{\{1,2\},\{3\},\{4\}\}$ in our example.

Second, the algorithm tests all the nearest neighbors of the current grouping. The neighbors mean the possible partitions in which only one original variable is changed from one group to another group of the current grouping. For example, one neighbor of the initial grouping $\{\{1,2\},\{3\},\{4\}\}$ is $\{\{1\},\{2,4\},\{3\}\}$ and it is obtained by assigning Variable 2 to another group. For each neighbor of the current grouping, a Bayesian network will be learned with the group variables defined by the partitioning, and the BIC score of the learned Bayesian network is used to measure the goodness of that partition.

Third, the algorithm chooses the neighbor with the highest BIC score as the new

current grouping, if the highest BIC score is greater than the BIC score of the current grouping. Suppose that the current grouping is {{1,2},{3},{4}} in the example. If its neighbor {{1},{2,3},{4}} has the highest BIC score among all the neighbors and this score is better than that from the current grouping, {{1},{2,3},{4}} will be assigned as the new current grouping.

Lastly, the grouping process stops when no neighbors have a higher BIC score than the current grouping. In the example, the current grouping is {{1},{2,3},{4}} and no neighbors have a better score, and the greedy grouping will stop.

Greedy grouping does the optimization locally and always reaches a local maximum. To escape from the local maximum, we can restart the greedy search several times with different initial groupings and select the best result we can obtain. In the second step of greedy grouping, only one variable's assignment is changed from one group to another group and two groups are involved. The results for other groups can be cached to speed up the process.

3.2.3.2 Determine the Values for Each Group Variable

In this algorithm, group variables are introduced as hidden variables to represent each group in this step. Determining the values of group variables is essential, since Bayesian network structure learning is based on group variables. In our work, we have tried different ways to determine the values of the group variables, such as the average of the variables in the group and the values from the first principal component of the variables in a group for continuous variables, and Autoclass package [24] for

discrete variables. For example, if the average of the variables in a group is used as the value of the group variable, Variables 2 and 3 are in a group, and the value of Variable 2 is 0.1 and the value of Variable 3 is 0.2 in one instance, then the value of the group variable is 0.15 (=(0.1+0.2)/2) in this instance.

In the previous paragraph, we showed an example to determine the values of group variables when the variables are Gaussian variables. The algorithm can be applied to the cases when the variables are non-Gaussian or discrete. In those cases, the values of the groups can be determined with domain knowledge or other learning methods, such as the unsupervised Bayesian classification method Autoclass [24].

3.2.3.3 Learn a Bayesian Network Based on the Group Variables

The third main step in the algorithm is to learn a Bayesian network with the group variables only. In this step, we adopt the greedy search with BIC score for Bayesian network structure learning. The important issue in this step is that Bayesian network structure learning is based on group variables only, and no original variables are used in this step. We name the learned Bayesian network the **group Bayesian network**.

Suppose the grouping is {{1},{2,3},{4}}, and we name {1} as group 1, {2,3} as group 2, and {4} as group 3. The learned group Bayesian network is shown in Figure 5:

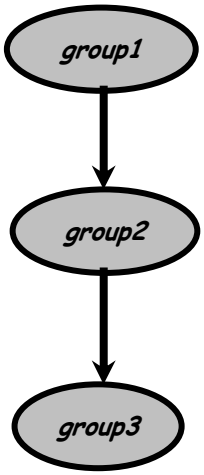


Figure 5 The learned group Bayesian network

3.2.3.4 Recover the Structure of the Entire Variables from the Group Bayesian Network

The fourth main step is to recover the structure for all variables. We adopt a strategy to keep the group variables as the skeleton in the recovered structure. In the process to determine the values of the group variables, a local structure is defined as the structure between the group variable and the original variables in each group, which can be used for potential structure recovery purpose. When the values of the group are from the average or the first principal component of the original variables in the group, the local structure is a tree structure – the original variables are independent of each other given the group variables. For example, group 2 is a root of the local structure with Variables 2 and 3 as in Figure 6.

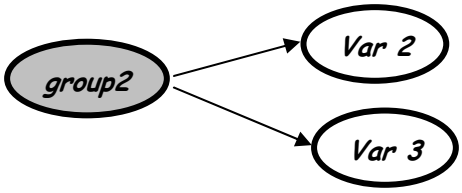


Figure 6 An example of the local structure

In the recovery strategy, the local structures are concatenated to the group

Bayesian network to form an entire Bayesian network. The structure between the group variables is the main frame of the recovered Bayesian network. For example, the recovered structure in the example is shown in Figure 7:

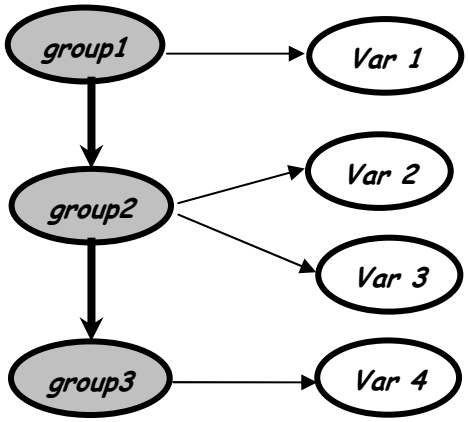


Figure 7 The recovered structure of the group Bayesian network

If another strategy is adopted to remove the group variables from the final structure, the Bayesian network structure with the original variables should be learned with the constraints from the group Bayesian network. The group Bayesian network structure will be used as the skeleton of Bayesian network of the original variables. The original variables in a group will only choose their parents from the original variables which are in the parent groups of this group. For example, group 1 is the parent of group 2 in the learned group Bayesian network in Figure 5, and then the original variables in group 2 will only choose the original variables in group 1 as their parent variables. In this case, the recovered Bayesian network in our example is the same as the original one.

3.2.4 Important Issues in the Proposed Algorithm

There are two important issues to be emphasized. First, there are two search spaces in

our algorithm – one for variable grouping and the other for Bayesian network structure search with group variables. Although both spaces are exponential (one in the number of original variables and the number of groups, and the other in the number of group variables), the combination space is much smaller than the space of possible Bayesian network structures with the entire set of original variables. For example, there are 4.2×10^{18} possible Bayesian network structures with ten original variables in a domain. If these ten variables will be partitioned into five groups, there are 42525 possible partitions[14], and the number of possible Bayesian network structures with five variables is 29,281. The combined search space is 1.24×10^{9} (=29,281*42525), which is much smaller than the original search space of Bayesian networks with ten variables. Therefore, variable grouping can narrow the search space and speed up the learning process.

Second, several heuristics are used in the learning process. One heuristic is the greedy search for variable grouping and the cache of the unchanged groups in greedy grouping. Another heuristic is the greedy search in Bayesian network structure learning. In these heuristics, we always choose a grouping and group Bayesian network with a higher BIC score as the next group assignment and Bayesian network structure. The BIC score never decreases in the search process. When the algorithm stops, it guarantees to reach a local maximum. These heuristics make the process reach a local maximum faster.

---

[14] $42525=\{C(5,5)\times5^{10}-C(5,4)\times4^{10}+C(5,3)\times3^{10}-C(5,2)\times2^{10}+C(5,1)\times1^{10}\}/P(5,5)$. Here $C(n,m)$ means the possible choices to choose m items from n items, and $P(n,m)$ means the possible permutation of m items from n items.

3.2.5 Experiments with Variable Grouping

The proposed algorithm has been tested in experiments with synthetic examples and a real microarray data. In the experiments with the synthetic examples, we build a synthetic Bayesian network first and sample data from the Bayesian network. With the sampled data, we apply the proposed algorithm to learn a group Bayesian network. The learned group Bayesian network will be compared with the expected group Bayesian network to evaluate the proposed algorithm. In the experiments with the real microarray data, we chose some genes in the domain of interest and compared the learned group Bayesian network with biological domain knowledge.

The first synthetic example First we tested the proposed algorithm with the example in Figure 4. In Figure 4, Variables 2 and 3 follow the same conditional probability distribution and are similar to each other, and should be grouped together in the group Bayesian network based on our assumption. The group with Variables 2 and 3 should be dependent on the group with Variable 1, and affect the group with Variable 4.

We drew different number of samples from this synthetic Bayesian network to learn a group Bayesian network. In our experiment, we tested with one thousand samples first and the expected group Bayesian network can be learned reliably. To determine the minimal number of samples required to estimate the expected group Bayesian network reliably, we reduced the number of samples gradually. In the end, we found that eleven was the smallest number of samples to make the group Bayesian network reliably learned in the experiments.

In the experiment, exhaustive search over all possible groupings was tested first for the comparison sake. The grouping problem here is to partition 4 variables into 3 groups. There are 6 different cases in total. Figure 7 shows the structure of the group Bayesian network with the highest BIC score. Group 1 contains Variable 1, Group 2 contains Variables 2 and 3, and Group 3 contains Variable 4. This grouping result is the same as what we expect.

For greedy search over grouping, we ran the program for 12 hours and finished 221 repeated experiments. In each experiment, we drew eleven samples from the synthetic model and learned a group Bayesian network from the instances with greedy grouping. In 82.8% of the repeated experiments, the learned grouping and the structure of the group Bayesian network are the same as expected result in Figure 7.

Another synthetic example

Figure 8 shows another synthetic Bayesian network, which has two copies of the first example with extra edges. In the example, Variables 3 and 4 follow the same conditional probability distribution and should be grouped together; Variables 5 and 6 follow the same conditional probability distribution and should be grouped together. The group with Variables 3 and 4 should be dependent on the group with Variable 1, and should affect the group with Variable 7 and the group with Variable 8. The group with Variables 5 and 6 should be dependent on the group with Variable 2, and should affect the group with Variable 8.

The eight original variables are partitioned into six groups and there are 266

different cases of grouping[15]. From the assumption, the expected result is to partition Variables 3 and 4 into one group, partition Variables 5 and 6 into another group, and partition other variables into individual groups. Figure 9 shows the result with the highest BIC score, which is the same as what we expected.

The synthetic Bayesian networks above show the combination of the diverging and converging patterns in the Bayesian networks, which is the difficult part to learn. If there is no combination of diverging patterns and converging patterns in a Bayesian network, the Bayesian network structure will be a chain or a tree-like structure, which is easier to learn. In the above two synthetic examples, the proposed algorithm can partition the variables which follow the same conditional probability distributions into one group, and the learned group Bayesian networks summarize the relationships between the original variables in a high-level abstraction.

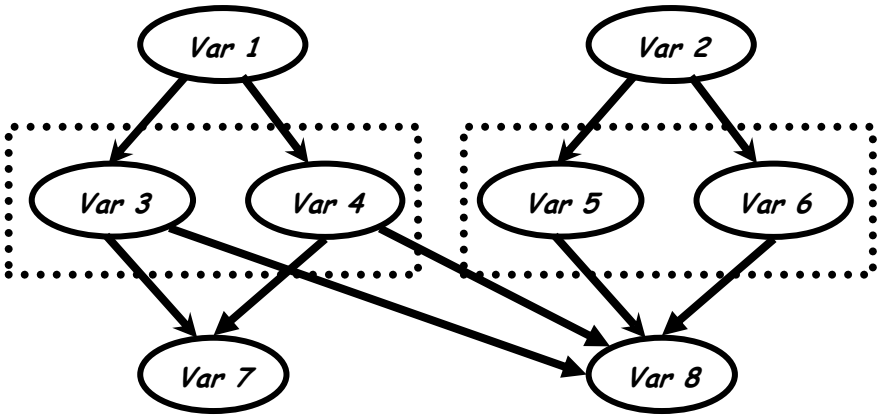


Figure 8 Another synthetic example with eight Gaussian variables

[15] $266 = C(8,2) \times C(6,2) / P(2,2) + C(8,3)$. Here $C(n,m)$ and $P(n,m)$ have explained in Footnote 14.

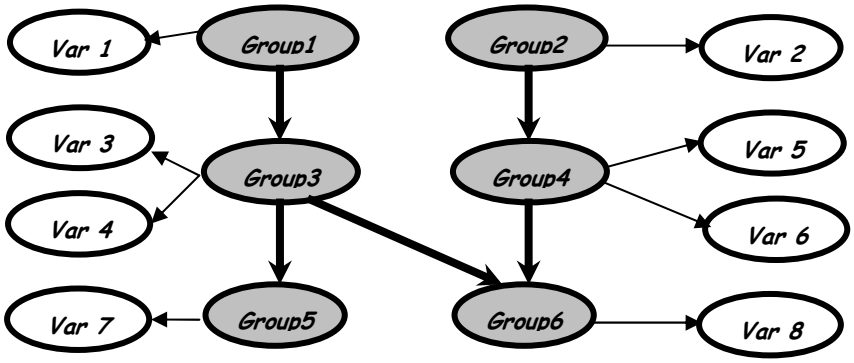


Figure 9 The expected group Bayesian network with eight Gaussian variables

Microarray data The microarray data set used in this work was from Gasch *et al*. [69], which measured the response of yeast cells to environmental changes under different conditions. The data set contains 6157 genes and 173 experiments. From this data set, we selected ninety known genes in Actin cytoskeleton group to learn a group Bayesian network for testing purpose. The missing values in the data set were filled in with the average of the known values for each gene. Based on domain knowledge, there are averagely six genes in one group to perform a function, and the number of groups is set to fifteen in our experiment.

We ran the experiments ten times to test whether the learned groups and group Bayesian networks are consistent in the majority of the experiments. In the result, genes ARC15, ARC19, ARP3 and the other three genes are in one of the learned groups in all the experiments. By checking with biological knowledge, we found that these genes are from one gene complex and are functionally related. Another group that contains gene PFY1 is dependent on the group with ARC19. The partial graph is shown in Figure 10. With the learned groups and group Bayesian networks, a domain expert checked the learned group Bayesian network and dependencies between groups of genes. He confirmed that most of the genes in the same groups and the

dependencies between groups are consistent with domain knowledge.

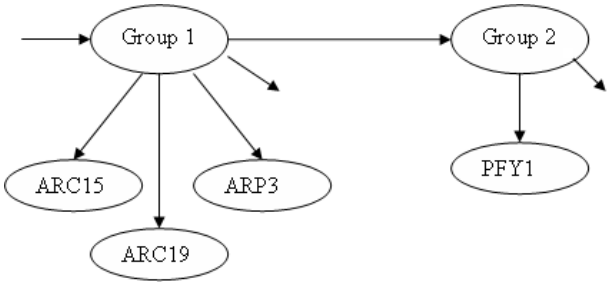


Figure 10 A partial graph from the learned model with genes from Actin cytoskeleton group
Note: Group 1 and 2 contain more genes than those in the figure. The edges without variables at the beginning or the end mean that some variables are not shown here.

3.2.6 Discussion

In this section, a new algorithm is proposed to learn Bayesian network from data in which some variables have similar conditional probability distributions. Within the limits of the experiments and investigations, we have shown that for a class of problems with practical implications, the proposed algorithm could discover groups of variables which follow the same conditional probability distributions, and could identify possible dependencies between groups simultaneously. The learned group Bayesian network is the skeleton of the relationships between the original variables in the domain.

This algorithm has several advantages. First, it will reduce the number of variables in Bayesian network structure learning, since only group variables are used to learn the Bayesian network structure. Reducing the number of variables can narrow down the Bayesian network structure space and speed up the learning process. Second, it will relieve the requirement for the number of instances. Moreover, the learned group Bayesian network is a high-level abstraction of the relationships between the

original variables, such as the activity level of the groups of genes in a cell from the microarray data. Group variables are more reliable in representing the biological functions in a cell, and the dependency relationships between group variables are more stable than the dependency relationships between individual variables. Although our algorithm will lose some details in the sense of the direct interactions between original variables by introducing group variables and restricting the edges between variables in the different groups, it can capture the main interactions between groups, and such high-level abstraction of interactions between gene functions is common in Biology.

In our work, the idea of variable grouping is motivated by the observations from microarray data. The algorithm can be applied to other domains with similar assumptions, such as different industrial sectors in stock market.

There are some future directions for this work. One is overlapped grouping. In the current algorithm, each gene is only assigned to one group. From biological knowledge, however, we know that some genes can perform several functions and belong to different groups. Overlapped grouping is a natural way to model this phenomenon. Another important future direction is to collaborate with domain experts who work on wet-bench biological experiments. The proposed algorithm can generate hypotheses of dependency between genes for future research.

3.3 Summary of Hypothesis Generation

This chapter discussed hypothesis generation from Bayesian network structure

learning. Two general hypothesis forms are introduced first: individual Bayesian networks and individual edges in Bayesian networks. Hypothesis generation is to select the statistically-significant individual Bayesian networks or individual edges in Bayesian networks based on the available data with Bayesian network structure learning. Variable grouping is proposed to generate hypotheses with Bayesian network structure learning in the domain where some variables follow similar conditional probability distributions. In our experiments, the proposed algorithm can partition similar variables into the same groups and learn the dependency between the groups. The concepts and algorithms for hypothesis generation via variable grouping developed in this chapter represent a new effort in this direction. Since there are no available methods for refinement and verification of group variables as hidden variables, the hypotheses generated with variable grouping will not be considered in the subsequent chapters in this thesis. Individual edges in Bayesian networks will be mainly used for hypothesis refinement and verification in Chapter 4 and Chapter 5.

Chapter 4 Hypothesis Refinement for Knowledge Discovery with Bayesian Networks

- Learning Bayesian Networks with Observational Data and Topological Constraints from Domain Knowledge

In Chapter 3, we have discussed hypothesis generation from observational data, which is a Bayesian network structure learning problem in our discussion. In this chapter, we will address one of the common problems in the learned Bayesian networks. This problem is that some edges in the learned Bayesian networks are inconsistent with domain knowledge. Generally, domain knowledge has been verified by experiences or interventional experiments and is considered correct. In this case, we need to adjust the hypotheses of direct influence relationships between variables as edges in Bayesian networks generated with observational data and make them consistent with domain knowledge.

In this chapter, we will discuss the representation of topological domain knowledge, the refinement of the generated hypotheses with the available topological domain knowledge, and the effect of topological domain knowledge on the learned Bayesian network structure.

4.1 Background and Motivation

In Section 3.1.3, we have observed the inconsistency between the learned Bayesian network and domain knowledge in the heart disease problem, where variable "*AGE*" is dependent on other variables. From common sense, we know that variables like "*AGE*" are not affected by other variables in the heart disease problem. Such variables should be root nodes in the related causal Bayesian networks. However, in the learned Bayesian networks, these variables can be the children of other variables, which make the learned Bayesian networks inconsistent with domain knowledge. Similarly, some variables should be leaf nodes in the causal Bayesian networks. For example, in medical domains, the lab test results will not affect other variables in the domain and should be leaf nodes. Moreover, some edges may be known before learning, such as a known edge from variable "*having a cold*" to variable "*running nose*".

This inconsistency problem can be addressed manually, and the inconsistent edges can be deleted from the learned Bayesian networks. This strategy can reduce the inconsistency; however, such modified structure may not be the one with the highest score given the available data and domain knowledge.

Alternatively, topological domain knowledge can be taken into consideration in Bayesian network structure learning to constrain the structure space. Many authors have tried in this direction. Cooper and Herskovits required the complete causal ordering of the variables in a domain and proposed the K2 algorithm [38]. Heckerman *et al.* [87] used a prior network for Bayesian network structure learning. These methods work well theoretically when the required systematic domain knowledge is

available. In practice, however, only partial domain knowledge is available in most cases. *Partial domain knowledge* means that, certain variables are known as roots or leaves in Bayesian networks from the time constraints or other sources, or there are known edges between some variables.

To utilize the available partial domain knowledge, we need to represent domain knowledge as topological constraints to restrict the structure space of Bayesian networks. In this case, domain knowledge should be represented in appropriate ways to facilitate Bayesian network structure learning. Certain kinds of partial domain knowledge have been considered in Bayesian network structure learning ([155], Section 5.4.5) in packages like LibB, TETRAD and Bayesian network PowerConstructor[16]. Experience has shown that the partial domain knowledge can be very helpful in improving both the efficiency and accuracy of the learned Bayesian network structures.

However, as far as we know, there is no systematic representation, analysis and evaluation on incorporating partial topological domain knowledge into Bayesian network structure learning, and the explicit effects and influences of different kinds of topological constraints are unknown. When domain knowledge is not well-expressed, domain experts may specify inconsistent domain knowledge, which may not be easily detected when there are many variables in a domain. Some other issues should also be addressed in Bayesian network structure learning with domain knowledge, such as

[16] LibB: http://www.cs.huji.ac.il/labs/compbio/LibB/
TETRAD: http://www.phil.cmu.edu/projects/tetrad/
BN PowerConstructor: http://www.cs.ualberta.ca/~jcheng/bnsoft.htm

random generation of Bayesian network structures and Bayesian network structure rejection with topological domain knowledge.

In this chapter, we propose two canonical formats to represent the partial topological domain knowledge as topological constraints in Bayesian network structure learning, and examine the effect of topological constraints on the accuracy of the learned Bayesian network structures.

We assume in this chapter that domain knowledge for Bayesian network structure learning is available. The source of domain knowledge is a big issue, and efficiently eliciting domain knowledge from domain experts is an active research topic. Causal knowledge elicitation has been proposed by Nadkarni and Shenoy [124] for Bayesian network construction. The general knowledge elicitation in artificial intelligence domain has been discussed extensively by Firlej & Hellens [56] and references therein.

4.1.1 Related Work

Domain knowledge considered in this chapter is qualitative domain knowledge, such as whether there is a direct edge from one variable to another variable. Donoho & Rendell [45] and Han *et al*. [82] have discussed some general categories of domain knowledge and previous efforts [11,94,95,126] have examined quantitative domain knowledge for Bayesian network learning. Another topic related to hypothesis refinement in Bayesian networks is the general knowledge refinement [72,162,163], where meta-knowledge is used to refine some specific domain knowledge. The work

mentioned above discussed the quantitative domain knowledge or general knowledge. Such knowledge is different from the direct causal influence relationships between variables and not directly applicable to the problem we address here.

4.2 Representation of Topological Domain Knowledge in Bayesian Networks

In this chapter, we consider qualitative domain knowledge for hypothesis refinement in Bayesian network structure learning. **The types of qualitative domain knowledge** considered are: root variables, leaf variables, known and forbidden edges, partial ordering of variables, (conditional) independence relationship between variables, known parents and children, possible parents and children, and the maximal number of parents and children.

These types of qualitative domain knowledge are derived from the understanding of causal Bayesian networks and how qualitative domain knowledge can be applied to causal Bayesian network construction. The root variables are usually determined by time constraint or common sense. As mentioned in the medical domain in the previous chapter, variables like "*AGE*", "*RACE*", and "*SEX*" have their values fixed before other variables and their values are not affected by other variables. Therefore, these variables should be root variables in a causal Bayesian network. Similarly, variables like the lab test results are dependent on other variables and do not affect other variables, and should be leaf variables in a causal Bayesian network.

The known edges and forbidden edges can be from common sense. For example,

there should be an edge from "*having a cold*" to "*runny nose*". There should be no edge from the lab X-ray test results to variable "*Tuberculosis*". Causal ordering is from time constraints and manipulation results. Similar examples can be applied to other qualitative domain knowledge.

We will represent qualitative domain knowledge as topological constraints in Bayesian networks in two formats: the rule format and the matrix format. Table 8 summarizes a common set of topological constraints in the rule format. The column "types of topological domain knowledge" lists all the types of topological domain knowledge we have considered. The column "meaning" explains the rules in the ordinary language. In general, these rules are easy to understand and elicit from domain experts. However, if there are conflicts and cycles in the elicited domain knowledge, it is difficult to detect them in the rule representation.

To facilitate consistency checking, we propose to convert the topological constraints from the rule format into the matrix format: one matrix for the known edges, one matrix for the forbidden edges, one matrix for the partial ordering, one vector for the maximal number of parents, and one vector for the maximal number of children, as summarized in Table 9. If there is a known edge from variable i to variable j, the element (i, j) in the known edge matrix will be 1; otherwise, the element will be 0. If there is a forbidden edge from variable i to variable j, the element (i, j) in the forbidden edge matrix will be 1; otherwise, the element will be 0. If it is known that variable i is before variable j, the element (i, j) in the partial ordering matrix will be 1; otherwise, the element will be 0. The values of the maximal

number of parents and children are non-negative natural numbers. For known edge matrix, forbidden edge matrix and partial ordering matrix, only the elements with value 1 will be used in the learning process; the elements with value 0 just mean that we do not have such knowledge and they will not be used in the learning process.

Types of topological domian knowledge	Meaning
Roots	Variables without parents. Such variables influence other variables, but are not influenced by any other variables
Leaves	Variables without children. Such variables are influenced by other variables, but do not affect other variables.
Known edges	Fixed edges before learning
Forbidden edges	Definitely no such edges
Partial ordering	Variables before some other variables in the causal ordering
(Conditional) independence	Variables conditional independent
Known parents	The parents of some variables are known
Known children	The children of some variables are known
Possible parents	The parents of variables are restricted to a subset of variables
Possible children	The children of variables are restricted to a subset of variables
Maximal number of parents	Numbers of parents of variables can be different and limited
Maximal number of children	Numbers of children of variables can be different and limited

Table 8 Summary of topological domain knowledge in the rule format

Names of components	Meaning
Matrix_k	Matrix for known edges
Matrix_f	Matrix for forbidden edges
Matrix_p	Matrix for partial ordering
V_maxParents	Vector for the maximal parents
V_maxChildren	Vector for the maximal children

Table 9 Summary of topological domain knowledge in the matrix format

These components of topological domain knowledge in the matrix format

summarize all possible topological constraints of the rule format in Table 8. The first ten rules in Table 8 suggest the known edges, forbidden edges and the partial orderings in Table 9. The last two rules suggest the limits on the numbers of parents and children of individual variables.

4.2.1 Compilation of Domain Knowledge from the Rule Format to the Matrix Format

Each rule of domain knowledge in Table 8 corresponds to different values in the matrix format. For example, if a variable is a root in a Bayesian network, it means that there are no edges pointing to it, and the values of elements in the row of the forbidden matrix corresponding to this variable will be 1. For another example, if we know the partial ordering of some variables, we can specify a set of variables before another set of variables or a set of variables arranged in their causal order. For every rule, we have performed such a mapping from the rule format to the matrix format.

4.2.2 Checking the Consistency of Topological Constraints

After the compilation, we have domain knowledge in the matrix format. Before we apply it in Bayesian network structure learning, we need to check the consistency in the specified domain knowledge. By analyzing the properties of the topological constraints, we identified five types of inconsistency:

1) Cycles in the known edges. There should be no cycle of the directed edges from topological domain knowledge;

2) Conflicts in the known edges and the forbidden edges. There should be no overlapping between the known edges and the forbidden edges in valid domain knowledge;

3) Cycles introduced by the partial ordering and known edges. If there are cycles, it means that certain paths of the known edges conflict with the known partial orderings;

4) The number of the maximal parents is smaller than the number of the known parents. The sums of the parents of the known edges to each variable represent the number of the known parents; and

5) The number of the maximal children is smaller than the number of the known children. The sums of the children of the known edges from each variable represent the number of the known children.

These five types of inconsistency are checked step by step. If there are inconsistencies in topological domain knowledge, our program will report them. Currently, we do not computationally resolve the inconsistencies in topological domain knowledge. We leave the work to domain experts or further experiments. In the following sections, we assume that the topological domain knowledge used is consistent and correct.

Running Time for Consistency Checking

As we mentioned above, the matrix format of domain knowledge is easy for consistency checking. For comparison, we have implemented the consistency checking in both the rule format and the matrix format with MATLAB. In the rule format, we need to enumerate all possible paths to check the circles, and the possible

conditions for overlapping. In the matrix format, the consistency checking is done by matrix manipulation. We conducted experiments to compare the running time of the rule format and matrix format for consistency checking.

