Medical Data Mining using Bayesian Network and DNA Sequence Analysis

LEE Kit Ying

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Master of Philosophy
in
Computer Science and Engineering

August 2004


香港中文大學
大學
圖書館系統
12 APR 2005
UNIVERSITY
LIBRARY SYSTEM
THE CHINESE UNIVERSITY OF HONG KONG

Abstract of thesis entitled:

Medical Data Mining using Bayesian Network and DNA Sequence Analysis

Submitted by LEE Kit Ying
for the degree of Master of Philosophy
at The Chinese University of Hong Kong in August 2004

Due to the use of information technology in medical care, data collected from clinical processes can be used for discovering useful patterns. These patterns can be analyzed automatically or by medical professionals in order to develop better strategies to improve the quality of medical treatment. Our work is based on the Hepatitis B Virus (HBV) Genome Project for investigating the use of various machine learning and data mining models in the medical domain. This work is mainly divided into three major parts.

The first part is the study of the application of Bayesian network classifiers (BNCs) on clinical data mining, which focuses on finding the interrelationship among the clinical attributes, as well as their contributions to the disease. Variations of BNCs, including the Bayesian-augmented Naïve Bayes (BAN) and General Bayesian Network (GBN), which have various degrees of constraints on the attribute dependency, are proposed for classification. The existing learning algorithms of them are mainly based on dependency analysis approach. In this thesis, the evolutionary learning algorithm (Hybridized Evolutionary Programming, HEP) is applied for learning these classifiers with satisfactory performance.

The second part is the development of a generic framework for virus DNA analysis and finding genetic markers. Based on the Genome project, a comprehensive framework including the data preprocessing, clustering, feature selection, classification and evaluation modules is proposed. Each module serves its functions in the architecture with a certain flexibility of customization. Using this framework, there are some important biochemical and medical findings, including the sub-

grouping of HBV genotype C and genetic markers with high accuracy and sensitivity.

The last part is the optimization of the HEP algorithm for structural learning of Bayesian networks. The idea of the novel Adaptive HEP algorithm (A-HEP) is based on the concept of adjusting the population size adaptively according to the dissimilarity of individuals in the current population. With the use of an increasing and a decreasing routines. The population expands for increasing diversity and contracts for reducing computation. The experimental results illustrate that our A-HEP has reduced the running time by half on average.

In this thesis, the feasibility and efficiency of using computer science technology to solve medical and biochemical problems are demonstrated.

論文摘要

由於在醫療保健中應用資訊科技，臨床過程中收集的資料可以被用來發掘有用的樣式。 這些樣式可以由醫療專家分析或自動化分析，從而開發更好的策略以改進醫學治療的質素。 我們的工作建基於乙型肝炎病毒(HBV)染色體項目，調查各種各樣的機器學習和數據挖掘模型在醫療領域的應用。 這工作主要被劃分成三大部分。

第一部份是關於研究貝葉斯網路分類器 (Bayesian network classifiers) 在臨床數據挖掘的應用，集中於發掘臨床屬性之間的相互聯繫及它們對疾病的影響。
我們提議利用一系列 BNC 的變種，包括貝葉斯擴大的貝葉(BAN)和一般貝葉斯網絡(GBN)來進行分類。 它們現有的學習算法主要根據依賴性分析方法。在這份論文中，混種演化編程算法(HEP)將被應用在這些分類器的學習，並得到滿意的效率。

第二部份是關於發展一個框架以分析 DNA 病毒和發現基因標記。 根據我們的項目，我們提出一個全面的框架，內含多個模組，包括資料預處理模組，簇群建立模組，特點選擇模組、分類模組和評估模組。 各個模組以一定靈活性的定製在框架內發揮它的作用。 透過這個框架，有一些重要生物化學和醫療研究結果被發現，包括 HBV 基因型 C 的附屬群集和具高準確性和敏感性的基因標記。

最後的一個部份是關於爲貝葉斯網路(Bayesian network)的結構化學習而提出的 Hybrid EP (HEP) 的優化。 優化混種演化編程算法(A-HEP)的概念是根據當前生態群體的個體不相似性而適當調整群體大小。 利用增加和減少的程序，當多樣性增加時群體會擴展，而當計算減少時群體會收縮。 實驗結果說明，我們的 A-HEP 運行時間平均減少了一半。

這份論文展示了利用計算機科學技術去解決 醫療和生物化學問題的可行性和效率。

Acknowledgement

I wish to express my sincere gratitude to my supervisors Prof. Kwong-Sak Leung and Prof. Kin-Hong Lee. They have given me excellent guidance from the inception of research directions to the approach of solving problems. The knowledge I acquired is not only beneficial to my research, but also to my future career. In addition, I would like to thank Dr. Man-Leung Wong for his constructive guidance and ideas, and Dr. Yong-Liang for his discussion and insightful advice on my A-HEP algorithm.

The support from my colleagues is also indispensable. In particular, I am grateful to Johnson Hung and Eddie Ng, who work with me on HBV Genome Project, for their friendly help and support. Last but not least, I would like to thank my fellow classmates, including Joe Lau, Gordon Lam, Edith Ngai, Chi-Hang Chan, Chi-Wai Leung, Chi-Hung Law and Denny Zhang for their continuous warmth in the midst of my setbacks. I also wish you give my warm thanks to Eric Ko and Kevin Yuen for their advice as well as emotional support throughout this thesis.

Lastly, I want to express my warm thanks to family. Their love and care are the energy of my life.

To my family

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Due to the use of information technology in medical care, data collected from clinical processes can be used for discovering useful patterns. These patterns can be analyzed automatically or by medical professionals in order to develop better strategies to improve the quality of medical treatment. Our work is based on the Hepatitis B Virus Genome Project for investigating the use of various machine learning and data mining models in the medical domain. This work is mainly divided into three major parts - the study of Bayesian network classifier on clinical data mining; the development of a framework for virus DNA analysis and finding genetic markers; and the optimization of existing efficient Bayesian network learning algorithm - HEP. An introduction of our work is given in the following sections.

1.1 Project Background

The Hepatitis B Virus Genome Project is a co-operated research project among Department of Medicine and Therapeutics, Department of Biochemistry and Department of Computer Science and Engineering, CUHK. In Asia, infection of Hepatitis B virus (HBV) is a major health prob-

lem. Near 20% of Chinese population are HBV carriers, and up to 25% of HBV carriers will die as a result of HBV-related complications including liver cirrhosis and hepatocellular carcinoma (HCC) which is commonly known as liver cancer. The aim of the project is to find the genomic markers of the HBV and clinical information which are useful to predict occurrence of HCC and response to therapy.

In this project, clinicians select patients for investigation that based on their expert knowledge and selection criteria. Data of patients are then collected from the Prince of Wales Hospital. At the same time, the latest or past blood samples of patients which contain Hepatitis B Virus are sent to Department of Biochemistry. They carry out advanced sequencing experiments to extract the whole genome of the HBV for each patient. Finally, computer science researchers are responsible for the data mining phase that finds the genetic and clinical markers which are useful for disease prediction and diagnosis. Clinical information and virus genome data are both used in the mining of significant markers of liver cancer (HCC).

According to the HBV project proposal, we need to carry out data mining part mainly for HBV genomic data. This project is divided into two phases : Finding genomic markers of HBV that related to the liver cancer, and investigate the genomic characteristics of HBV in response to the drug treatment - Laminvndine. At this stage, we are working on the first phase. In this study, we look into the clinical data prepared by the clinicians, and the HBV DNA genomes prepared by biochemists. Patients who take part in this study are selected by the clinicians carefully, according to their age, sex, and past clinical status. Clinical attributes for analysis are chosen by the clinicians with their expert knowledge. The selection process and criteria of patients and

the research experiments run by the Department of Biochemistry will not be discussed in detail here.

In biological point of view, the genome of an organism is all of the genetic information or hereditary material possessed by an organism; the entire genetic complement of an organism, i.e. HBV genomes are extracted by experiments and represented in a form of DNA. In this study, we have DNA sequences from 100 Control patients and 100 HCC patients. The DNA sequences of HBV are not exactly the same for each group, and they possess some individual nucleotide mutations that may or may not be related to HCC. In literature, HBV can be divided into seven genotypes where each of them have more than 8% difference of nucleotides to the others. In Hong Kong, genotypes B and C are the most common types, and all the samples we have are of these genotypes. To reduce the noise of genotypical difference between genotype B and C, we analyze the DNA samples separately.

The aim of this study is to develop a classification model for HCC based on HBV DNA and clinical data. This classification model should have high accuracy, specificity and sensitivity for HCC diagnosis and prediction.

1.2 Problem Specifications

This work is based on the HBV Genome Project and investigates the use of different machine learning and data mining models in medical domain. The background of this project is described in the previous section. It is mainly divided into clinical data mining and DNA analysis. The aim of this study is to find genetic and clinical markers for HCC, i.e. to develop a classification model for HCC based on HBV DNA and clinical data.

The clinical data mining focuses on finding the inter-relationship between clinical attributes, as well as their contributions to liver cancer. Among various machine learning models, we choose Bayesian network which can represent the casual dependency with probability. Bayesian network classifiers, such as Naïve-Bayes, are also popular classification models with satisfactory performance. However, those simple structural classifiers limit the dependency among the attributes that may not be realistic in real-life problems. Therefore, the BN-augmented Naïve-Bayes (BAN) and General Bayesian Network (GBN), which release the constraint on attribute dependency, are proposed for classification. Since the existing learning algorithms of them are mainly based on dependency analysis approach, we investigate the feasibility of applying evolutionary algorithm called Hybrid Evolutionary Programming (HEP) [66] into them.

Concerning the DNA analysis, it is a challenging and pioneering project the findings of which are very meaningful and valuable to the society. In medical and biochemical research field, the scale of this project is considered large and comprehensive. The whole genome of HBV DNA are extracted and analyzed. In the computer science point of view, the volume of data is too small while the data dimension is so large. Moreover, how to tackle such small data set carefully to ensure the statistical correctness, how to distinguish which genome sites may be meaningful to our analysis, how to reduce the noise (unrelated mutations) of data, and how to choose a suitable classification model, are all challenges to this project. We endeavor to devise a comprehensive framework by giving a closer inspection on the problem.

In addition, inspired by the adaptive elitist-population genetic algorithm (AEGA), the efficient HEP could still further improved. Since

the running time of evolutionary algorithms depends on the population size, we may adopt the dynamic population size concept in AEGA in HEP. However, the search spaces of HEP and AEGA are totally different, so that a feasibility study should be conducted. How to adjust the population size with performance enhancement should also be explored. Finally, applications of new algorithm on medical domain will be studied.

1.3 Contributions

The contributions of this thesis are summarized as follows:

- It investigates the real-life pioneer genome analysis project and design the approach to solve it by various machine learning and data mining models.
- It proposes learning algorithms of BN-augmented Naïve-Bayes classifier and General Bayesian Network classifier based on HEP. The modifications on HEP and introduction of Markov Blanket concept improve the performance of the classifiers.
- It discovers the easy-missed errors in Bayesian network parameter calculation and investigates possible causes and suggest feasible solutions on the problem.
- It applies the Bayesian network classifier on the clinical data of HBV genome project. The results have discovered the interrelationships among the clinical attributes which is very useful to the doctors.
- It introduces a comprehensive framework for DNA sequence analysis targeted on classification. With the proposed data prepro-

cessing, feature selection, classification and evaluation steps, useful information can be obtained.

- It analyzes the HBV DNA in the genome project and discovered a number of important biochemical and medical findings.
- It optimizes the evolutionary HEP by introducing the concept of adjusting the population size adaptively. The new Bayesian network learning algorithm A-HEP speeds up the original algorithm by two times with comparable performance.
- We have published the paper named "A-HEP : Adaptive Hybrid Evolutionary Programming for Learning Bayesian Networks" in the "Genetic and Evolutionary Computation Conference 2004" [42].

1.4 Thesis Organization

This thesis is organized as follows. In the next chapter, we describe the background relating to our work. This includes a brief introduction on medical data mining, DNA analysis and Hepatitis B virus research. The background information about Bayesian network, Bayesian network classifiers and their learning algorithms including HEP are also presented.

In chapter three, the proposed learning algorithms on BN-augmented Naïve-Bayes (BAN) and General Bayesian Network (GBN) are described in detail. They are designed based on the efficient evolutionary Bayesian network learning algorithm - HEP. Next, the new findings on Bayesian network parameter calculation are discussed. At the end of this chapter, the performance of the proposed algorithms are evaluated by benchmark data sets and real-life clinical data sets.

Chapter four concentrates on DNA data analysis which is an important phase in Hepatitis B virus genome project. Since we target on finding genetic markers of HCC from HBV DNA sequences, the complete framework includes the data pre-processing, feature selection, classification and evaluation steps. A detail description on each step are presented with examples and experimental results. Important research findings are also introduced at the end of the chapter.

In chapter five, the optimized version of HEP - A-HEP are proposed. Its optimization strategy is based on dynamic population size controlled by a newly designed increasing routine and decreasing routine. Its speed improvements are illustrated by experiments with comparisons to the original HEP. Since the algorithm involves a number of parameters, the effect of parameter settings are also investigated through experiments.

In the conclusion, the work is summarized with discussion on future directions.

□ **End of chapter.**

Chapter 2

Background

In this chapter, we introduce the background and previous works that are relevant to our research. In Section 2.1, we introduce the emergence and importance of medical data mining. The difficulties encountered for medical domain are stated as well. In our Hepatitis B Virus Genome project, DNA sequence analysis also plays an important part of it. The general information and related works are described in Section 2.2. For a better understanding of our real-life project and research, Section 2.3 provides more information on Hepatitis B Virus and related biochemical findings. On the other hand, Bayesian network is a major data mining model used in this thesis. In Section 2.4, we give a brief overview of it and its learning algorithms. Finally, in Section 2.5, we describe different types of Bayesian network classifiers and their existing learning algorithms.

2.1 Medical Data Mining

The theme of our work is medical data mining. In this section, the general information and related research of this area are presented. Next, the special features of data mining with medical data and difficulties

encountered are addressed.

2.1.1 General Information

Modern hospitals are well equipped with monitoring and other data collection devices which provide relatively inexpensive means to collect and store the data in inter- and intra-hospital information systems. Extensive amounts of data gathered in medical databases require specialized tools for storing and accessing data, for data analysis, and for effective use of data. In particular, the increase in data volume causes great difficulties in extracting useful information for decision support. Traditional manual data analysis has become inadequate, and methods for efficient computer-based analysis are indispensable. To satisfy this need, medical informatics may use the technologies developed in the new interdisciplinary field of knowledge discovery in databases (KDD), encompassing statistical, pattern recognition, machine learning, and visualization tools to support the analysis of data and the discovery of regularities that are encoded within the data. KDD typically consists of the following steps: understanding the domain, forming the data set and cleansing the data, extracting regularities hidden in the data and formulating knowledge in the form of patterns or rules (this step in the overall KDD process is usually referred to as data mining (DM)), post-processing of discovered knowledge, and exploiting the results [41]. Under this situation, various data mining techniques, together with different models of knowledge representations, are proposed for building medical diagnosis and prediction systems [35].

2.1.2 Related Research

Due to the use of information technology in medical care, data collected from clinical processes can be used for discovering useful patterns. These patterns can be analyzed automatically or by medical professionals in order to develop better strategies to improve the quality of medical treatment. There are plenty of examples, including discovering temporal-state transition in Hemodialysis [47], temporal pattern discovery in course-of-disease data (HIV) [34], knowledge discovery in fracture and Scoliosis database [65] [51], improving diagnosis of ischaemic heart disease [38], preoperative prediction of malignancy of ovarian tumors [48], multiple classifier system for early melanoma diagnosis [60], using Bayesian network and decision trees in diagnosis of female urinary incontinence [33], evolutionary computing for medical diagnosis [37], etc.

Let us look into some of them in detail. Discovering temporal-state transition in Hemodialysis is a research done by two Taiwanese researchers. They adopted the Bayesian network approach to encode the probabilistic relationships among medical treatments and transitions of patient's physiological states in the Hemodialysis process. The background theoretical research is based on another paper [46]. It allow them to find the time dependency patterns in the clinical pathway. The second medical data mining example is temporal pattern discovery in course-of-disease data. In this paper, authors did not apply the Bayesian network for data mining, but they clearly described the process of knowledge discovery in database (KDD). Their target is to discover patterns in Human Immunodeficiency Virus (HIV) database. In Hong Kong, researchers have applied evolutionary algorithms to discover knowledge from medical databases successfully. They used

MDLEP and genetic algorithm to learn the Bayesian network structure for the fracture database and Scoliosis database. Genetic algorithm has been used on the learning of discretization policies on variables [51]. Obviously, applying various data mining techniques on medical domain to discover knowledge in database and/or to develop a decision support system has become a trend. Bayesian network is also a popular choice for knowledge representation.

In recent years, Bayesian networks (BNs) have emerged as one of the most successful tools for medical diagnostics and many have been deployed in real medical environments or implemented in off-the-shelf diagnostic software [55]. BN is widely used because of its ability to encode the probabilistic relationships among variables, and efficiency and flexibility in inference. In certain domains such as medicine, planning and control, and industrial environments, the incorporation of temporal reasoning is crucial. Therefore, different variations of BNs are developed in order to represent the causal and temporal relationships among events or variables, like the above Hemodialysis and Course-of-Disease examples.

In our work, Bayesian network and its classifiers are chosen as the knowledge representation and machine learning models to solve our medical problem.

2.1.3 Characteristics and Difficulties Encountered

Data mining in realistic medical domain has a number of characteristics and also faces some common difficulties. They are briefly described in the following paragraphs.

Let us start from the uniqueness of medical data mining. Krzysztof J.Cios et al. proposed the special features of data mining with medi-

cal data [15]. In their paper, they pointed out that medical data are privacy-sensitive. We should collect them in an ethical and legal way, and administrate them in a secure way. The aim of collection should be primarily directed to patient-care activity but not solely used for research resource. Data from medical sources are sometimes voluminous and with different structures and quality. In this case, the physician's interpretations are essential.

Building classification and prediction models are the common purpose of doing medical data mining. Typically, the goodness of the classification models depends on the accuracy. For medical applications, the sensitivity and specificity are also important for measuring the errors.

In real medical domains, the problems of insufficient data and data with missing values are very common. It increases the difficulty of finding an accurate model. There are a number of researchers working on this. Here we focus on constructing Bayesian network from missing and inadequate data. X. Wu et el. did research on the learning of Bayesian network topologies using the algorithm they developed [67]. They explained why normal statistical models cannot be used, and applied their learning method on an example - stroke. They constructed a causal model with the help of an expert clinician. On the other hand, Nikovski also did similar research on this field [55]. He suggested the way to construct BN from incomplete and partial correct statistics. His key point was to introduce the domain dependent constraints. It is a similar idea to using expert knowledge. Another interesting paper is from Bellazzi and Riva in 1998 [57]. They investigated the way to deal with longitudinal data. One example of longitudinal data is the continuous assessments on the patient's clinical conditions. A diabetes

data set is used in its experiments.

Medical data mining becomes more and more important in the research field and clinical situation. The results of studies are expected to be beneficial to the society.

2.2 DNA Sequence Analysis

DNA sequence analysis is a wide research area in biological and medical fields that becomes more and more important. There are plenty of research on DNA sequence analysis. They belong to the field of Bioinformatics. Bertone's paper gives a general picture of this field [4]. In the paper, the machine learning for analyzing genome-wide expression profiles and proteomics data sets is described. Current and future research directions are introduced.

DNA is the acronym for Deoxyribo-Nucleic Acid. It is the basic hereditary material in all cells and contains all the information necessary to make proteins. The structure of a DNA is composed of two complementary nucleotide strands aligned in a double-helix form. DNA is a linear polymer that is made up of nucleotide units. The nucleotide unit consists of a base, a deoxyribose sugar, and a phosphate. There are four types of bases: adenine (A), thymine (T), guanine (G), and cytosine (C). In normal DNA, the bases form pairs: A to T and G to C. This is called complementarity.

In an organism, the order of amino acids in a protein produced is defined by the DNA in the cells. The order of Amino acids can affect the functions of the protein, and thus the development of an organism. Therefore, any mutations in the DNA of an organism may cause damages or benefits to its corresponding species.

Our Hepatitis B Virus Genome project introduced in the previous

section aims on finding genetic markers of HCC. This is an project working on DNA sequence analysis. We define the genetic markers as those nucleic mutations or characteristics of HBV DNA which are related to the HCC occurrence. Biochemical and medical knowledge are essential for this analysis.

2.3 Hepatitis B Virus

Hepatitis B is caused by the hepatitis B virus. The virus is very common in China, Asia, Africa and the Middle east. It is estimated that there are over 350 million hepatitis B carriers worldwide which represents 5% of the worlds population and it is estimated that 10 to 30 million people become infected with the virus each year. Hepatitis B virus (HBV) is transmitted by the exchange of body fluids e.g. Blood, Semen, Breast Milk and in some circumstances saliva. It is possible to be infected with the HBV and experience no illness or symptoms whatsoever. Commonest is an acute attack of hepatitis during which one may have the Hepatitis B symptoms. In some cases hepatitis B can be fatal, in the elderly. Around 90% of people infected with hepatitis B recover completely and become immune to the virus. Blood tests will show antibodies to hepatitis B (HBeAg) indicating you have had hepatitis B but are now immune and will not get hepatitis B again. However 10% of people infected with hepatitis B develop chronic infection, may have ongoing symptoms and they continue to be infectious for a variable length of time. Chronic infection is defined as having hepatitis B present for 6 months or more. People with a chronic hepatitis infection are at risk of liver damage and around 20-30% of these cases progress to cirrhosis [25][26].

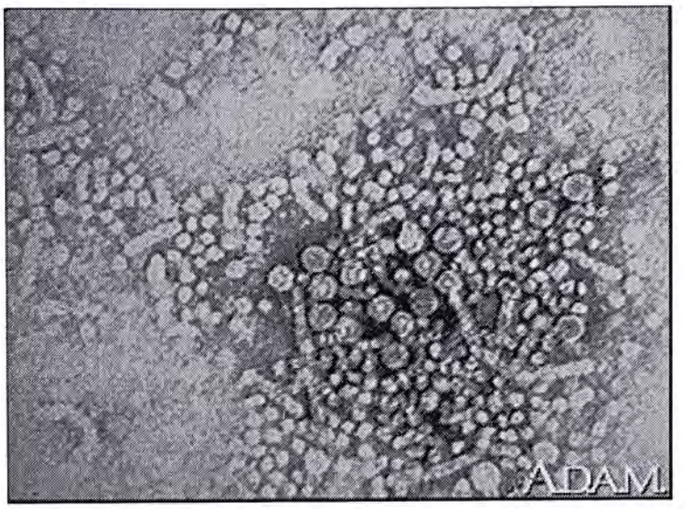


Figure 2.1: Hepatitis B virus under microscope. Copyright is owned by A.D.A.M. Inc[28]

2.3.1 Virus Characteristics

Hepatitis B is a DNA virus of the hepadnaviridae family of viruses. It replicates within infected liver cells (hepatocytes). The infectious particle consists of an inner core plus an outer surface coat. In real life the virus is a spherical particle with a diameter of 42nm (1nm = 0.000000001 metres), as shown in Fig. 2.2. Its outer shell (or envelope) composes of several proteins known collectively as HBs or surface proteins. This outer shell is frequently referred to as the surface coat. The outer surface coat surrounds an inner protein shell which is composed of HBc proteins. This inner shell is referred to as the core particle or capsid. Surrounded by the core particle, there are the viral DNA and the enzyme DNA Polymerase [25].

There are various detectable clues on the infection of HBV, including Hepatitis B DNA (HBV DNA), Hepatitis B DNA polymerase (HBV DNAp), Hepatitis B Core protein (HBcAg), Hepatitis B Surface antigen (HBsAg), HBe Protein (HBeAg or 'e'antigen) and HBx Protein.

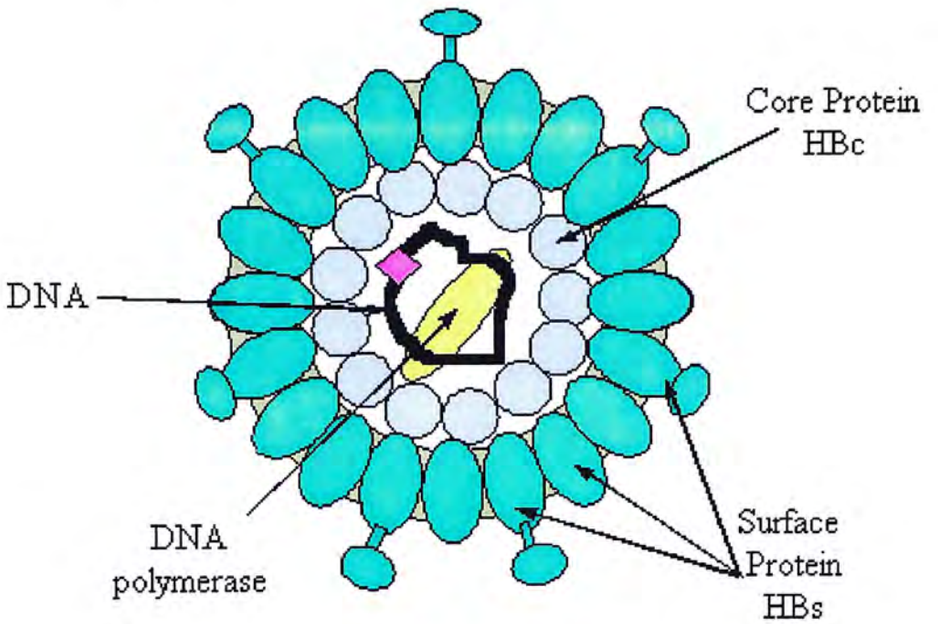


Figure 2.2: Hepatitis B virus structure [25].

Some of these components enter the blood stream and cause detectable changes, for example, HBV DNA and HBeAg. Blood tests can be done to check the state of infections of Hepatitis B virus.

HBV is the smallest DNA virus infecting humans. Its genome contains one strand with about 3200 nucleotides that is complementary to a shorter strand with 1700-2800 nucleotides. The two strands have cohesive ends over a stretch of about 200 nucleotides, which enable a circle to be formed, resulting in a unique, circular double-stranded genome with a single stranded gap of variable [70]. The HBV genome encodes proteins that constitute the external viral envelope and the viral capsid by its gene pre-S1, pre-S2, S and C. The small size of the DNA genome limits the number of proteins that can be encoded. However, the HBV employs all three reading frames and overlaps them to encode four proteins. About half of the genome codes for two proteins at one time, using different reading frames. Regulatory signals are

also included in protein encoding genes. Therefore, the virus makes economical use of its genome. It also introduce great difficulties for analyze the genome in nucleotide-level.

2.3.2 Important Findings on the Virus

The studies on HBV are an active research area in both medical and biochemistry fields, since the infection of Hepatitis is one the major health problems in the world. Here, the related discoveries are introduced briefly.

Concerning the progression to cirrhosis, it is proved that the double promoter mutation, A1762T/G1764A is an important clue. The G-to-A change at nucleotide 1896 (G1896A) which creates a stop codon at codon 28 also confirmed in relationship with the progression [70]. In our later study, we also discovery these benchmark findings from our genome data.

Hepatitis B virus (HBV) has been classified into seven genotypes (A to G) based on a nucleotide divergence within the complete genome of greater than 8%. Recent research shows that genotypes are related to the degree of the liver disease, eruption of virus gene mutation, and drug effectiveness. In Asia-Pacific region, the most common genotypes are genotypes B and C. The HBV from our patients are all in this two genotypes. Since there are genotypical differences, we separate the samples according to their genotype in our study.

2.4 Bayesian Network and its Classifiers

Bayesian network (BN) is a popular knowledge representation for machine learning and data mining. Owing to its ability of representing causality and uncertainty, it is widely used in various domains. With

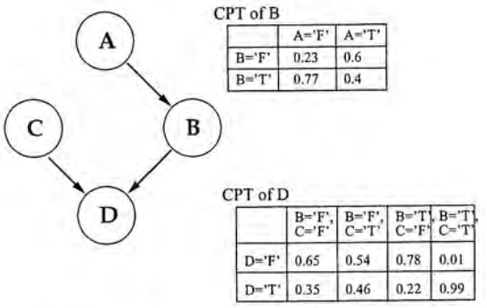


Figure 2.3: A Bayesian network example.

the use of Bayesian networks, many medical and operational diagnostic and prediction systems can be developed. Bayesian network classifiers are models of using BN for classification. There are over ten different models while Naïve-Bayes and TAN are the most popular ones. Research on learning algorithm of BN classifiers is in progress with a view of improving their performance. On the other hand, evolutionary computation is an active research area in soft computing. It is especially efficient to find optimal or nearly optimal solution from large search-space. This section gives an introduction on them.

2.4.1 Formal Definition

A Bayesian network, G, has a directed acyclic graph (DAG) structure. As shown in Figure 2.3, each node in the graph corresponds to a discrete random variable in the domain. An edge, $X \leftarrow Y$, on the graph, describes a parent and child relation in which X is the child and Y is the parent. All parents of X constitute the parent set of X which is denoted by Π_X. In addition to the graph, each node has a conditional probability table (CPT) specifying the probability of each possible state of the node given each possible combination of states of its parents. If a node contains no parent, the table gives the marginal probabilities of the node [56].

Since Bayesian networks are founded on the idea of conditional independence, it is necessary to give a brief description here. Let U be the set of variables in the domain and P be the joint probability distribution of U. Following Pearl's notation [56], a conditional independence (CI) relation is denoted by $I(X, Z, Y)$ where X, Y, and Z are disjoint subsets of variables in U. Such notation says that X and Y are conditionally independent given the *conditioning set*, Z. Formally, a CI relation is defined as [56]:

$$P(x \mid y, z) = P(x \mid z) \quad \text{whenever} \quad P(y, z) > 0, \tag{2.1}$$

where x, y, and z are any value assignments to the set of variables X, Y, and Z respectively. A CI relation is characterized by its *order*, which is simply the number of variables in the conditioning set Z.

By definition, a Bayesian network encodes the joint probability distribution of the domain variables, $U = \{N_1, \ldots, N_n\}$:

$$P(N_1, \ldots, N_n) = \prod_i P(N_i \mid \Pi_{N_i}). \tag{2.2}$$

2.4.2 Existing Learning Algorithms

Typically, a Bayesian network can be constructed by eliciting knowledge from domain experts. To reduce imprecision due to subjective judgments, many algorithms are designed for learning Bayesian networks from collected data and past observations in the domain.

In the literature of Bayesian network learning, we could roughly divide the works into two categories: the dependency analysis and the score-and-search approaches [7]. Since BN is viewed as a model underlying the dependency, it suggests the use of dependency information for the BN construction. On the other hand, BN can be considered as

encoding a joint probability distribution. As a result, various kinds of score metrics and functions are designed to evaluate the quality of a given network. Therefore, constructing the BN can be formulated as searching the best network structure. Both approaches have respective problems and difficulties to be solved.

Dependency Analysis Approach

Since the BN structure encodes a group of conditional independence relationships among the nodes, according to the concept of d-separation [56]. This suggests learning the BN structure by identifying the conditional independence relationships among the nodes. The dependency relationships are measured by using some kind of conditional independence (CI) test. Cheng et al. applied information theory concept on CI test. [7].

In general, the dependency analysis approach has three typical problems. First, it is difficult to determine whether two nodes are dependent. Examining every possible combinations of the conditioning set requires an exponential number of tests. Second, result from CI tests may not be reliable especially for high order CI tests. Also, an earlier mistake during the execution of the construction algorithm is consequential [43].