First we randomly generated domain knowledge with ten variables in rule format to test whether our program can work properly. We manually checked the inconsistency in the generated domain knowledge as the base cases for consistency checking. In our testing, the program can report the same inconsistency in rule format and matrix format as that in manual checking, if applicable.

Next, we ran experiments to compare the time required for consistency checking in two different topological formats. We tested the consistency checking with ten to one hundred variables. For each specified number of variables, we randomly generated fifty different topological constraint sets in the rule format. For each topological constraint set, the consistency checking was performed both in the rule format and the matrix format. The time for consistency checking in the matrix format includes the time to compile the topological constraints from the rule format to the matrix format. The average time in two different cases is reported in Figure 11.

Figure 11 shows that, in our experiment, the time required for consistency checking in the rule format seems exponential in the number of variables in the domain, while the time required for consistency checking in the matrix format seems linear to the number of variables in the domain. On average, the consistency checking in the matrix format takes only about 10% of time required in the rule format. A potential reason is that, in the matrix format, consistency checking is done by matrix

operation, which can check multiple circles in a single manipulation; but in the rule format, it takes more time to enumerate all possible situations for consistency checking.

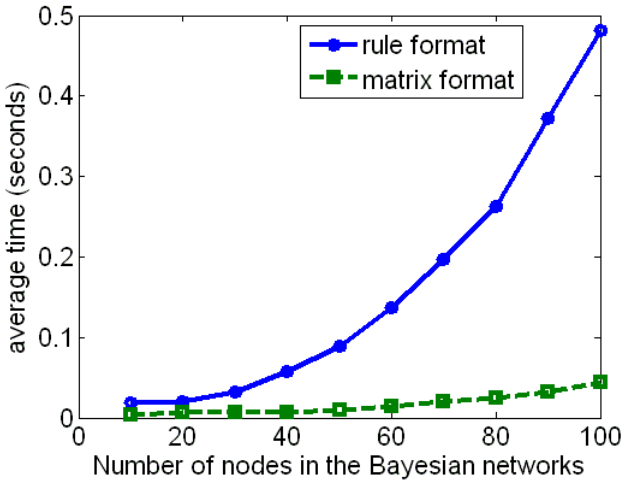


Figure 11 Average time required for consistency checking with different constraint formats

4.2.3 Induction with Topological Constraints

Some topological constraints in Bayesian networks are related to the conditional independence. Dawid [41] examined the axiomatic rules to characterize the conditional independence in a probability distribution: symmetry, decomposition, weak union, contraction, and intersection. In these rules, only the conditional independence information is considered.

Here, we can utilize the graphical properties of Bayesian networks to deduce independence relationships with d-separation (Refer to Appendix A.B for the definition of “D-separation”). A new rule is proposed to derive new topological constraints from the available topological constraints which are not explicitly mentioned by domain experts. This will minimize the expert’s effort in domain

knowledge elicitation stage. Moreover, if there are inconsistencies in the specified topological constraints, the derived topological constraints from this rule may show the direct conflict in the elicited domain knowledge and make the inconsistencies clearer. The rule and the proof are as follows.

Theorem 4.1 Suppose X, Y and Z are three different variables in a domain which can be modeled with a Bayesian network (G, P) where the structure of Bayesian network G is unknown, and P is the probability distribution in the domain that satisfies the causal Markov assumption and corresponds to G. If X is independent of Y, and Z is a parent of X, then Z and Y will be independent.

$$\textit{If} \quad X \perp Y \quad \textit{and} \quad Z \rightarrow X, \quad \textit{then} \quad Z \perp Y$$

- Proof:
  - Suppose that Z is not independent of Y
  - Then there will be a path from Z to Y without v-structure (This is an application of the d-separation criterion)
  - If this path includes X, it means that the part of this path between X and Y will not have v-structure
  - If this path does not pass through X, adding an edge $Z \rightarrow X$ to the path will not introduce v-structure at Z. In this case, there is a path between X and Y without v-structure
  - In both cases, there will be a path between X and Y without v-structure. Then we can conclude that, X and Y are not independent (This is another application of the d-separation criterion)
  - Contradiction! ■

Take the benchmark Asia network (refer to Figure 12) as an example. If we know that variable "*Tuberculosis*" is independent of variable "*Smoking*" and there is an edge

from variable "*Visit to Asia*" to "*Tuberculosis*", then "*Visit to Asia*" should be independent of variable "*Smoking*". This relationship can be easily checked with the structure in Asia network.

4.3 Bayesian Network Structure Learning with Domain Knowledge

After domain knowledge is represented in the matrix format, we can apply it in Bayesian network structure learning. The topological domain knowledge can be used to reject the DAGs which are inconsistent with domain knowledge, and it is applicable to all the Bayesian network structure learning methods, including score-and-search-based approach and constraint-based approach. In the following section, we will use a score-and-search-based approach for illustration.

Greedy Search Algorithm with Topological Constraints

In this work, we adopt greedy search and Bayesian Information Criterion (BIC) score to learn Bayesian networks and Table 10 shows the pseudo code.

Compared with the general greedy search method for Bayesian network structure learning, there are two main differences. First, the initial DAG generated should be consistent with domain knowledge. The base DAG under topological constraints is the one with the known edges only. Other edges can be randomly added to the base DAG under the acyclic constraint in Bayesian networks.

Second, an additional step should be applied to reject the neighbors of the current DAG that are inconsistent with the topological constraints. When we have a set of

DAGs as the candidate structures of Bayesian network, we need to check them and reject the ones inconsistent with topological constraints. The consistency checking is similar to the consistency checking process discussed in Section 4.2.2. This will guarantee that the selected Bayesian network structures are consistent with the topological domain knowledge, and will help to narrow the structure space and speed up the learning process.

Generate an initial DAG ***consistent with the topological constraints*** *as the current DAG*

Done = false

While ~Done

Generate all possible neighbors of the current DAG

Reject the neighbors which are inconsistent with the topological constraints

Evaluate the remaining neighbor DAGs

If the best score of the remaining neighbor DAGs is better than that of the current DAG

Set the neighbor DAG with the best score as the current DAG

Else

Done = true

Table 10 Algorithm for Bayesian network learning with topological domain knowledge

4.4 An Iterative Process to Identify Topological Constraints with Bayesian Network Structure Learning

Domain knowledge elicitation is an important step for hypothesis refinement. In practice, we may not be able to identify all possible topological constraints in one round, due to time constraints and knowledge limitation. We may identify some topological constraints first and then identify others later after we learn the Bayesian

network structures.

In Section 3.1.2, we have mentioned how to estimate the probabilities of individual edges (and other features). After learning, the edges with the highest probabilities and the lowest probabilities can be shown to domain experts. The domain experts will evaluate the significance of these edges with their expertise, and decide whether the edges with high probabilities are real edges and the edges with low probabilities are forbidden edges. If the domain experts confirm that these edges are known edges or forbidden edges, these edges will be included in the topological constraint set. Then the new topological constraint set can be used to learn the probabilities of other edges. This process can repeat until the domain experts confirm that there are no more topological constraints from domain knowledge.

Likelihood of Individual Topological Constraints

Another issue with domain knowledge is whether the specified topological constraints are correct or not. For each individual topological constraint, we can estimate its probability with data and other topological constraints. The known edges in the topological constraint set should have high probabilities. The forbidden edges in the topological constraint set should have low probabilities. If not so, more justification is needed for the known edges with low probabilities and the forbidden edges with high probability, and we leave this issue to the domain experts.

4.5 Empirical Evaluation of Topological Constraints on Bayesian Network Structure Learning

In this section, we examine the effects of topological constraints on Bayesian network structure learning. One expected effect is the speed-up of Bayesian network structure learning with domain knowledge. Another expected effect is the improvement of the learned Bayesian network structure with domain knowledge. We want to know which kind of topological constraints can lead to more correct edges in the learned Bayesian network structure. Such knowledge will help us in data collection, domain knowledge elicitation, and interventional experiment design for causal knowledge discovery.

In this section, the benchmark **Asia network** [104] is used to examine the effects of topological constraints on Bayesian network structure learning, since all possible constraint types can be represented in it. Asia network is a Bayesian network with eight variables that models the situation to determine the likelihood of a person having a disease, given his/her visiting history and smoking habit. The structure of the Asia network[17] is shown in Figure 12.

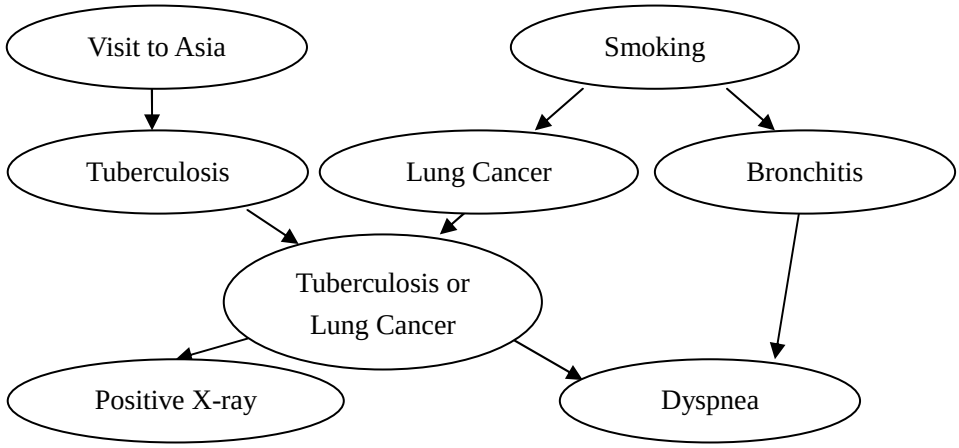


Figure 12 Asia network

[17] Reprinted with the permission from Wiley-Blackwell.

The experiment process is as follows.

1) *Specify the topological constraint sets (empty, with a single topological constraint, or with multiple topological constraints)*
2) *For i=1:n,*
3) *Sample data from the original Bayesian network*
4) *Learn Bayesian networks with the data and the specified topological constraint sets*
5) *Count the edges in the learned Bayesian networks*

Topological constraints are generated based on the structure of Asia network. The single topological constraints are generated systematically, which are the possible roots, leaf nodes, the existing edges as known edges, the non-existing edges as forbidden edges. Topological constraint sets with multiple individual constraints are randomly generated and some topological constraint sets are manually generated to examine the effects of the topological constraint types of interest.

The evaluation criteria are the correct edges in the learned Bayesian network and the Hamming distance between the learned Bayesian network and the original Bayesian network. In addition, the number of the correct structures learned and the number of the learned structure in the Markov equivalent class (as the CPDAG) of the original Bayesian network are counted in the repeated experiments. The correct structure means that the algorithm can identify all the influence relationships between variables from the data. The number of the learned structure in CPDAG is used to measure how the algorithm extracts the conditional independence relationships between the variables from data.

4.5.1 Without Constraints

We first tested the case without topological constraints. We sampled data from the

Asia network to learn Bayesian network structure. Experiments show that it is very difficult to learn the correct complete structure from the data alone, and it is a little bit easier to learn the CPDAGs. This is consistent with theoretic analysis and findings from other researchers [33].

4.5.2 With Individual Topological Constraints

In this experiment, we tested the effects of all possible individual topological constraints in Asia network on the correctness of the learned Bayesian network structure. Individual topological constraint means only one constraint in the constraint set. Totally, sixty-one individual topological constraint sets were generated from Asia network – 1 without constraints, 2 with one root, 2 with one leaf, and 56 with 1 edge as known or forbidden edge. The experiment setup is as follows:

1) *The program ran 36 hours and finished 100 experiments.*
2) *100 different randomly sampled data sets were generated from the original Bayesian network*
3) *Each data set has 1000 randomly sampled instances*
4) *One Bayesian network is learned with each data set and one of the 61 different constraint sets.*

The total number of the learned Bayesian networks is 6100 (=100*61). Some findings from the experiment results are:

1) In the total learned Bayesian networks, only 3 are the same as the original ones. However, more learned Bayesian networks are in the CPDAG of the original Bayesian network. It shows that it is more likely to learn a Bayesian network in the Markov equivalent class of the original Bayesian network [33] from data and

individual topological constraints, other than the exact structure of the original Bayesian network.

2) A constraint set with a single edge from node "*Visit to Asia*" to node "*Tuberculosis*" leads to the highest average correct edges and the minimal average Hamming distances. This edge is an undirected edge in the CPDAG of the original Bayesian network, and such type of edges is **distribution-indistinguishable**. The direction of such edges cannot be determined by the observational data. This result means that, when the constraint set contains the edges that are distribution-indistinguishable, the constraint set can lead to more accurate structure in the learned Bayesian networks.
3) The other two constraints also lead to the learned structures with high accuracy. One constraint is an edge from node "*Lung Cancer*" to node "*Tuberculosis or Lung Cancer*", which is an edge in a v-structure of the original Bayesian network. The other constraint is a leaf node "*Dyspnea*". These results suggested that we need to pay more attention to certain types of topological constraints in practice for knowledge discovery, such as the roots, leaves, edges in v-structure, and edges which are distribution-indistinguishable in Bayesian networks. If possible, we need to determine such types of topological constraints with interventional experiments. This is the task of hypothesis verification in Chapter 5.

4.5.3 With Multiple Randomly-sampled Constraints

In this experiment, we tested the effects of multiple constraints in one constraint set

on the correctness of the learned Bayesian network structure. We want to know which kind of topological constraints or their combinations can lead to Bayesian networks with better accuracy. We randomly selected one to seven possible edges in the original Bayesian network as known edges or forbidden edges of the constraint sets, and totally generated 43 constraint sets. The experiment setup is:

1) *We ran the program for 14 hours and finished 93 experiments*
2) *93 different data sets were randomly sampled from the original Bayesian network in this period*
3) *Each data set has 1000 random instances*
4) *One Bayesian network was learned with each data set and one of the 43 randomly generated constraint sets*

The total number of the learned Bayesian networks is 3999 (=93*43). In the total learned Bayesian networks, 453 (11%) of them were the same as the original one. Compared to the results with individual constraints, where almost no learned Bayesian networks were the same as the original ones, this means that it is more likely to learn the correct Bayesian networks with more topological constraints. This coincides with our belief that, the more the topological constraints we know, the easier to learn the correct edges from the data.

4.5.4 With Multiple Manually-generated Constraints

In this experiment, we want to know the effects of some specific topological constraints and their combinations on the learned Bayesian networks. From the previous experiments, we observed that roots, leaves, distribution-indistinguishable edges and edges in v-structure of the original Bayesian networks are important for the

correct Bayesian network structure learning. Here we want to examine how these topological constraints affect the accuracy of the learned Bayesian network structure and the learning process. We manually generated 12 different constraint sets:

1) Without topological constraints;

2) "*Visit to Asia*" and "*Smoking*" as roots, "*Positive X-ray*" and "*Dyspnea*" as leaves, "*Lung Cancer*" to "*Tuberculosis or Lung Cancer*" and "*Tuberculosis*" to "*Tuberculosis or Lung Cancer*" as known edges;

3) "*Positive X-ray*" and "*Dyspnea*" as leaves;

4) "*Visit to Asia*" and "*Smoking*" as roots, "*Positive X-ray*" and "*Dyspnea*" as leaves;

5) "*Smoking*" to "*Lung Cancer*" as known edge;

6) "*Smoking*" to "*Bronchitis*" as known edge;

7) "*Lung Cancer*" to "*Tuberculosis or Lung Cancer*" as known edge;

8) "*Smoking*" as root, and "*Bronchitis*" to "*Dyspnea*" as known edge;

9) "*Positive X-ray*" and "*Dyspnea*" as leaves, and "*Lung Cancer*" to "*Tuberculosis or Lung Cancer*" as known edge;

10) "*Lung Cancer*" to "*Tuberculosis or Lung Cancer*" and "*Tuberculosis*" to "*Tuberculosis or Lung Cancer*" as known edges;

11) "*Lung Cancer*" to "*Tuberculosis or Lung Cancer*" and "*Bronchitis*" to "*Dyspnea*" as known edges; and

12) "*Visit to Asia*" and "*Smoking*" as roots, "*Positive X-ray*" and "*Dyspnea*" as leaves, "*Tuberculosis or Lung Cancer*" to "*Dyspnea*" and "*Bronchitis*" to "*Dyspnea*" as known edges.

We ran the program for eighteen hours and finished 328 repeated experiments. The results are summarized in Table 11. In Table 11, the rows represent different evaluation criteria, and the columns represent different topological constraint sets. From Table 11, we know that, when the topological constraint sets (sets 2, 4, and 12) contain roots or leaves from the original Bayesian networks, more candidate DAGs will be rejected (refer to row (5) in Table 11), and more correct edges will be recovered in the learned Bayesian networks (refer to row (4) in Table 11).

	Set1	Set2	Set3	Set4	Set5	Set6	Set7	Set8	Set9	Set10	Set11	Set12
(1)	1	0	1	0	5	0	0	0	2	0	0	0
(2)	1	0	3	8	**16**	0	0	0	5	0	0	0
(3)	11	5	6	4	9	11	9	10	4	11	8	5
(4)	3	5	5	5	4	3	4	4	6	4	5	6
(5)(%)	0	**43**	20	**37**	4	4	4	15	23	7	8	**40**

Table 11 Results of Bayesian network structure learning with topological constraints

Note: The row numbers represent (1) number of learned DAGs as expected; (2) number of learned CPDAGs as expected; (3) average Hamming distance; (4) average correct edges; and (5) average percent of DAGs rejected. The columns represent the topological constraint sets from the Asia network: (1) set1 has no constraints, (2) set2, set4, set8 and set12 have the roots specified, (3) set2, set3, set4, set9, and set12 have the leaves specified, and other topological constraint sets have some edges specified.

The constraint set 5 contains an edge from "*Smoking*" to "*Lung Cancer*", which is a distribution-indistinguishable edge in the CPDAG of the original Bayesian network. This constraint set leads to the maximum number of correct CPDAG. It emphasizes that the distribution-indistinguishable edges in the original Bayesian networks are important for Bayesian network structure learning.

4.6 Application of Bayesian Network Structure Learning with Domain Knowledge in Heart Disease Problem

In Section 3.1.3, we applied the hypothesis generation methods to a real heart disease data. In this section, we will apply domain knowledge for hypothesis refinement on this same data set.

Two Bayesian networks are learned from the heart disease data, one without topological constraints (Figure 13, shown in Section 3.1.3 before) and one with topological constraints (Figure 14). In Figure 13, the variables "*AGE*", "*RACE*" and "*SEX*" have parents, which is inconsistent with common sense as these variables are not affected by other variables in the domain. This motivated us to combine topological domain knowledge in Bayesian network structure learning.

In our work, we applied three types of topological domain knowledge. First, variables "*RACE*", "*AGE*" and "*SEX*" are specified as roots in the causal Bayesian networks, since we know from common sense that the probability distributions of these three variables are not dependent on other patient profile information. Second, the family health history variables "*FHY*", "*FDM*" and "*FCAD*" precede all other variables in the partial ordering, since family health history precedes the patient profile in time. Third, there is a known edge from "*Smoker*" to "*CAD*", which is from our current domain understanding. The Bayesian network learned with such topological domain knowledge is shown in Figure 14. As compared with the Bayesian network in Figure 13, the causal Bayesian network structure in Figure 14 is more

meaningful as judged by commonsense: The disease "*CAD*" is dependent on the variable "*AGE*", and the variable "*Smoker*" is dependent on "*SEX*" and "*RACE*". The results in this section and related research with single nucleotide polymorphism (SNP) information were published in World Congress of Health(Medical) Informatics[18] [27,107].

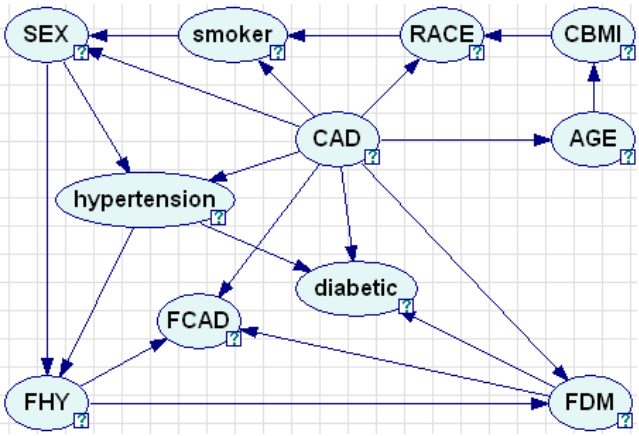


Figure 13 Bayesian network learned without domain knowledge

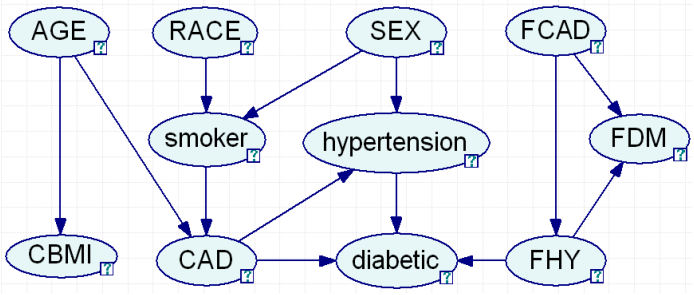


Figure 14 Bayesian network learned with domain knowledge

To be noted, the edges of the learned Bayesian networks in the heart disease problem may not be causal relationships between the variables, although we prefer discovering causal relationships from the data. The learned relationships need to be verified with manipulation criterion, and we cannot manipulate the values of variables in the heart disease domain due to ethical reasons. In the next chapter, we will

[18] The permission for re-printing the materials refers to Footnote 4.

consider the relationships which can be verified with manipulation.

4.7 Application of Bayesian Network Structure Learning with Domain Knowledge and Bootstrapping in Heart Disease Problem

To verify the significance of the edges in the learned Bayesian networks, we applied the bootstrap approach [50] to learn the probabilities of individual edges. We sampled data from the original data with replacement, and two Bayesian networks were learned from each sampled data set – one without topological domain knowledge and one with the topological domain knowledge specified before.

Table 12 and Table 13 show the significant pairs of variables from the bootstrap approach in the learned Bayesian networks with and without domain knowledge. The results showed that almost all the edges in the learned Bayesian networks were quite significant and appeared more than 80% of times in the 500 bootstrap experiments. The pair of variables "*Smoker*-*CAD*" appeared 100% in the learned Bayesian networks as the known edges in Table 12. The top pair of variables in the learned Bayesian networks with bootstrap approach and domain knowledge is "*SEX*-*Smoker*" in Table 12, which appeared surprisingly 100% in the 500 bootstrap experiments. This pair of variables is deemed to be related to each other based on the current domain understanding. Other evaluation methods such as chi-square, mutual information and the Bayesian network learned without domain knowledge, however, did not rank this pair of variables highly. The third top pair of variables in Table 12 is "*RACE*-*Smoker*",

which shows that smoking habits are correlated with race, similar to the research on the adolescents in the United States of America [166].

The fourth top pair of variables in Table 12 is "*AGE-CAD*", which is consistent with common sense: the likelihood of having heart disease depends on the age of the patient. Other highly ranked pairs of variables in Table 12 appeared in the learned Bayesian network with topological domain knowledge (Figure 14). It means that the edges in the learned Bayesian network with domain knowledge are statistically significant.

Both Table 12 and Table 13 show that "*CAD*" is related to smoking habit, diabetes and race. But how one variable will affect another is not clear, and the clinical meaning of these pairs of variables needs further examination.

Order	Variable 1	Variable 2	Occurrences (%)
1	Smoker	CAD	500 (100.0%)
2	SEX	Smoker	500 (100.0%)
3	AGE	CAD	411 (82.2%)
4	RACE	Smoker	406 (81.2%)
5	CAD	diabetic	401 (80.2%)

Table 12 Top edges learned with bootstrap and topological constraints

Note: The percentage in the Occurrences column means the percent of the edges appearing in the 500 learned Bayesian networks with domain knowledge from bootstrap approach.

Order	Variable 1	Variable 2	Occurrences (%)
1	CAD	Diabetic	462 (92.4%)
2	CAD	FDM	443 (88.6%)
3	CAD	Hypertension	439 (87.8%)
4	CAD	RACE	408 (81.6%)
5	CAD	Smoker	402 (80.4%)

Table 13 Top edges learned with bootstrap but no topological constraints

Note: The percentage in the Occurrences column means the percent of the edges appearing in the 500 learned Bayesian networks without domain knowledge from bootstrap approach.

Figure 15 shows the histograms of the running time of Bayesian network

structure learning with and without topological constraints. As indicated, the running time with topological constraints are much shorter than those without topological constraints. The average running time with topological constraints is 65.9 seconds. Compared to the average running time without topological constraints (140.1 seconds), the speed-up of Bayesian network structure learning with topological constraints is more than two times in our experiments.

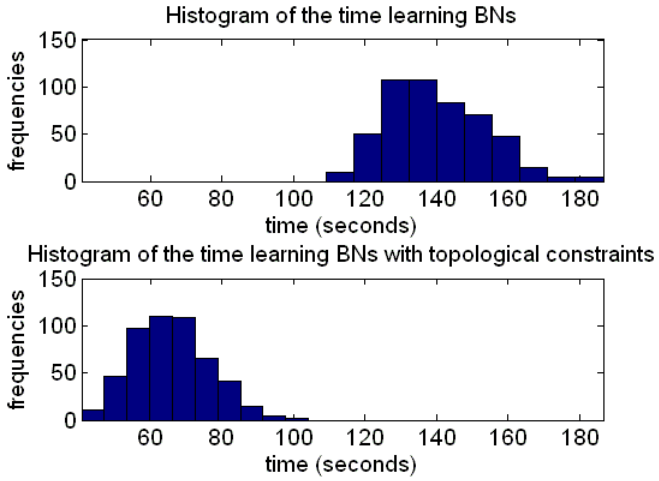


Figure 15 Histograms of times taken to learn Bayesian networks with/without domain knowledge

We notice that the speed-up of the learning process is dependent on the available domain knowledge. In our example, there are eleven variables and one hundred and ten possible edges (two directions for fifty-five pairs of variables). The available domain knowledge in our example is that three variables are roots, three variables are before other eight variables in causal ordering, and one edge is known. Based on such constraints, there are totally fifty-five forbidden edges. The Bayesian network structure space with the constraints was much smaller, which led to the speed-up in Bayesian network structure learning. If the number of the known topological constraints is different, the speed-up will be different too.

4.8 Summary of Hypothesis Refinement

Inconsistency between the learned Bayesian networks and domain knowledge is a big issue in applications of Bayesian networks. When the learned Bayesian networks are consistent with domain knowledge, it will be much easier for domain experts to accept the learned Bayesian networks and apply them in their work. To make the hypotheses generated from data consistent with domain knowledge, we need to incorporate domain knowledge into Bayesian network structure learning.

In this chapter, we have proposed two canonical formats to represent qualitative domain knowledge as topological constraints in Bayesian networks, and identified that some topological constraints are important for Bayesian network structure learning, such as roots, leaves and distribution-indistinguishable edges in the CPDAG of the original Bayesian network.

The two types of domain knowledge representations have different properties. The rule format is easy for domain knowledge elicitation from domain experts. However, the relationship of a specific pair of variables may be specified in several rules. This repeated information may lead to conflicts in the specified domain knowledge if the rules are not well-specified. In addition, it is difficult to detect such conflicts of domain knowledge in the rule format. Alternatively, the matrix format is easy for checking the consistency in domain knowledge. And it is easy to apply the matrix format of domain knowledge in Bayesian network structure learning. We suggest using the rule format to elicit domain knowledge from domain experts and using the matrix format in Bayesian network structure learning. To fill in the gap from

the rule format of domain knowledge to the matrix format, we have proposed the compilation methods for this purpose.

In our experiments, we combined domain knowledge with the score-and-search-based method for Bayesian network structure learning. Experiments on the benchmark Asia network show that topological constraints can increase the validity of the learned Bayesian network structure, especially when the constraint sets consist of roots, leaves, and distribution-indistinguishable edges in the Markov equivalent class of the original Bayesian network. The direction of the distribution-indistinguishable edges cannot be determined with observational data alone but can be identified with interventional experiments.

A case study on a real heart disease data shows that both efficiency and "meaningfulness" of Bayesian network learning can be improved with topological constraints. The significance of the identified direct dependency relationship between variables is estimated with the bootstrap approach. The direct edges in the learned Bayesian network with topological constraints are statistically significant, which can in turn be used as new hypotheses for further analysis.

Chapter 5 Hypothesis Verification in Knowledge Discovery with Bayesian Networks

In the last two chapters, we have discussed hypothesis generation and hypothesis refinement with Bayesian network structure learning. If the goal of our knowledge discovery is for causal prediction and control, one major concern is that these generated hypotheses are merely some kinds of associations and not applicable to situations with causal prediction. This problem is more important when we want to re-engineer the current system to achieve some expected functions, since the associations from observational data cannot provide useful information when the mechanism of the system changes. To determine whether the generated causal hypotheses are real causal knowledge, we need to verify the hypotheses with interventional experiments.

In this chapter, we will discuss causal knowledge discovery with interventions, and consider the situation where multiple data instances are collected in each active learning step. We propose node entropy and edge entropy from the current data to rank the hypotheses, first propose non-symmetrical entropy to select hypotheses for verification and propose an entropy-based criterion to stop the active learning process. The results from simulation show that hypothesis selection with non-symmetrical

entropy requires minimal interventions to achieve the Bayesian network structure with the specified structure entropy than hypothesis selection with symmetrical entropy or random node selection.

Some significant issues need to be emphasized. The first issue is the distinction between observational data and interventional data – whether the data is observed under manipulation (Refer to Section 1.1.4 for details). The second issue is causal knowledge – we adopt the manipulation criterion for causal knowledge (Refer to Section 1.1.1 for details) and apply it to hypothesis verification in the knowledge discovery process.

5.1 Background and the Problem

5.1.1 Roles of Interventional Data in Bayesian Network Structure Learning

In the last two decades, there have been many research efforts to learn Bayesian networks from observational data [38,65,73,86,88,130,155]. However, with observational data alone, it is difficult (if not impossible) to determine the structure of causal Bayesian networks. In most of the cases, only a Markov equivalent class can be learned from observational data [33], which is not sufficient for domains where causal knowledge is required.

A simple example, in which the causal structure cannot be learned from the observational data, is the model with two variables X and Y. From observational data, we may conclude that these two variables are highly correlated. However, we

cannot determine which variable will affect which variable, even with infinite observational data.

Figure 16 shows another example where the Bayesian network structure cannot be reliably learned from observational data. In the model, the binary variables X_1, X_2, and X_3 are independent of each other and all have the value "*true*" with the prior probability 0.5. The value of variable X_4 is determined by the values of X_1, X_2, and X_3. If there are one or three variables of X_1, X_2, and X_3 with values "*true*", the value of X_4 is "*true*" with probability 1.0. If there are zero or two variables of X_1, X_2, and X_3 with values "*true*", the value of X_4 is "*true*" with probability 0.

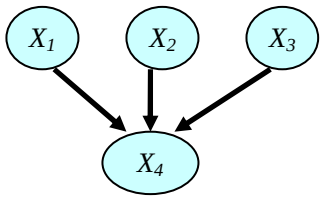


Figure 16 An example which cannot be recovered from observational data reliably

P(X_1=F)=0.5	P(X_1 = T)=0.5
P(X_2=F)=0.5	P(X_2 = T)=0.5
P(X_3=F)=0.5	P(X_3 = T)=0.5
P(X_4=F | X_1=F, X_2=F, X_3=F)=1.0	P(X_4=T | X_1=F, X_2=F, X_3=F)=0
P(X_4=F | X_1=F, X_2=F, X_3=T)=0	P(X_4=T | X_1=F, X_2=F, X_3=T)=1.0
P(X_4=F | X_1=F, X_2=T, X_3=F)=0	P(X_4=T | X_1=F, X_2=T, X_3=F)= 1.0
P(X_4=F | X_1=F, X_2=T, X_3=T)=1.0	P(X_4=T | X_1=F, X_2=T, X_3=T)=0
P(X_4=F | X_1=T, X_2=F, X_3=F)=0	P(X_4=T | X_1=T, X_2=F, X_3=F)= 1.0
P(X_4=F | X_1=T, X_2=F, X_3=T)=1.0	P(X_4=T | X_1=T, X_2=F, X_3=T)=0
P(X_4=F | X_1=T, X_2=T, X_3=F)=1.0	P(X_4=T | X_1=T, X_2=T, X_3=F)=0
P(X_4=F | X_1=T, X_2=T, X_3=T)=0	P(X_4=T | X_1=T, X_2=T, X_3=T)= 1.0

Table 14 The probabilities associated with Figure 16

In this example, the variables in any true sub-sets of the four variables will pass the independence test. However, when all the four variables are considered together, the variables are not independent any more – The value of any variable can be

determined by other three variables. In this case, the true structure of the model cannot be learned from observational data, even with infinite number of instances. This is the reason why we need interventional data for causal Bayesian network structure discovery.

5.1.2 Different Interventions

Interventional data can be obtained by manipulating one or more variables and observing the effects on other variables in a domain. In this chapter, we need to distinguish different kinds of interventions: **node-based interventions** and **edge-based interventions**.

In **node-based interventions**, we will set the values of some variables by manipulation and observe the effects on other variables in a domain. When only one variable is manipulated in a node-based intervention, we say that the non-manipulated variables are the descendants of the manipulated variable if the conditional probability distributions of these non-manipulated variables change as the effect of the manipulated variable. Therefore, with node-based interventions, we can establish the ancestor-descendant relationships as causal ordering of the variables.

From a Bayesian network perspective, manipulating a variable is to mutilate the Bayesian network by cutting the edges to this variable in the original structure and assigning one independent exogenous variable as its parent. The original parents of this variable will not affect the probability distribution of this variable anymore. As an example, the model in Figure 17 will be used for illustration.

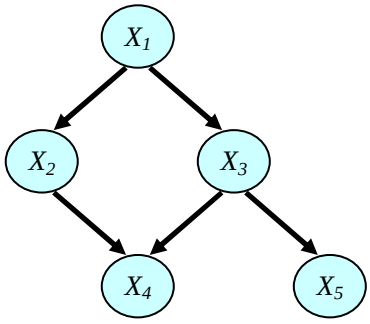


Figure 17 Cancer network

Suppose that variable X_2 is to be manipulated. The mutilated graph of Figure 17 in this situation is shown in Figure 18 and an independent exogenous variable[19] is added as the parent of variable X_2. In Figure 18, variable X_2 is dependent on the exogenous variable, and will not depend on the original parent X_1 anymore. When variable X_2 is manipulated to different values, the probability distribution of X_4 will change, while the probability distributions of other variables will not be affected by the change of variable X_2. From the change of the probability distribution of X_4, we can conclude that variable X_4 is dependent on X_2 and is a descendant of X_2.

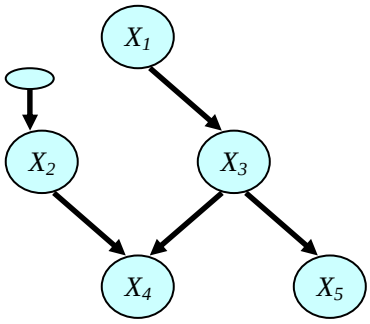


Figure 18 A case of the node-based intervention

Note: The small oval represents an exogenous variable.

Node-based interventions have been explored for knowledge discovery with Bayesian networks recently [121,161]. Another type of interventions **edge-based interventions** has not been explored to our best knowledge. In edge-based

[19] Exogenous variables are shown as small ovals in this and the following examples.

interventions, the interest is the direct causal relationship from one variable (say A) to another variable (say B). To verify the direct causal influence relationship from variable A to variable B, the values of other $n-2$ variables need to be fixed in one of their exponential number of configurations by manipulation, and the value of variable A is changed and the effect on variable B is observed. If the probability distribution of variable B changes when variable A is manipulated to different values under any configuration of other variables, variable B is dependent on variable A in the domain. Since all other variables have been set to specific values by manipulation, there is no indirect path from variable A to variable B – all indirect paths via other variables have been blocked. In this case, the only explanation to the change of variable B is that there is a direct edge from variable A to variable B, and variable A is a parent of variable B in the causal Bayesian network. When variable A is determined to be a parent of variable B in any configuration of other variables, the edge-based intervention will stop the verification of direct relationship from variable A to variable B.

If the probability distribution of variable B does not change when variable A is manipulated to different values under all the configurations of other variables, variable B is not dependent on variable A, and there is no edge from variable A to variable B. In summary, the result from an edge-based intervention can determine whether there is an edge from one variable to another variable.

In the case when there is no edge from variable A to variable B, the data set collected in the edge-based intervention is the effect of variable B when all other

$n-1$ variables (including variable A) are manipulated to different values[20]. This same data set can be treated as the data to examine the causal relationship from any one of the $n-1$ variables to variable B. Following the strategy in the last two paragraphs, the parent set of variable B can be determined with such a data set.

For illustration, suppose that the direct causal influence from variable X_1 to variable X_2 in Figure 17 will be examined with an edge-based intervention. Figure 19 shows the mutilated graph for an edge-based intervention. From the mutilated graph, we can find that variable X_2 is dependent on variable X_1 in some configuration of X_3, X_4 and X_5, and conclude that variable X_1 is a parent of variable X_2. In this case, we can stop the edge-based intervention for the causal influence relationship from X_1 to X_2.

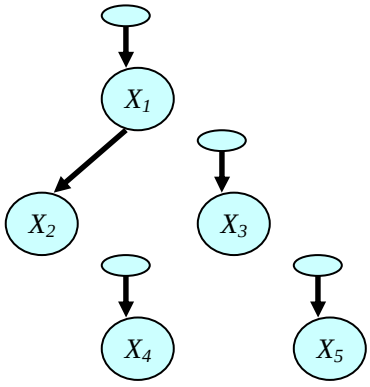


Figure 19 A case of the edge-based intervention

Note: Small ovals represent different independent exogenous variables.

For another example, suppose that the causal relationship from variable X_1 to variable X_4 is selected for an edge-based intervention. Figure 20 shows the mutilated graph. In this case, we need to test the effect of X_1 on variable X_4 under

[20] Note: The concept of Markov blanket in Bayesian networks is not applicable to this situation to reduce the number of variables to be manipulated, since at this stage, we do not know the structure of the underlying Bayesian network and the Markov blanket of the variable to be manipulated.

all possible configurations of X_2, X_3 and X_5. In all the configurations of X_2, X_3 and X_5, variable X_4 is independent of variable X_1. In the end, we can conclude that variable X_1 is not a parent (or a direct cause) of variable X_4 in the domain.

In Figure 20, we have manipulated variables X_1, X_2, X_3 and X_5 to all their possible configurations to determine whether there is a direct edge from X_1 to X_4. The data set collected in this step can be used to determine the relationship from X_2 to X_4, since variable X_2 has been changed to different values under all configurations of X_1, X_3 and X_5 in the same data set. In this example, we can determine that variable X_2 is a parent of variable X_4 from the data. The same procedure can be applied to the relationships from all other variables to variable X_4 with the same data. In the end, we can achieve the result: variables X_2 and X_3 are parents of variable X_4, and variable X_5 is not a parent of X_4. Therefore, with an edge-based intervention, we can identify the parent set of the target variable.

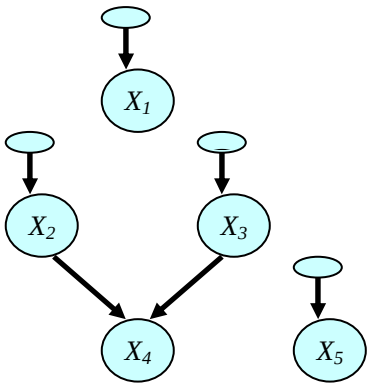


Figure 20 Another case of the edge-based intervention

Note: Small ovals represent different independent exogenous variables.

From causal Bayesian network perspective, the result of an edge-based intervention is whether there is an edge from one variable to another variable. We

treat the determination of such an edge as one edge-based intervention. Regardless whether there is an edge or not, the information can be used as a known edge or a forbidden edge and added to the topological constraint set for further causal Bayesian network structure learning. If all possible edges between variables have been determined by edge-based interventions, the results can be combined to build the complete Bayesian network structure.

Compared with node-based interventions, the advantage of edge-based interventions is that it is possible (probably the only method) to verify the direct causal influence relationship from one variable to another variable. The disadvantage of edge-based interventions is that the number of the observed instances can be exponential, since we need to consider all possible configurations of the $n-1$ variables in edge-based interventions. Although an exponential number of instances are needed, Fisher [57,59] claimed that complex experiment designs (such as factorial designs in edge-based interventions) were more efficient than studying one factor at a time for causal knowledge discovery.

When we compare the observational experiments, node-based interventions and edge-based interventions, we can see that node-based interventions are a general case of the experiments: an observational experiment is a special type of node-based interventions without any variable manipulated, and an edge-based intervention is a special type of node-based interventions with $n-1$ variables manipulated. To distinguish the different experiments, we will name the node-based interventions in the following sections as the experiments with at least one variable manipulated and

at most $n-2$ variables manipulated.

Based on different interventions, we can get the observational data from experiments without manipulation, the node-based interventional data from node-based interventions, and the causal influence relationship between two variables from the edge-based interventions, respectively.

5.1.3 Related Work

With different types of data, there are requirements to combine them for effective and efficient causal knowledge discovery. Recently, some new methods have been proposed to combine observational data with interventional data for this purpose [39,47,85,121,161].

Cooper and Yoo [39] examined the assumptions by combining the observational and interventional data for Bayesian network probability updates. They extended the Bayesian method for observational data by Cooper and Herskovits [38] and Heckerman *et al.* [87] to the mix of observational and interventional data. In particular, when one variable is manipulated to specific values in some data instances, these data instances will not be used to update the probabilities of the family[21] in which this variable is the child variable of the family. Under the assumptions of complete data and no hidden variables and other assumptions [39], the likelihood of a data set can be estimated in a closed form if the Bayesian network structure is known. Yoo *et al*. [175] applied the extension of this method to gene regulatory pathway discovery with

[21] In Bayesian networks, a family means a partial structure that consists of a variable and all its parents.

simulated observational and interventional microarray data, and generated some hypotheses of influence relationships between genes which are supported by the results in the scientific literature.

Sachs *et al*. [145] applied Bayesian network structure learning method to a real biological domain – the intracellular signaling networks of human primary naïve $CD4^+$ T cells. They conducted real biological experiments to collect observational data and node-based interventional data, and applied the methods from Heckerman *et al.* [87] and Pe'er *et al.* [128] to learn Bayesian network structures. A representative network with seventeen high-confidence edges was chosen from the average of five hundred high-scoring structures. After searching the literature, they claimed that fifteen of the seventeen high-confidence edges had been reported in literature (and three real edges are missing in their learned structure). Then they conducted the real biological experiments to verify whether the remaining two edges are causal relationships, and the experiment results were statistically significant. This is a real success of Bayesian network learning in real application, although the selection of the high-scoring structures and the selection of the threshold for high-confidence edges are arguable.

Eberhardt *et al.* [48,49] proved that, under ideal conditions with causal Markov assumption and faithfulness assumption (and ideal probability distributions), the number of the experiments required to identify the causal relationships between n variables is $n-1$ when at most one variable is manipulated each time, and $\log_2(n+1)$ when multiple variables can be manipulated simultaneously.

Meganck *et al.* [119] assumed that their method starts with the correct CPDAG from the observational data and determines the directions of the un-directed edges in the CPDAG with interventional data. Eaton and Murphy [47] discussed different kinds of manipulations and proposed uncertain intervention for Bayesian network structure learning with interventional data.

Although the effects mentioned above [39,47-49,119] achieved significant results, only passive learning is considered in the learning process. **Passive learning** works with a set of readily available data; the data set does not change in the learning process. More interventions and more data are needed to achieve the required criteria, which can be quite expensive.

5.1.3.1 Active Learning

Active learning is a method that samples new data during the learning process. It tries to collect new data with the help from the existing data. Typically, its goal is to reduce the uncertainty in the model. Therefore, active learning is more effective and efficient than random sampling, and requires a smaller number of data instances for knowledge discovery [35,113,138].

The general active learning framework ([110], page 19-20) is shown in Figure 21. It starts with some prior information (including data and domain knowledge). Next, it estimates the probability of each possible observation under every action, and the posterior of the selected measure with each observation under every action. Then it estimates the expected posterior loss for every action. The action with the maximal

expected posterior loss is selected for an experiment. After the experiment, new data are collected and combined with the existing data for the next round of active learning. In active learning, the action space, the observation space under each action and the estimation of the posterior are usually exponential. The product of these three spaces is exponential too. Heuristics from domain knowledge are needed to reduce the space and speed-up the learning process.

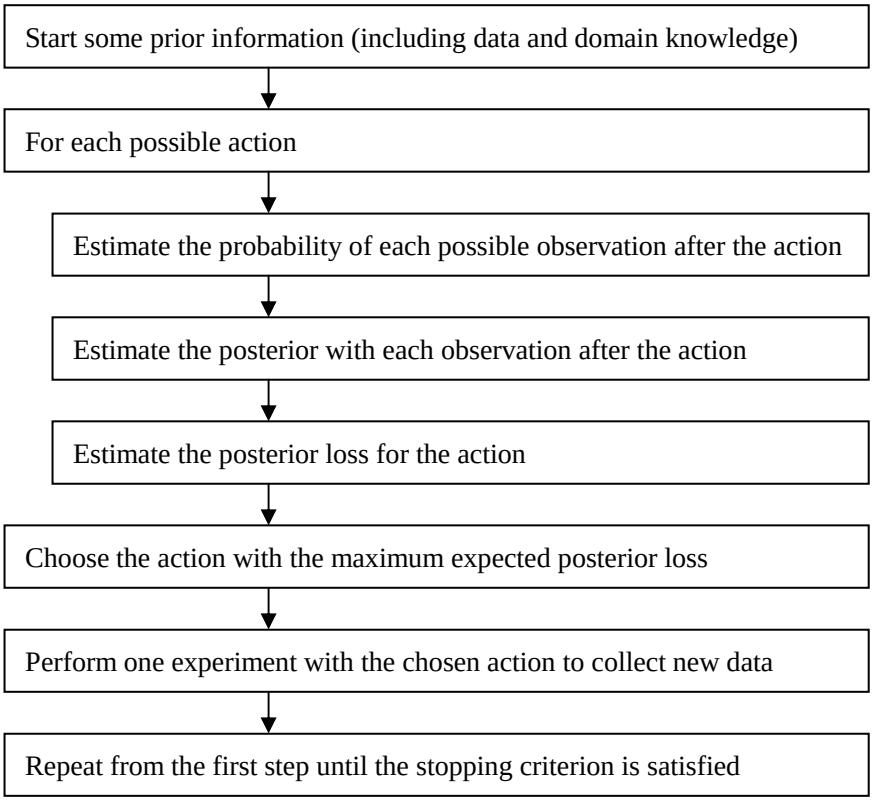


Figure 21 The general framework for active learning

Tong and Koller [161] and Murphy [121] applied the above active learning framework to guide the experiments to collect interventional data for probability update in Bayesian networks. In active learning of causal Bayesian networks [121,161], the learning process starts with an available data set, and the expected posterior loss of each possible intervention is used as a criterion to select nodes for node-based interventions. Suppose the domain has n binary variables. The number

of possible interventions is $3^n - 2^n$ if the variables can be manipulated simultaneously[22]. If only one variable is manipulated in each intervention, the number of possible interventions is $2n$. When one variable is manipulated to a specific value, there are 2^{n-1} possible observations from other $n-1$ binary variables. Each possible observation should be combined with the existing data to estimate the edge probabilities and the structure entropy of the Bayesian network. With the structure entropy, the node with the maximal expected posterior loss is selected for intervention, and a new data instance is collected. This process for the expected posterior loss estimation and the new data collection can be repeated until the maximal number of interventions is reached.

The computational complexity to select one node for an intervention is $o(n*2^nT)$, where T is the time to estimate the edge probabilities in one situation. Estimating the edge probabilities need to sum over all possible Bayesian networks, which is already a big computational challenge. The complexity of the best approach currently available to estimate all the edge probabilities is $o(n*2^n)$ [97]. The computational complexity to select one node for an intervention is $o(n^2*2^{2n})$. Monte Carlo method was used to sample Bayesian networks for approximate edge probability estimation in [121,161], but the convergence is very slow.

We have shown that the complexity with one instance being collected at an

[22] In a manipulation, each binary variable can be in one of 3 possible conditions: manipulated to two different values or not manipulated. Totally, there are 3^n possible combinations of manipulations for n binary variables. The 2^n cases with all n binary variables manipulated should be excluded, since there are no observational variables in these cases which cannot provide any information for knowledge discovery.

intervention is computationally intensive. In real applications [145], multiple data instances can be collected during an intervention. The active learning method based on expected posterior loss is not applicable to multiple data instances for a few reasons. First, if only one of the multiple data instances from each intervention is used to estimate the expected posterior loss, it will be a waste of other instances in the available data. Second, if all multiple data instances are considered to estimate the expected posterior loss, the edge probabilities need to be estimated in $2^{m(n-1)}$ possible observations (assume that there are m instances collected in each intervention in the domain with n binary variables and only one variable is manipulated). Monte Carlo method can be used to sample possible observations and possible Bayesian networks for edge probability estimation. But a small sample of possible observations will give a very biased result and a big sample may not be feasible in a reasonable time.

The methods mentioned above are not practical, even if only one instance can be collected from one intervention. To achieve a reliable estimation of edge probabilities (or the probabilities of the Bayesian network structures), many data instances and many interventions are required. Since intervention is usually expensive and needs more time, the required interventional instances cannot be easily collected in practice. Therefore, active learning methods based on multiple data instances from one intervention need to be studied.

5.1.4 The Problem and Our Proposed Solution

The problem to be addressed in this chapter is to identify the complete causal Bayesian network structure with observational data, node-based interventional data and the results from the edge-based interventions when multiple data instances can be collected in each intervention. This is a common goal in many reverse engineering areas, such as the biological pathway research. The application domain is where we can collect observational data economically and can manipulate the variables with more cost for causal knowledge discovery. Since more cost will be involved in interventional experiments, it depends on the application's objective to adopt interventional experiments for the complete causal Bayesian network structure. If the true causal structure is very important, the node-based and edge-based interventions are compulsory for causal knowledge discovery. In the following sections, we assume that the node-based and edge-based interventions are needed.

Our objective is to minimize the number of interventions, subject to the identification of the complete causal Bayesian network structure. We will utilize the available data and topological constraints to generate hypotheses of causal influence relationships between variables for interventions. The data from node-based interventions will be used to update the probabilities of hypotheses. The results from edge-based interventions will be used as topological constraints in Bayesian network learning. The main challenges in this task are: 1) how to choose a node for node-based interventions, or how to choose the pair of variables for edge-based interventions, and 2) when to stop the intervention process.

We propose a non-symmetrical-entropy-based method to select nodes or edges for new interventions and propose an entropy-based criterion to stop the active learning process. Our simulation results show that, on average, our non-symmetrical-entropy-based method requires the minimal number of interventions to identify the complete causal Bayesian networks. For the stopping criterion, we found that the structure entropy is the best method in the sense that the learned structure is very similar to the original structure when the learning process stops. These results are promising and instructive to many reverse engineering tasks where the goal is to identify the causal structure in the domain.

The relationships between our work and some related efforts are as follows.

5.1.4.1 Relationship to Experiment Design

Traditional **experiment design** [18] is a discipline that has broad application across all the natural and social sciences. In traditional experiment design, the experimenters are interested in the effect of some interventions on certain objects, which are the hypotheses in the experiments. The objective of experiment design is to organize the experiments to facilitate the data collection and hypothesis evaluation. In Bayesian experimental design [22], the experimenters have the prior probabilities of the parameters in the experiment, and try to optimize the parameters based on the expected posterior. In optimal experimental design [22], the experimenters try to optimize the experiment parameters without prior information, which is a limiting case of Bayesian experimental design when the data is sufficiently large. In all these

settings, the experimenters obtain the hypothesis from some sources, which is external to the experiment design process.

In our work, the learning process with data and domain knowledge is to select significant hypotheses of causal influence relationships between variables for new experiments. With the selected hypotheses, we can apply traditional experiment design methods to test the hypotheses. Our work for edge or node selection is one step ahead of the traditional experiment design, which will provide more informative hypotheses to test and make the identification of the complete causal structure more efficient.

5.1.4.2 Relationship to Closed-loop Data Mining

Traditional data mining is usually an open-loop process [111,112]. After we generate and deploy the conclusions from data mining methods, the data mining process will usually stop. However, the conclusions from data mining are not the end of the story. We need to know the effects of results from data mining methods in real applications. If the results are not good enough, we should try data mining methods again with the feedback from the real applications, and verify the hypotheses generated with data mining methods for further improvement. The entire data mining process is repeated and the closed-loop data mining is required.

In our work, node selection and edge selection are for hypothesis verification. The node-based intervention is a way to verify causal orderings in Bayesian networks. The edge-based intervention is a way to verify direct causal relationships between

variables. With such verification, the validity of hypotheses will be improved.

5.2 Assumptions for Applying Active Learning with Interventions

To apply active learning for causal Bayesian network discovery with interventions, we use the following assumptions.

Assumption 1. The underlying causal mechanism in the domain is stable. This assumption requires that observational data are collected from the same system mechanism and the interventional experiment is working on the same mechanism. If a causal Bayesian network is used to represent the mechanism of the domain, this assumption means that the structure and parameters of the underlying Bayesian network do not change during the data collection and experiment periods. This is a basic assumption for all the research where we need repeated experiments and observations.

Assumption 2. There is no feedback in the domain. Feedback can lead to directed cycles, which are not allowed in Bayesian networks. If there are feedbacks in a domain, it is not appropriate to represent the mechanism in the domain with a general Bayesian network. Dynamic Bayesian networks have been proposed to extend Bayesian networks to the situations with feedback [70,123].

Assumption 3. The underlying mechanism in the domain can be represented as a causal Bayesian network. Only under this condition, we can apply causal Bayesian networks to the problem.

Assumption 4. There are no hidden variables in the domain. This assumption is also known as the causal sufficiency assumption. It means that all variables are directly observable. If there are hidden variables, we cannot observe and manipulate them for causal knowledge discovery, and the causal relationship between hidden variables and other variables cannot be determined directly. The group variables in Section 0 are hidden variables. So they are not considered in this chapter.

Assumption 5. All variables are atomic and can be directly manipulated. This assumption means that no variables are logical functions of other variables and the values of the variables can be manipulated directly, instead of changed by some intermediate variables in the domain.

Assumption 6. When there are manipulations on some part of the structure or the values of some variables in a domain, the causal mechanism of the other parts in the system, including structure and parameters, do not change except the edges to the manipulated variables and the values of the manipulated variables. This is the invariance requirement on the causal relationships between the manipulated variables in the domain and other parts of the structure and parameters. Only under this assumption, the results from the interventional experiments can be applicable to the original system.