Score-and-search Approach

Recalling that BN encodes the joint distribution of the attributes, we could devise a measure for assessing the goodness of such encoding. Using the score metric, a search algorithm can be used to find a network structure with a good score. In literature, greedy search algorithms were firstly employed for learning the BN structure. At the same time,

computer scientists and statisticians worked on the local structure and the score metric which determine the goodness of the candidate structures [69] [11] [12] [20].

Heckerman et al. compared two learning approaches, and show that the scoring-based methods often have certain advantages over the CI-based methods in terms of modelling a distribution. The common score metrics which the searching algorithms try to optimize include Bayesian Dirichlet score, Kullback-Leibler (KL) entropy scoring function and Minimum Description Length (MDL) [20].

Among the above metrics, MDL is widely used because it is a good tradeoff between the model complexity and model accuracy. The MDL score of a network B given a training set D is defined as follows:

$$MDL(B|D) = \frac{1}{2} \cdot \log N|B| - LL(B|D) \tag{2.3}$$

where N is the number of data in the training set, $|B|$ is the number of parameters in the network, and LL(B|D) is the *log-likelihood* of B given D. The first term simply counts how many bits we need to encode the specific network B, where we stored $\frac{1}{2} \cdot \log N$ bits for each parameter in Θ. The second term measures how many bits are needed for the encoded representation of D. For details of log-likelihood, please refer to the Heckerman's tutorial paper [20]. By minimizing the MDL score, the structure complexity is minimized and log-likelihood (i.e. the accuracy of the structure) is maximized. Limiting the structure complexity is required to prevent the *overfitting* of the training data, while maximizing the log-likelihood can ensure the structure can represent the data properly.

As a property common to other metrics, the MDL metric is node-decomposable and expressed as a summation of the independent eval-

uation on the parent set, Π_{N_i}, of every node N_i in the domain U.

$$MDL(B|D) = \sum_{N_i \in U} MDL(N_i, \Pi_{N_i}) \tag{2.4}$$

However, using the greedy search heuristics with any metric may yield sub-optimal solutions. Like branch-and-bound, exhaustive and systematic searching can find optimal solution. At its worst, the time complexity consumed would be exponential. These drawbacks can be reduced by using evolutionary algorithms.

Existing evolutionary algorithms for learning Bayesian networks is introduced in next section. In addition, a number of researches have been conducted on related fields which include incomplete data [52], dynamic Bayesian network [58], node topologies, etc.[5][13]

2.4.3 Evolutionary Algorithms and Hybrid EP (HEP)

Bayesian network structure learning can be done by two approaches: score-and-search and dependency analysis. Finding the best structure of BN which can perfectly represent the interrelationship among attributes are proved to be NP-hard [17]. The search space is a very large, multi-dimensional and multi-modal landscape, so that *evolutionary algorithms (EA)* are good solutions to this problem.

Evolutionary Algorithm

Evolutionary computation is a general stochastic search methodology. The principal idea derives from natural evolution mechanisms suggested by Charles Darwin. Since the evolutionary computation is very powerful, it is often applied to solve large-scale optimization problems in different area. Although it is a stochastic computation, its performance is always reasonable in many global optimization problems.

In general, there are four typical categories of EA: Genetic Algorithm (GA), Genetic Programming (GP), Evolutionary Programming (EP) and Evolutionary Strategies (ES). They share the same concept - group search with guidance.

A group of candidate solutions are randomly generated as the initial *population*. Each of the candidate solution is evaluated by a *fitness function* which can determine the quality of the solution. For each iteration (*generation*), some individuals are selected to reproduce *offsprings* by some genetic *operators*, such as crossover and mutation. The new offspring are evaluated by the fitness function, then the better offspring replace the less fit candidates in the old population.

The variation of genetic composition in each generation can be regarded as exploration of search-space. Selection comes into play where the weaker ones will eliminated while the stronger ones will have a higher chance to survive into the next generation. Only the better ones will survive, it is expected that a global, or near optimal solution can be obtained. Unlike the greedy searching methods, stochastic searching, as EA, can avoid trapping in local optimal solutions.

Existing EA Bayesian Network Learning Algorithms

Previous researches on Bayesian network learning by evolutionary algorithms were mainly conducted in two directions: Genetic Algorithm (GA) and Evolutionary Programming (EP) approaches.

Larrañaga et al. used GA for structure learning. They represented BNs as connectivity matrix and encoded them in chromosomes. Fitness function is the Bayesian score. As genetic operators could create illegal structures, cycle repairing operator was introduced as a response. For more details, please refer to [39].

Wong et al. [64] used EP to tackle the search problem. Their algorithm is called MDLEP, as they use the Minimum Description Length (MDL) as the fitness function. The mutation operators that they used include simple mutation, reversion mutation, move mutation and knowledge-guided mutation. The last operator is similar to single mutation except that an edge is selected by comparing the corresponding MDL score of the connecting nodes. Heaviest edges tend to be removed and lightest edges tend to be added. For more details, please refer to [64].

Owing to the inefficiency of GA mutations and slow convergence of MDLEP, Wong et al. continued their work and introduced a HEP for BN structure learning [66] described below.

Hybrid EP (HEP)

As mentioned before, researchers treat the network learning problem by two very different approaches. They are the dependency analysis and the search-and-scoring approaches. Both approaches have their own drawbacks. Hybrid EP (HEP), an extension of MDLEP [64], was designed to incorporate the dependency information into the searching process. The combination of the two approaches achieves better efficiency and improves the solution quality with a smaller number of generations [66].

The evolutionary part of HEP is similar to MDLEP, except that HEP has an additional CI test Phase immediately before the EP (Evolutionary Programming) Search Phase. For every pair of nodes, the order-0 and order-1 CI tests are used to find the *p-value* which indicates the dependency level between them. The search space is refined in each generation by checking the *alpha value*, the dependency threshold,

against the p-value matrix.

Mutation operators used in HEP include simple mutation, reversion mutation, move mutation and knowledge-guided mutation. The last operator is similar to single mutation except that an edge is selected by comparing the corresponding MDL score of the connecting nodes. Edges with larger MDL scores tend to be removed while edges with smaller MDL scores tend to be added. In the HEP framework, a new operator *merge* is also introduced for better evolution. Taking a parent network G_a and another network G_b as input, the merge operator attempts to produce a better network by modifying G_a with G_b. If no modification can be done, G_a is returned. The use of merge operator has proved to improve the effectiveness and the efficiency of HEP, which is outlined in Algorithm 1.

2.4.4 Bayesian Network Classifiers

Knowledge discovery from database constitutes an important part of computer science technology. *Classification* is one of the problems we face. Many tasks, including fault diagnosis, pattern recognition and forecasting can all be viewed as classification. By definition, classification is the task to identify the class label for instances, while each of them is described by a set of attributes.

The learning of accurate classifiers has been an active research area in the past two decades. Many representation models, such as decision trees, neural networks and association rules, have been used for classification.

Since the Bayesian network (BN) has been formally defined by Pearl [56], it is widely used as a knowledge representation model because of its powerful casual representation with uncertainty. Related research

Algorithm 1 Algorithm of HEP

CI Test Phase

for each pair of nodes (X, Y) **do**
Perform order-0 and all order-1 CI tests;
Store the highest p-vale in the matrix P_v;
end for

Evolutionary Programming Search Space

Set t, the generation count, to 0;
Initialize and evaluate the population with size m;
for each individual G_i in the population $Pop(t)$ **do**
Initialize the α value randomly;
Refine the search space by checking the α value against the P_v matrix;
Create a DAG randomly in the reduced search space;
end for
Each DAG in the population is evaluated using the MDL metric;
while t is less than the maximum number of generations **do**
Randomly select $m/2$ individuals from $Pop(t)$, the rest are marked NS;
for each of the selected ones **do**
Merge with a random pick from the dumped half in $Pop'(t-1)$;
If merge does not produce a new structure, mark the individual with NS;
Otherwise, regard the new structure as an offspring;
end for
for each individual marked NS **do**
Produce an offspring by cloning;
Alter the α value of the offspring by a possible increment or decrement of Δ_α;
Refine the search space by checking the α value against the P_v matrix;
Change the structure by performing a number of mutation operations;
end for
The DAGs in $Pop(t)$ and all new offspring are stored in the intermediate population $Pop'(t)$ with size $2 * m$;
Conduct pairwise competitions over all DAGs in $Pop'(t)$. For each G_i in the population, its fitness is compared against q individuals. The score of G_i is the number of individuals (out of q) that are worse than G_i;
Store the m highest score individuals from $Pop'(t)$ with ties broken randomly in $Pop(t+1)$;
Increment t by 1;
end while
Return the final structure with the lowest MDL score in any generation of a run.

in its applications includes fault diagnosis [19] and management [58], medical database [65], document classification [63], etc.

BN can also be used as a classifier that it gives the *posterior probability distribution* of the class node C given the values of other attributes $A_1, A_2, \ldots, A_n$. A major advantage of BN classifiers over other types of predictive models, such as neural networks, is that the BN structure represents the inter-relationships among the data set attributes. Human experts can easily understand the network structures and where necessary modify them to obtain better predictive models. Therefore, a series of BN classifiers are designed for classification [6] [7] [18].

Among the Bayesian Network Classifiers, Naïve-Bayes and Tree-Augmented Naïve-Bayes (TAN) are the simpler ones. However, their performance is limited by their restricted structure. For the unrestricted models, their learning involves high complexity and computation cost. Recently, researchers start designing and improving their learning algorithms. Their approaches are CI-based algorithms which mainly base on dependency tests [8] [9]. Score-and-search is another approach for learning BNs, and the hybrid approach using evolutionary programming (HEP) is one of the most efficient learning algorithms [66]. It is a score-based searching algorithm, and has adopted CI tests information into the approach, so that it can overcome the existing defects of score-based approach and generate good classifiers.

By applying HEP learning algorithm, two classification models - BN-augmented Naïve Bayes and General Bayesian Network classifier can be obtained. In our work, several modifications are applied on HEP in the learning process for improvement. A series of experiments are conducted to evaluate the performance of these models which show that they have comparable performance as that of existing models with

some improvements.

Bayesian Network Classifiers have been applied to many real life domains where the medical field is one of them. The Hepatitis B Virus Genome Project conducted by researchers in CUHK is currently working on data analysis on its HBV DNA genome and clinical data. They are valuable medical data sets for classification, especially the DNA genome data which are rare in medical and biochemical research fields. Partial analysis are accomplished by using BN classifiers in Chapter 3.

In this section, mathematical background of the Bayesian network classifiers are presented. An introduction of some common types of Bayesian network classifiers is given, followed by their learning algorithms.

Mathematical background

Bayesian network classifiers are one of the Bayesian classifier which follows the *Bayes decision rule*. The Bayes decision rule estimates the conditional probability of the class variable for a given instance, and returns the class which yields the greatest value. Let an instance $I = a_1, \ldots, a_n$ is assigned to class c_i, if

$$P(C = c_i|I) > P(C = c_j|I) \quad \text{for all} \quad j \neq i \tag{2.5}$$

By Bayes rule, the class posterior probability could be expressed as,

$$P(C|A_1, \ldots, A_n) = \frac{P(A_1, \ldots, A_n|C)P(C)}{\sum P(A_1, \ldots, A_n|C)P(C)} \tag{2.6}$$

Since the denominator in equation 2.6 is the same for every $P(C|A_1, \ldots, A_n)$, the decision function can be rewritten as,

$$P(I|C = c_i)P(C = c_i) \geq P(I|C = c_j)P(C = c_j) \quad \text{for all} \quad j \neq i \quad (2.7)$$

However, the theoretically sound idea creates a difficulty. Normally, the training set is not large enough to store the entire distribution. Therefore, it is impossible to learn the true distribution from the training data. Thus, various assumption is used to approximate the estimation of the true distribution [43].

Since Bayesian networks can be used to represent a joint probability distribution, we can apply them to approximate the estimation of $P(A_1, \ldots, A_n, C)$. For each instance, the predicted class c_p is the class that gives the greatest value in $P(I|C = c_i)P(C = c_i)$.

Naïve-Bayes

A Naïve Bayes is a simple structure that has the class node as the parent node of all other nodes, as shown in Fig. 2.4. No other connections are allowed in a Naïve-Bayes structure.

An independence assumption among the attributes is made resulting in its simple structure - every attribute has the class node as its only parent. Such independence assumption enables the likelihood probability be represented as a product of $P(A_i|C)$:

$$P(A_1, \ldots, A_n|C) = \prod P(A_i|C) \quad (2.8)$$

Unlike other classifiers, it is easy to construct Naïve-Bayes, as the structure is given a priori. Although its structure is simple, it can surprisingly outperform some other more sophisticated classifiers over

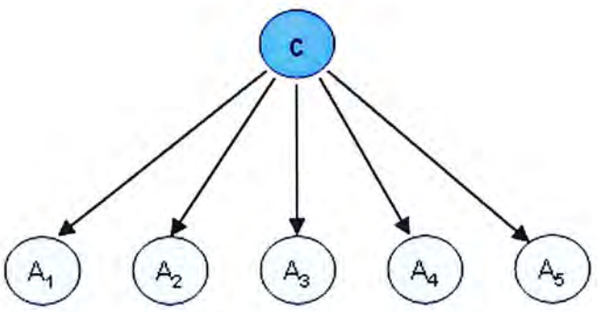


Figure 2.4: Naïve-Bayes BN classifier

a lot of data sets. However, the independence assumption is rarely hold for real world problems.

In recent years, a lot of effort has focussed on improving Naïve-Bayes classifier, following two general approaches: selecting feature subset and relaxing independence assumptions [18]. The variations of Naïve-Bayes are introduced in the following parts.

Tree-augmented Naïve-Bayes classifier (TAN)

TAN classifier extends Naïve-Bayes by allowing the attributes to form a tree, as shown in Fig. 2.5, so that the independence assumption of attributes is relaxed. TAN is a compromise between accuracy and simplicity. It is defined by the following conditions:

- Each attribute has the class attribute as its parent
- Attributes may have at most one other attribute as its parent

The latter condition means that if there is an arc from A_i to A_j, the two attributes are not independent given the class. Learning the

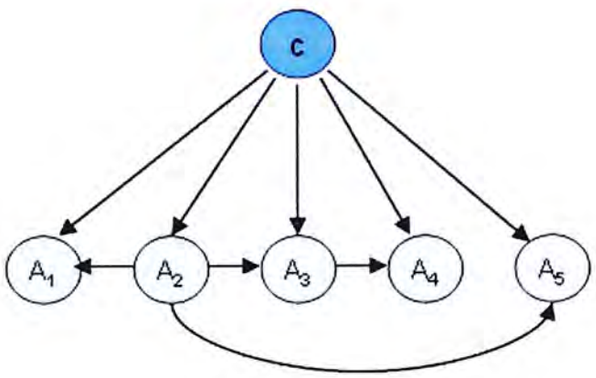


Figure 2.5: Tree-augmented Naïve Bayes(TAN)

tree-structural interrelationships among attributes in TAN is studied extensively.

BA Augmented Naïve Bayes classifier (BAN)

BAN classifier is an variation of TAN which has looser constraint on the dependence among attributes. Unlike the TAN, attributes can form an arbitrary directed acyclic graph rather than just a tree. In Fig. 2.6, the node A_4 has the class node C, A_1 and A_3 as its parents.

General Bayesian Network (GBN)

GBN is an unrestricted BN classifier which can be regarded as a normal Bayesian network. It treats the class node as an ordinary node when the structure is learned, so that it is not necessary to be the parent of all the attributes. The performance of this classifier is not as good as expected in preliminary research, because it highly depends on the performance of the BN structure learning algorithm.