Assumption 7. It is possible to conduct the node-based and edge-based interventional experiments and observe the effects of the manipulated variables on other variables.

Assumption 8. The results from the edge-based interventions are concrete knowledge of the examined edges. The result is deterministic about the edge, i.e., there is an edge

or not; if there is an edge, the direction of the edge is also known.

Assumptions 3 and 4 in the list have been discussed by Cooper and Yoo [39]. Other assumptions are derived from our understanding of domain knowledge and commonsense. For the domains where causal knowledge is required and causal Bayesian networks are the appropriate models, these assumptions are general and applicable. For example, in agricultural research, Wright [172] was probably the first to use a graphical model in analysis of crop failure. Recently, Sachs *et al.* [145] applied causal Bayesian network to protein-signaling networks in biological domain.

5.3 Hypothesis Verification with Node-based Interventions

In this section, we will discuss the node-based intervention when one variable is manipulated and multiple data instances can be collected in each intervention step. In Section 5.1.3, we mentioned that the previous work is not applicable in this situation due to the computational complexity.

One computationally-intensive problem in previous work is to estimate the expected posterior loss. If one variable is manipulated and m instances are observed in one intervention from a domain with n binary variables, edge probabilities should be estimated under $2^{m(n-1)}$ possible observations. With the current best method [97] for edge probability estimation, the computational complexity is $o(n^2 * 2^{mn+n-m})$ for each active learning step, which is infeasible even for very small n and m. A possible way to solve this problem is to select variables for intervention based on the

node uncertainty from the current data, rather than the expected posterior loss estimation. Our observation is that, when a big interventional data set is collected, the influence relationship from the manipulated variable to the non-manipulated variables can be determined and the node uncertainty from the manipulated variable to non-manipulated variables can be reduced significantly (even totally). This means that node uncertainty from the current data can be used as an indicator to select nodes for node-based intervention.

Another observation is that intervention is non-symmetrical in nature. In an intervention, we can only manipulate one variable in a pair of variables to derive the causal information between this pair of variables: whether the manipulated variable affects the non-manipulated variable; we cannot derive the causal information from the non-manipulated variable to the manipulated variable. If both variables are manipulated, we cannot derive any causal information between this pair of variables from the interventional data.

Therefore, we propose node uncertainty and non-symmetrical entropy from the current data for node selection. In this way, the exponential number of possible observations for the expected posterior loss estimation can be avoided. After a node is selected for intervention, equal numbers of instances will be collected when this node is manipulated to different values.

There are two main issues in node-based interventions – **node selection criteria** and **stopping criteria**. Before further discussion, we will discuss some uncertainty measures in Bayesian networks.

5.3.1 Bayesian Network Uncertainty Measures

The intuitive option is to use probabilities of all possible DAGs to measure the uncertainty of the Bayesian network structure given the available data. In the ideal condition, the DAG of the true structure has probability 1, and other DAGs have probability 0. However, in practice, we cannot obtain the probability 1 for one DAG[23], since the data is not ideal. Even if the data is ideal, we need to enumerate all possible DAGs to find the optimal DAG, which is infeasible for a reasonably large number of variables, since the number of DAGs is exponential in the number of variables.

Another option is to measure the uncertainty of each pair of variables and use the sum of uncertainties from pairs of variables as the structure uncertainty measure. In Section 3.1.2, we have discussed how to estimate the edge probabilities. Suppose that we have two variables A and B. There are three possible conditions between A and B in a Bayesian network: 1) there is an edge from A to B, $A \to B$; 2) there is an edge from B to A, $A \leftarrow B$; and 3) there is no edge between A and B, $A \perp B$. With these three conditions, the entropy between variables A and B is calculated with the following formula [161]:

$$\begin{aligned} H_S(A,B) = & -p(A \to B)\log p(A \to B) \\ & - p(A \leftarrow B)\log p(A \leftarrow B) \\ & - p(A \perp B))\log p(A \perp B) \end{aligned} \quad (1)$$

The entropy of the Bayesian network structure is the sum of the entropy of all

[23] The Markov equivalent class of Bayesian networks means that some Bayesian networks are distribution-equivalent and some edges can be in either direction. Alternatively, manipulation can determine the direction of edges by experiments, and all Bayesian networks can be distinguished from each other by manipulation.

possible pairs of variables:

$$H_S(G) = \sum_{A \neq B} H_S(A, B)$$

In an ideal condition, only one of the three conditions between variables A and B is with probability 1 and other two are with probability 0. The entropy between variables A and B will be 0. If all pairs of variables are ideal, the entropy of the real DAG will be 0.

Another Option for Edge Entropy

In the case mentioned above, the edge entropy between two variables is symmetrical. The word "symmetrical" means that the edge entropy between A and B is calculated with the three conditions between A and B: $A \to B$, $A \leftarrow B$ and $A \perp B$. The conditions $A \to B$ and $A \leftarrow B$ are treated equally. In this case, when a pair of variables are selected with high entropy, it does not tell us which variable to manipulate: A or B.

To know which variable to be manipulated, we need to distinguish the conditions of $A \to B$ and $A \leftarrow B$ for edge entropy calculation. We propose to calculate the entropies of the two situations separately.

$$H_{NS}(A \to B) = -p(A \to B)\log p(A \to B) - (1 - p(A \to B))\log(1 - p(A \to B)) \quad (2)$$

To distinguish these two entropy definitions, we call the one in Formula (1) with three edge conditions as **symmetrical edge entropy**, and the one in Formula (2) with two edge conditions as **non-symmetrical edge entropy**. The non-symmetrical edge entropy is from the observation that intervention is non-symmetrical in nature.

Estimating the edge probabilities is important for edge entropy calculation in

active learning. In the previous active learning work [121,161], edge probabilities are estimated approximately with Markov Chain Monte Carlo (MCMC). In this section, we propose to estimate the edge probabilities with an exact method by Koivisto [97], since the exact edge probabilities can provide accurate information. When the interventional data is combined with observational data, the instances with the manipulated variable will not be used in calculating the probability of the family with the manipulated variable as the child (the assumptions and the method can be referred to Cooper and Yoo [39]) .

5.3.2 Selecting Nodes for Node-based Interventions

We propose to choose the node with maximal node uncertainty for intervention. The node uncertainty between a variable and all the other variables can be estimated from edge entropy.

$$H_{NS}(A) = \sum_{B} H_{NS}(A,B) \tag{3}$$

$$H_{S}(A) = \sum_{B} H_{S}(A,B) \tag{4}$$

Where $H_S(A,B)$ and $H_{NS}(A,B)$ are defined in formulas (1) and (2). Similar to edge entropy $H_S(A,B)$ and $H_{NS}(A,B)$, we refer to $H_{NS}(A)$ as **non-symmetrical node entropy** and $H_S(A)$ as **symmetrical node entropy**.

5.3.3 Stopping Criteria for Causal Structure Learning

Another main practical problem in applying Bayesian network learning for causal knowledge discovery is when to stop the learning process – when do we think that the

learned causal Bayesian network is good enough?

The intuitive way is to choose the number of interventions as the stopping criterion. The disadvantage of this approach is that there is no guarantee on the quality of the learned Bayesian network structure. We propose to use certain "acceptable" entropy of the learned structure as the stopping criterion. The ideal entropy of the learned structure is 0; however, it is difficult to reach in practice. We consider the effects of the different entropies from the learned structure as the stopping criteria on the accuracy of the learned structures.

5.3.4 Topological Constraints

In practice, we may have domain knowledge which can be used as topological constraints in causal Bayesian network structure learning, as discussed in Chapter 4. In Koivisto's method [97] for edge probability estimation, the families of variables will be set as impossible ones if some corresponding edges are not allowed in the topological constraints.

5.3.5 Experiments for Node-based Interventions

The proposed method has been tested in experiments with five different Bayesian networks: two Bayesian networks created by ourselves (**Study network** and **Cold network**), and three benchmark Bayesian networks used for active learning in Tong and Koller [161] (Cancer network, Asia network, and Car network). Cancer network has five variables (Figure 17), Asia network has eight variables (Figure 12), and Car

network has twelve variables for car trouble-shooting, respectively. Our Bayesian networks (Study network and Cold network) are shown in Figure 22 and Figure 23, and the corresponding hypothetical CPDs are in Table 15 and Table 16, respectively. Study network and Cold network have the canonical structures of Bayesian networks, and the direction of the edges in these two networks cannot be learned with observational data alone, even with infinite number of instances.

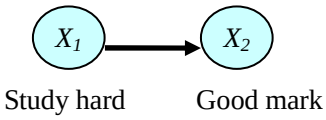


Figure 22 A hypothetic Study network

$P(X_1=F)=0.2$	$P(X_1 = T)=0.8$
$P(X_2=F \mid X_1=F)=0.6$	$P(X_2=T \mid X_1=F)=0.4$
$P(X_2=F \mid X_1=T)=0.2$	$P(X_2=T \mid X_1=T)=0.8$

Table 15 The corresponding CPDs of Study network

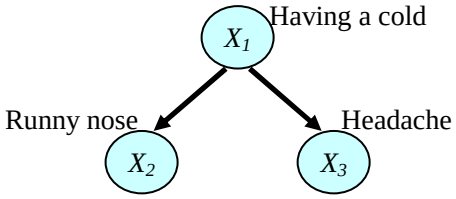


Figure 23 A hypothetic Cold network

$P(X_1=F)=0.7$	$P(X_1 = T)=0.3$
$P(X_2=F \mid X_1=F)=0.95$	$P(X_2=T \mid X_1=F)=0.05$
$P(X_2=F \mid X_1=T)=0.1$	$P(X_2=T \mid X_1=T)=0.9$
$P(X_3=F \mid X_1=F)=0.92$	$P(X_3=T \mid X_1=F)=0.08$
$P(X_3=F \mid X_1=T)=0.75$	$P(X_3=T \mid X_1=T)=0.25$

Table 16 The corresponding CPDs of Cold network

The experiment setup is as follows and the flowchart is shown in Figure 24:

1) *Choose a Bayesian network from Cancer network, Asia network, Car network, Study network, or Cold network as the ground truth Bayesian network;*
2) *Sample a data set with N_obs observational instances from the ground truth Bayesian network;*

3) *Estimate the edge probabilities, node entropy and structure entropy with the available data (and domain knowledge, if any);*
4) *Check the stopping criterion. If the stopping criterion is satisfied, stop the learning process; otherwise, continue;*
5) *Select one node for intervention based on the criteria in formula (3) or (4), random node selection for intervention, or without manipulated node; and*
6) *Generate a new data set with N_int interventional instances from the ground truth Bayesian network with the selected variables manipulated to different values; return to step 3).*

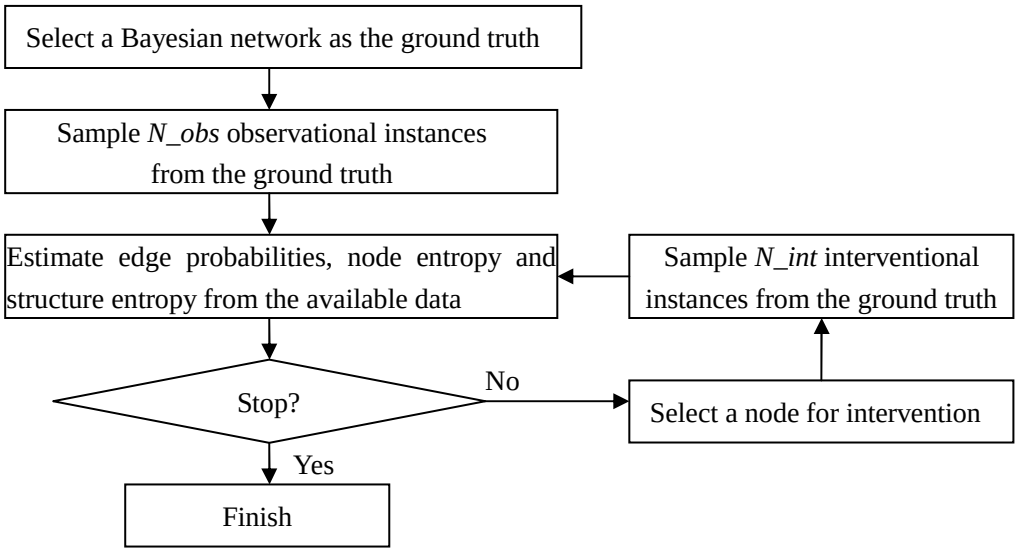


Figure 24 Flowchart of active learning with node-based interventions

In the experiments, the uniform prior is used for Markov equivalent classes as in Heckerman *et al*. [87] to create the structure prior. The edge probabilities are estimated by Koivisto's exact method [97] with the extension to the combination of observational and interventional data as discussed in Cooper and Yoo [39]. The size of the initial observational data *N_obs* is set to 20, and the size of the interventional data in each intervention *N_int* changes from 1 to 200 instances. Such size of data for each intervention is more realistic than an ideal probability distribution required in Eberhardt *et al.* [49].

Two different stopping criteria are tested in our experiments - the number of interventions and the structure entropy of the learned Bayesian networks. Different

numbers of interventions have been tested as stopping criteria in our experiments. The maximal number of total interventional instances is set to 1000 for Study network and Cold network, 2000 for Cancer network and 5000 for Asia network and Car network. There are two reasons to set the maximal number of total interventional instances: 1) after trying different numbers of the total interventional instances, we found that the structure entropy of the learned Bayesian networks can converge with the specified number of interventional instances from non-symmetrical-entropy-based node selection in our experiments; and 2) we had observed that the learned Bayesian network would not reach certain small structure entropy when node selection is based on symmetrical node entropy, even if a very large data set is sampled. The maximal number of interventional instances is used to stop the learning process when the structure entropy is used as the stopping criteria.

Besides node selection with symmetrical node entropy and non-symmetrical node entropy, we consider random node selection for intervention and consider the situation without manipulation (i.e., there is no manipulated variable in new data collection at each step, and the data is observational data).

When one variable is selected for intervention, the edges pointing to this variable will be removed from the ground truth Bayesian network and this variable will be manipulated to specific values. The values of other variables are sampled based on the Bayesian network structure and the original conditional probabilities. In addition, one variable can be selected for more than one round of intervention in the active learning process.

In our experiments, we have tested: 1) which method requires the minimal number of node-based interventions to achieve required structure entropy? 2) what are the relationships between the number of interventions and the entropy of the learned structure? 3) what is the relationship between the number of interventions and Hamming distance between the learned structure and the ground truth Bayesian networks? 4) which stopping criterion can achieve a structure with smaller structure entropy?

The experiments show that non-symmetrical entropy is the best method for node selection to learn causal Bayesian networks with the minimal structure entropy. The conclusions from different Bayesian networks are similar, and the results from the different sizes of interventional data from each intervention are similar. In the following section, the results will be demonstrated with Cancer network with the size of the interventional data as 200. More results are listed in Appendix A.C.

We first used the original conditional probabilities in the Bayesian networks for test. To examine whether the results from the specific values of the conditional probabilities in the original Bayesian networks can be generalized to different conditional probabilities, we conducted experiments with the same Bayesian network structures but with randomized conditional probabilities. The conclusions from the experiments with the randomized conditional probabilities are similar to the results with the original conditional probabilities. In the following sections, we will only discuss the results from the original conditional probabilities.

5.3.5.1 Number of Interventions vs. Structure Entropy

In the first experiment, we tested the relationship between the number of interventions and the entropy of the learned structures. The objective is to show how the entropy of the learned structures varies with the different number of interventions. In the experiment, we found that, in order to reach small structure entropy in the learned Bayesian networks, the required number of interventions is dependent on the number of instances to be collected in each intervention. The maximal number of intervention is set to the division of the total instances and the number of instances to be collected in each intervention. For Cancer network, the maximal number of interventions[24] is set to 6 when the size of the interventional data is 200 in each active learning step.

The programs ran eight hours and finished 608 repeated experiments[25] on the Cancer network (about 48 seconds for one experiment). The results are shown in Figure 25, where the lines represent the change of the average structure entropy with the number of interventions. Figure 25 shows that, with the same number of interventions, node selection with non-symmetrical node entropy can derive a Bayesian network with the lowest entropy (also with the smallest variance) on average, which means that the structure learned with non-symmetrical node entropy has less uncertainty. This is consistent with our expectation, since the interventions

[24] The maximum number of interventions depends on the number of variables in the domain and the conditional probabilities. In the Cancer network, the maximum number is set to 6 when the size of the interventional data is 200 in each active learning step.

[25] We distinguish between the terms "intervention" and "experiment" here. "Intervention" means to manipulate some variables and observe the effects on other variables. "Experiment" means to run the method for testing.

are non-symmetrical in nature and the interventional data can provide more causal information about the probabilities between the manipulated variable and non-manipulated variables. If there is a real edge from the manipulated variable to one non-manipulated variable, the probability of this edge should increase with the interventional data, and the non-symmetrical entropy of this edge will decrease.

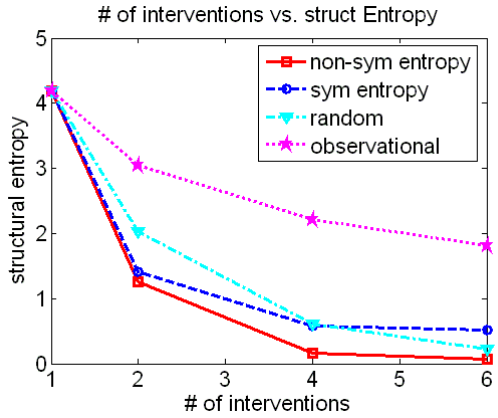


Figure 25 Number of interventions vs. average structure entropy of the learned Bayesian network from Cancer network

Note: The "non-sym entropy" and the "sym entropy" refer to node selection with non-symmetrical node entropy and symmetrical node entropy defined in formulas (3) and (4). "Random" refers to random node selection for node-based interventions. "Observational" means only observational data used in the learning. The same terms are used in the later figures of this section.

The highest structure entropy is derived from observational data when the same number of data instances is collected at each step. The entropy of the Bayesian network structure learned with the random node selection and node selection with the symmetrical node entropy fall between those of the node selection with non-symmetrical node entropy and the observational data.

The significance of the structure entropy differences from different node selection measures was evaluated by one-sided t-test. The p-values between the entropy of the learned Bayesian network structure from non-symmetrical node entropy and other methods are all smaller than 10^{-10}. This means that the structure entropy from node

selection with non-symmetrical entropy is significantly smaller than others.

From Figure 25, we have a surprising observation. When the number of interventions is smaller than 6 in the Cancer network, the entropy of the learned structure with nodes selected from the symmetrical node entropy is lower than that from random node selection. When the number of interventions is equal to or greater than 6, the entropy of the learned structure by node selection with symmetrical node entropy is higher than that from random node selection. It means that, in the first several interventions, symmetrical node entropy selects the nodes to reduce the structure uncertainty significantly when compared with random node selection. However, when the number of interventions is greater than or equal to 6, the leaf nodes (nodes X_4 and X_5 in Figure 17) are always selected by symmetrical node entropy. The data with leaf nodes as manipulated nodes can reduce the probabilities of the edges from the leaf nodes to other nodes. But, the data cannot provide information about the causal influence relationships from other nodes to the leaf nodes. The uncertainty of the leaf nodes calculated from symmetrical node entropy can still be quite large. However, the random method may select other nodes for intervention, which could generate subsequent interventional data with more causal information about the edges from other nodes to the leaf nodes. Such information will reduce the total structure entropy.

Figure 25 also shows that, with more interventions (which means more data), the entropy of the learned structure decreases with all the node selection criteria. The entropy of the learned Bayesian network structure generally decreases more in the

first few interventions. In the later stages, the entropy of the learned structure seems to converge to certain values. These results are general across all the Bayesian networks tested.

5.3.5.2 Number of Interventions vs. Distance of the Learned Structure to the Ground-truth Bayesian Network

In this experiment, we compared the learned structures with the ground-truth Bayesian networks. The difference between the learned structure and the ground truth is measured with Hamming distance. Figure 26 shows that node selection with non-symmetrical node entropy leads to the smallest average Hamming distance to the ground truth, as compared with other methods for node selection. With 6 or more interventions when nodes are selected by non-symmetrical node entropy, the average distance is 0 and the variance is near 0 with Cancer network. The variances of the Hamming distances from the symmetrical node entropy and observational data only are quite high (about 0.55 and 0.33, respectively). In addition, Figure 26 shows that the average Hamming distance decreases with the number of interventions. With more interventional data, the average distance from the learned structure to the ground truth will be smaller.

From Figure 25 and Figure 26, we can observe that, when the number of the interventions increases, the structure entropy converges to a certain low value with either node selection with non-symmetrical node entropy or random node selection. The reason is that, the true causal Bayesian network structure can be identified with

sufficient interventional data from any node selection method. We note that, however, when the number of interventions is small, non-symmetrical node entropy could outperform all other methods for node selection in active learning. The difference in performance could be significant in applications where only a few interventions are feasible. For example, in practice there are resource constraints (time, cost, and man power) in biological experiments, and we may only conduct a small number of interventional experiments to collect data for causal relationship verification.

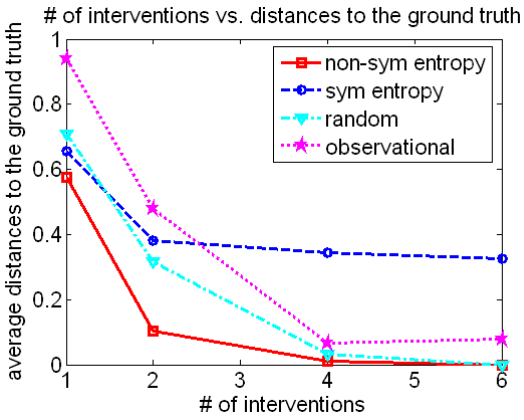


Figure 26 Number of interventions vs. average Hamming distance from the learned Bayesian network structure to the ground truth Cancer network

5.3.5.3 Structure Entropy vs. Distance of the Learned Structure to the Ground-truth Bayesian network

In practice, we do not know the structure of the underlying Bayesian networks in the domain, and cannot use the Hamming distance from the learned structure to the ground truth structure as the stopping criteria in causal Bayesian network learning. A different strategy is needed to stop the learning process. This experiment will examine the relationship between the structure entropy and the Hamming distance from the learned structure to the ground truth Bayesian network. Figure 27 shows how the

entropy of the learned structure approximates the average Hamming distance from the learned structure to the ground truth. When the entropy of the learned structure is small, the average Hamming distance is also small, which means that the entropy of the learned structure is a good approximation of the distance of the learned structure to the ground truth Bayesian network and can be used as a stopping criterion for the structure learning.

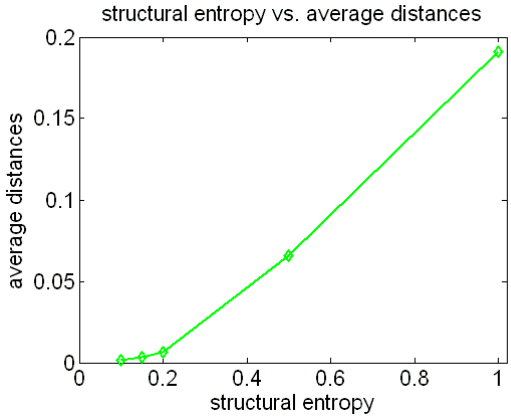


Figure 27 Relationship between average structure entropy of the learned Bayesian network and the average Hamming distance to the ground truth Cancer network

5.3.5.4 Structure Entropy as Stopping Criterion

In the subsequent experiment, we tested the effect of the structure entropy as the stopping criterion. Figure 28 shows that, with non-symmetrical node entropy as the node selection criterion, the program can reach the required structure entropy with fewer interventions. When the manipulated node is selected with symmetrical node entropy, a large number of interventions are needed. The results from observational data do not show in Figure 28, as the program with observational data cannot reach the required structure entropy in the maximal steps allowed in this set of experiments.

Appendix A.C shows more results when structure entropy is used as the stopping

criterion. One observation from these results is that, when the size of the data collected from one intervention is bigger, the performance of node selection with non-symmetrical entropy is better. This is consistent with our expectation: a data set with more instances from one intervention can reduce the uncertainty from the selected node to other nodes the most, and fewer interventions are needed to achieve the required structure entropy. If the data set collected from each intervention is small, the change of the structure uncertainty with the new data will be limited, and the performance of node selection with non-symmetrical entropy could be similar to the performance of random node selection.

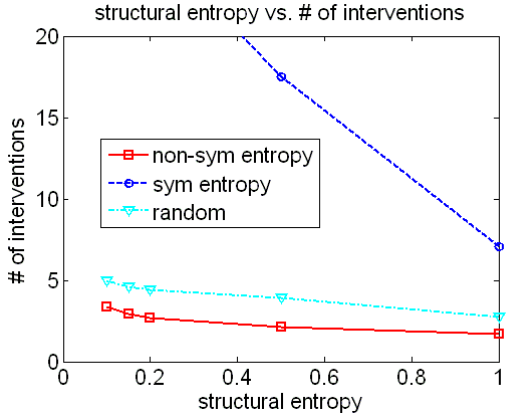


Figure 28 Structure entropy vs. number of interventions required from Cancer network

5.3.5.5 Comparison with the Expected Posterior Loss Method

For comparison purpose, we have implemented the method based on the expected posterior loss [121,161]. The expected posterior loss considers possible interventions and possible observations after interventions and should give better structure entropy with the same number of interventions theoretically. But it will take a very long time to estimate the probabilities of possible observations and the edge probabilities under different observations, as we mentioned in the beginning of this section. In our

implementation, the expected posterior loss is similar to that in Tong and Koller [161]: we sample the orderings of variables from the current data and estimate the probabilities of possible observations. The edge probabilities are estimated with both the exact method by Koivisto [97] and MCMC method. Experiments show that the MCMC methods take more time to converge to the probabilities estimated with the exact method. So, only the edge probabilities from the exact method will be discussed here.

We have tested our method with Study network and Cold network[26]. In the experiment, the number of instances collected from each intervention is set to 1 when the selected node is manipulated to a distinct value. Due to the computational complexity, the multiple data instances from each intervention are not tested.

Figure 29 shows the results from Study network. Figure 29 (a) shows that all the methods with interventional data can reach the required structure entropy with smaller than 50 interventional instances, while the observational data alone cannot reach the requires structure entropy with the maximal instances allowed. Figure 29 (b) shows the detailed results from the node selection methods with interventional data. In this example, node selection with the expected posterior loss requires the minimal number of instances to reach the structure entropy on average. The method next to the expected posterior loss is node selection with non-symmetrical entropy. Node selection with symmetrical entropy and random node selection requires a larger number of instances to reach the required structure entropy. Figure 29 (c) shows the

[26] We have tried with Cancer network, but the program cannot finish one experiment in 12 hours.

average running time the different methods spent. We can see that the expected posterior loss requires much more time than other methods for node selection. The time for observational data converges when the maximal number of instances is reached.