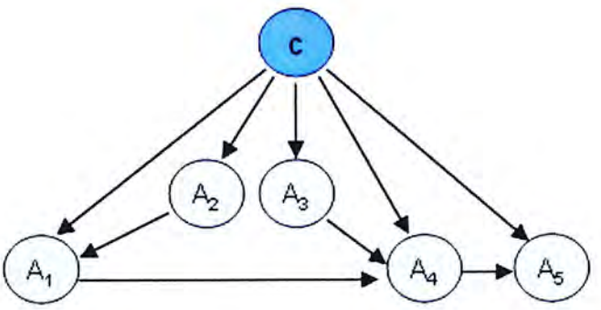


Figure 2.6: BN-augmented Naïve Bayes(BAN)

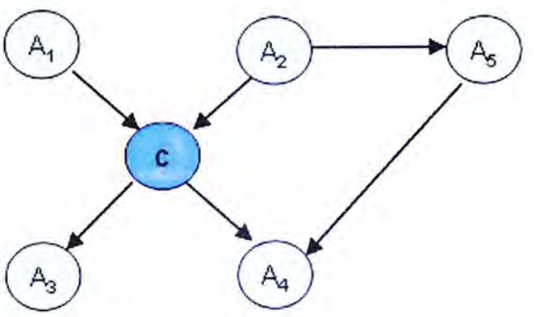


Figure 2.7: General Bayesian Network Classifier (GBN)

2.4.5 Learning Algorithms for BN Classifiers

Since the performance of Bayesian network classifiers are comparable to other popular classifiers, there are a lot of research attempting to improve on their learning performance. In this section, general information and the learning algorithms of typical models of BN classifiers are introduced.

Naïve-Bayes and TAN

Naïve-Bayes has outstanding performance for classification with a simple structure. It is easy to construct, as the structure is given a priori. Hence, no structure learning procedure is required.

Learning Tree-augmented Naïve-Bayes Classifier (TAN) is relative easy and efficient by using dependency analysis and tree-learning algorithms in graph theory. Friedman et al. developed one which returns the maximum likelihood estimate of tree-augmented structures [18]. Here is the TAN learning procedure:

Algorithm 2 Algorithm of TAN learning

Compute the conditional mutual information $I_P(A_i, A_j|C)$ between each pair of attributes, $i \neq j$.

Build a complete undirected graph in which the vertices are the attributes $A_1, \ldots, A_n$. Annotate the weight of an edge connecting A_i to A_j by $I_P(A_i, A_j|C)$.

Build a maximum weighted spanning tree.

Transform the resulting undirected tree to a directed one by choosing a root variable and setting the direction of all edges to be outward from it.

Construct the classifier network by adding the class node, label by C, and adding an edge from C to each A_i.

Calculating the weights of the edges has complexity of $O(n^2N)$, and constructing the maximum weighted spanning tree has complexity of $O(n^2 \log n)$. Since N is usually larger than $log\ n$, the overall complexity is $O(n^2N)$, which is computationally efficient.

In the literature, some TAN learning algorithms are also developed to reduce the space complexity. Lucas [49] developed the Forest-augmented BN classifier (FAN) which is similar to TAN, except that the attributes are allowed to form a k-edge forest rather than a single

spanning tree. The learning algorithm of FAN is the same as TAN, except the spanning learning part.

Other learning algorithms

BN classifiers with more complex structures are less popular, thus fewer learning algorithms are developed.

Cheng et al. tried to incorporate their BN learning algorithms into a TAN learning algorithm for BAN learning. Their own CI-based BN algorithms, CBL_i use information theory for dependency analysis [6] [7]. CBL_1 is used for case that node ordering is given, and CBL_2 is used for the case that node ordering is unknown. A learning algorithm is also designed for learning General Bayesian Network. Their approaches are presented in Chapter 3. Recently, they introduced a wrapper algorithm for combining multi-net and GBN for classification [9].

There are a number of research work done on BN classifier learning, including Madden's Markov Blanket Bayesian Classifier Algorithm [50], and SuperParent appraoch by Eamonn et al. [36], K. Huang's Semi-Naive Bayesian network classifier [32], etc.

□ **End of chapter.**

Chapter 3

Bayesian Network Classifier for Clinical Data

This chapter focuses on the use of an evolutionary Bayesian network learning algorithm (HEP), on learning Bayesian network classifiers, and its applications on clinical data classification. As discussed in the previous sections, Bayesian networks can be applied on classification in various ways. Models with simpler structures, including Naïve-Bayes and Tree-augmented Naïve Bayes (TAN), are good classifiers with efficient learning algorithms. However, the limitation on independence among attributes is unrealistic. BN classifiers with more complex structure shows better performance, but they require more computation. Existing learning algorithms for BN-augmented Naïve Bayes classifier (BAN) and General BN classifier (GBN) mainly concentrate on dependence analysis with given node topology. Therefore, there are still rooms for improvement on the learning algorithms.

On the other hand, HEP shows good BN-learning performance with good convergence. The incorporation of dependency analysis greatly reduces the network structure searching space. Modifications on it are

proposed here for learning the BAN and GBN.

This chapter is structured as follows. Section 3.1 describes the previous algorithms designed by Cheng et al. and their shortcomings. Sections 3.2 and 3.3 present the proposed learning algorithms for BAN and GBN. In Section 3.4, we describe the possible errors found in parameter calculation in Bayesian network classifier learning. The proposed algorithms are then evaluated and compared with the existing learning algorithms on benchmark and real clinical data in Section 3.5. Finally, a summary of the evaluation is presented.

3.1 Related Work

In this section, the related work on BN-augmented Naïve Bayes classifier (BAN) and General Bayesian Network (GBN) learning algorithms are reviewed. Cheng et al. spent a great effort in this field.

BAN learning

Cheng et al. tried to incorporate their BN learning algorithms into a TAN learning algorithm for BAN learning. Their own CI-based BN algorithms, CBL_i use information theory for dependency analysis [6] [7]. CBL_1 is used for cases where node ordering is given, and CBL_2 is used for the case that node ordering is unknown. For the dependency analysis approach, the number of CI-test is an important concern on efficiency of the algorithm. In Cheng's case, CBL_1 requires $O(n^2)$ mutual information tests , while CBL_2 requires $O(n^5)$ mutual information tests. The BAN learning algorithm is shown as below:

Like the TAN-learning algorithm, this BAN learning algorithm does not require additional mutual information tests, but it requires $O(n^2)$ mutual information tests. However, the complexity is much higher if

Algorithm 3 Algorithm of BAN learning by Cheng et al.

Take the training set and $X \setminus c$ (along with the node ordering) as input

Call a modified CBL_1 algorithm - modified by replacing every mutual information test $I(x_i, x_j|c)$, and replacing every conditional mutual information test $I(x_i, x_j|Z)$ with $I(x_i, x_j|Z + c)$, where $Z \subset X \setminus c$.

Add c as a parent of every x_i where $1 \leq i \leq n$.

Learn the parameters and output the BAN.

the node topology (ordering) is not given.

GBN learning

General Bayesian Network (GBN) is a normal Bayesian network with both a class node and attributes. Learning a GBN can be considered as learning a Bayesian network structure. Cheng et al. proposed constructing GBNs with CBL_i algorithm [8].

Algorithm 4 Algorithm of GBN learning by Cheng et al.

Take the training set S,and feature sets F with node ordering as input

Call BN-structure learning algorithm CBL_i

Find the *Markov Blanket* of the classification node.

Delete all the nodes that are outside the Markov Blanket.

Learn the parameters and output the GBN.

Discussion

As discussed in Section 2.4.2, the structure of a Bayesian network can be learned by two approaches - dependency analysis and search-and-score approach. Cheng's algorithms on BN classifier learning is founded

on the Information Theory based dependency analysis. The authors have proved that the proposed algorithm is efficient for finding a desirable Bayesian Network classifier. Besides, efficient BN learning algorithm using evolutionary approach - HEP is published. The objective of our work is to analyze the performance of HEP on learning BAN and GBN for classification.

S.Y.Lee [43] did something similar in his thesis. He analyzed the performance of multi-net and augmented Bayesian network directly learned by HEP with little modifications. In our proposed approach, HEP is slightly modified at CI-test phase for learning the structure of BAN and GBN for classification.

3.2 Proposed BN-augmented Naïve Bayes Classifier (BAN)

The proposed learning algorithm is based on the state-of-the-art evolutionary Bayesian Network learning algorithm - HEP. In this section, the details of the proposed algorithm and performance evaluation are presented.

3.2.1 Definition

BAN classifier is a variation of Naïve-Bayes which has looser constraint on the dependence among attributes. Unlike the TAN, attributes can form an arbitrary directed acyclic graph rather than just a tree. Fig. 2.6 shows an example of BAN classifier. In the figure, the class node C is the parent of every node. However, the node A_4 is allowed to have A_1 and A_3 as its parents, besides the class node C.

The advantage of BAN structure is that the limitation on attribute

independency is released. It can model a more realistic causal relationship among attributes. However, the search space of suitable structure is greatly increased when compared with that of the learning algorithm of Naïve-Bayes and TAN.

3.2.2 Learning Algorithm with HEP

Cheng's algorithm for learning BAN stipulates that node ordering must be given as input. It is not practical in many cases. In addition, while predefined node ordering reduces the search space, it can also introduce errors. In the related work, Wong et al. did not investigated the use of HEP for learning BAN. Our proposed algorithm is replacing Cheng's mathematical CBL_1 algorithm with Wong's evolutionary HEP.

Let C be the class node and $A_1, \ldots, A_n$ be the set of attributes. Here is the outline of proposed BAN learning algorithm:

Algorithm 5 Proposed algorithm of BAN learning

Use HEP to learn the structure among the attributes $A_1, \ldots, A_n$ (without the class node C)

Add C into the parent set of every attribute A_i.

Learn the parameters (conditional probability table) and output the BAN

3.2.3 Modifications on HEP

HEP is originally designed for learning unrestricted Bayesian network structure, rather than learning BAN classifier. Slight modifications are proposed on the HEP, to obtain a more accurate classifier. In the CI test Phase of HEP, order-0 and order-1 CI tests are performed. The CI test result (p-value) of each pair of nodes is stored in the matrix P_v which are used to reduce the search space.

Recalling the conditional independence assertion $I(X, Z, Y)$ of any two nodes X,Y and a conditioning set Z defined in HEP algorithm is calculated using χ^2 test. The result (p-value) is checked against the cutoff value α. If p-value is greater or equal to α, the hypothesis would not be rejected and $I(X, Z, Y)$ would be taken as valid. Consequently, these two nodes cannot have any edge between them, and vice versa.

For the BAN model, an edge is added from the class node to each attribute node in Step 2 of Algorithm 5. Therefore, the class node C must be an element of conditioning set Z. In our algorithm, the CI-Test Phase of HEP should be changed as follow:

Algorithm 6 Proposed modified CI Test Phase of HEP

For every pair of nodes (A_i, A_j) where A_i, A_j are attribute nodes

Perform order-1 and order-2 CI tests and class node C is an element in the conditioning set Z

Store the highest *p-value* in the matrix P_v.

This change is expected to refine the CI-Test Phase for learning BAN classifier. In the experiment section, the effect of modification is tested by comparing the learning algorithms with and without this change.

3.3 Proposed General Bayesian Network with Markov Blanket (GBN)

In this section, the classifier learned by HEP on the training set is used for classification. Proposed learning algorithm incorporates the concept of Markov Blanket in it.

3.3.1 Definition

Let $A_1, \ldots, A_n$ denotes the set of attributes and let C denotes the class variable. We can apply any Bayesian network learning algorithm on the training set, which consists of $A_1, \ldots, A_n, C$, and use the network returned as a classifier with unrestricted structure for classification. This is what we call General Bayesian Network (GBN). We use the decision function, Equation 2.7, for predicting the class of an instance.

3.3.2 Learning Algorithm with HEP

To construct unrestricted network like GBN, we can use BN learning algorithms. However, the accuracy may not be as good as expected. The BN learning algorithm has to be suitable for constructing classifier.

Cheng's algorithm is efficient when the node ordering is given, but the case is different when node ordering is absent. They tried to use wrapper algorithms to improve their classifier-learning algorithms. Although wrapper algorithms can combine the advantages of different classifiers, the computation effort is doubled or more.

By taking Cheng's algorithm as reference, *Markov Blanket* concept is added into our GBN learning algorithm. Markov boundary of a node in a BN is defined as the subset of nodes that "shields" n from being affected by any node outside the boundary. One of n's Markov boundaries is its Markov Blanket[8]. In general, Markov Blanket of a node n is the union of n's parents, n's children and its children's parents. In Fig. 3.1, the purple nodes are the Markov Blanket of the class node C. The pink nodes are removed in our algorithm. Nodes outside the Markov Blanket can be deleted without affecting the classification accuracy, because they are conditionally independent from the class node when the the value of intermediate node between them are known.

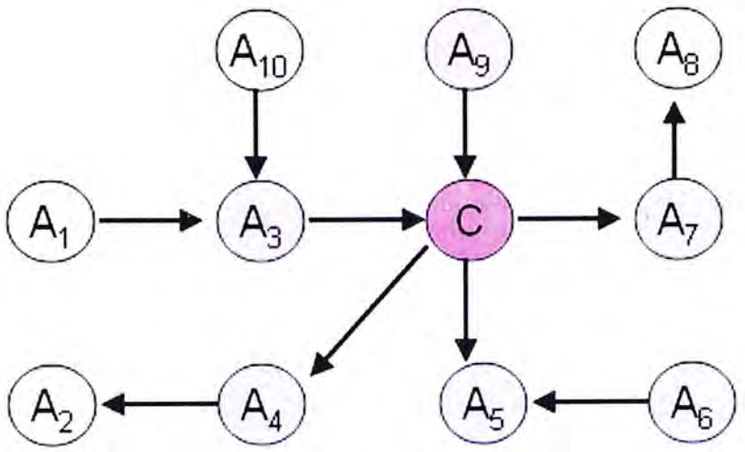


Figure 3.1: The concept of Markov Blanket

Here is the proposed algorithm of GBN using HEP:

Algorithm 7 Proposed algorithm of GBN learning

Use HEP (BN-learning algorithm) to learn the structure.

Find the *Markov Blanket* M of the class node C.

Remove the nodes that outside the Markov Blanket.

Learn the parameters and output the classifier.

With the use of Markov Blanket, the attributes outside the Markov Blanket are removed in the classifier. That means they can be ignored during classification without affecting the accuracy. This simpler structure can highlight the attributes related to the class, and reduce the computation effort.

3.4 Findings on Bayesian Network Parameters Calculation

Learning Bayesian network includes learning its structure and its parameters, i.e. the conditional probability table (CPT) for each node. The second task is relatively easier when the data is complete and abundant. Various algorithms are proposed for computing the parameters of BN when the data are incomplete. For example, EM approach and evolutionary approach.

In our problem, we assume the data are complete, and can reflex the entire distribution. As the result, learning the conditional probability tables becomes a trivial task. After the BN structure is found, we just need to compute the conditional probability of each value of the node, given the values of its parents. In this section, we report the findings on the possible error arising in the parameter calculation, and propose an error handling method.

3.4.1 Situation and Errors

When we run experiments on the evaluation of the proposed GBN learning algorithm, the result of the model with Markov Blanket concept is better than that of the model without it. However, according to the literature, the nodes outside the Markov Blanket of the class node is independent to the class node given the value of the node in-between them. Therefore, the removal of nodes outside the Markov Blanket of the class node should not lead to any accuracy gain. That means the calculated conditional probability tables have some errors, thus the class prediction is wrong. We try to investigate the reasons or causes of these errors.

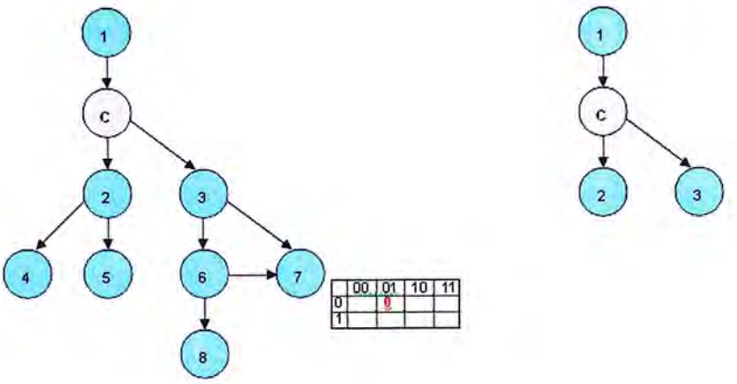


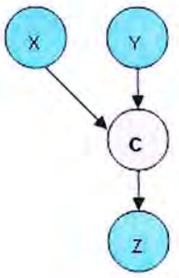
Figure 3.2: Left: Without the removal of the nodes outside the Markov Blanket of the class node. Right : Removed the nodes outside the Markov Blanket of the class node.

After checking the parameters calculation and testing by simple test data sets, the reason of this "pseudo" accuracy improvement was due to the zero entry(entries) in CPT(s). Fig 3.2 is a case of our debugging example. Both the GBN classifiers are learned from the same training set with eight attributes and one class node. For the left GBN, if the nodes that outside the Markov Blanket of the class node C are removed in our algorithm, it becomes the right GBN.

By equation 2.7, the predicted class is the class with the larger joint probability distribution of the testing instance than the other class values. For example, there are two class values c_1 and c_2.

$$P(A_1 = a_1, A_2 = a_2, \ldots, C = c_1) \geq P(A_1 = a_1, A_2 = a_2, \ldots, C = c_2) \tag{3.1}$$

In this case, the predicted class is c_1. Since the joint probability distribution (JPD) of a Bayesian network is encoded as its structure and calculated by equation 2.2, a zero entry in the CPT may yield a zero value of JPD. Referring to Fig. 3.3, attributes X and Y are the par-



(X,Y)	(0,0)	(0,1)	(0,2)	(1,0)	(1,1)	(1,2)
C = 0	2/3	0	3/4	0	2/3	0
C = 1	1/3	1	1/4	0	1/3	1

Figure 3.3: Zero entry in the CPT of GBN classifiers

ents of the class node C. The absent of instances with $(X = 1; Y = 0)$, $(X = 0; Y = 1)$ and $(X = 1; Y = 2)$ in the training set causes the zero entries in the CPT of node C. If there is a testing instance with $(X = 1; Y = 0)$, the calculated JPD for both classes become zero. In the program, the default class $C = 0$ is predicted. We may then get the wrong answer. The case is similar to other zero entries, provided that the training set cannot reflex the entire distribution. One point we need to clarify is that the zero entry may not cause error, but it may be reflexing the real distribution. It depends on the case and the assumptions on the data set made. If the entry of all class values are zero, like the column $(X = 1; Y = 0)$ in the CPT shown in Fig. 3.3, we need to handle it carefully.

Going back to our "pseudo" improvement of classification accuracy, the causes can be summarized in the following points:

1. The training set is not large enough to store the entire distribution, so that the entries in the CPTs may become zero when some value patterns of parents do not exist in the training set. The value pattern of the parents refers to the set of value of all the parents of a particular node.

2. Unskillful programming. The programmer should prevent the

zero entries of CPT and handle the exceptional case, e.g. division by zero.

3.4.2 Proposed Solution

The zero entry in the CPT of a node is due to the absent of its value pattern of parents in the training set. The first solution is to use a larger training data set that can represent the entire distribution. The more training data is used, the more accurate the model can represent the true distribution. We do not aim at avoiding zero entry in CPT, as it may be the true case. One solution is to make assumption on the completeness of the data set. Another solution is modifying the program to return a small number (e.g. 0.0001) instead of zero in the CPT when that value pattern of parents with that class value is not found in the training set. The second solution is more practical for the case of insufficient data.

When zero entries appear for every class value in a given value pattern of parents, no class can be predicted by the equations. One possible solution is to assign each value an equal probability. Here is a statistically sound solution to deal with it:

Assume there are two possible value for the class node. Let one of the entry be p, while another entry be $1 - p$. Since p is unknown, we can assume it as a random variable. Since the probability distribution of p is unknown, we can let it having a uniform probability distribution. That means equal probability is assigned to the value from 0 to 1. In this case, the expected probability of p is $1/2$. In general, for a problem with n class values, the expected probability for each class value should be $1/n$.

In our experiments carried out for evaluation, we use 0.0001 to

replace zero entry equal probability in the second situation as proposed.

3.5 Performance Analysis on Proposed BN Classifier Learning Algorithms

In this section, the proposed learning algorithms on BAN and GBN are evaluated by experiments. In Section 3.5.1, we describe our experiment methodology. In Section 3.5.2, the performance of the classifiers are studied and compared with common Bayesian Network Classifiers - Naïve-Bayes, TAN and FAN. The experiments are run on benchmark data sets. In Section 3.5.3, real clinical data sets from an HBV Genome Project are used in the experiments to illustrate their performance on real-life medical data mining. Finally, the results are summarized in the discussion part.

3.5.1 Experimental Methodology

In the first part of the experiments, we concentrate on the benchmark data sets which can show the average performance of our classifiers. Two types of experiments are carried out. The first type aims at examining the improvement of learning algorithms after modifications on CI-Test Phase for BAN and Markov Blanket for GBN. The second type is for comparing the overall classification performance of different BN classifiers. Benchmark data sets in UCI Machine Learning Repository [27] are used for evaluation. Fig. 3.1 shows the summary of them.

For simplicity, records with missing values are not considered. Those data sets with continuous attributes are preprocessed by MLC++ which is a popular API for Machine Learning experiments [31]. Some of the UCI data sets are obtained from the web site of MLC++ as they are

Data Set	Discrete or Continous	No. of instance	No. of attribute	No. of class	Train	Test
DNA	D	3186	60	3	CV10	
Flare	D	1066	10	3	CV10	
Vehicle	C	846	18	4	CV10	
Vote	D	435	16	2	CV10	
Chess	D	3196	36	2	2130	1066
German	D	1000	20	2	CV10	
Lymophography	D	148	19	4	CV10	
Mushroom	D	8124	22	2	5416	2708

Table 3.1: UCI Data sets used for experiments

in the format ready for preprocessing. We adapt the default entropy discretizor for preprocessing.

In all the experiments, ten-fold cross validation (CV10) are used for running small data sets. For large data sets, we simply use the default training set and testing set for performance evaluation. The experiments are run for ten times for each data set. The average accuracy of each classifier is the percentage of average correct prediction on the testing data of each data set.

In the performance comparison experiments, the Naïve-Bayes, TAN and FAN classifiers are implemented by the jBNC - Bayesian Network Classifier Toolbox [22]. It is a popular Java toolbox used for performance evaluation of Bayesian network classifier, machine learning and data mining applications.

3.5.2 Benchmark Data

Performance on improved learning algorithms

In the learning algorithm of BAN, the CI-Test Phase of HEP are modified from order-0 and order-1 to order-1 and order-2 (with class node

in the conditioning set). Such change is reasonable as the class node is assigned as the parent of every node. Experiments are carried out to show the improvement of such modification. Only the DNA data set is used in this test. We used the default training and testing sets obtained from MLC++ web site. Table 3.2 shows the average accuracy on the CV10 experiments.

Order of CI Test used	Classification Accuracy (%)
Order-0 and Order-1	93.17
Order-1 and Order-2	93.68

Table 3.2: Accuracy improvement by the modification on HEP

This experiment shows that the accuracy is slightly improved. It is because the CI-Test is corrected to take consideration on BAN structure - existing edges between the class node and the attribute nodes. Therefore, such change is essential for the learning the structure of BAN. In the later experiments, improved version of HEP with new CI-Test is used for learning BAN structure. However, the order of CI-Test for learning GBN is unchanged.

In the learning algorithm of GBN, we adopt HEP to learn the structure and cut the Markov Blanket of the class node as the resultant classifier. In the following experiments, the use of Markov Blanket is examined by comparing the classification accuracy of each models. Table 3.3 shows the summarized results on classification of Flare and Vote data sets.

In the experiment results, there are improvements on the GBN classification performance after pruning out the nodes outside the Markov Blanket of the class node. This is the "pseudo"-improvement we described in last section. Although we cannot prove the classification

Dataset	Classification Accuracy(%)	
	Before extracting Markov Blanket	After extracting Markov Blanket
Flare	82.48 ± 0.48	82.62 ± 0.14
Vote	93.73 ± 0.18	94.65 ± 0.18

Table 3.3: Performance of GBN with/without the Markov Blanket extraction

ability of GBN is improved, we can demonstrate the use of Markov Blanket extraction under the existence of zero entry in CPT.

Comparison with existing BNC models

In order to study the performance of BAN and GBN learned by the new algorithms, experiments are carried out to compare the learned models with other common BN classifiers. Table 3.4 shows the classification accuracy of different models on the selected data sets. The classification accuracy and standard deviation of different classifiers are evaluated on the nine data sets.

Experimental results show that the performance of BAN and GBN learned by the HEP algorithm are satisfactory. BAN has better performance on Lymophography, German, and Mushroom data sets, and GBN has outstanding performance on DNA, Lymophography and Mushroom data sets which have larger number of attributes. Referring to the structure of GBN, it is favorable classifier for the data sets with larger number of attributes but not all of them are related to the class node.

3.5.3 Clinical Data

Different Bayesian network classifiers have been evaluated by benchmark data sets in the previous section. In this section, they are ap-

Date set	> 1000 instances	No. of attribute	Naïve-Bayes (%)	TAN(%)	FAN(%)	BAN(%)	GBN(%)
DNA	x	60	95.4 ±1.22	92.33 ±0.77	N/A	95.41 ±1.08	**96.2** ±0.96
Flare	x	10	79.96 ±3.28	**83.15** ±1.99	82.02 ±2.04	82.83 ±3.68	82.64 ±3.19
Vehicle		18	59.23 ±2.49	69.39 ±3.54	**69.74** ±3.50	68.91 ±3.15	60.41 ±3.24
Vote		16	90.06 ±4.15	**94.82** ±1.92	94.07 ±2.04	93.55 ±3.32	94.65 ±3.46
Chess	x	36	80.48 ±2.17	92.4 ±0.81	**93.15** ±0.77	90.95 ±1.92	92.72 ±2.87
German	x	20	75.47 ±4.17	71.56 ±2.47	74.85 ±2.38	**75.50** ±4.47	71.27 ±4.34
Lymophography		19	84.00 ±5.24	82 ±5.49	82 ±5.49	98.43 ±3.88	**99.28** ±2.23
Mushroom	x	22	98.59	99.85	**100.00**	**100.00**	**100.00**
Nursery	x	8	90.44	**93.58**	93.66	90.46	91.06
Average			83.74	86.56	86.18	**88.2**	**87.15**

Table 3.4: A summary of performance of different classifiers.

plied to real-life classification and prediction problems in medical domain. Since the Hepatitis B Virus Genome Project coordinated by researchers in CUHK is currently doing data analysis on its HBV DNA genome data and clinical data, different BN classifiers are used to find genetic and clinical markers of HCC from the HBV DNA genome data and clinical data.

In Feb 2003, clinical database of this project was setup under the supervision of the medical doctors. According to their expert knowledge, thirteen attributes are chosen for preliminary experiments and listed in Table 3.5. Most of the chosen attributes are laboratory test of blood sample. The bold value(s) for each attribute is the abnormal value(s), while the values in the bracket indicate the normal value/range for that attribute.

Attribute Name	Type	Details
Age	5	$< 30/30 - 40/40 - 50/50 - 60/ > 60$
Gender	2	M/F
Hemoglobin	2	**Below**/ Normal (#M:13.2-16.7 # F:11.5-14.3)
White Cell	2	**Below**/ Normal (#4.0-10.8)
Platelet	2	**Below**/ Normal (#140-380)
INR	3	< 1.4 / **1.4-1.8**/ > 1.8
Albumin	3	< 28 / **28-35** / > 35
Bilirubin	3	< 35 / **35-50** / > 50
HBeAg	3	+, -, Eq
HBeAb(Anti-HBe)	3	+, -, Eq
ALP	5	<1x / **1x-2x** / **2x-5x** / **5x-10x** / >**10x** (#100)
ALT	5	<1x / **1x-2x** / **2x-5x** / **5x-10x** / >**10x** (#58)
AFP	3	<20 / **20-50** / **50-100** / **100 500** / >**500**

Table 3.5: Clinical Attributes for HBV genome experiments

Data Preparation

Since most of the attributes are numerical laboratory test results, they are either numeric or continuous, thus discretization is required. We adopt the conventional discretization ranges used by clinicians, please refer to Table 3.5.

As for the other medical data sets, there are also missing values in the experiment data. They belong to attribute HBeAg and HBeAb. In other data mining methodology, various techniques are used for predicting the missing values. However, it is not applicable in these two attributes. There are two major reasons. First, the data set is too small in our project. The second reason is nearly 50% of values for these 2 attribute are missing. It is not statistical significant for learning the missing values. Therefore, we assign the value N/A for the missed value.

Experiments

There are 100 Control patients and 100 liver cancer (HCC) patients in our study. All of them are included in our project study. The goal of this test is to build up a classification model which can correctly classify the testing data into Control class or HCC class. BAN and GBN models are tested as they can discover the interrelation between attributes which is useful for clinicians. The experiments is repeated 5 times 10-fold cross-validation for evaluation.

Preliminary Results

For the GBN experiments, we obtain several Bayesian network structures that representing the inter-attribute relationship in the training set. Among these learned structures, only six attributes are included as nodes of the networks. Referring to the algorithm of learning GBN, nodes outside the Markov Blanket of class node are removed from the network. Therefore, we can prove that these six attributes are highly related for class prediction. They are Hemoglobin, Albumin, HBeAg, AFP and ALT. Figure 3.4 shows one of the learned structure. The following table shows the classification performance of BAN and GBN models.

	BAN	GBN
Classification Accuracy (%)	92.6 ± 5.82	92.10 ± 5.72

Table 3.6: Performance of BAN and GBN in HBV genome experiments

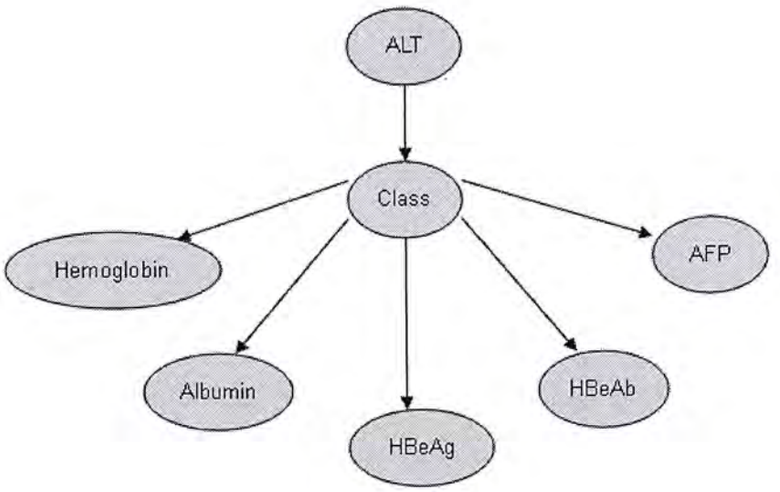


Figure 3.4: One of the GBN result in the experiments

Discussion

High classification accuracy can be obtained from the above experiments by GBN, shown in Table 3.6. However, this result is not clinically useful for doctors after they review it for certain reasons. Referring to medical literature, age and gender are important factors related to developing HCC. In our study, the age and gender of control and HCC patients are matched statistically in patient selection process for equalizing the variance of virus mutation due to age. Therefore, age and gender factors are not significant in our experiments. For those six attributes included in the network structure, AFP is a standard test used for HCC prediction with around 70% correctness. The clinicians suggest to us to exclude AFP from our experiment if we want to check the classification power of other related clinical attributes. More experiments can be run using with the data set AFP excluded. The clinical result is going to used with genetic data to get a more accuracy classification model.