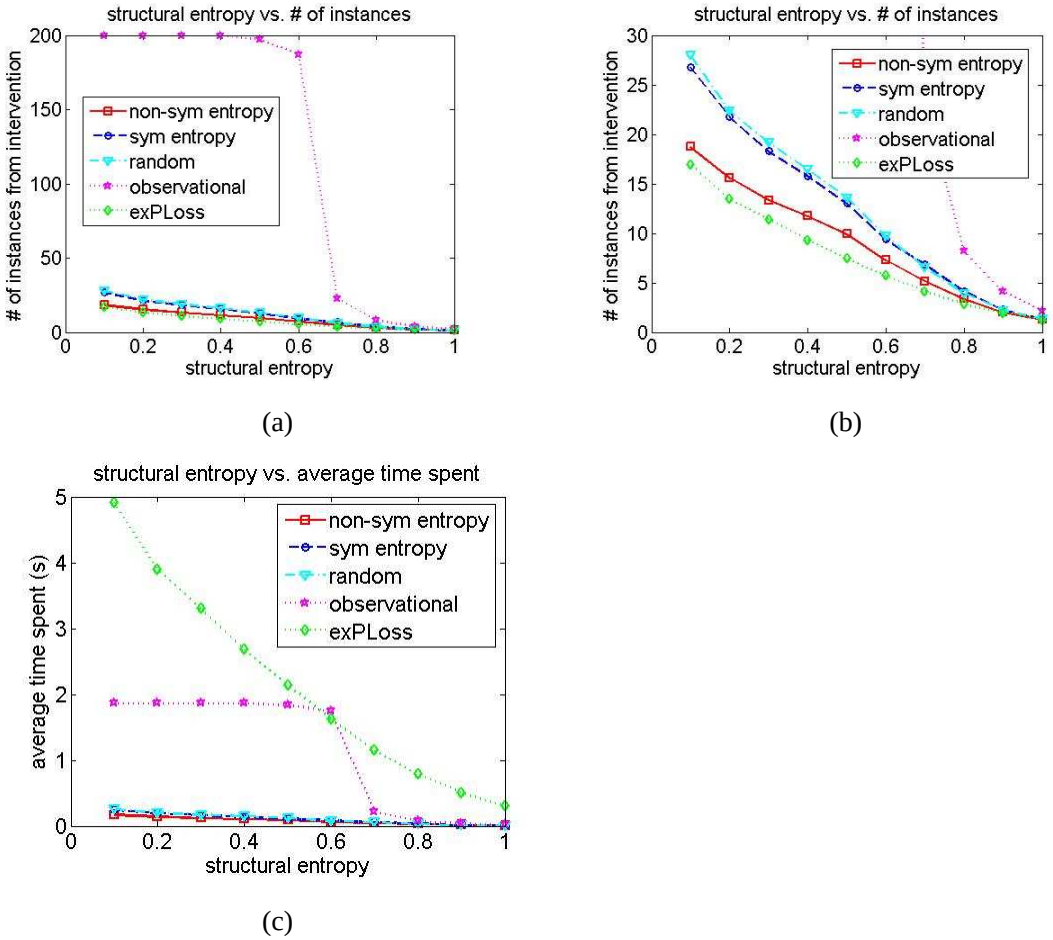


Figure 29 Comparison of different node selection methods for intervention on Study network

5.3.5.6 Positive Findings in Subsequent Interventions

In the final experiment for node-based interventions, we considered the situation with resource constraints. In the previous experiments, the objective is to identify the complete causal structure with multiple interventions and we have enough resources to reach this objective. In practice, there are usually resource constraints for

interventions, and sometimes we can only conduct one interventional experiment. In this case, we may hope to get a positive finding in one interventional experiment, where a positive finding means that there is really a causal relationship between the manipulated variable and one of the non-manipulated variables.

The problem in this experiment is defined as follows: given the available data, domain knowledge and resource constraints, what is the likelihood to get a positive finding in a single interventional experiment? In this case, we assume we can only conduct one interventional experiment. This problem has not been considered in any previous active learning work with Bayesian networks [121,161], since one instance collected from an intervention cannot change the probability of the hypotheses much. A positive finding is only possible when a data set is collected from one intervention.

There is no guarantee to obtain a positive finding in a single intervention, but some strategies are available to increase the chance for a positive finding. In the experiment, we generated the observational data and interventional data randomly first. Then, we sampled the possible edges in the Bayesian network as topological constraints with probabilities 0.1, 0.2, 0.3 and 0.4, respectively. We estimated the edge probabilities with the available data, and chose the parent node of the edge with the highest probability as the node to be manipulated. We repeated the experiments 1000 times in the different scenarios.

The results show that in 98.5% cases and above, the edges with the highest probability from the available data and the known edges (as domain knowledge) are the true edges. It empirically shows that the edges with the highest probability are the

best choice for a positive finding if we have resource constraints and only can conduct one interventional experiment.

5.3.6 Discussion

In this section, we propose an active learning algorithm for causal Bayesian network structure learning when multiple data instances are collected from one intervention. The current node entropy is used to select nodes for intervention, not the expected posterior loss in Tong and Koller [161] and Murphy [121]. Therefore, there is no need to consider the exponential number of possible observations after each intervention, and the algorithm can be sped up.

Non-symmetrical entropy is proposed for node selection, since the intervention is non-symmetrical in nature. The experiment results show that non-symmetrical entropy is much better than symmetrical entropy in all the cases, and better than random node selection when more instances are sampled in one intervention. The performance of node selection with non-symmetrical entropy is comparable to the random node selection sometimes when one instance is sampled from an intervention, which is consistent with Murphy's observation in Car network [121]. The possible reason is that when one instance is collected from one intervention, limited information is provided by this instance.

Tong and Koller [161] considered domain knowledge and only root nodes in the domain were manipulated in active learning, while other nodes could be selected in random node selection. From manipulation criterion and our experiment results, we

know that the manipulation on the leaf nodes cannot provide sufficient causal influence information from the leaf node to other nodes, while the manipulation on the root nodes can be used to establish the causal ordering between the variables. Although root nodes can be known from domain knowledge in some domains, the reported results with only root nodes manipulated in Tong and Koller [161] are biased.

L1 edge error is used in Tong and Koller [161] as the goodness criterion of the learned structure, which requires the knowledge of the true Bayesian network structure. This is suitable for a simulation, but not for a real application. This is why we choose the structure entropy as the criterion to evaluate the quality of the learned structure. Experiments show that the structure entropy is a good approximate to the Hamming distance from the learned structure to the ground truth, and can be used as the stopping criterion.

5.4 Hypothesis Verification with Edge-based Interventions

In some situations, we need the concrete knowledge of causal relationships between variables, such as the situations for system re-engineering. However, the concrete causal knowledge cannot be achieved with observational data and node-based interventional data sometimes, especially when several variables interact together to affect one variable. In this case, we need edge-based interventions to verify the relationships between the variables. In this section, we try to identify the complete

structure of causal Bayesian networks in a domain with the minimal number of edge-based interventions.

5.4.1 Active Learning with Edge-based Interventions

Active learning with edge-based interventions starts with a data set (possibly with observational data and node-based interventional data) and the capability to conduct edge-based interventions. The data (with topological constraints, if applicable) is used to estimate edge probabilities, edge entropy and structure entropy. One edge is chosen with certain criterion for an edge-based intervention, and the edge-based intervention determines whether there is a causal influence relationship from the parent node to the child node in the selected edge. The result of the edge-based intervention is combined with the available topological constraints for another round of edge probability estimation. The learning process will repeat until the stopping criterion is satisfied. The process is summarized in Table 17 and the flowchart is shown in Figure 30.

In the second step of the process, we apply Koivisto's exact method [97] to estimate the edge probabilities with the available data and topological constraints. The edges are predicted as the learned edges[27] when their probabilities are greater than 0.5. In the following steps of the learning process, there are two challenges similar to those for node-based interventions. One challenge is how to select a pair of variables for an edge-based intervention. Another challenge is when to stop the learning process. In the following two sub-sections, we will discuss these two challenges in detail.

[27] Predicting edges from edge probabilities is different from the complete Bayesian network structure learning.

Input of the algorithm:

1) *A data set (possibly with observational data and node-based interventional data); and*
2) *The set of topological constraints (can be empty in the beginning), which can be from domain knowledge or from edge-based interventions.*

Output of the algorithm:

1) *Intermediate results: the chosen edges for the subsequent edge-based interventions;*
2) *The results of the edge-based interventions for selected hypotheses; and*
3) *The final result: the structure of the causal Bayesian network.*

1. *Set the initial topological constraint set* **C** *(can be empty)*
2. *Learn edge probabilities with the data and the topological constraint set* **C**
3. *Check whether to stop the learning process*
4. *If not, select an edge for an edge-based intervention. Assume the result of the edge-based intervention as E. Set* $\mathrm{C} \cup E \rightarrow \mathrm{C}$ *, and return to step 2*
5. *If yes, stop.*

Table 17 Active learning of Bayesian networks with edge-based intervention

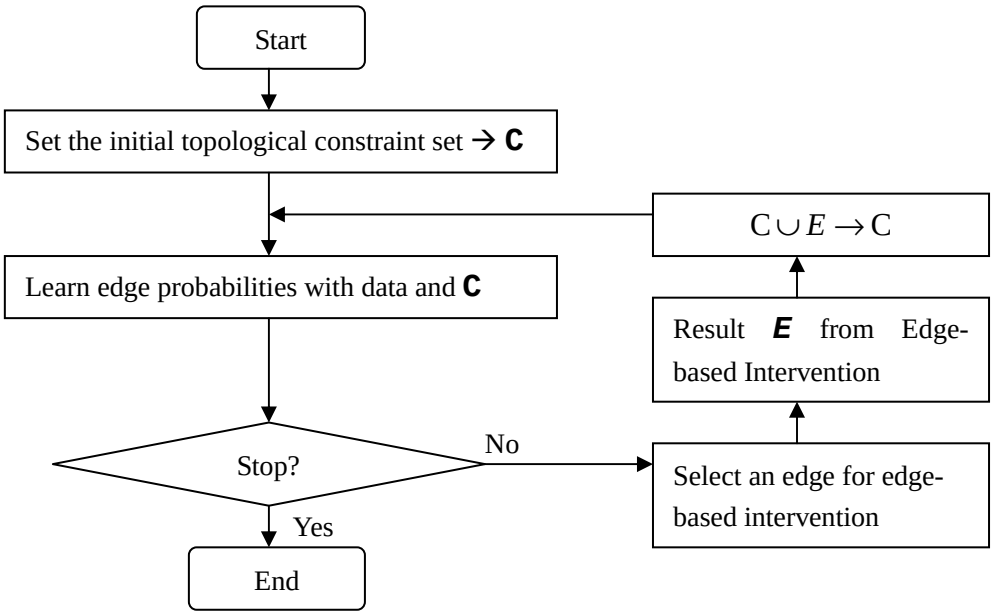


Figure 30 Flowchart of active learning with edge-based intervention

5.4.2 Edge Selection for Edge-based Interventions

The objective of edge selection is to choose the edge that is most informative for

causal knowledge discovery with edge-based interventions. After we determine the relationship between the chosen pair of variables, the uncertainty of the Bayesian network structure is expected to reduce the most.

Edge selection can be determined by the expected posterior loss from decision theoretical approach. In a domain with n variables, there are $n*(n-1)$ edges and the expected posterior loss from every edge needs to be estimated for edge selection. Due to the computational complexity, the expected posterior loss for edge selection will not be considered.

We will use the edge entropy as the criterion for edge selection. In Section 5.3.1, we have discussed two ways to measure the edge uncertainties: symmetrical edge entropy in Formula (1) and non-symmetrical edge entropy in Formula (2). The reason to use edge entropy from the available data and topological constraints for edge selection is as follows. When we perform edge-based interventions on two variables, we will manipulate one variable and observe the effect on another variable. Suppose we manipulate variable A in a pair of variables A and B in the edge-based intervention.

1) When the edge-based intervention tells us that there is really a directed edge from A to B ($A \rightarrow B$), the probability $p(A \rightarrow B)$ will be 1 and the probabilities of other two conditions between A and B will be 0. In this case, the edge entropy between A and B will be 0, and the total entropy of the DAG will reduce with the amount of the entropy between A and B estimated before the edge-based intervention.

2) When the edge-based intervention tells us that there is no directed edge from A to B, the probability $p(A \to B)$ will be 0 and the probabilities for other two conditions between A and B will change. In this case, the edge entropy between A and B may not be 0, and we are not sure about the change of the total entropy.

We have tried the following methods to select edges for edge-based interventions.

1) Random edge selection. This is a straightforward method to select an edge for an edge-based intervention.
2) Edge selection based on chi-square[28] values between any pair of variables from the available data. The uncertain edge with the highest chi-square value will be selected for an edge-based intervention.
3) Edge selection based on mutual information between any pair of variables from the available data. The uncertain edges with the highest mutual information values will be selected for an edge-based intervention.
4) Random selection of edges from the learned Bayesian network with the data and topological constraints. If all the edges in the learned Bayesian network have been determined by edge-based interventions, we will randomly select one uncertain edge.
5) Edge selection based on symmetrical entropy as in Formula (1). The edge with the maximal edge entropy will be selected for an edge-based intervention.
6) Edge selection based on maximal non-symmetrical entropy as in Formula (2).
7) Edge selection based on the edge with the highest probability from the data

[28] Refer to Appendix A for a brief description of chi-square and mutual information.

and topological constraints.

Our experiments show that edge selection with non-symmetrical entropy requires the minimal number of interventions to identify the true structures of the Bayesian networks (refer to Section 5.4.4 for details).

5.4.3 Criteria to Stop the Learning Process

A stopping criterion is used to evaluate the learned Bayesian networks, and decide whether to stop the learning process. In the simulation test, we can compare the learned Bayesian network with the ground-truth Bayesian network and stop the learning process if the learned Bayesian network is the same as the ground-truth. In practice, however, we do not know the true structure of the Bayesian networks, and cannot compare the learned Bayesian networks with the underlying Bayesian networks (If the underlying Bayesian networks are known, there is no need to learn the Bayesian network structure). In this case, we need some strategies to stop the learning process.

The possible strategies to stop the learning process are: 1) when the maximal absolute edge entropy is small enough; 2) when the maximal relative edge entropy is small enough; and 3) when there is no change in the learned structure for several iterations.

5.4.4 Experiments for Edge-based Interventions

We conducted experiments for hypothesis verification with edge-based interventions

on four Bayesian networks: two Bayesian networks created by ourselves (Study network and Cold network), and two benchmark Bayesian networks[29] (Cancer network and Asia network). These Bayesian networks include the canonical structures and the results can be generalized to other Bayesian networks.

In the simulation experiments, the results of the edge-based interventions are obtained from the ground-truth Bayesian networks. These results will be used as topological constraints in the next round of the learning process.

The performance of the learning process is measured by the following criteria:

1) The number of interventions required;
2) The number of correct edges identified in the final learned Bayesian network;
3) The Hamming distance between the final learned structure and the original Bayesian network;
4) Product of (#Interventions+1)*(HammingDistance+1), where #Interventions means the number of interventions required in the learning process. We proposed this measure to combine the number of interventions required and the Hamming distance from the learned structure to the original Bayesian network structure. The addition of one to each variable is to avoid the situation when one variable is 0 and the product is 0. The smaller this measure, the better the learning strategy.

[29] Car network is not used since the experiment cannot finish in a reasonable time.

5.4.4.1 Best Strategy for Edge Selection

In the first experiment, we tested the edge selection strategies until the learned process identifies the ground truth structure. The experiment setup is: Given a known Bayesian network, we sample data instances from the given Bayesian network as the observational data, and apply the active learning algorithm described in Section 5.4.1. When a Bayesian network is learned from the available observational data and the topological constraints, we will compare it with the given Bayesian network. If two Bayesian networks are the same, the learning process stops and the number of edge-based interventions conducted will be recorded; otherwise, the learning process will continue.

We ran the program on Study network for two minutes with 261 experiments, on Cold network for two minutes with 152 experiments, on Cancer network for six minutes with 155 experiments, and on Asia network for two hours with 76 experiments. The medians and averages of the required edge-based interventions from different methods are shown in Table 18 and Table 19.

Table 18 and Table 19 show that, for Cancer network and Asia network, the required interventions by edge selection with symmetrical edge entropy, non-symmetrical edge entropy, and edges with the highest probability is much smaller than those by random edge selection, chi-square value, mutual information and random selection from the learned edges. For Study network and Cold network, the required number of edge-based interventions is similar in different edge selection strategies, since these two networks are very small, and only one or two edge-based

interventions are needed.

Bayesian network	Study network	Cold network	Cancer network	Asia network	Average
# data sets generated	261	152	155	76	
Random selected edges	1	2	6	39.5	12.125
Chi-square	1	2	4	37	11
Mutual information	1	2	4	41	12
Randomly learned edge	1	2	4	36.5	10.875
Max symmetrical entropy	1	1	6	25.5	8.375
Max non-symmetrical entropy	1	2	4	28	8.75
Edge with the highest prob	1	2	4	27	8.5

Table 18 The median of the interventions required to identify the true structure

Bayesian network	Study network	Cold network	Cancer network	Asia network	Average
# data sets generated	261	152	155	76	
Random selected edges	0.99	1.73	6.7	39	12.105
Chi-square	0.99	2.07	6.64	35.2	11.225
Mutual information	0.99	2.07	6.86	39.95	12.4675
Randomly learned edge	0.99	2.28	5.68	34.46	10.8525
Max symmetrical entropy	0.99	1.49	6.34	23.78	8.15
Max non-symmetrical entropy	0.99	1.62	5.23	26.05	8.4725
Edge with the highest prob	0.99	1.72	5.23	27.22	8.79

Table 19 The average of the interventions required to identify the true structure

Among edge selection strategies, the random edge selection does not use any available information from data and domain knowledge. The chi-square and mutual information only measure the pair-wised dependency between variables from the data. They do not consider other variables in the domain, and cannot take advantage of the information from other variables and the available topological domain knowledge for edge selection, such as the acyclicity constraints in Bayesian networks. This is why these methods cannot compete with the entropy-based methods for edge selection.

After this preliminary experiment, we will keep the following three edge selection methods for further testing: 1) symmetrical edge entropy; 2) non-symmetrical edge

entropy; and 3) edge with the highest probability, and will not consider other edge selection strategies anymore.

5.4.4.2 Best Strategy for Edge Selection and Stopping Criterion

In this experiment, we tested the full learning process with edge selection and stopping criterion. The results from different networks are summarized in Table 20, Table 21 and Table 22.

Bayesian network		Study network	Cold network	Cancer network	Asia network	Average
# data sets generated		470	241	345	161	
Edge selection	Stopping criterion					
Symmetrical entropy	Absolute entropy	1	2.84	10.81	34.64	12.32
	Relative entropy	1	2	6.41	9.64	4.76
	No structure change	1.51	3.6	5.03	4.09	3.56
Non-symmetrical entropy	Absolute entropy	1	2.88	10.68	35.55	12.53
	Relative entropy	1	2.05	4.89	10.36	4.58
	No structure change	1.48	3.59	5.24	4.73	3.76
Edge with the highest prob	Absolute entropy	1	2.63	9.43	33	11.52
	Relative entropy	1	2.33	4.38	8.27	4.00
	No structure change	1.47	3.91	4.46	3.82	3.42

Table 20 Average interventions required in active learning of Bayesian network structure

On average, three edge selection methods require the similar number of edge-based interventions to reach the required stopping criterion. The stopping criterion based on the absolute entropy requires the maximal number of edge-based interventions to achieve the stopping criterion, while the learned structure has the minimal Hamming distance. The stopping criterion based on no structure change requires the minimal number of edge-based interventions to stop the learning process, and the average of (#interventions+1)*(Hamming distance+1) is the smallest.

Therefore, different stopping criteria have different effects. If there are resource constraints, the stopping criterion with no structure change is preferable. If the correct structure of the domain is more important, the stopping criterion with absolute structure entropy is desirable.

Bayesian network		Study network	Cold network	Cancer network	Asia network	Average
# data sets generated		470	241	345	161	
Edge selection	Stopping criterion					
Symmetrical entropy	Absolute entropy	1	0.48	0.08	0.27	0.46
	Relative entropy	1	1.22	1.62	2.55	1.60
	No structure change	0.49	0.12	1.76	3.73	1.53
Non-symmetrical entropy	Absolute entropy	1	0.6	0.08	0.18	0.47
	Relative entropy	1	1.12	1.49	3	1.65
	No structure change	0.52	0.15	0.78	3.27	1.18
Edge with the highest prob	Absolute entropy	1	0.72	0.11	0.27	0.53
	Relative entropy	1	0.86	1.03	2.91	1.45
	No structure change	0.51	0.07	0.86	4.45	1.47

Table 21 Average Hamming distance from the learned Bayesian networks to the ground-truth Bayesian networks

Bayesian network		Study network	Cold network	Cancer network	Asia network	Average
# data sets generated		470	241	345	161	
Edge selection	Stopping criterion					
Symmetrical entropy	Absolute entropy	4	5.62	12.81	45.55	17.00
	Relative entropy	4	6.74	19.46	38.91	17.28
	No structure change	3.49	5.02	13.05	22.36	10.98
Non-symmetrical entropy	Absolute entropy	4	6.05	12.68	43.36	16.52
	Relative entropy	4	6.42	14.62	49	18.51
	No structure change	3.51	5.13	9.95	23.55	10.54
Edge with the highest prob	Absolute entropy	4	6.02	11.78	43.27	16.27
	Relative entropy	4	6.13	10.84	35.55	14.13
	No structure change	3.49	5.15	8.97	23	10.15

Table 22 Average of (#interventions+1)*(Hamming distance + 1) required in active learning of Bayesian network structure

5.5 Conclusion and Discussion

Causal Bayesian network learning is a big challenge for knowledge discovery. The problem we addressed in this chapter is on how to determine the causal structure with observational data and interventional data. In this thesis, we assume that we can manipulate the variables and can collect the interventional data in the application domain. Our objective is to minimize the number of interventions while identifying the correct structure of the Bayesian network.

We have proposed a type of active learning for causal Bayesian networks: combining the observational data with the node-based interventional data and the results from the edge-based interventions. The method can utilize the available data and domain knowledge to guide the interventional experiments for efficient causal knowledge discovery.

Two different intervention types have been discussed in this chapter: the node-based intervention and the edge-based intervention. The node-based interventions would help establish the causal ordering of variables. The advantage of the node-based interventions is that it may only require linear number of interventions when one variable is manipulated each time. This is more applicable in practice. The disadvantage of the node-based interventions is that some direct causal relationship may not be tested. Therefore, if some direct causal relationships are really important, we first proposed the *edge-based intervention* to examine the direct relationships between variables.

The edge-based interventions would help establish the parent sets of variables,

and distinguish the different structures in the Markov equivalent class, which cannot be done with the observational data and node-based interventional data sometimes. However, an exponential number of instances may be needed in edge-based interventions. So, the choice of the methods is dependent on the objective in the applications and the resources available.

There are two main problems in the active learning process: how to select the hypotheses for intervention, and when to stop the learning process. Non-symmetrical entropy is first proposed to select nodes for interventional experiments. Compared with other methods, non-symmetrical entropy requires the minimal number of interventional experiments to achieve the required structure entropy.

In node-based interventions, entropy-based stopping criterion is better than the stopping criterion based on the number of interventions, since stopping the learning process with the number of intervention cannot guarantee the learned structure quality. In edge-based interventions, we can see the compromise between the accuracy of the learned Bayesian network structure and the number of interventions required. If the accuracy of the learned Bayesian network structure is more important, entropy-based stopping criterion is a better choice.

Chapter 6 An Example in a Biological Domain

In this thesis, we have proposed a framework for knowledge discovery with Bayesian networks which includes three steps: hypothesis generation with Bayesian network structure learning, hypothesis refinement with topological constraints, and hypothesis verification with interventional experiments. We have examined the technical challenges and practical issues in the framework in last three chapters.

In this chapter, we will show how to apply the framework of knowledge discovery with Bayesian networks in a biological domain – the intracellular signaling network of human primary naïve $CD4^+$ T cells, downstream of CD3, CD28, and LFA-1 activation. Figure 31 shows the network structure of signaling molecule interactions (from [145]). Eleven variables in the structure represent eleven proteins measured. The twenty edges represent the causal influence relationships between the proteins from the consensus of the current domain understanding. Among the twenty edges, eighteen of them have been verified with biological experiments in the literature. Another two edges (*PKC*→ *PKA* and *Erk*→*Akt* in dashed lines) were recently hypothesized with Bayesian network techniques and confirmed with biological experiments by Sachs *et al*. [145].

In the following sections, we will show the application of the proposed framework with this example network. Section 6.1 will show that Bayesian networks

can generate reasonable hypotheses of influence relationships between variables from the sampled data, although some edges are not exactly the same as the original network. Section 6.2 will show that topological domain knowledge can refine the generated hypotheses of influence relationships between variables and improve the meaningfulness of the hypotheses. Section 6.3 will show how to conduct node selection for hypothesis verification with interventional experiments.

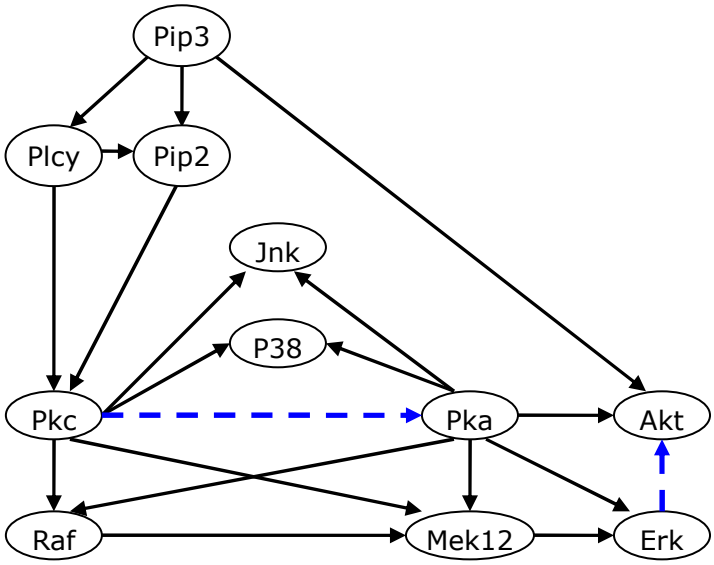


Figure 31 The consensus intracellular signaling networks of human primary naïve CD4+ T cells, downstream of CD3, CD28, and LFA-1 activation

6.1 Hypothesis Generation: Learning the Structure with Observational Data

In the first experiment, we will show how the hypotheses of influence relationships between variables are generated from observational data. The observational data is sampled from the Bayesian network in Figure 31. The parameters are randomly generated from Dirichlet priors. Two thousand of instances were sampled from the network by direct sampling. The edge probabilities were estimated with Koivisto's

exact method [97]. If the edge probabilities are greater than 0.5, the edges are regarded as predicted edges. The learned Bayesian network is shown in Figure 32.

Compared with Figure 31, eleven edges are the same as those in the original structure. Seven edges have reverse directions (the dotted lines in Figure 32), and two edges are missing (*Plcy* → *Pip2* and *Pka* → *Jnk*). This means that most of the undirected edges between variables are learned correctly, and Bayesian networks can generate the reasonable hypotheses of influence relationships between variables from the data for the domain of interest. The edges in this learned Bayesian network can now serve as the initial hypotheses for further refinement and verification.

Note that we only consider the hypotheses of direct influence relationships between variables in this example. While this example does not show the potential application of variable grouping, variable grouping could be useful in a situation where the aggregate functions of some proteins are to be studied.