Owing the the bias patient selection, the clinical experiment discovery is not useful in clinical situation now. However, the use of BN classifiers can be demonstrated to be effective for classification on real-life medical data.

3.5.4 Discussion

On average, BAN and GBN have better performance than other popular BN classifier models, especially on data sets with larger no. of attributes. Referring to the structure of GBN, it is favorable for data set with large number of attributes and not all of them are related to the class node. For other BN classifier models, the compulsory edges between class node and attributes sometimes yield a worse classification performance. Although BAN and GBN are not the best classifiers for each data sets, they are classifiers with the highest average accuracy in those nine benchmark data sets. For the clinical application of BAN and GBN classifiers, their performance is evaluated and presented in Section 3.5.3.

At the same time, the future research direction has been suggested for further improvements. The core of learning algorithms of BAN and GBN are HEP which using MDL as score metric. As Friedman showed that using MDL (or other nonspecialized scoring functions) for learning unrestricted Bayesian networks may result in poor classifier [18]. Therefore, using evolutionary algorithm for BN classifier structure learning with MDL score metric as fitness function may yield a poor classifier. Hence, further research can concentrate on improving the use of MDL for learning a better BN classifier.

3.6 Summary

In this chapter, the learning algorithms for BN-augmented Naïve-Bayes classifier (BAN) and General Bayesian Network classifier (GBN) have been proposed. These learning algorithms are developed based on the Hybrid EP (HEP) which is a state-of-the-art Bayesian network learning algorithm. With some modifications on HEP and the introduction of the Markov Blanket concept, the proposed learning algorithms are effective to learn a satisfactory structure of classifier.

The classifiers learned by proposed algorithms have been analyzed by a comprehensive set of experiments on UCI benchmark data sets, as well as a real-life clinical data set. The experimental results show that both models are satisfactory for classification, and can discover the inter-relationship between the attributes. The BAN and GBN classifiers are especially useful for data sets with larger number of attributes.

In addition, an easily-missed error on conditional probability table calculation are reported in this chapter. This error is mainly caused by the zero entry of CPT. Feasible solutions are proposed and used in our experiments.

For the future direction, further investigation on the fitness evaluation metric for HEP or other evolutionary Bayesian network learning algorithm can be tried.

□ **End of chapter.**

Chapter 4

Classification in DNA Analysis

Our work is based on the HBV Genome Project and investigates the use of different machine learning and data mining models in the medical domain. The details of this project are described in the previous sections. The project is mainly divided into clinical data mining and DNA analysis. The aim of the study is to find genetic and clinical markers for HCC, i.e. to develop a classification model based on HBV DNA and clinical data. This chapter concentrates on the DNA analysis part.

This chapter is structured as follows. Section 4.1 describes the related work briefly. Then, we define the problem clearly in Section 4.2. Starting from Section 4.3, we present the proposed methodology architecture and its modules in detail. Sections 4.4 and 4.5 focus on the feature selection and classification model selection modules respectively. In our experiments, we find out a critical error which can lead to a completely incorrect evaluation of the model. It is discussed in Section 4.6. Our proposed framework is evaluated on the HBV DNA

of our project. Important results are presented in Section 4.7. Finally, a summary of the evaluation is presented.

4.1 Related Work

The focus of this project is to find genetic marker(s) for liver cancer (HCC) from our Hepatitis B Virus (HBV) DNA sequences. There are similar medical researches in the literature, but all of them just focus on the specific gene positions, proteins or part of a virus genome. Therefore, this research project is a pioneer study on the complete viral genome. One of the past research is a HIV genomic study [2]. The researchers align each DNA sequence with a reference sequence first, then select the genes by their expert knowledge, and use decision tree and Support Vector Machine for analysis. In our project, we develop a new framework for finding genetic markers of HCC in HBV genome data.

Another interesting publication is on the identification of HBV DNA sequences that are predictive to the response to Lamivudine therapy [14]. In the paper, authors identified certain gene and mutations patterns that can be used to predict the drug response to Lamivudine. Their experiments are carried out among 26 patients who is consecutively enrolled in hospital and under Lamivudine treatment. It is similar to the second part of our HBV genome project on drug response. However, the scale of their study is rather small, compared to ours. They just concentrated on 3 nucleotides and 2 polymerase, but our study focuses on the whole viral genome of hundreds of patients. The paper gives us an introduction on the methodology that works on DNA sequences.

4.2 Problem Definition

Genome of an organism is all of the genetic information or hereditary material possessed by an organism, and it includes the entire genetic complement of an organism. HBV genomes are extracted by laboratory processes and represented in a form of DNA. Section 2.3 has given you a general picture on Hepatitis B infection and virology background of HBV. In this study, we have DNA sequences from 100 Control patients and 100 HCC patients. The DNA sequences of HBV are not exactly the same for each patient, as they possess some individual nucleotide mutations that may or may not related to HCC. In the literature, HBV can be divided into seven genotypes (A to G) where each of them have more than 8% difference of nucleotides from the others. In Hong Kong, genotypes B and C are the most common types, and all the samples we have belong to these genotypes. To reduce the noise of genotypic difference between genotypes B and C, we analyze their DNA samples separately.

This project is a pioneering project. In the medical and biochemical research field, the scale of this project is considered large and comprehensive. The whole genome of HBV DNA are extracted and analyzed. In the computer science point of view, the volume of data is too small while the data dimension is so large. At the same time, how to tackle such a small data set carefully to ensure the statistical correctness, how to distinguish which genome sites may be meaningful to our analysis, how to reduce the noise (unrelated mutations) of data, and how to choose a suitable classification model, are all challenges to this project. Our goal is to devise a comprehensive framework that can be used in classification of HCC based on the DNA sequences. This classification model should have high accuracy, specificity and sensitivity for HCC

diagnosis and prediction.

4.3 Proposed Methodology Architecture

According to the problem definition and characteristics of Hepatitis B virus DNA sequences, we have proposed a general framework for solving our problem defined.

4.3.1 Overall Design

Our proposed framework composed of several modules to handle data preprocessing, feature selection and data mining respectively. Fig. 4.1 shows the overall architecture of our framework.

We start from the modelling of DNA sequences with over thousands to millions of nucleotides. In this stage, we assume each nucleotide is independent from the other adjacent nucleotides. Each nucleotide is treated as an attribute. However, every DNA sequence does not have fix length because of the insertion and deletion of nucleotides. We have to align the DNA sequences with a reference DNA sequence obtained from GeneBank [53], before any comparison is made. As a result, we can consider the DNA sequences as records with a fix number of attributes.

Going back to our HBV genome project, our group discovers that there exists subgroups in genotype C sequences by observing the phylogenetic tree of them. This is an important finding on biochemistry field. As analyzing different subgroups separately is a reasonable and effective approach, it is included in our proposed framework.

For each subgroup, the training data are used to search the useful features for classification and learn the classification model. New testing data must be assigned to the corresponding subgroup first, then

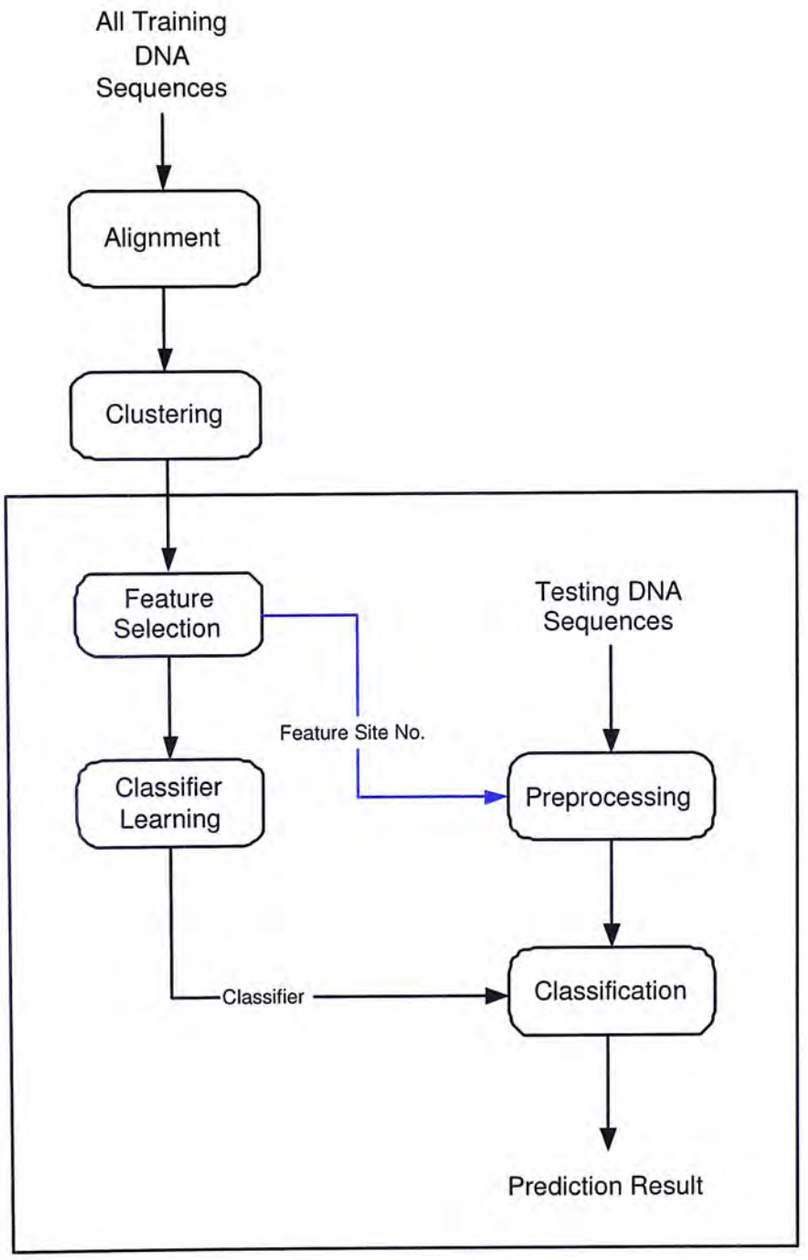


Figure 4.1: Overall design architecture

extracted the chosen features for analysis, which are fed into the classification model learned.

4.3.2 Important Components

The important components steps of our framework are introduced as follows.

1. Alignment

   Every DNA sequence does not have a fix length because of the nucleotide insertion and deletion. We have to align the DNA sequences with a reference DNA sequence before making any comparison. The public tool ClustalW is used for multiple sequences alignment in our experiments [24].

2. Clustering

   In this module, the subgroups existing in the data set will be discovered. Separating different subgroups for analysis can enhance the accuracy of the model. The signatures of each subgroup discovered can be used for subgroup classification for the prediction phase.

3. Feature Selection

   The length of DNA sequences of a virus or an organism can range from thousands to billions of base-pairs. It is impossible to consider every nucleotide as an attribute, because the complexity is too large and there may exist some noise features among the data. A tailor-made feature selection algorithm should be designed for each DNA analysis problem.

4. Classifier Learning

   In our framework, different classification models can be applied

with general or problem-specific learning algorithms.

5. Preprocessing

   Testing data must be assigned to the corresponding subgroups, if there exists subgroups in the problem. Then, some attributes are extracted from the testing sequences according to the result of feature selection step.

6. Classification

   Testing data is applied into the learned classifier for class prediction.

4.4 Clustering

In our HBV genome data set, there are 86 sequences in genotype B and 110 sequences in genotype C. Before applying any classification models to these two data sets, the phylogenetic tree results show that there exists 3 subgroups in the genotype C data set. Fig. 4.2 is the *phylogenetic tree* for genotype C sequences, where C1, C2 and C3 are the subgroups we found.

After we subdivide the sequences into C1, C2 and C3 groups, we can generalize some site positions as the signatures of each group. However, these signature sites number cannot be presented here as the latest results are being patented. When we have a new DNA sequence, we can align it with the reference sequence and check the signature site positions of each subgroup. Since this part is my research partner Y. T. Ng's work, please refer to his term paper for details [54].

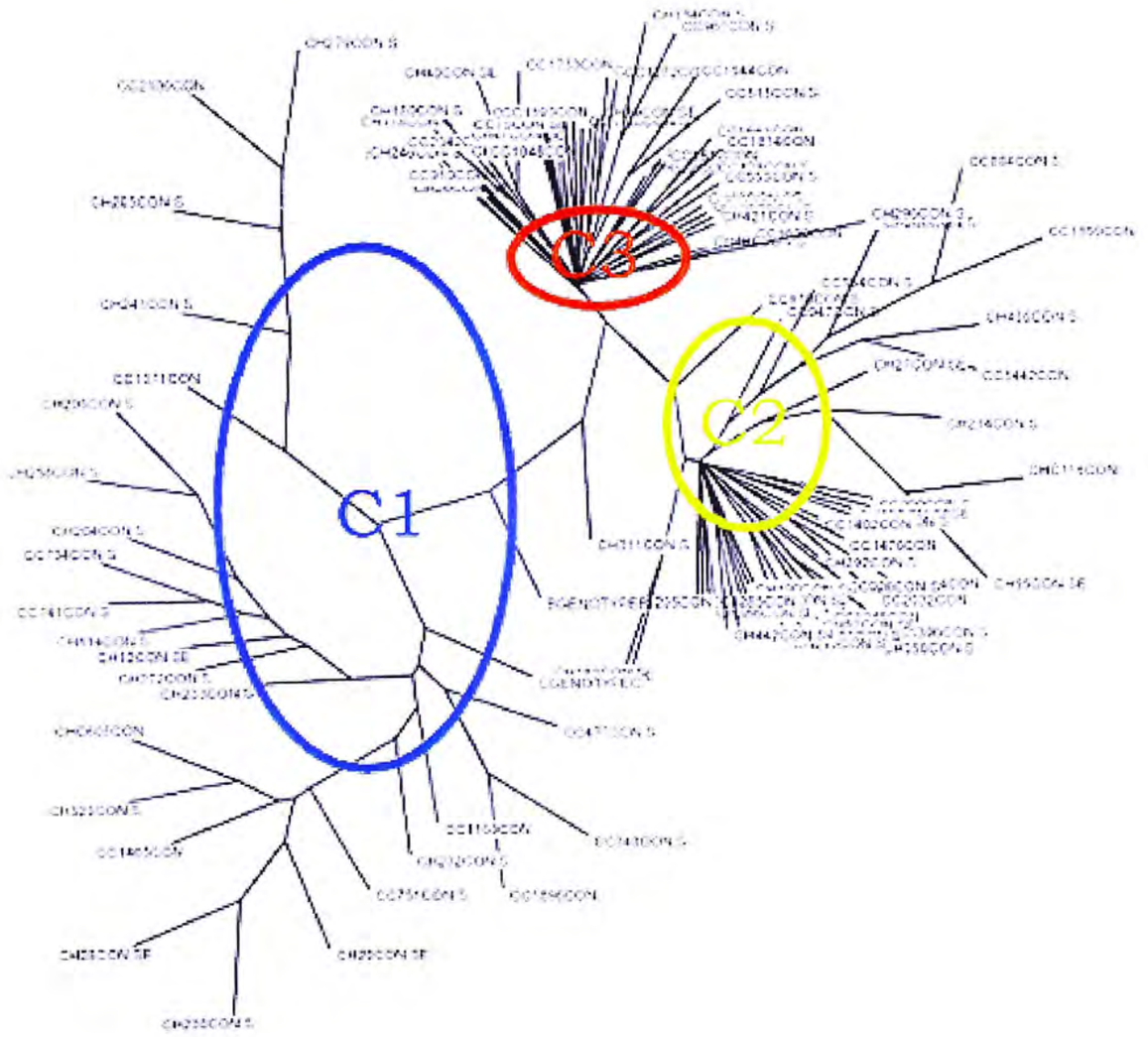


Figure 4.2: Phylogenetic tree showing the three subgroups in Genotype C

4.5 Feature Selection Algorithms

The length of HBV DNA sequence is between 3200 to 3300 base-pair(bp). In our current approach, we cannot use all of them for classification. As we just have 200 records in total, the number of records is insufficient for the statistical correctness. Therefore, feature selection must be performed, so that the gene positions that are more "useful" for distinguishing between Control and HCC groups are selected for analysis. In our project, we have tried the following criteria for the feature selection :

1. InfoGain :
   Rank all the nucleotide positions with *information gain* and choose the top rank positions. This approach can sort out the positions with the most distinguishing power for classification.