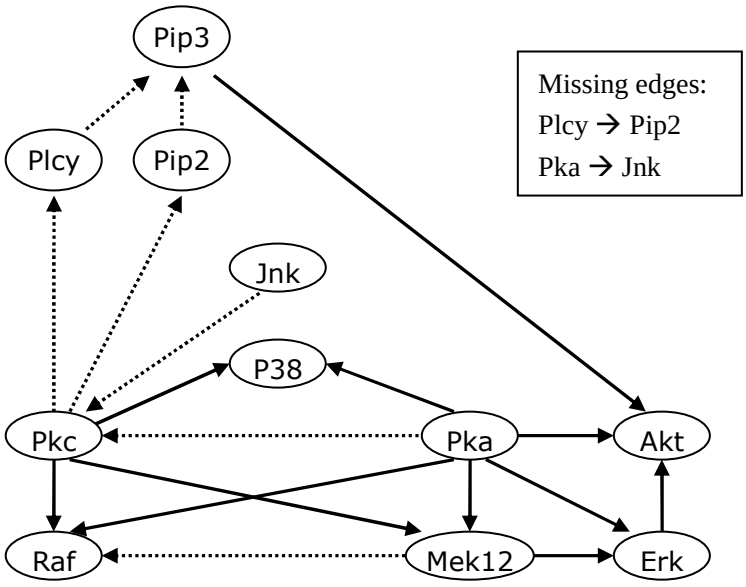


Figure 32 The learned BN with data sampled from the intracellular signaling network

6.2 Hypothesis Refinement: Learning the Structure with Observational Data and Topological Constraints

The second experiment will show the effect of topological constraints on the hypothesis refinement. We will re-estimate edge probabilities with the available observational data and topological constraints. In this experiment, we use the following topological constraints: *Pip3* is a root in the domain, and *Jnk* and *P38* are leaf nodes in the domain. Such topological constraints are from biological domain knowledge: *Pip3* is an upstream protein and is not affected by other proteins in the domain, and can be treated as a root in a Bayesian network; *Jnk* and *P38* are two downstream proteins and will not affect other proteins in the domain, and can be treated as leaf nodes in a Bayesian network. After combining the topological constraints with observational data, the generated graph is shown in Figure 33.

Compared with the original structure in Figure 31, eighteen edges are learned correctly with the data and topological constraints. Only one edge (*Plcy* → *Pip2*) is missing and one edge (*Raf* → *Mek12*, dotted line in Figure 33) is reversed. We can see that topological constraints can improve the hypotheses of influence relationships between variables generated from data in this example.

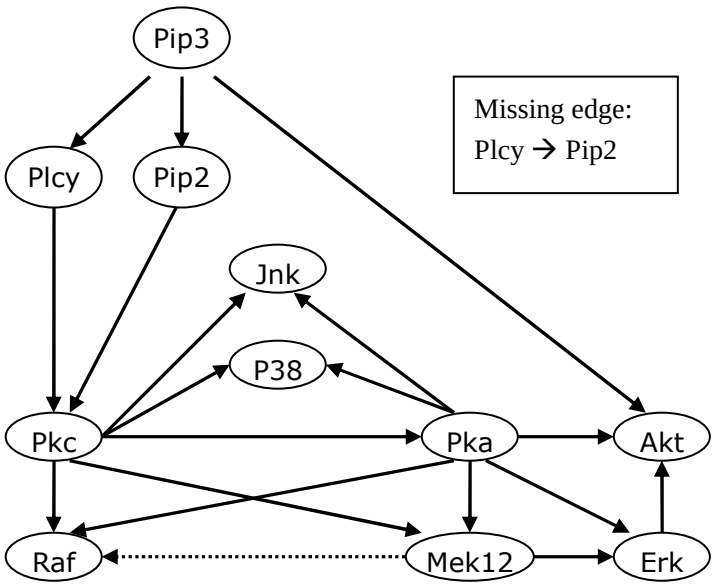


Figure 33 The learned BN with data and topological constraints from the intracellular signaling network

6.3 Hypothesis Verification: Node Selection for Interventional Experiments

With the available observational data and topological constraints, we can estimate the node uncertainty with our first proposed non-symmetrical entropy in Formula (3) and (4) of Section 5.3.2. The estimated node uncertainties without/with topological constraints are shown in Table 23 and Table 24. If only the observational data is used for edge probability estimation, variable *Pkc* has the highest node entropy with value 1.2, and will be selected for the subsequent node-based intervention. If both the observational data and the topological constraints are used for edge probability estimation, variable *Mek12* has the highest node entropy with value 1.25, and will be selected for the subsequent node-based intervention.

With the generated node entropies, we can choose the protein with the highest entropy for intervention and conduct the real biological experiments to collect

interventional data to verify the causal relationships. Since the biological experiments are out of the scope of our research, we will not continue the experiment. We hope that there would be biologists who are interested in this domain and would use the hypotheses from computational methods for further experiments in future.

Protein	Node uncertainty
Pkc	1.1996
Pka	1.1559
Mek12	0.94244
Raf	0.80507
Plcy	0.79517
Pip2	0.79517
Jnk	0.48444
p38	0.10604
Erk	0.033956
Pip3	0.020267
Akt	1.04E-29

Table 23 Node uncertainty from observational data for the intracellular signaling network

Protein	Node uncertainty
Mek12	1.2515
Pka	1.157
Raf	1.0853
Plcy	0.69315
Pip2	0.69315
Pkc	7.96E-07
Erk	4.13E-07
Pip3	3.00E-10
Akt	8.96E-29
p38	0
Jnk	0

Table 24 Node uncertainty from observational data and topological constraints for the intracellular signaling network

6.4 Summary

In this chapter, we applied our proposed framework to a biological problem using simulation. The results show that Bayesian networks could generate reasonable hypotheses of influence relationships between variables from the data for the domain of interest. The combination of domain knowledge could improve the quality of the hypotheses of influence relationships between variables generated from data. Both the missing edges and reversed edges are reduced with the specified root and leaf nodes as topological constraints. Node entropy can be derived from the available data and domain knowledge, and the node with the highest node entropy can be selected for hypothesis verification with the node-based interventional experiments; the list of results has correctly suggested the most promising nodes for further experiments.

Chapter 7 Conclusion

Causal knowledge is helpful for comprehension, diagnosis, prediction and control in many complex situations. In this thesis, I consider causal knowledge from the probabilistic perspective with a manipulation criterion. A mixture of observational data and interventional data is used for causal knowledge discovery with causal Bayesian networks.

7.1 Summary of Contributions

Identification of causal knowledge is an important research topic with a long history ([130], Epilogue) and many challenging issues. Neyman [125] and Fisher [57] pioneered causal knowledge discovery with randomized experiments. Rubin [143,144] initiated the study of causal knowledge discovery with observational data and statistical methods, while Spirtes *et al*. [155,156] and Pearl [130] led the way of inductive learning of causal knowledge from observational data with Bayesian networks.

One of the main differences between the traditional statistical methods and Bayesian network methods for causal knowledge discovery is how the hypotheses of causal influence relationships between variables are modeled and generated. In the traditional statistical methods, the hypotheses of causal influence relationships between variables are generated from domain experts without sufficient mathematical

support. In this case, the results of knowledge discovery heavily depend on domain experts to generate the hypotheses of causal influence relationships between variables. However, in the inductive learning of causal knowledge with Bayesian networks, the hypothesis space is assumed before learning, and the best hypothesis is automatically searched through the hypothesis space based on the data.

In this thesis, I use causal Bayesian networks [130,131,155,156] as the basic tool for causal knowledge discovery. I assume that observational data can be collected economically and interventional data will be collected with higher cost, such as in Biological Science [145]. Interventional data is surely useful for causal knowledge discovery, while it is controversial to discover causal knowledge from observational data, since observational data mainly gives correlation information between variables. In this thesis, a combination of observational data and interventional data is used for causal knowledge discovery.

The hypotheses of direct causal influence relationships between variables will be generated as edges in causal Bayesian networks from the available data. The hypotheses will be updated with topological domain knowledge. Interventional experiments can verify the generated hypotheses. Our objective is to reduce the number of interventions required to identify the underlying causal structure of the domains of interest, while keeping the computational complexity affordable.

In this thesis, I proposed an iterative and interactive framework for causal knowledge discovery with observational data and interventional data to close the knowledge discovery loop. My main contributions include: 1) proposal of an iterative

and interactive framework for causal knowledge discovery with three components: *hypothesis generation*, *hypothesis refinement* and *hypothesis verification*; and within the framework: 2) proposal of a new hypothesis generation method with variable grouping; 3) proposal of a new hypothesis refinement method with topological constraints; and 4) proposal of a new hypothesis verification with node-based and edge-based interventional experiments and non-symmetrical entropy for hypothesis selection. I have also illustrated how to integrate the different tasks in a systematic way to support cost-effective causal knowledge discovery. Promising results are shown in a set of applications with practical implications.

7.1.1 Framework for Knowledge Discovery with Bayesian Networks

The proposed framework for knowledge discovery with Bayesian networks is an iterative and interactive process with modular components: hypothesis generation, refinement and verification. These three components are generally studied separately and a unified framework is needed. In the proposed framework, the details of the three components can be updated or extended further in future without affecting the structure of the framework.

7.1.2 Hypothesis Generation

The main kind of hypotheses used in this thesis is the direct causal relationships between variables in a domain, which are the edges in Bayesian networks. Another

kind of hypotheses is the complete Bayesian network structure. These two kinds of hypotheses can be generated from Bayesian network structure learning. The edges between variables are mainly used for hypothesis refinement and hypothesis verification.

A new method is proposed to generate hypotheses by Bayesian network structure learning with variable grouping. This method is applicable in the domains where some variables follow similar conditional probability distributions. Experiments show that this method could identify the group variables and dependency between the groups simultaneously. This would be particularly useful in the domains where the group information and the dependency between the group variables are important, *e.g.*, microarray data from gene expressions and stock price from the stock market.

7.1.3 Hypothesis Refinement

Two canonical formats are proposed to represent topological constraints for Bayesian network structure learning. The rule format is easy for domain knowledge elicitation and the matrix format is easy for Bayesian network structure learning and domain knowledge consistency checking. Experiments show that topological constraints could improve the relevance of the hypotheses of causal influence relationships between variables generated from data.

7.1.4 Hypothesis Verification

Node-based and edge-based interventions are proposed for active learning of causal

Bayesian network structures. Non-symmetrical entropy is proposed to select nodes or edges for interventional experiments. Compared with the decision theoretic approach (the expected posterior loss), our method is more computationally affordable while the learned structure entropy is comparable. Compared with other methods, non-symmetrical entropy requires a minimal number of node-based interventions and a comparable number of edge-based interventions for causal knowledge discovery. Entropy-based method is proposed as the stopping criterion in causal Bayesian network structure learning process.

7.1.5 Limitations

There are some limitations in the current work. The definition of causal knowledge in this thesis is based on manipulation criterion. Therefore, it may not be applicable to the conditions with other causal knowledge definitions. And, since the proposed framework needs the interventional experiments for hypothesis verification, it is not applicable to the domains where the interventional experiments are not feasible, *e.g.*, in Social Science. Even if the interventional experiments are possible, the resources needed in the hypothesis verification process can be substantial. The availability of the resources will limit the application of the proposed framework. When the resources are available, it could still be time-consuming to collect the interventional data. However, we believe that the proposed framework would be useful in cases where real experiments or the dire consequences of inaccurate diagnosis, prediction, or other applications of causal knowledge are extremely costly.

7.2 Related Work

This work builds on and extends the existing Bayesian network theory and active learning methods for causal knowledge discovery. It also integrates many ideas from knowledge discovery in database and experiment design in Statistics.

Causal knowledge discovery started from the very beginning of human history. Our ancestors learned causal knowledge from their experiences and manipulations in natural exploration process. Aristotle spoke of the doctrine of four causes, while others proposed different forms of causality afterwards [90,106,130,155,171]. David Hume [90] thought that causality was just from our habit and doubted whether we could identify the certain laws of cause and effect. David Lewis [106] suggested the counterfactual causality. Cheng [29], Pearl [130], Spirtes *et al.* [155], and Woodward [171] considered causality from a probabilistic perspective. Pearl [130], Price [135], Spirtes *et al.* [155], and Woodward [171] discussed causality with a manipulation criterion. In this thesis, I follow the definition from Spirtes *et al*. [155] and consider causal knowledge from probabilistic perspective with manipulation criterion.

The main scientific method for causal knowledge discovery from data is randomized experiments in Statistics [58,125,144]. The interventional data is collected in randomized experiments to infer the causal strength of the randomized variables on other variables. Bayesian experiment design and optimal experiment design [22] have been explored to optimize the parameters in the experiments to make the interventional data collection and analysis more effective for causal knowledge discovery. However, hypothesis generation is not integrated in traditional experiment

design, and only one variable is considered as the target variable at a time. To model the causal relationships between multiple variables for hypothesis generation, causal Bayesian networks should be considered.

Wright [172] is among the first to use a graphical model for causal knowledge discovery. Rubin [143,144] is one of the pioneers to infer causal knowledge from observational data with statistical methods. Pearl [130,131] and Spirtes *et al*. [155,156] have developed a comprehensive theory for causal knowledge discovery from observational data with Bayesian networks. Pearl [129] proposed three basic rules to make it possible to infer the probabilities under manipulation from observational data with graphical models.

The knowledge discovery process has been discussed in general (i.e., in expert systems [74,133] and data mining [13,23], and the survey [101]). Fayyad, Piatetsky-Shapiro and Smyth [54] discussed the general knowledge discovery tasks, the typical methods and the knowledge discovery process. The large amount of work in knowledge discovery focuses on *observational data* for correlational knowledge discovery. However, hypothesis refinement and hypothesis verification have not been sufficiently considered in the knowledge discovery processes mentioned above, especially little work on hypothesis verification with *interventional data*.

Knowledge discovered from observational data has been applied in many domains. However, it may not help causal knowledge discovery in many situations. For example, observational data cannot distinguish the simple causal models with two variables, such as A→B or A←B, since these two models imply the same conditional

(in)dependence from observational data. Alternatively, interventional data can distinguish such models for causal knowledge discovery. If the concrete causal knowledge is required in some reverse engineering projects, we need to conduct the interventional experiments. But in some domains, we cannot conduct interventional experiments due to financial, legal or ethical reasons, and cannot collect interventional data. In this case, we need to resort to causal knowledge discovery with observational data. In summary, causal knowledge discoveries from interventional data and observational data have their own advantages and disadvantages, and their application domains. Generally, observational data can be collected economically, and interventional data will be collected costly. Causal knowledge discovery with observational data and interventional data is complimentary to each other.

Our framework includes three components (*hypothesis generation*, *hypothesis refinement* and *hypothesis verification*), exploits the available resources (i.e., observational data, interventional data, topological domain knowledge, and interventional experiments) to discover new causal knowledge, and minimize the number of experiments required for new interventional data collection. The comparisons of our proposed methods with the related work in each component are discussed in the following sub-sections. More related references, brief comments and comparisons with the proposed methods are listed in Appendix A.E.

7.2.1 Related Work for Hypothesis Generation with Variable Grouping

Hypothesis generation with variable grouping in Bayesian network structure learning is one method for variable aggregation. The group variables as hidden or latent variables are dependent on each other as a Bayesian network in our proposed method, and the original variables are independent of each other given the group variables. The idea of variable aggregation is not really new and has been considered in many situations, inside and outside of Bayesian network area. The general variable clustering [105] is one way to detect the redundant variables or highly-correlated variables in the data. However, it does not consider the dependency between the different clusters of the original variables.

Hidden variable discovery can be identified with maximal cliques [117] or semi-maximal cliques in the learned Bayesian networks [52]. However, the hidden variables identified in this way are difficult to interpret.

Module networks [148] defined the sets of variables with the similar behaviors as a module. The variables in the same modules have the same parents and the same conditional probability distributions. By enforcing such constraints, the complexity of the Bayesian network space is significantly reduced as well as the number of parameters. Different from our proposed Bayesian network structure learning with variable grouping, no hidden variables were explicitly introduced in module networks: the variables in module networks are interacting with each other directly and the search space is still very big.

Hierarchical Bayesian networks [79] consider the Cartesian product of the original variables as composite variables, which is similar to the clustering method[30] for Bayesian network inference [21]. This is another way for variable aggregation in Bayesian networks. However, the Cartesian product of the original variables may lead to an exponential number of states in the composite variables.

Multiply Sectioned Bayesian Networks (MSBNs) [173] were proposed to identify groups of variables as sections, and a local Bayesian network could be built with the variables in one section. The local Bayesian networks can be connected together through d-sep nodes among the sections to build MSBNs. Network fragments [102] were proposed to build partial Bayesian networks from domain knowledge as blocks for big Bayesian network construction. These two methods focus on the Bayesian network construction and no known work has been developed for learning.

When the classes or population of variables are considered, Bayesian network can be extended to object-oriented Bayesian networks [98], probabilistic frame-based systems [99], probabilistic relational models [63], and the first-order probabilistic models [134]. The variables in these models are class variables as in object-oriented languages, the class variables can be instantiated as objects, and the structures and parameters of the objects can be instantiated and reused for many times. Most of these models are new methods for knowledge representation and inference based on Bayesian networks. In the learning, objects and their relations have to be specified in skeletons. In our work, the group variables can be treated as class variables in the

[30] The clustering method has been proposed to cluster the variables together to transform the Bayesian networks with un-directed cycles into poly-trees for efficient inference.

first-order probabilistic model and each member in the groups can be treated as an individual in a population. Then our work provides a way to learn a first-order probabilistic model from data.

7.2.2 Related Work for Hypothesis Refinement

It is not new to facilitate the causal structure inference with domain knowledge [77] and the Bayesian network learning with domain knowledge [38,67,87]. The quantitative domain knowledge has been explored to learn the conditional probabilities [11,94,95,126]. The qualitative domain knowledge can be represented as the topological constraints in Bayesian networks [38,87].

Physical theories are required in Griffiths *et al.* [77] as domain knowledge to infer the causal structure. A causal ordering of variables is required to learn the Bayesian network structure in Cooper and Herskovits [38]. An initial Bayesian network is required in Heckerman *et al.* [87]. The degree of the node connected to other nodes is required in Friedman *et al.* [67]. Although these methods work well with the required domain knowledge, such knowledge is not available in many cases. Also, in these methods, the elicitation of domain knowledge is ad hoc and not in the systematic way for causal Bayesian network learning.

Certain kinds of partial domain knowledge have been considered in Bayesian network structure learning ([155], Section 5.4.5) in packages like LibB, TETRAD and Bayesian network PowerConstructor[31]. However, as far as we know, there is no

[31] Same as Footnote 16.

systematic representation, analysis and evaluation on incorporating partial topological domain knowledge into Bayesian network structure learning, and the explicit effects and influences of different kinds of topological constraints are unknown.

7.2.3 Related Work for Hypothesis Verification

Hypothesis verification with interventional data is important before applying the discovered knowledge to real causal prediction and control. The mixture of observational data and interventional data has been explored for causal knowledge discovery. Cooper and Yoo [39] first examined the assumptions to combine observational data and interventional data for knowledge discovery with Bayesian networks. Active learning [35,113,138] has been tried to guide the new data collection with the available resources to reduce the variance of the model. Recently, Tong and Koller [160,161] and Murphy [121] applied active learning to causal Bayesian network learning. They applied the decision theoretic approach to estimating the expected posterior loss to select variables for manipulation. Every possible intervention and their corresponding possible outcomes should be considered to estimate the expected posterior loss. However, the computational complexity involved is very high and only the case with one instance collected in each intervention is considered in their work. The case with multiple data instances collected in each intervention is not considered, although this is a general situation in practice.

We propose the node-based interventions and edge-based interventions for causal knowledge discovery. The hypothesis selection is based on non-symmetrical entropy

from the current data, and the possible outcomes from each intervention do not need to be considered, which will reduce the computational complexity. The stopping criterion is based on the structure entropy from the learned Bayesian networks. The detailed comparison of our proposed methods and active learning with expected posterior loss is listed in Table 25.

	one instance collected from each intervention	m instances collected from each intervention	comments
Active learning with expected posterior loss for intervention selection [121,160,161]	computational complexity in each active learning step is $o(n^2 * 2^{2n})$	computational complexity in each active learning step is $o(n^2 * 2^{mn+n-m})$	Theoretically, active learning with expected posterior loss is the best one to achieve the smallest structure entropy. However, the computational cost makes it infeasible in most of the situations. Can be compared with node-based intervention when only one instance can be collected in each intervention. In this case, the learned Bayesian networks have the smaller structure entropy than those from node-based intervention. Can not finish the experiments in a reasonable time with multiple data instances collected in each intervention
Our proposed node-based intervention with non-symmetrical entropy from the current data	Computational complexity in each active learning step is $o(n * 2^n)$.	Computational complexity in each active learning step is $o(n * 2^n)$.	Can be used to establish the causal ordering of variables with interventional data. The learned structure entropy is near that from expected posterior loss when one instance is collected in each intervention. Can be applied to the case when multiple data instances are collected from each intervention. But may not determine some direct causal relationships.
Our proposed edge-based intervention with non-symmetrical entropy from the current data	Not applicable	It depends. Generally not applicable	Can identify the direct causal relationships. But need to consider the exponential numbers of configurations of manipulated variables for data collection to establish the direct causal relationship from one variable to another variable. Computational complexity in each active learning step is $o(n * 2^n)$.

Table 25 Comparisons of the active learning methods for causal Bayesian network learning

7.3 Future Work

The possible future work includes: 1) extending the topological constraints to soft topological constraints; 2) variable selection for causal Bayesian network building; and 3) hidden variable discovery.

7.3.1 Extending to Soft Topological Constraints

Currently, we consider domain knowledge as concrete topological constraints: whether there is a root, a leaf node or an edge. In some situations, however, we are not sure about the available domain knowledge. For example, from domain knowledge, we may have 80% confidence that variable A affects variable B. In this case, we cannot specify that there is an edge from A to B as in Chapter 4. We need other methods to deal with soft or uncertain topological constraints.

7.3.2 Variable Selection for Causal Bayesian Networks

The first step in Bayesian network building is to determine variables in a domain of interest. In our work, we assume that variables in a domain have been determined beforehand. However, in many real applications, the variables related to knowledge discovery objective may not be known in advance. We need to select variables in the process of Bayesian network learning [40] for knowledge discovery. From Simpson's Paradox [130], we know that, the conclusion of a statistical test can be reversed under some occasions when one extra variable is included into a model. In this case, we must pay more attention to the variable selection problem and determine when it is

appropriate to include a variable into the model or exclude a variable from the model for knowledge discovery.

7.3.3 Hidden Variable Discovery

Hidden variables are a difficult and complex issue in knowledge discovery and have been explored with Bayesian networks [8,34,36,52,61,77,117,174]. However, some important questions related to hidden variables need to be further examined: 1) when is a hidden variable really needed for knowledge discovery? and 2) what is the real meaning of the hidden variable, if applicable? Introducing a hidden variable into the model for knowledge discovery will change the hypothesis space, which will significantly change the problem complexity in many domains. As pointed out by Tiles ([158], page 12) nineteen years ago, automated programs cannot restructure the problem space and introduce new (or hidden) variables into the model for knowledge discovery. How to introduce a new variable to change the hypothesis space is still a problem we face today.

Appendix

A. Hypothesis Generation with Two Variables

This section is a brief review of some methods for hypothesis generation. The possible methods are correlation, chi-square, and mutual information. These methods are used to determine the dependencies between the variables. Whether the dependencies are causal or associational, it is dependent on the characteristic of the data. If the data is from intervention, the estimated dependencies will be causal; otherwise, the dependencies will be associational.

The hypotheses generated with two variables are only the total dependencies between two variables. The total dependencies between two variables can come from the direct dependency between two variables, and from the indirect dependencies – through paths along other variables. Therefore, the total dependencies from two variables cannot be used to determine the direct influence; however, they can be used as indicators of the direct dependencies – high total dependencies sometimes mean the high direct dependencies. Low total dependencies, however, do not guarantee the low direct dependencies, since there can be multiple paths between two variables and the dependencies through different paths can reduce the effect of each other.

i. Correlation for Continuous Variables

In probability theory and Statistics, **correlation**, also called **correlation coefficient**

[141], indicates the strength and direction of a linear relationship between two random variables. The correlation coefficient $\rho_{X,Y}$ between two random variables X and Y with expected values μ_X and μ_Y and standard deviations σ_X and σ_Y is defined as:

$$\rho_{X,Y} = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E((X-\mu_X)(Y-\mu_Y))}{\sigma_X \sigma_Y}$$

where $E()$ is the expectation function of a random variable and $\text{cov}(X,Y)$ is the covariance function of the random variables.

The maximum of the absolute correlation coefficient value is 1. If the correlation coefficient is +1, it means that two variables change linearly in the same directions. If the correlation coefficient is -1, it means that two variables change linearly in the opposite directions.

When two variables are independent, their correlation should be 0. However, when the correlation is 0, it does not mean that two variables are independent, since correlation only measures the linear dependency between two variables.

ii. Chi-square Test for Discrete Variables

The **chi-square** value between two variables [141] is defined as

$$\chi^2 = \sum_{i=1}^{m}\sum_{j=1}^{n}\frac{(A_{ij}-E_{ij})^2}{E_{ij}}$$

Where m is the number of the possible states of variable 1, n is the number of the possible states of variable 2, A_{ij} is the number of instances with i-th value for variable 1 and j-th value for variable 2, R_i is the number of instances with i-th

value for variable 1, C_j is the number of instances with j-th value for variable 2, N is the number of the total instances, and $E_{ij} = R_i * C_j / N$ is the expected frequency of A_{ij}. The chi-square value measures the difference of the expected frequencies and the actual frequencies in different categories.

We have tried the chi-square measure on a data set sampled from the Asia network. Most of the real edges in the Asia network achieved high chi-square values. In addition, some pairs of variables, without direct edges between them in the Asia network, also achieved high chi-square values. The possible explanation is that chi-square only measures the total dependency between two variables, both from the direct edges and from any indirect paths. The top chi-square values are shown in Table 26.

Order	Variable 1	Variable 2	Occurances
1	Lung_Cancer	Tuberculosis_or_Lung_Cancer	874.84
2	Tuberculosis_or_Lung_Cancer	X-ray_result	517.32
3	Bronchitis	Dyspnea	486.16
4	Lung_Cancer	X-ray_result	450.52
5	Tuberculosis	Tuberculosis_or_Lung_Cancer	110.53
6	Smoking	Bronchitis	71.81
7	Tuberculosis	X-ray_result	59.14
8	Smoking	Dyspnea	39.2
9	Smoking	Lung_Cancer	35.44
10	Visit_to_Asia	Tuberculosis	35.3

Table 26 High chi-square values between variables from data sampled from Asia network

iii. Mutual Information for Discrete Variables

Mutual Information (MI) [114] is an entropy-based measure of the dependency between two variables. It is the difference between the prior entropy of variable C and the posterior entropy of variable C given values of another variable F:

$$MI = entropy(C) - \sum_{F} \left(- P(F) * entropy(C \mid F) \right)$$

We have tried the mutual information on the data set sampled from the Asia network. Similar to the results from chi-square values, most of the real edges in the Asia network achieved high mutual information values, and some pairs of variables, without direct edges between them in the Asia network, achieved the high mutual information values too. The reason is the same – mutual information only measures the total dependency between two variables. The top mutual information values are shown in Table 27.