2. C-Pure
   Select the nucleotide positions that have the same nucleic acid for all the Control sequences but have mutations in HCC sequences. Rank the selected positions by information gain, and choose the top ranking ones. We suspect that the virus mutation may contribute to the risk of HCC, thus this feature selection approach can validate our hypothesis.

3. H-Pure
   Select the nucleotide positions that have the same nucleic acid for all the HCC sequences but have mutations in Control sequences. Rank the selected positions by information gain, and choose the top ranking ones. We suspect that the virus mutation may resist the progression of HCC, thus this feature selection approach can validate our hypothesis.

4.5.1 Information Gain

Information Gain is a common criterion for feature selection. It is frequently used in decision tree learning. Feature with higher information gain is the one which can reduce more uncertainty (entropy) in the target attribute. The following is some background information about it.

Equation 4.1 is the entropy E, of an attribute X with n values $X_1 \ldots X_n$. $P(X_j)$ is the probability of the value X_j.

$$E(X) = \sum_{j=1}^{n} -P(X_j) \log_2 P(X_j) \tag{4.1}$$

Specific to a typical DNA classification problem, we assume the data have M classes $C_1 \ldots C_M$. For each aligned site position, it has N possible nucleotides $V_1 \ldots V_N$. We define $|C_m|$ be the number sequences in class C_m. $|C_{mi}|$ be the number of sequence in Class C_m, whose character at the aligned site is V_i.

The reminder of X, $R(X)$ is defined as follows :

$$R(X) = \sum_{i=0}^{N} \frac{\sum_{k=1}^{M} |C_{ki}|}{\sum_{k=1}^{M} |C_k|} E(P(C_{1i}), \ldots, P(C_{Mi})) \tag{4.2}$$

Information Gain $IG_j(S)$ of aligned site j is the difference between the original information content of the data set and the amount of information to classify all the data in the data set :

$$IG_j(S) = E(C) - R(j) \tag{4.3}$$

Calculated information gain of each aligned site can be displayed with the aligned sequences by our viewer tools. Using different feature selection criteria, top ranked sites are chosen for experiments.

4.5.2 Other Approaches

Apart from using information gain, different feature selection approaches can be applied. Expert knowledge is the primary way to do it. Biochemists and doctors have knowledge on the virus virology and immunology, so that they can pick out suspected genes, sites or specific proteins for analysis. However, their knowledge on the virus is limited and should only be used to aid the research and automatic knowledge discovery. Human justification is sometimes imprecise and inaccurate. Moreover, autonomous feature selection is another approach. For example, using evolutionary algorithm like genetic algorithm (GA) to seek the set of features which is most favorable for classification [68] [61].

Feature selection on biological and medical data sets is more challenging than that from normal domains. We should consider the nature, characteristics and biological meanings of each attribute. For example, our attributes are sites taking different nucleic acid with mutations as their values, but the different nucleic acid may not imply that the mutation contributes to the disease. Consecutive sites may have linked relationship among them. At this time, expert knowledge may be useful. The most efficient approach is to combine the autonomous feature selection with expert knowledge.

4.6 Classification Algorithms

Our goal is to discover genetic markers of HCC from HBV DNA. In other words, we are building up a classification model for HBV DNA for predicting cancer. In fact, the choose of classification algorithm is crucial for the model. In this section, we describe the classification

models we have used for analysis and other possible choices in detail.

4.6.1 Naïve Bayes Classifier

Naïve Bayes is the first model we tried in our analysis. Its structure is simple but efficient for computation and classification. The details of Naïve Bayes can be found in Section 2.4.4. Using it as our classification model, we must assume that each nucleotide position (site) is independent from each other. Although this may not be the real case, we can investigate the independent contribution of each site to the class value. Another advantage of this model is its scalability. It can handle large number of attributes without great computational effort.

4.6.2 Decision Tree

Decision tree is a popular model for classification. It takes an object or situation described by a set of attributes as input, then give out yes/no decision as output. It can also represent functions with larger range of outputs [59]. In our framework, we can choose decision tree as our classification model. Popular decision tree constructing algorithms include ID3 and C4.5. In our project, we use C5.0 which is the latest efficient decision tree constructing algorithm for our experiments [30]. One advantage of using decision trees is its readability to the researchers who are not in computer science field. The decision tree can also be translated into decision rule which is useful for future clinical use.

4.6.3 Neural Networks

Neural network is another popular model for classification which models the biological nervous system. It is composed of a large number

of interconnected processing elements (neurons) working in union to solve specific problems. It has remarkable ability to derive an implicit prediction model from complicated or imprecise data. It can be used to extract patterns and detect trends that are too complex to be extracted by either humans or other computer techniques [23].

In our problem, the interrelationship between each attribute (site) is unknown. Neural network may be a good choice for DNA sequence classification. Experiments are conducted for evaluation of our hypothesis in the later part of this chapter.

4.6.4 Other Approaches

Apart from above models, other classification models can be tested including association rules, nonlinear multi-regression networks [45],etc. Classification rules are an ideal knowledge representation of this analysis for the doctors, because of its human readability. Association rules learning is useful to discover relationship or patterns between different attributes. Nonlinear multi-regression networks are also used as a classification model in the framework. The preliminary results shows that it can get outstanding results.

4.7 Important Points on Evaluation

In our experiments, we find out a critical error which leads to a completely incorrect evaluation of the model. In the preliminary stage of our analysis, we designed this framework with information gain as feature selection approach and Naïve Bayes as classification model. At that time, the evaluation experiments on real HBV DNA sequences showed that the classification accuracy is up to 85% or more. We were curious to know why the accuracy was so high for this real and complex

clinical problem. Consequently, we discovered this error of evaluation methodology.

4.7.1 Errors

In fact, the mistake is quite tricky and easily-made. In our experiment methodology, we used ten-fold cross-validation (CV10) for evaluation as the volume of data sets are just 20-80. The original data set is divided into ten groups and there are ten experiments for each round. For each round, one group is taken as testing set while the other nine groups left are taken as the training set of the model. Each group take turns to be the testing set in these ten experiments.

Before we discovered this error, we divided the data set into ten groups immediately after feature selection step and before the classification step. Then, the training and testing sets were used for model training and testing respectively. As the testing set was not involved in the model training, we assumed that it was a valid experiment because the testing set was independent from the training. However, this is not the case.

The testing set had already participated in the feature selection learning step with the training set. It had contributed to the modelling process. Therefore, the sites picked for use was partially selected by the testing data. As a result, we could obtain a higher accuracy in the later classification step because of the sites selected.

4.7.2 Independent Test

Once the error is identified, the evaluation methodology is corrected. The experimental results show that the inclusion of testing data in feature selection step will enhance the accuracy of the model, i.e. over-

training. It is also statistically biased in real classification problem. Since the class of unclassified data is unknown, how can they contribute to the feature selection step which deciding which sites to pick?

For any similar study, the independence of testing data is very important for model evaluation. The testing data must be excluded from the experiments from the very beginning.

4.8 Performance Analysis on Classification of DNA Data

In this section, we present the results of applying different classification models to classify the HBV DNA data into liver cancer (HCC) and normal cases.

4.8.1 Experimental Methodology

After preprocessing the HBV DNA sequences by multiple sequence alignment, clustering and feature selection, we try to use Naïve-Bayes, decision tree, neural network models and expert rules for classification. Table 4.1 shows the details of our HBV genome data.

Genotype\Datasets	CON	HCC	Total	%
B	49	37	86	43.878
C1	10	16	26	13.265
C2	18	22	40	20.408
C3	19	25	44	22.449
Total	96	100	196	

Table 4.1: Summary of HBV DNA data

Genotype B and genotype C data were separated for analysis. Biochemists applied data cleansing process on the data after alignment.

Two genotype C and one genotype B sequences were removed from our data set. Our experiments used ten-fold cross-validation (CV10) for each experiment setting to obtain an accurate evaluation of the model.

The HBV genome project is a medical project the results of which will be used clinically for cancer prediction in the future. In medical diagnosis and disease predication problems, the algorithm or model performance is not only judged by *accuracy*, but also *sensitivity* and *specificity*. According to the expert opinion from doctors, sensitivity is much more important than specificity and accuracy, because they do not want to miss any patients with diseases. Extra diagnosis and tests can be done to confirm their prediction. Therefore, we evaluate our model in all these three measurements.

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{True Negative}} \tag{4.4}$$

$$\text{Sensitivity} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{4.5}$$

$$\text{Specificity} = \frac{\text{True Negative}}{\text{True Negative} + \text{False Positive}} \tag{4.6}$$

The *true positives* are the number of all the patients with the disease and positive test results, whereas the *true negatives* are the number of all the patients without the disease and negative test results. The *false positives* are the number of all the patients without the disease and positive test results, whereas the *false negatives* are the number of all the patients with the disease and negative test results. In medical diagnosis, a false negative is the most undesirable case.

4.8.2 Using Naïve-Bayes Classifier

Naïve Bayes is the first classifier we used for analysis. It has a simple but efficient structure that can save computation effort on model learning. Table 4.2 shows the summary of using Naïve Bayes as the classifier of liver cancer cases in our framework.

	Genotype B	Genotype C1	Genotype C2	Genotype C3
Sensitivity (%)	53	96	70	50
Specificity (%)	64	40	58	36
Accuracy (%)	60	77	65	44

Table 4.2: Performance of model with Naïve Bayes as classifier

Experimental results show that the accuracy of classification of genotypes C1 and C2 is better than other subgroups. This pattern also applies to other kinds of classifiers. However, the above results are not satisfactory enough for real medical use.

4.8.3 Using Decision Tree

The next set of experiments are conducted by using decision trees as the classifier. We adopt the C5.0 which is the latest efficient decision tree learning algorithm for our classifiers [30].

	Genotype C1	Genotype C2
Sensitivity (%)	100.00	72.20
Specificity (%)	50.00	66.60
Accuracy (%)	80.00	70.00

Table 4.3: Performance of model with decision tree as classifier

In this model, we concentrate the testing on genotypes C1 and C2. Table 4.3 shows the performance of our model using decision tree as classifier. In the doctors' opinion, the sensitivity and accuracy are high enough for clinical use, but the specificity is rather low. Further

verification and improvements on the model should be done in the future.

4.8.4 Using Neural Network

As described in the previous section, neural networks are chosen as our classifier in our framework. Table 4.4 shows a summary of the experimental results.

•	Genotype B	Genotype C1	Genotype C2	Genotype C3
Sensitivity (%)	71.00	100.00	87.00	87.00
Specificity (%)	71.00	60.00	86.00	50.00
Accuracy (%)	71.00	85.00	86.00	71.00

Table 4.4: Performance of model with neural network as classifier

The performance of neural networks working with our framework is pretty good. It can obtain at least 71% accuracy on genotype B and genotype C3 data sets, and over 85% on genotypes C1 and C2. The sensitivity is quite high, while the specificity is also satisfactory. This is the best model among the three classifiers we have tested.

Fig. **??** shows the classification performance of different classifiers on each genotype. By comparing the average classification performance of all models in different genotypes, genotype B and C3 have relative low accuracy and sensitivity which are not satisfied for being clinical tests. On the other hand, every classifier gets over 70% of accuracy and up to 100% of sensitivity in Genotype C1 and C2. These results are very promising and encouraging. Among different classifier models, neural network is the best model for classification. The accuracy and sensitivity are up to 85%, except in genotype B. In addition, the specificity of every test is quite low. Although sensitivity is more important than specificity in medical tests and diagnosis, it is still a room for

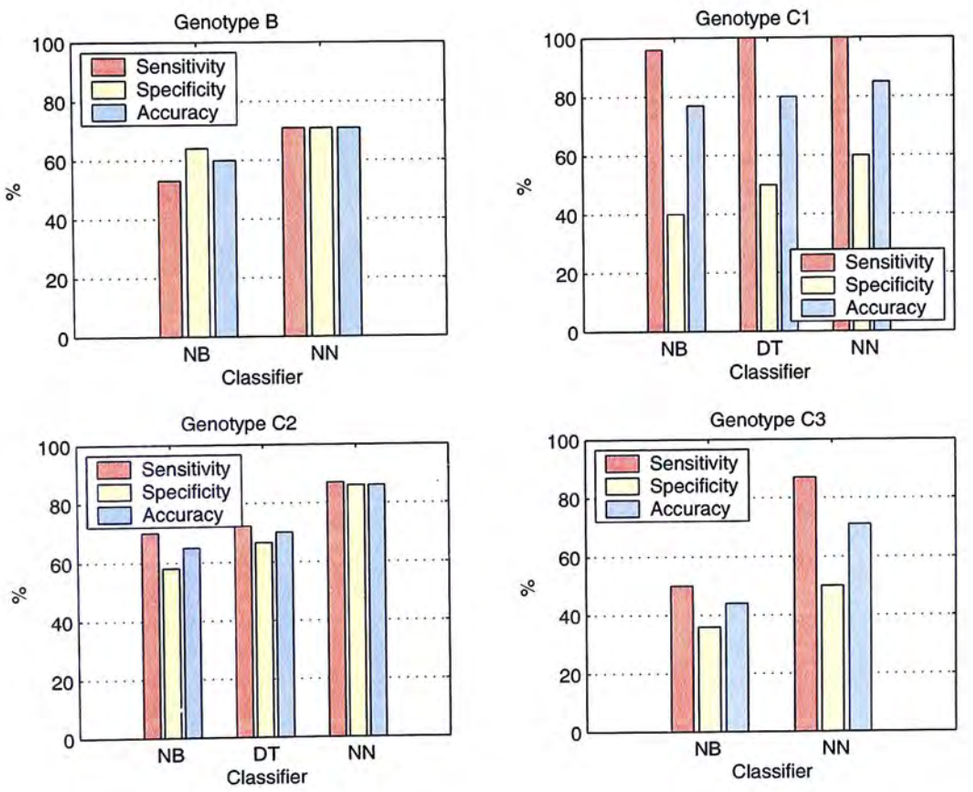


Figure 4.3: Comparison on different genotypes using different classifier models (NB - Naïve Bayes, DT - Decision Tree, NN - Neural Network)

improvement.

4.8.5 Discussion

The proposed framework with particular feature selection approach and classification models work well for DNA analysis. For example, the accuracy and sensitivity of the framework using neural networks as its classifier can reach up to 80-100% for several subgroups of sequences. The findings are validated and accepted by the biochemists and doctors in our project.

In our methodology, we have made some assumptions that may have missed some useful information or introduced some bias into our model. There is still room for improvement.

Firstly, we interpret each nucleotide in a DNA sequence as an individual attribute. In fact, there may exist interrelationship between adjacent nucleotides which we have not considered in our approach. In future analysis, representations at gene level and protein level can be tried, or a new model which can take this relationship into consideration could be proposed.

Secondly, not all the mutations of a HBV DNA sequence affect the genetic functions of a virus and its activity. These mutations may be random processes and do not contribute to the HCC progression. Our study makes an assumption on the direct relationship between mutation and HCC progression.

However, from our previous experiments, the above assumptions do not have big impacts on our model. The results obtained are validated by biochemists and doctors with reference to the related researches and knowledge-base.

4.9 Summary

The use of a full viral DNA sequence for computational data mining is quite new and unique in the field of bioinformatics. After working for the past two years, we have several great findings such as the existence of subgroups in HBV virus. Moreover, we have achieved the project goal in genotypes B, C1 and C2 and the classifiers developed have over 70% accuracy. As our project team is now undergoing the stage of patent application, we believe that our project will be completed with great success.