Order	Variable 1	Variable 2	Occurances
1	Bronchitis	Dyspnea	0.27
2	Lung_Cancer	Tuberculosis_or_Lung_Cancer	0.2
3	Tuberculosis_or_Lung_Cancer	X-ray_result	0.16
4	Lung_Cancer	X-ray_result	0.14
5	Smoking	Bronchitis	0.04
6	Tuberculosis	Tuberculosis_or_Lung_Cancer	0.02
7	Smoking	Lung_Cancer	0.02
8	Smoking	Dyspnea	0.02
9	Smoking	Tuberculosis_or_Lung_Cancer	0.02
10	Tuberculosis	X-ray_result	0.02

Table 27 High mutual information values between variables from data sampled from Asia network

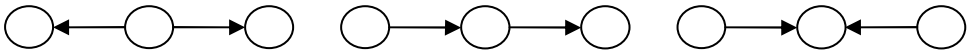
B. D-separation

Bayesian networks encode the dependencies and independencies between variables. Under the causal Markov assumption, each variable in a Bayesian network is independent of its non-descendants given the values of its parents. With the causal Markov assumption, we can check some conditional independence in Bayesian networks. For the general conditional independence in a Bayesian network, Pearl [131]

proposed a graphical criterion: d-separation. **D-separation** in Bayesian networks has the following implication: If two sets of variables X and Y are d-separated in Bayesian network by a third set Z (excluding X and Y), the corresponding variable sets X and Y are independent given the variables in Z. The definition of d-separation is: two sets of variables X and Y are d-separated in Bayesian network by a third set Z (excluding X and Y) if and only if every un-directed path between X and Y is "blocked", where the term "blocked" means that there is an intermediate variable W (distinct from X and Y) such that:

- The connection through W is "tail-to-tail" or "tail-to-head" and W is in Z;

- Or, the connection through W is "head-to-head" and neither W nor any descendant of W is in Z. The graph patterns of "tail-to-tail", "tail-to-head" and "head-to-head" are shown in Figure 34.



(a) tail-to-tail, or diverging pattern

(b) tail-to-head, or serial pattern

(c) head-to-head, or converging pattern

Figure 34 Patterns for paths through a variable

C. Results of Node-Based Interventions

For node-based interventions, we have tested two of our created Bayesian networks (Study network and Cold network) and three benchmark Bayesian networks (Cancer network, Asia network and Car network). The test conditions are:

1) Five different node selection criteria: node selection with non-symmetrical node entropy, symmetrical node entropy, the expected posterior loss, random node selection,

or observational data;

2) Two stopping criteria: the number of interventions and the structure entropy of the learned Bayesian networks;

3) The original conditional probabilities in the tested Bayesian networks or randomized conditional probabilities; and

4) Different numbers of instances from each intervention, which are from 1 to 200.

There are many different combinations of these conditions. Some representative results are shown here. More results will be available online.

i. Study Network

Figure 35 shows the active learning results from Study network. The stopping criterion is the structure entropy of the learned Bayesian networks. The original conditional probabilities from Study network are used. In Figure 35, node selection with non-symmetrical node entropy requires the minimal number of instances to achieve structures with the specified entropy in active learning with Bayesian networks.

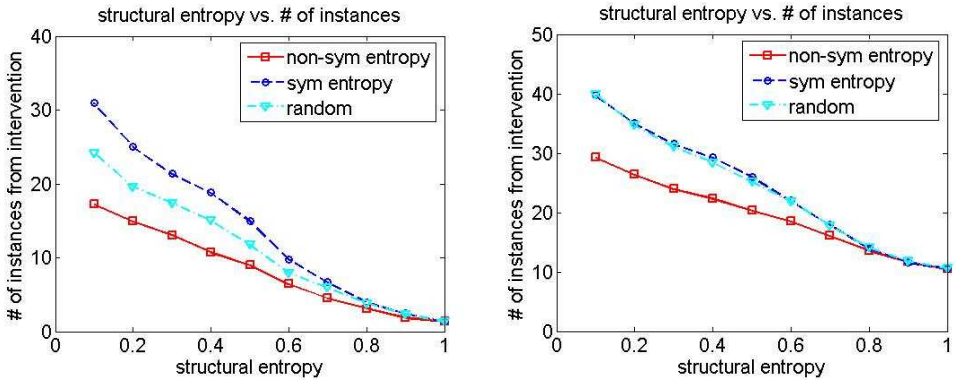


Figure 35 Active learning results from Study network

Note: The left panel shows the results when one instance is sampled in each intervention and the right panel shows the results when ten instances are sampled in each intervention

ii. Cold Network

Figure 36 shows the active learning results from Cold network. The stopping criterion is the structure entropy of the learned Bayesian networks. The original conditional probabilities from Cold network are used. In Figure 36, node selection with non-symmetrical node entropy requires much smaller number of total instances to reach the required structure entropy. When the number of instances collected in each intervention is large (100 or 200 in our example), non-symmetrical entropy performs much better than other node selection methods.

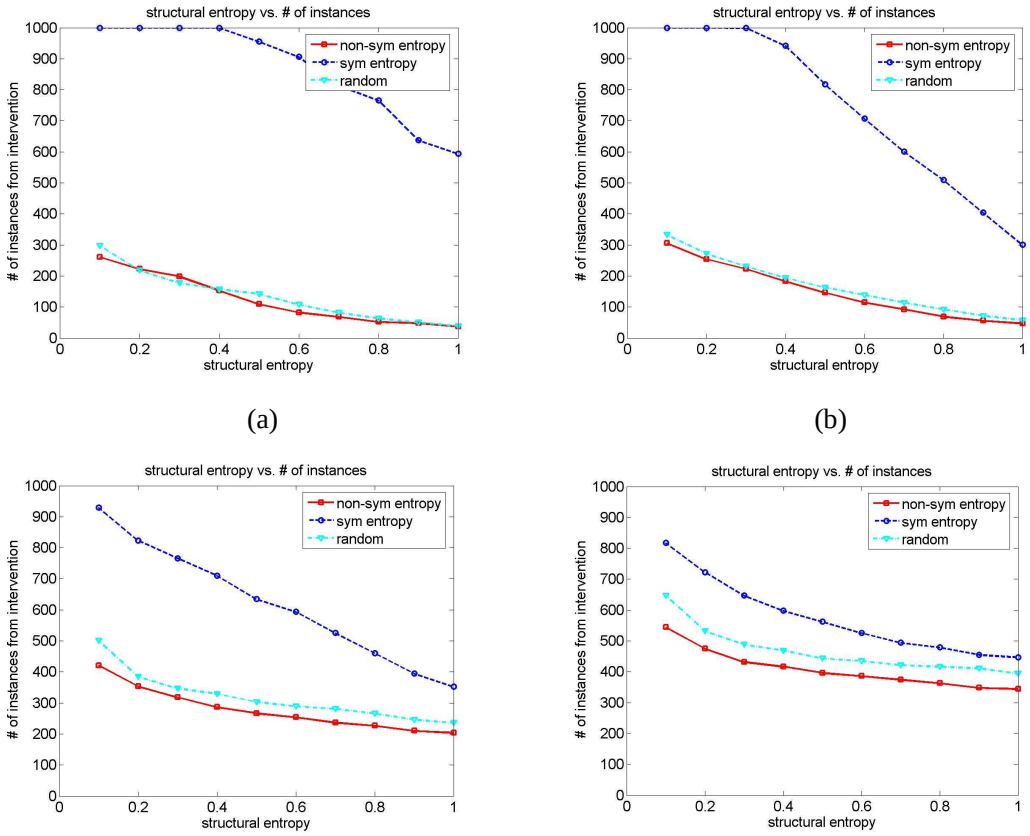


Figure 36 Active learning results from Cold network

Note: The numbers of the sampled instances in each intervention are 1, 10, 100, and 200 in (a), (b), (c), and (d), respectively.

iii. Cancer Network

Figure 37 shows the active learning results from Cancer network. The stopping criterion is the structure entropy of the learned Bayesian networks. The original conditional probabilities from Cancer network are used. In Figure 37, node selection with non-symmetrical entropy requires much smaller number of total instances to reach the required structure entropy. When the number of instances collected in each intervention is large (100 or 200 in our example), non-symmetrical entropy performs much better than other node selection methods.

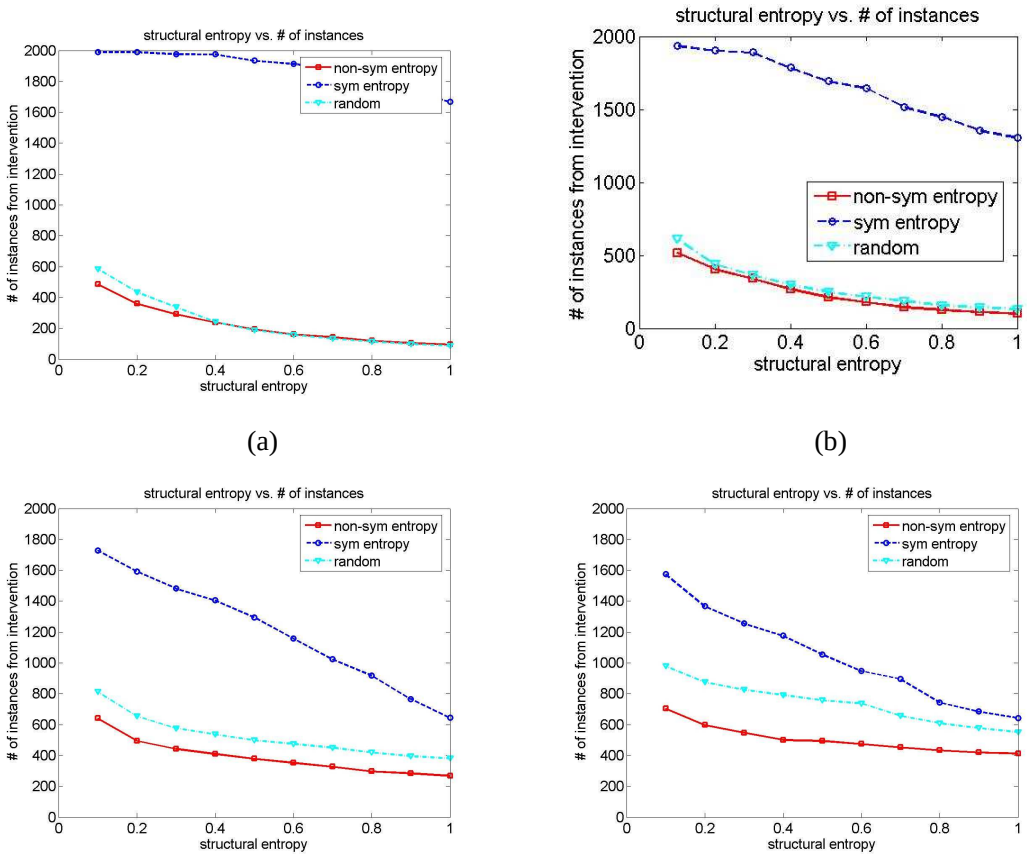
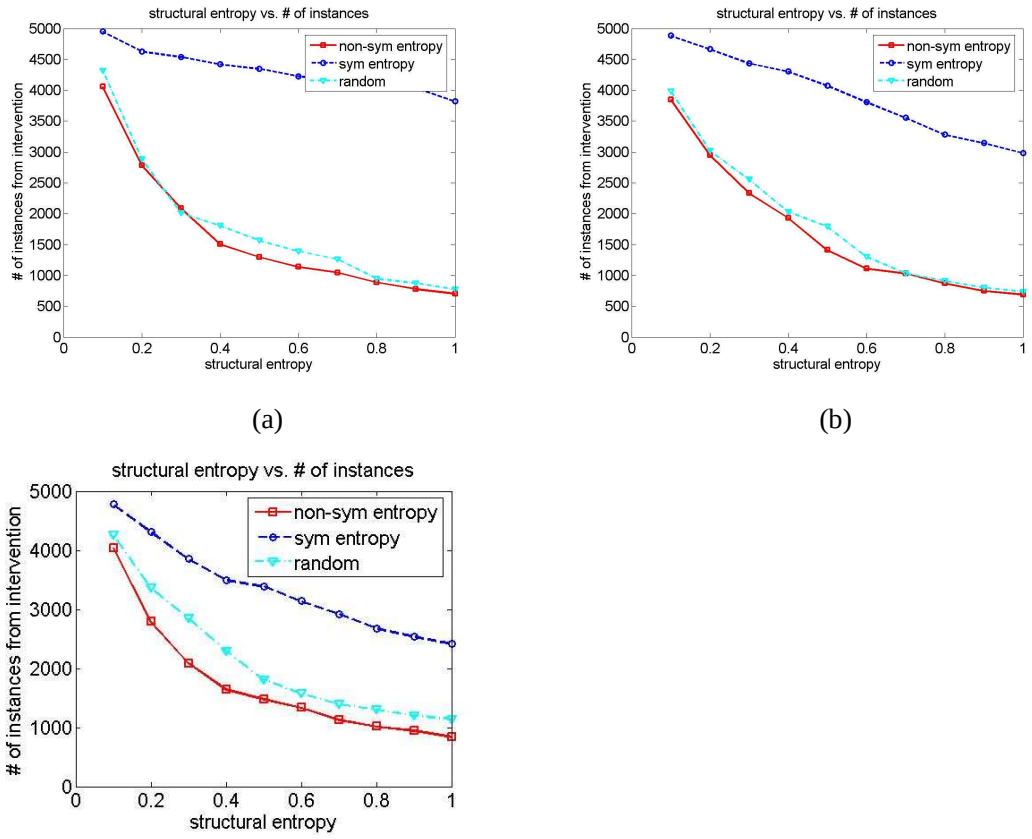


Figure 37 Active learning results from Cancer network

Note: The numbers of the sampled instances in each intervention are 1, 10, 100, and 200 in (a), (b), (c), and (d), respectively.

iv. Asia Network

Figure 38 shows the active learning results from Asia network. The stopping criterion is the structure entropy of the learned Bayesian networks. Randomized conditional probabilities are used for this example. In Figure 38, node selection with non-symmetrical entropy requires much smaller number of total instances to reach the required structure entropy. When the number of instances collected in each intervention is large (400 in our example), non-symmetrical entropy performs much better than other node selection methods.



(a) (b)

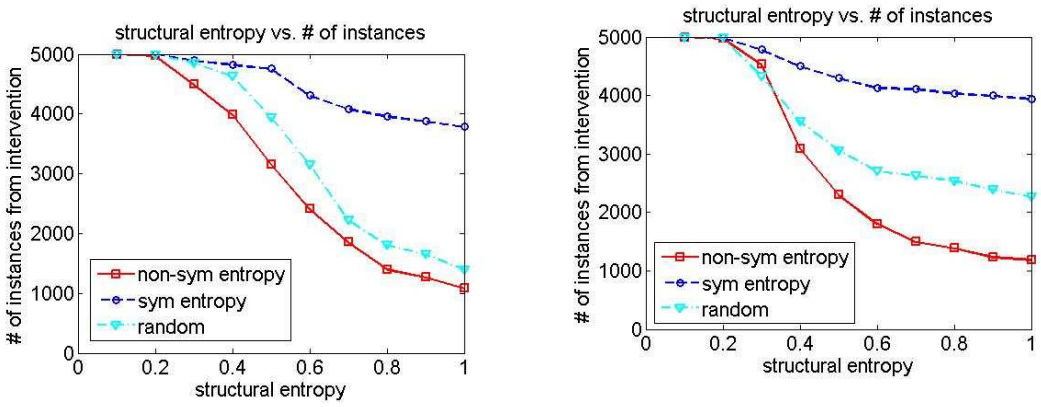
(c)

Figure 38 Active learning results from Asia network

Note: The numbers of the sampled instances in each intervention are 1, 10, and 400 in (a), (b), and (c), respectively.

v. Car Network

Figure 39 shows the active learning results from Car network. The stopping criterion is the structure entropy of the learned Bayesian networks. The original conditional probabilities are used for this example. In Figure 39, node selection with non-symmetrical entropy requires much smaller number of total instances to reach the required structure entropy. When the number of instances collected in each intervention is large (100 and 200 in our example), non-symmetrical entropy performs much better than other node selection methods. When the required structure entropy is small, all the node selection methods need to sample the maximal number of instances, which explains why the number of the total instances from intervention is 5000 when the structure entropy is 0.1 or 0.2.



(a) (b)

Figure 39 Active learning results from Car network

Note: The numbers of the sampled instances in each intervention are 100 and 200 in (a) and (b), respectively.

D. Selected Publications

The followings are the selected publications during my PhD study period:

- Li, Guoliang, Tze-Yun Leong, Active Learning for Causal Bayesian Network

Structure with Non-symmetrical Entropy, The 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), LNAI 5476, Springer-Verlag, 2009, pp. 290-301.

- Li, Guoliang, Steel Mike, Louxin Zhang, More Taxa Are Not Necessarily Better For the Reconstruction of Ancestral Character States, Systematic Biology 57 (4) (2008) 647-653.
- AH. Morris, J. Orme,Jr., JD Truwit, J. Steingrub, C. Grissom, KH Lee, Guoliang Li, BT Thompson, R. Brower, M. Tidswell, G. Bernard, D. Sorenson, K. Sward, H. Zheng, D. Schoenfeld, H. Warner, A replicable method for blood glucose control in critically ill patients, Critical Care Medicine. 36(6):1787-1795, June 2008
- Li, Guoliang, Tze-Yun Leong, Biomedical Knowledge Discovery with Topological Constraints Modeling in Bayesian Networks: A Preliminary Report, in: World Congress on Health (Medical) Informatics (MedInfo) (IOS Press, Brisbane, Australia, 2007) 560-565.
- Li, Guoliang, J. Ma, L. Zhang, Selecting Genomes for Reconstruction of Ancestral Genomes, Proceedings of the Fifth Annual RECOMB Satellite Workshop on Comparative Genomics (RECOMB-CG), LNBI 4751, 2007, pp. 110-121.
- Li, Guoliang, Tze-Yun Leong, and Louxin Zhang, Translation Initiation Sites Prediction with Mixture Gaussian Models in Human cDNA Sequences. IEEE Transactions on Knowledge and Data Engineering, 2005. 17(8): p. 1152-1160.

- Li, Guoliang and Tze-Yun Leong, Feature Selection for the Prediction of Translation Initiation Sites. Genomics, Proteomics & Bioinformatics, 2005. 3(2): p. 73-83.
- Li, Guoliang and Tze-Yun Leong, A framework to learn Bayesian Networks from changing, multiple-source biomedical data. in Proceedings of the 2005 AAAI Spring Symposium on Challenges to Decision Support in a Changing World. Stanford University, CA, USA, 66-72.
- Li, Guoliang, Tze-Yun Leong, L. Zhang, Translation Initiation Sites Prediction with Mixture Gaussian Models, the Proceedings of the 4th Workshop on Algorithms in Bioinformatics (WABI 2004), LNBI 3240, Bergen, Norway, 2004, pp. 338-349.

E. Summary of Related Work and Comments

The followings are some selected references related to this research, some brief comments and the comparisons with the methods proposed in this thesis.

Topic	References	Comments
Knowledge discovery framework	the general process of knowledge discovery [13,23,54,74,133], and the survey [101]	More emphasis on hypothesis generation
Our three-step iterative framework	Our proposed framework	More emphasis on hypothesis refinement and hypothesis verification

Table 28 References for knowledge discovery process

Topic	References
Bayesian network theory	Pearl [130,131], Spirtes *et al.* [155,156]
Bayesian network construction from domain knowledge	Druzdzel and van der Gaag [46], Heckerman [89], Nadkarni and Shenoy [124]
Bayesian network parameter learning	With complete data [15,153], with incomplete data by gradient method [8,157], the EM method [103] and Monte Carlo methods such as Gibbs sampling [71].
Bayesian network structure learning	The representative methods in score-and-search-based category are K2 algorithm [38], Greedy search, Markov Chain Monte Carlo (MCMC), and Structural EM [60]. The representative methods in constraint-based category are SGS algorithm and PC algorithm [155]
Bayesian network structure learning with the mixture of observational and interventional data	Cooper and Yoo [39], Tong and Koller [161], Murphy [121]
Proponents on causal knowledge discovery with Bayesian networks	Pearl [130], Spirtes *et al.* [155,156], Korb and Wallace [100]
Opponents on Causal knowledge discovery with Bayesian networks	Cartwright [19,20], Humphreys and Freedman [91], and McKim and Turner [118]

Table 29 Selected references for Bayesian networks

Topic	References	Comments
Variable clustering	Lee *et al.* [105]	No dependency between the variable clusters
Hidden variable discovery in Bayesian networks	with maximal cliques [117] or semi-maximal cliques [52] in the learned Bayesian networks	difficult to interpret the hidden variables
Module networks	The variables in the same modules have the same parents [148]	No hidden variable introduced. The search space is still very large
Hierarchical Bayesian networks	Cartesian product of the original variables as composite variables [79]	Possibly an exponential number of states in the composite variables
Multi-sectioned Bayesian Network	Xiang *et al.* [173]	Mainly for Bayesian network construction
Network fragment	Laskey and Mahoney [102]	Mainly for Bayesian network construction
First-order probabilistic models and the variants	first-order probabilistic models (Poole, 2003), object-oriented Bayesian network (Koller & Pfeffer, 1997), or probabilistic frame-based systems (Koller & Pfeffer, 1998)	Objects and relations have to be specified in skeleton
Latent Tree Models	Wang *et al.* [168]	Hidden variables are dependent on each other in a tree structure
Bayesian network structure learning with variable grouping	Our proposed method	Hidden variables are dependent on each other in a network structure. No need specify the relations in skeleton as required in PRMs.

Table 30 References for variable aggregation – Related to hypothesis generation

Topic	References	Comments
General domain knowledge	Donoho and Rendell [45] and Han *et al.* [82]	The representation is not for Bayesian network
general knowledge refinement	general knowledge refinement [72,162,163]	meta-knowledge is used to refine the specific domain knowledge
quantitative domain knowledge in Bayesian networks	Boutilier *et al.* 1996; Joshi and Leong 2006; Niculescu *et al.* 2006; Joshi *et al.* 2007 [11,94,95,126]	Not our research focus in this thesis
qualitative domain knowledge in Bayesian networks	Cooper and Herskovits [38], and Heckerman *et al.* [87], LibB, TETRAD and Bayesian network PowerConstructor[32]	The proposed topological constraints. The systematic domain knowledge such as the full causal ordering of variables may not be available. The effects of different topological constraints are unknown.

Table 31 References for domain knowledge – Related to hypothesis refinement

Topic	References	Comments
Causal knowledge	Aristotle's doctrine of four causes; logical perspective, probabilistic perspective, Granger causality, counterfactual causality, [90,106,130,155,171]	I follow the definition of causal knowledge from Spirtes *et al.* [155]: causal knowledge from probabilistic perspective with manipulation criterion
causal knowledge discovery with randomized experiments	Neyman [125], Fisher [57], Rubin [144]	The established method for causal knowledge discovery in scientific research. Manipulation-based
causal knowledge discovery with observational data	Pearl [130], Spirtes *et al.* [155], Rubin [143]	With causal Markov assumption, causal sufficiency assumption, and faithfulness assumption
Knowledge discovery with observational data	knowledge discovery in database [53,86]: classification, regression, clustering, and association rule mining with observational data	Correlational information from observational data. May not be causal knowledge
Causal knowledge discovery with the mixture of observational and interventional data	Probability update [39], active learning [121,160,161]	My proposed method in this category. Active learning with Bayesian networks

Table 32 References for causal knowledge and causal knowledge discovery – Related to hypothesis verification

[32] Same as Footnote 16.

Index

References

[1] K. Aas, Microarray Data Mining: A Survey. Note, Norsk Regnesentral, SAMBA/02/01, 2001.

[2] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, I. Verkamo, Fast Discovery of Association Rules, in: U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy (Eds.), Advances in Knowledge Discovery and Data Mining, AAAI Press, 1996, pp. 307-328.

[3] R. Agrawal, R. Srikant, Fast Algorithms for Mining Association Rules in Large Databases, VLDB 1994, pp. 487-499.

[4] D.W. Aha, D. Kibler, M. Albert, Instance-based learning algorithms, Machine Learning 6 (1991) 37-66.

[5] T.V. Allen, R. Greiner, Model Selection Criteria for Learning Belief Nets: an Empirical Comparison, ICML, 2000, pp. 1047-1054.

[6] E. Bauer, D. Koller, Y. Singer, Update rules for parameter estimation in Bayesian networks, Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence, Providence, Rhode Island, USA, 1997, pp. 3-13.

[7] P. Berkhin, Survey Of Clustering Data Mining Techniques, Technical Report, Accrue Software Inc., 2002, pp. 1-56.

[8] J. Binder, D. Koller, S. Russell, K. Kanazawa, Adaptive probabilistic networks with hidden variables, Machine Learning 29 (1997) 213-244.

[9] C.M. Bishop, Neural networks for pattern recognition, Clarendon Press, Oxford, 1995.

[10] R. Bouckaert, Properties of Bayesian belief network learning algorithms, Tenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, 1994, pp. 102-109.

[11] C. Boutilier, N. Friedman, M. Goldszmidt, D. Koller, Context-specific independence in Bayesian Networks, In Proceeding of 12th Conf. on Uncertainty in Artificial Intelligence (UAI-96), 1996, pp. 115–123.

[12] G. Box, G. Jenkins, Time series analysis: Forecasting and control, Holden-Day, San Francisco, 1970.

[13] R.J. Brachman, T. Anand, The process of knowledge discovery in databases, Advances in knowledge discovery and data mining, American Association for Artificial Intelligence, Menlo Park, CA, USA, 1996, pp. 37-57.

[14] S. Brin, R. Motwani, J.D. Ullman, S. Tsur, Dynamic Itemset Counting and Implication Rules for Market Basket Data, Proceedings ACM SIGMOD International Conference on Management of Data, 1997, pp. 255-264.

[15] W. Buntine, Operations for learning with graphical models, Journal of Artificial Intelligence Research 2 (1994) 159-225.

[16] W. Buntine, Theory refinement on Bayesian networks, Proceedings of the seventh conference (1991) on Uncertainty in artificial intelligence, Morgan Kaufmann

Publishers Inc., Los Angeles, California, United States, 1991, pp. 52-60.
[17] C.J.C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery 2 (1998) 121-167.
[18] D. Campbell, J. Stanley, Experimental and quasi-experimental designs for research (reprinted from Handbook of Research on Teaching, 1963), Houghton Mifflin Co., Boston, 1966.
[19] N. Cartwright, Against modularity, the causal Markov condition, and any link between the two: Comments on Hausman and Woodward, British Journal for the Philosophy of Science 53 (2002) 411-453.
[20] N. Cartwright, What is wrong with Bayes Nets?, The Monist 84 (2001) 242-264.
[21] E. Castillo, J.M. Gutiérrez, A.S. Hadi., Expert systems and probabilistic network models, Springer, New York, 1997.
[22] K. Chaloner, I. Verdinelli, Bayesian experimental design: A review, Statistical Science 10 (1995) 273-304.
[23] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, R. Wirth, CRISP-DM 1.0 Step-by-step data mining guide http://www.crisp-dm.org/, 2000, pp. 1-78.
[24] P. Cheeseman, J. Stutz, Bayesian classification (AutoClass): Theory and results, in: U.M. Fayyad, G. Diatetsky-Shapiro, P. Smyth, R. Uthurusamy (Eds.), Advances in Knowledge Discovery and Data Mining, AAAI Press/The MIT Press, 1996, pp. 153-180.
[25] C.-C. Chen, K.M. Koh, Principles and techniques in combinatorics, World Scientific, Singapore 1992.
[26] Q. Chen, G. Li, B. Han, C.K. Heng, T.-Y. Leong, Coronary Artery Disease Prediction with Bayesian Networks and Constraint Elicitation, Technical report TRC9/06, School of Computing, National University of Singapore, September 2006, 2006.
[27] Q. Chen, G. Li, T.-Y. Leong, C.-K. Heng, Predicting Coronary Artery Disease with Medical Profile and Gene Polymorphisms Data, World Congress on Health (Medical) Informatics (MedInfo), IOS Press, Brisbane, Australia, 2007, pp. 1219-1224.
[28] T. Chen, H.L. He, G.M. Church, Modeling gene expression with differential equations, Pacific Symposium Biocomputing (1999) 29-40.
[29] P.W. Cheng, From Covariation to Causation: A Causal Power Theory, Psychological Review 104 (2) (1997) 367-405.
[30] D.M. Chickering, Learning Bayesian networks is NP-complete, AI & STAT V, 1996.
[31] D.M. Chickering, Learning Equivalence Classes of Bayesian Network Structures, in: E. Horvitz, F.V. Jensen (Eds.), Proceedings of the Twelfth Annual Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, Reed College, Portland, Oregon, USA, 1996, pp. 150-157.
[32] D.M. Chickering, Optimal Structure Identification with Greedy Search, Journal of Machine Learning Research 3 (2002) 507-554.
[33] D.M. Chickering, A transformational characterization of Bayesian network structures, in: S. Hanks, P. Besnard (Eds.), Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, 1995, pp. 87-98.
[34] D.M. Chickering, D. Heckerman, Efficient approximations for the marginal likelihood

of Bayesian networks with hidden variables, Machine Learning 29 (1997) 181-212.
[35] D.A. Cohn, Z. Ghahramani, M.I. Jordan, Active Learning with Statistical Models, Journal of Artificial Intelligence Research 4 (1996) 129-145.
[36] G.F. Cooper, A Bayesian Method for Learning Belief Networks that Contain Hidden Variables, Journal of Intelligent Information Systems 4 (1) (1995) 71-88.
[37] G.F. Cooper, An overview of the representation and discovery of causal relationships using Bayesian networks, in: C. Glymour, G.F. Cooper (Eds.), Computation, Causation, and Discovery, AAAI Press and MIT Press, 1999, pp. 3-62.
[38] G.F. Cooper, E. Herskovits, A Bayesian method for the induction of probabilistic networks from data, Machine Learning 9 (1992) 309-347.
[39] G.F. Cooper, C. Yoo, Causal discovery from a mixture of experimental and observational data, Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, 1999, pp. 116-125.
[40] D. Danks, Learning the Causal Structure of Overlapping Variable Sets, in: S. Lange, K. Satoh, C.H. Smith (Eds.), Discovery Science: Proceedings of the 5th International Conference, Springer-Verlag, Berlin, 2002, pp. 178-191.
[41] A.P. Dawid, Conditional independence in statistical theory (with discussion), Journal of the Royal Statistical Society, Series B 41 (1979) 1-31.
[42] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum Likelihood from Incomplete Data via The EM Algorithm, Journal of Royal Statistical Society 39 (1977) 1-38.
[43] T.J. DiCiccio, R.E. Kass, A. Raftery, L. Wasserman, Computing Bayes Factors by Combining Simulation and Asymptotic Approximations, Journal of the American Statistical Association 92 (439) (1997) 903-915.
[44] F.J. Diez, Parameter adjustment in Bayes networks: The generalized noisy or-gate, in: D.H.a.A. Mamdani (Ed.), Proceedings of Ninth Conference on Uncertainty in Artificial Intelligence (UAI '93), Morgan Kaufmann, 1993, pp. 99-105.
[45] S.K. Donoho, L.A. Rendell, Constructive Induction Using Fragmentary Knowledge, Proceedings of International Conference on Machine Learning (ICML'96), Morgan Kaufmann, Bari, Italy, 1996, pp. 113-121.
[46] M.J. Druzdzel, L.C. van der Gaag, Building Probabilistic Networks: Where Do the Numbers Come From? Guest Editors' Introduction, IEEE Transactions on Knowledge and Data Engineering 12 (4) (2000) 481-486.
[47] D. Eaton, K. Murphy, Exact Bayesian structure learning from uncertain interventions, AI & Statistics, Vol. 2, 2007, pp. 107-114.
[48] F. Eberhardt, C. Glymour, R. Scheines, N-1 Experiments Suffice to Determine the Causal Relations Among N Variables, In Department of Philosophy, Carnegie Mellon University, Technical Report CMU-PHIL-161, 2004.
[49] F. Eberhardt, C. Glymour, R. Scheines, On the Number of Experiments Sufficient and in the Worst Case Necessary to Identify All Causal Relations Among N Variables, UAI-05, AUAI Press, 2005, pp. 178-184.
[50] B. Efron, R. Tibshirani, An Introduction to the Bootstrap, Chapman & Hall/CRC, 1994.
[51] M.B. Eisen, P.T. Spellman, P.O. Browndagger, D. Botstein, Cluster analysis and display of genome-wide expression patterns, The Proceedings of the National Academy of Sciences (PNAS) USA 95 (25) (1998) 14863-14868.

[52] G. Elidan, N. Lotner, N. Friedman, D. Koller, Discovering Hidden Variables: A Structure-Based Approach, Neural Information Processing Systems (NIPS), 2000, pp. 479-485.
[53] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, From Data Mining to Knowledge Discovery in Databases, AI Magazine 17 (3) (1996) 37-54.
[54] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, From data mining to knowledge discovery: An overview, in: U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy (Eds.), Advances in knowledge discovery and data mining, AAAI Press, Menlo Park, CA, USA, 1996, pp. 1-30.
[55] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy (Eds.), Advances in knowledge discovery and data mining, AAAI Press, Menlo Park, Calif., 1996.
[56] M. Firlej, D. Hellens, Knowledge Elicitation: a practical guide, Prentice Hall, New York, 1991.
[57] R.A. Fisher, The Arrangement of Field Experiments, Journal of the Ministry of Agriculture of Great Britain 33 (1926) 503-512.
[58] R.A. Fisher, Statistical Methods for Research Workers, Oliver & Boyd, London, 1925.
[59] R.A. Fisher, Sir, The design of experiments, Oliver and Boyd, Edinburgh 1937.
[60] N. Friedman, The Bayesian Structural EM Algorithm, in: G.F. Cooper, S. Moral (Eds.), Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, University of Wisconsin Business School, Madison, Wisconsin, USA, 1998, pp. 129-138.
[61] N. Friedman, Learning belief networks in the presence of missing values and hidden variables, in: D.H. Fisher (Ed.), Proceedings of the Fourteenth International Conference on Machine Learning (ICML 1997), Morgan Kaufmann, Nashville, Tennessee, USA, 1997, pp. 125-133.
[62] N. Friedman, D. Geiger, M. Goldszmidt, Bayesian Network Classifiers, Machine Learning 29 (1997) 131-163.
[63] N. Friedman, L. Getoor, D. Koller, A. Pfeffer, Learning Probabilistic Relational Models, Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI '99), Morgan Kaufmann Publishers, 1999, pp. 1300-1309.
[64] N. Friedman, M. Goldszmidt, A. Wyner, Data Analysis with Bayesian Networks: A Bootstrap Approach, Proceeding of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI), 1999, pp. 206-215.
[65] N. Friedman, D. Koller, Being Bayesian About Network Structure: A Bayesian Approach to Structure Discovery in Bayesian Networks, Machine Learning 50 (1-2) (2003) 95-125.
[66] N. Friedman, M. Linial, I. Nachman, D. Pe'er, Using Bayesian networks to analyze expression data, Journal of Computational Biology 7 (3-4) (2000) 601-620.
[67] N. Friedman, I. Nachman, D. Pe'er., Learning Bayesian network structure from massive datasets: The "sparse candidate" algorithm, Proceedings of 15th Conference on Uncertainty in Artificial Intelligence, Stockholm, Sweden, 1999, pp. 206-215.
[68] T.S. Furey, N. Cristianini, N. Duffy, D.W. Bednarski, M. Schummer, D. Haussler, support vector machine classification and validation of cancer tissue samples using microarray expression data, Bioinformatics 16 (10) (2000) 906-914.

[69] A.P. Gasch, P.T. Spellman, C.M. Kao, O. Carmel-Harel, M.B. Eisen, G. Storz, D. Botstein, P.O. Brown, Genomic expression programs in the response of yeast cells to environmental changes, Molecular Biology of the Cell 11 (12) (2000) 4241-4257.
[70] Z. Ghahramani, Learning Dynamic Bayesian Networks, in: C.L. Giles, M. Gori (Eds.), Adaptive Processing of Sequences and Data Structures. Lecture Notes in Artificial Intelligence, Springer-Verlag, Berlin, 1998, pp. 168-197.
[71] W. Gilks, S. Richardson, D. Spiegelhalter, Markov Chain Monte Carlo Methods in Practice, Chapman and Hall, London, 1996.
[72] A. Ginsberg, S.M. Weiss, P. Politakis, Automatic knowledge base refinement for classification systems, Artificial Intelligence 35 (2) (1988) 197-226.
[73] C. Glymour, G.F. Cooper (Eds.), Computation, Causation, and Discovery, MIT Press, Cambridge, MA, USA, 1999.
[74] G. Gorry, G. Barnett, Experience with a model of sequential diagnosis, Computers and Biomedical Research 1 (1968) 490-507.
[75] J. Grabmeier, A. Rudolph, Techniques of Cluster Algorithms in Data Mining, Data Mining and Knowledge Discovery 6 (4) (2002) 303-360.
[76] C.W.J. Granger, Testing for Causality: A personal viewpoint, Journal of Economic Dynamics and Control. 2 (1980) 329-352.
[77] T.L. Griffiths, E.R. Baraff, J.B. Tenenbaum, Using physical theories to infer hidden causal structure, Proceedings of the 26th Annual Conference of the Cognitive Science Society, 2004.
[78] T.L. Griffiths, J.B. Tenenbaum, Structure and strength in causal induction, Cognitive Psychology 51 (334-384) (2005).
[79] E. Gyftodimos, P. Flach, Hierarchical Bayesian Networks: A Probabilistic Reasoning Model for Structured Domains, Proceedings of the ICML-2002 Workshop on Development of Representations, 2002, pp. 23-30.
[80] J. Hall, G. Mani, D. Barr, Applying Computational Intelligence to the Investment Process, Proceedings of CIFER-96: Computational Intelligence in Financial Engineering, IEEE Computer Society, Washington, D.C., 1996.
[81] J. Han, M. Kamber, Data mining: concepts and techniques, Morgan Kaufmann, San Francisco, 2001.
[82] J. Han, L.V.S. Lakshmanan, R.T. Ng, Constraint-Based, Multidimensional Data Mining, Computer 32 (8) (1999) 46-50.
[83] D.J. Hand, K. Yu, Idiot's Bayes: Not So Stupid after All?, International statistical Review 69 (3) (2001) 385-398.
[84] S. Haykin, Neural networks: a comprehensive foundation, 2nd ed., Prentice Hall, Upper Saddle River, 1999.
[85] D. Heckerman, A Bayesian Approach to Learning Causal Networks, Proceedings of Eleventh Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, San Francisco, CA, 1995, pp. 285-295.
[86] D. Heckerman, A Tutorial on Learning with Bayesian Networks, in: M. Jordan (Ed.), Learning in Graphical Models, MIT Press, Cambridge, MA, 1998, pp. 301-354.
[87] D. Heckerman, D. Geiger, D.M. Chickering, Learning Bayesian networks: The combination of knowledge and statistical data, Machine Learning 20 (1995) 197-243.

[88] D. Heckerman, C. Meek, G.F. Cooper, A Bayesian approach to causal discovery, in: C. Glymour, G.F. Cooper (Eds.), Computation, Causation, Discovery, MIT Press, Cambridge, MA, USA, 1999, pp. 141-165.
[89] D.E. Heckerman, Probabilistic similarity networks, MIT Press, Cambridge, Mass., 1991.
[90] D. Hume, (1711-1776), A treatise of human nature, Oxford University Press, Oxford; New York, 2000.
[91] P. Humphreys, D. Freedman, The Grand Leap, The British Journal for the Philosophy of Science 47 (1) (1996) 113-123.
[92] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice Hall, 1988.
[93] A.K. Jain, N.M. Murty, P.J. Flynn., Data Clustering: A Review, ACM Computing Survey 31 (3) (1999) 264-323.
[94] R. Joshi, T.Y. Leong, Patient-specific Inference and Situation-dependent classification using Context-sensitive Networks, Proceedings of AMIA Annual Symposium, 2006, pp. 404-408.
[95] R. Joshi, G. Li, T.-Y. Leong, Context-aware Probabilistic Reasoning for Proactive Healthcare, Work Notes of 2nd Workshop on Artificial Intelligence Techniques for Ambient Intelligence (AITAmI 07), jointed with IJCAI2007, 2007.
[96] L. Kaufman, P.J. Rousseeuw, Finding groups in data: an introduction to cluster analysis, Wiley, New York, 1990.
[97] M. Koivisto, Advances in exact Bayesian structure discovery in Bayesian networks, in: R. Dechter, T. Richardson (Eds.), UAI 2006, AUAI Press, 2006, pp. 241-248.
[98] D. Koller, A. Pfeffer, Object-Oriented Bayesian Networks, in: D. Geiger, P.P. Shenoy (Eds.), Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, Brown University, Providence, Rhode Island, USA, 1997, pp. 302-313.
[99] D. Koller, A. Pfeffer, Probabilistic Frame-Based Systems, Proceedings of the Fifteenth National Conference on Artificial Intelligence and Tenth Innovative Applications of Artificial Intelligence Conference, AAAI 98, IAAI 98, AAAI Press / The MIT Press, Madison, Wisconsin, USA, 1998, pp. 580-587.
[100] K.B. Korb, C.S. Wallace, In search of the philosopher's stone: Remarks on Humphreys and Freedman's critique of causal discovery, British Journal for the Philosophy of Science 48 (1997) 543-553.
[101] L.A. Kurgan, P. Musilek, A survey of Knowledge Discovery and Data Mining process models, The Knowledge Engineering Review 21 (2006) 1-24.
[102] K.B. Laskey, S.M. Mahoney, Network Fragments: Representing Knowledge for Constructing Probabilistic Models, Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence (UAI 1997), Morgan Kaufmann, 1997, pp. 334-341.
[103] S.L. Lauritzen, The EM algorithm for graphical association models with missing data, Computational Statistics and Data Analysis 19 (1995) 191-201.
[104] S.L. Lauritzen, D. Spiegelhalter, Local computations with probabilities on graphical structures and their application to expert systems, Journal of the Royal Statistical Society, Series B 50 (2) (1988) 157-224.

[105] T. Lee, D. Duling, S. Liu, D. Latour, Two-stage variable clustering for large data sets, SAS Global Forum 2008, 2008.
[106] D. Lewis, Causation, The Journal of Philosophy 70 (17) (1973) 556-567.
[107] G. Li, T.-Y. Leong, Biomedical Knowledge Discovery with Topological Constraints Modeling in Bayesian Networks: A Preliminary Report, World Congress on Health (Medical) Informatics (MedInfo), IOS Press, Brisbane, Australia, 2007, pp. 560-565.
[108] G. Li, T.-Y. Leong, A framework to learn Bayesian Networks from changing, multiple-source biomedical data, Proceedings of the 2005 AAAI Spring Symposium on Challenges to Decision Support in a Changing World Stanford University, CA, USA, 2005, pp. 66-72.
[109] G. Li, T.-Y. Leong, L. Zhang, Translation Initiation Sites Prediction with Mixture Gaussian Models in Human cDNA Sequences, IEEE Transactions on Knowledge and Data Engineering 17 (8) (2005) 1152-1160.
[110] D.V. Lindley, Bayesian statistics, a review, Society for Industrial and Applied Mathematics, Philadelphia 1971.
[111] G. Livingston, J. Rosenberg, B. Buchanan, Closing the Loop: an Agenda- and Justification-Based Framework for Selecting the Next Discovery Task to Perform, Proceedings of the 2001 IEEE International Conference on Data Mining, IEEE Computer Society Press, 2001, pp. 385-392.
[112] G. Livingston, J. Rosenberg, B. Buchanan, Closing the Loop: Heuristics for Autonomous Discovery, Proceedings of the 2001 IEEE International Conference on Data Mining, IEEE Computer Society Press, 2001, pp. 393-400.
[113] D.J.C. Mackay, Information-Based Objective Functions for Active Data Selection, Neural Computation 4 (1992) 590-604.
[114] D.J.C. MacKay, Information theory, inference, and learning algorithms, Cambridge University Press, Cambridge, UK, 2003.
[115] D. Madigan, J. York, Bayesian graphical models for discrete data, International statistical Review 63 (1995) 215-232.
[116] M. Manago, M. Auriol, Mining for OR, ORMS Today (Special Issue on Data Mining) (1996) 28-32.
[117] J.D. Martin, K. VanLehn, Discrete factor analysis: learning hidden variables in Bayesian networks. Technical Report LRGC ONR-94-1, Department of Computer Science, University of Pittsburgh, 1994.
[118] V.R. McKim, S.P. Turner (Eds.), Causality in crisis?: statistical methods and the search for causal knowledge in the social sciences, University of Notre Dame Press, Notre Dame, 1996.
[119] S. Meganck, P. Leray, B. Manderick, Learning Causal Bayesian Networks from Observations and Experiments: A Decision Theoretic Approach, Proceedings of Modelling Decisions in Artificial Intelligence (MDAI 2006), LNAI 3885, 2006, pp. 58-69.
[120] A. Morris, C. Wallace, R. Menlove, T. Clemmer, J.J. Orme, L. Weaver, N. Dean, F. Thomas, T. East, M. Suchyta, E. Beck, M. Bombino, D. Sittig, S. Böhm, B. Hoffmann, H. Becks, N. Pace, S. Butler, J. Pearl, B. Rasmusson, Randomized clinical trial of pressure-controlled inverse ratio ventilation and extracorporeal CO2 removal for ARDS

[erratum 1994;149(3, Pt 1):838, Letters to the editor 1995;151(1):255-256, 1995;151(3):1269-1270, and 1997;156(3):1016-1017], Am J Respir Crit Care Med 149 (2) (1994) 295-305.
[121] K. Murphy, Active Learning of Causal Bayes Net Structure. Technical report, Computer Science Division, University of California, Berkeley, CA, 2001.
[122] K. Murphy, Bayes Net Toolbox for Matlab, http://www.cs.ubc.ca/~murphyk/Software/BNT/bnt.html, 2007.
[123] K. Murphy, Dynamic Bayesian Networks: Representation, Inference and Learning, Computer Science Division, University of California, Berkeley 2002.
[124] S. Nadkarni, P.P. Shenoy, A Causal Mapping Approach to Constructing Bayesian Networks, Decision Support Systems 38 (2) (2004) 259-281.
[125] J. Neyman, On the Application of Probability Theory to Agricultural Experiments. Essay on Principles. Section 9, Roczniki Nauk Rolniczych Tom X [in Polish]; translated in Statistical Science, 5, 465-480 (1923).
[126] R.S. Niculescu, T.M. Mitchell, R.B. Rao, Bayesian Network Learning with Parameter Constraints, Journal of Machine Learning Research 7 (2006) 1357-1383.
[127] A. Onisko, M.J. Druzdzel, H. Wasyluk, Learning Bayesian network parameters from small data sets: Application of Noisy-OR gates, International Journal of Approximate Reasoning 27 (2) (2001) 165-182.
[128] D. Pe'er, A. Regev, G. Elidan, N. Friedman, Inferring subnetworks from perturbed expression profiles, Bioinformatics 17 (2001) S215-S224.
[129] J. Pearl, Causal diagrams for empirical research, Biometrika 82 (4) (1995) 669-688.
[130] J. Pearl, Causality: models, reasoning, and inference, Cambridge University Press, New York, 2000.
[131] J. Pearl, Probabilistic Reasoning in Intelligent Systems, Morgan Kaufmann, San Francisco, California, 1988.
[132] J. Pearl, T. Verma, A theory of inferred causation, in: J. Allen, R. Fikes, E. Sandewall (Eds.), Principles of Knowledge Representation and Reasoning: Proceeding of the Second International Conference, Morgan Kaufmann, San Mateo, CA, 1991, pp. 441-452.
[133] M. Peot, R. Shachter, Learning From What You Don't Observe, Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence (UAI-98), Morgan Kaufmann, 1998, pp. 439-444.
[134] D. Poole, First-order probabilistic inference, Proceedings of the 18th International Joint Conference on Artificial Intelligence, IJCAI 2003, Acapulco, Mexico, 2003, pp. 985-991.
[135] H. Price, Agency and causal asymmetry, Mind 101 (403) (1992) 501-520.
[136] J.R. Quinlan, C4.5: programs for machine learning, Morgan Kaufmann, San Mateo, Calif., 1993.
[137] L.R. Rabiner, A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Proceedings of the IEEE 77 (2) (1989) 257-286.
[138] T. Raychaudhuri, L.G.C. Hamey, Minimisation of data collection by active learning, IEEE ICNN, 1995.
[139] C. Reid, Neyman From Life, Springer-Verlag, New York, 1982.

[140] R.W. Robinson, Counting unlabeled acyclic digraphs, in: C.H.C. Little (Ed.), Combinatorial Mathematics V, volume 622 of Lecture Notes in Mathematics, Springer-Verlag, Australia, 1977, pp. 28-43.
[141] S.M. Ross, Introduction to probability and statistics for engineers and scientists, Elsevier Academic Press, San Diego, Calif., 2004.
[142] D. Rubin, Inference and missing data, Biometrika 63 (1976) 581-592.
[143] D.B. Rubin, Causal Inference Using Potential Outcomes: Design, Modeling, Decisions, Journal of American Statistical Association 100 (469) (2005) 322-331.
[144] D.B. Rubin, Estimating Causal Effects of Treatments in Randomized and Nonrandomized Studies, Journal of Educational Psychology 66 (1974) 688-701.
[145] K. Sachs, O. Perez, D. Pe'er, D.A. Lauffenburger, G.P. Nolan, Causal Protein-Signaling Networks Derived from Multiparameter Single-Cell Data, Science 308 (5721) (2005) 523-529.
[146] L.E. Schulz, A. Gopnik, Causal Learning Across Domains, Developmental Psychology 40 (2) (2004) 162-176.
[147] G. Schwarz, Estimating the dimension of a model, The Annals of Statistics 6 (1978) 461-464.
[148] E. Segal, D. Pe'er, A. Regev, D. Koller, N. Friedman, Learning Module Networks, Proceedings of the 19th Conference in Uncertainty in Artificial Intelligence (UAI), Acapulco, Mexico, 2003, pp. 525-534.
[149] T. Senator, H.G. Goldberg, J. Wooton, M.A. Cottini, A.F. Umarkhan, C.D. Klinger, W.M. Llamas, M.P. Marrone, R.W.H. Wong, The Financial Crimes Enforcement Network AI System (FAIS): Identifying Potential Money Laundering from Reports of Large Cash Transactions, AI Magazine 16 (4) (1995) 21-39.
[150] M. Shwe, B. Middleton, D. Heckerman, M. Henrion, E. Horvitz, H. Lehmann, G. Cooper, Probabilistic diagnosis using a reformulation of the INTERNIST-1/QMR knowledge base I. The probabilistic model and inference algorithms, Methods of Information in Medicine 30 (1991) 241-255.
[151] R. Singh, N. Palmer, D. Gifford, B. Berger, Z. Bar-Joseph, Active learning for sampling in time-series experiments with application to gene expression analysis, Proceedings of the 22nd international conference on Machine learning, 2005, pp. 832-839.
[152] P.T. Spellman, G. Sherlock, M.Q. Zhang, V.R. Iyer, K. Anders, M.B. Eisen, P.O. Brown, D. Botstein, B. Futcher, Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization, Molecular Biology of the Cell 9 (12) (1998) 3273-3297.
[153] D.J. Spiegelhalter, S.L. Lauritzen, Sequential updating of conditional probabilities on directed graphical structures, Networks 20 (1990) 579-605.
[154] P. Spirtes, C. Glymour, R. Scheines, causality from probability, Proceedings of the Conference on Advanced Computing for the Social Sciences, Williamsburg, VA., 1990.
[155] P. Spirtes, C. Glymour, R. Scheines, Causation, Prediction, and Search (Second Edition), MIT Press, Cambridge, MA, USA, 2000.
[156] P. Spirtes, C. Glymour, R. Scheines, Causation, Prediction, and Search. Number 81 in Lecture Notes in Statistics, Springer Verlag, New York, 1993.
[157] B. Thiesson, Accelerated quantification of Bayesian networks with incomplete data,

Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD-95), AAAI Press, 1995, pp. 306-311.
[158] J.E. Tiles, G.T. McKee, G.C. Dean (Eds.), Evolving knowledge in natural science and artificial intelligence, Pitman, London 1990.
[159] D.M. Titterington, A.F. Smith, U.E. Makov, Statistical Analysis of Finite Mixture Distributions, John Wiley & Sons, New York, 1985.
[160] S. Tong, D. Koller, Active Learning for Parameter Estimation in Bayesian Networks, in: T. Leen, T. Dietterich, V. Tresp (Eds.), Proceedings of the 13th Advances in Neural Information Processing Systems (NIPS), MIT Press, 2000, pp. 647-653.
[161] S. Tong, D. Koller, Active Learning for Structure in Bayesian Networks, in: B. Nebel (Ed.), IJCAI 2001, Morgan Kaufmann, Seattle, Washington, USA, 2001, pp. 863-869.
[162] G.G. Towell, J.W. Shavlik, M.O. Noordewier, Refinement of Approximate Domain Theories by Knowledge-Based Neural Networks, AAAI-90, Proceedings of the 8th National Conference on AI, 1990, pp. 861-866.
[163] M. Valtorta, Knowledge base refinement: A bibliography, Applied Intelligence 1 (1) (1990) 87-94.
[164] V.N. Vapnik, The nature of statistical learning theory (2nd edition), Springer, New York, 1999.
[165] T. Verma, J. Pearl, Equivalence and synthesis of causal models, Proceedings of Sixth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, Boston, MA, 1990, pp. 220-227.
[166] J.I. Vidrine, C.B. Anderson, K.I. Pollak, D.W. Wetter, Race/Ethnicity, Smoking Status, and Self-Generated Expected Outcomes From Smoking Among Adolescents, Cancer Control 12 (2005) Supplement 251-257.
[167] C.S. Wallace, K.B. Korb, H. Dai, Causal discovery via MML, Proceedings of the Thirteenth International Conference of Machine Learning (ICML'96), Morgan Kaufmann, San Francisco CA USA, 1996, pp. 516-524.
[168] Y. Wang, N.L. Zhang, T. Chen, Latent Tree Models and Approximate Inference in Bayesian Networks, AAAI-2008, 2008, pp. 1112-1118.
[169] G.I. Webb, J.R. Boughton, Z. Wang, Not So Naive Bayes: Aggregating One-Dependence Estimators, Machine Learning 58 (1) (2005) 5-24.
[170] G. Wiederhold, Foreword: On the Barriers and Future of Knowledge Discovery, in: U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy (Eds.), Advances in Knowledge Discovery and Data Mining, AAAI Press/The MIT Press, 1996, pp. vii-xi.
[171] J. Woodward, Making things happen: a theory of causal explanation, Oxford University Press, 2003.
[172] S. Wright, Correlation and causation, Journal of Agricultural Research 20 (1921) 557-585.
[173] Y. Xiang, D. Poole, M.P. Beddoes, Multiply Sectioned Bayesian Networks and Junction Forests for Large Knowledge-Based Systems, Computational Intelligence 9 (1993) 171-220.
[174] C. Yoo, G.F. Cooper, Causal Discovery of Latent Variable Models from a Mixture of Experimental and Observational Data. CBMI Research Report CBMI-173, 2001.
[175] C. Yoo, V. Thorsson, G.F. Cooper, Discovery of Causal Relationships in a

Gene-regulation Pathway from a Mixture of Experimental and Observational DNA Microarray Data, Proceedings of Pacific Symposium on Biocomputing, World Scientific, 2002, pp. 498-509.

[176] M. Zaït, H. Messatfa, A comparative study of clustering methods, Future Generation Computer Systems, special issue on data mining 13 (2-3) (1997) 149-159.