Our proposed framework for doing DNA sequence analysis have been shown to be comprehensive and effective to solve DNA sequence classification problems. Different feature selection approaches and classification models can be included in our framework with high flexibility. Users can tailor-make their models according to the characteristics of data and problems.

□ **End of chapter.**

Chapter 5

Adaptive HEP for Learning Bayesian Network Structure

This chapter describes an optimized algorithm for learning Bayesian Network structure by using adaptive population sized evolutionary programming (A-HEP). Bayesian network (BN) is a popular knowledge discovery model which represents the causal relationship of different events or attributes with uncertainty. Learning the structure solely by dependency analysis or search-and-score approach is not effective. The hybrid algorithm on evolutionary programming, HEP, has been shown to be effective and efficient to solve this learning problem [66]. By introducing the concept of adjusting the population size according to the individuals' dissimilarity, HEP is further optimized in respect of the execution time with comparable performance. The empirical results illustrate that the optimized algorithm has reduced the running time by half on average.

5.1 Background

As described in Chapter 2, there are two major approaches to this network learning problem - dependency analysis and score-and-search approach. However, the two approaches have their own drawbacks. In the previous work, a hybrid approach is used for learning Bayesian network structures by evolutionary programming (HEP) [66]. HEP searches the network structure with the help of the statistical dependency information. It is shown to be effective and efficient in this learning problem. On the other hand, Y. Liang have designed the adaptive elitist-population search method (AEGA) that locates all optima of multimodal problems [44]. By combining the concepts of both algorithms, an optimized version of Bayesian network learning algorithm can be designed. In this chapter, A-HEP is described as an extension of HEP adopting the dynamic population size concept of AEGA.

5.1.1 Objective

Since the running time of evolutionary algorithms depends on the population size, our algorithm (A-HEP) should be designed in a way that the population size can increase and decrease adaptively according to the dissimilarity of individuals. Once the algorithm converges to a certain degree, the population size can be decreased and computation is also reduced. As a result, execution time can be reduced significantly.

5.1.2 Related Work - AEGA

AEGA is a new technique used to solve multimodal function maximization problems. Elitist individuals are defined as the best ones on the respective peaks. With the help of elitist operators, the diversity of the population can be maintained and even improved by adjusting the

population size according to the dissimilarity and relative directions of individuals in the population. Eventually, the population can explore all optima of multimodal problems in parallel based on elitism. [44]. The principles of AEGA (assuming the objective is finding all the local and global maxima) can be summarized as follows:

- If the relative ascending directions of both individuals are *back to back*, these two individuals are dissimilar and locate on different peaks.
- If the relative ascending directions of both individuals are *face to face* or *one-way*, and the distance between two individuals are smaller than a threshold, they are similar and locate on the same peak.

Elitist crossover operator and elitist mutation operator are designed according to the above principles. As a result, each elitist individual is converging into each local or global maximum and solving multimodal problems efficiently.

5.2 Feasibility Study

Our original objective was to adopt the concepts of AEGA into HEP to enhance its performance. As AEGA can be used to find multi-local optima of a problem, we believed that similar concept might be used to find local optimal structures. Therefore, we tried to design a distance measurement function between individuals and population size variation routines.

As a matter of fact, the concepts of AEGA cannot be adopted directly into Bayesian network structure learning problem because of the following factors:

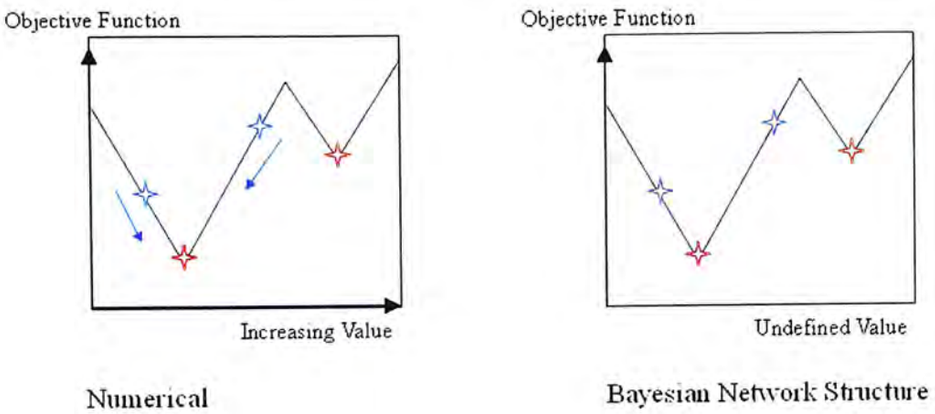


Figure 5.1: Search space

1. AEGA is basically used on *numerical data*, while HEP is used to find the structure of a Bayesian network which is represented as a *data structure*, which is discrete in nature.

2. The relative direction between individuals is undefined in Bayesian network structure learning problem. However, the core idea of AEGA is to vary the population by the relative directions between parents and offsprings.

3. It is difficult to determine if a structure is the local optimal one or not. Even we get some structures, we cannot confirm whether they are local optimal structures by varying each part of the Bayesian network.

Referring to Fig.5.1, red points represent the local optima and blue points represent the current individual. For the numerical cases, the relative direction between two individuals can be calculated, i.e. toward the same minima in this case. Then, the AEGA algorithm is able the remove the redundant individual and keep the elitist one. For the Bayesian network structure, the concept of relative direction does not

exist. We can only define the distance between two individuals and compare their objective function values. Therefore, the concepts and operators of AEGA cannot be used directly in our problem. The only idea we can adopt is the dynamic population size.

5.3 Proposed A-HEP Algorithm

The principle used for improving HEP is the adaptive population size concept based on the dissimilarity of individuals. Similar to other evolutionary algorithms, the population of HEP converges into one or several solutions with best fitness at the end of evolution. In the later part of evolution, most individuals are similar or exactly the same. Computation time can be reduced if these redundant individuals are removed from the population in later generations. On the other hand, the diversity of the population are also important for searching the optimal solution, especially at early generations. Based on the HEP algorithm, new routines for increasing and decreasing population size are designed, in order to increase the diversity and remove redundancy respectively. These routines work by comparing the dissimilarity of individuals in the population. We have also defined a structural dissimilarity comparison metric for comparing different Bayesian network structures. In this section, the techniques will be described in detail.

5.3.1 Structural Dissimilarity Comparison

The objective is to use the A-HEP algorithm to speed up the process of searching good network structures with small MDL scores. Therefore, a representation for network structures has to be defined. In A-HEP, a network structure is represented as a two-dimensional matrix (shown as in Fig. 5.2). The size of the matrix is $n \times n$, where n is the number

of nodes.

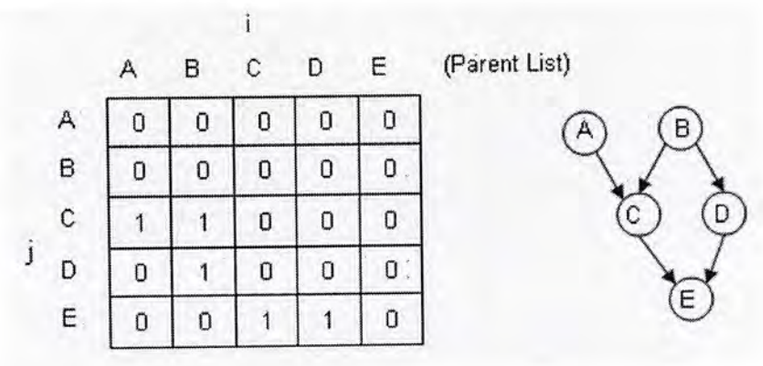


Figure 5.2: Representation of a Bayesian network structure

The value of the matrix is defined as:

$$StructureB_{ij} = \begin{cases} 1 & \text{if node } j \text{ is the parent of node } i \\ 0 & \text{if node } j \text{ is } \textit{not} \text{ the parent of node } i \end{cases}$$

Since the individuals are represented in this data structure, we can define a function to compare the distance between two individuals, i.e. the dissimilarity of two Bayesian networks.

$$Distance(B, B') = \sum_{i,j}^{n} x_{ij} \qquad \begin{cases} \text{where } x_{ij} = 0 \text{ if } B_{ij} = B'_{ij} \\ \text{where } x_{ij} = 1 \text{ if } B_{ij} \neq B'_{ij} \end{cases}$$

For example, the distance between the two Bayesian networks in Fig. 5.3 is 2, as one edge is added and one edge is deleted.

5.3.2 Dynamic Population Size

In the original HEP algorithm, the population size m is fixed. In each generation, each individual either cross over with another individual in the previous generation or mutates itself by different operators to get

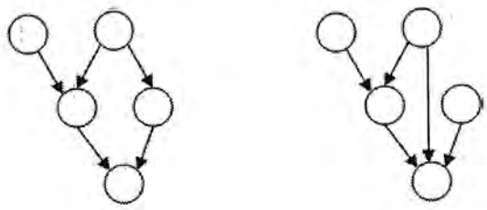


Figure 5.3: Two different Bayesian network structures with $distance = 2$.

its new offspring. New population size is then increased to $2m$. After that, a number of pairwise competitions are carried out and the fittest m individuals are kept for the next generation.

In A-HEP, the population size changes adaptively according to the dissimilarity of individuals in the current population and newly evolved offspring. The principles of this approach are:

1. If new offspring is quite different from the individuals of the current population (i.e. the distance is large), it is worthy to keep it, although it has worse fitness.

2. If some individuals are similar or exactly the same as other individuals (i.e. the distance is small or zero), it is reasonable to remove them.

Undoubtedly, the above principles cannot be applied strictly throughout the whole evolution process. Otherwise, the search space for better solutions will be limited by the second principle, and the algorithm will not converge under the first principle. Therefore, the stage of evolution must be considered as a factor of population size expansion and contraction. In the early stage of evolution, we can allow more degree of population size expansion and less degree of contraction. In the later stage of evolution, we can limit population size expansion and allow removing more redundant individuals in the population. At the same

time, the population size is maintained within a range of values.

A-HEP is developed based on the original HEP with several modifications. The original CI Test Phase and different operators are still used in the new algorithm. An increasing routine and a decreasing routine are introduced for processing mutated individuals to change the population size adaptively. In A-HEP, the pairwise competition is no longer used because the new routines are employed to decide which individuals should be kept.

Algorithm 8 outlines the operations of A-HEP. The following notations are used to describe A-HEP throughout this chapter:

- p_c is the current population size in this generation.
- p_{new} is the new population size for next generation.
- p_{max} is the maximum population size.
- p_{min} is the minimum population size.
- Gen_c is the current generation number.
- Gen_{total} is the total generation number.
- $AvgDis_i$ is average distance between the individual I_i to all other individuals.
- R_1, R_2, R_3 are three random numbers between 0 and 1.

Algorithm 8 Algorithm of A-HEP

CI Test Phase

Same as the one in HEP (Algorithm 1)

Evolutionary Programming Search Space

Set $Gen_c = 0$;
Initialize and evaluate the population with size p_{init};
while $Gen_c < Gen_{total}$ **do**
  Randomly select $p_c/2$ individuals for merge operator
  Each unselected and unmerged individual produce one offspring by different mutation operators
  Increasing Routine (See Routine 1);
  Decreasing Routine (See Routine 2);
  Update population size, $p_c = p_{new}$
end while
Return the final structure with the lowest MDL score.

Increasing Routine

In order to increase the diversity at the early stage of evolution, the population size is increased adaptively by examining the newly mutated offspring. If the newly mutated offspring is very different from the individuals in current population, the parent and itself will be kept, regardless of their fitness values. This technique can prevent premature convergence. However, the population size expansion must be controlled by considering the following factors:

1. Ratio of the current population size to the maximum population size, p_c/p_{max}
2. Ratio of the current generation number to the total generation number, Gen_c/Gen_{total}

Two random numbers are generated and compared against the above ratios. If they are larger than the ratios, newly mutated offspring is considered to be added into the next population. When the current population size or the generation number is large, it is less likely to further increase the population size.

In order to decide the fate of the newly mutated offspring and its parent, calculation is performed on the average distance between them and all the other individuals in the current population by the dissimilarity metric defined in the previous section. If it is larger than the threshold, *far-factor* $\times$ *no. of nodes*, both of them will be preserved for the next generation. Since the size of matrix for representing for a Bayesian network structure depends on its number of attributes, the distance threshold should also depend on the number of nodes.

With the population size increasing routine, the diversity increases with the search space in the early generations. Experimental results have shown that better network structures can be obtained.

Routine 1 For increasing population size

$p_{new} \leftarrow p_c$
$i \leftarrow 0$;
while $p_c < p_{max}$, and $R_1 > p_c/p_{max}$, and $R_2 > Gen_c/Gen_{total}$, and $i < p_c$ **do**
  for each mutated offspring I_i **do**
    Calculate $AvgDis_i$;
    if $AvgDis_i >$ *far-factor* $\times$ *no. of nodes* **then**
      Both its parent and itself are kept for next generation;
      $p_{new} \leftarrow p_{new} + 1$;
    end if
  end for
  $i \leftarrow i + 1$;
end while

Decreasing Routine

The main objective of A-HEP is to reduce the running time of the original HEP. This can be achieved by removing the *redundant* individuals in the population at the later part of evolution. Based on AEGA [44], a routine is designed for decreasing the population size adaptively by considering the dissimilarity between individuals.

Similar to the increasing routine, ratio of the current generation number to the total generation number is used as a parameter to delay the time of population size contraction. There are two cases when population size is going to decrease:

1. Where two mutated offsprings are fitter, and are more similar between themselves than their parents do between themselves, and their distance fall short of the threshold, the*cutoff-distance.*

2. The pair of chosen individuals for the next generation are exactly the same.

Before the individual is removed from the population, a random is generated

Routine 2 For decreasing population size

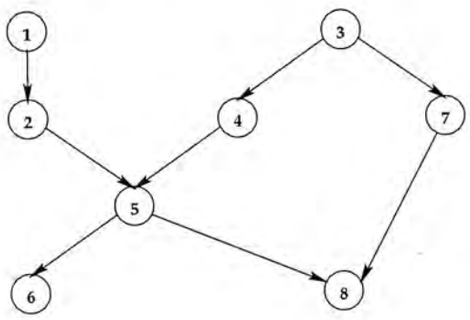
for each pair of mutated individuals I_i,I_j and their parents I_i^p,I_j^p **do**
  Calculate $d_1 = Distance(I_i^p, I_j^p)$;
  Calculate $d_2 = Distance(I_i, I_j)$;
  Compare the fitness of I_i with I_i^p and I_j with I_j^p, and take the fitter one in each pair for next generation;
  if Both children,I_i, I_j, are chosen **then**
    if $p_c > p_{min}$ and $d_2 < d_1$ and $d_2 <$ *cutoff-distance* **then**
      Choose the fitter child and remove another one;
      $p_{new} \leftarrow p_{new} - 1$;
    end if
  end if
  if *Distance* between chosen pair $= 0$, and $p_c > p_{min}$, and $R_3 > 1 - Gen_c/Gen_{total}$ **then**
    Remove one of them;
    $p_{new} \leftarrow p_{new} - 1$;
  end if
end for

and compared against one minus the above ratio, $1 - Gen_c/Gen_{total}$. In the later stage of evolution, the ratio is so large that the random number is always large enough to satisfy the condition of deleting the redundant individuals.

With the decreasing routine, redundant individuals are removed in the population at the later stage of evolution. As population size decreases, the computation effort required for each generation is also reduced. Consequently, the time it takes to obtain the final structure is greatly reduced. This will be demonstrated by experiments in next section.

5.4 Evaluation on Proposed Algorithm

The performance of A-HEP was evaluated and compared with the original HEP by the following set of experiments. An optimized algorithm should obtain a Bayesian network with comparable or even better quality in shorter time. Therefore, the following experiments are used to examine the performance of A-HEP on running time and fitness of final network structure



1. Visit to Asia
2. Tuberculosis
3. Smoking
4. Lung Cancer
5. Tuberculosis or Cancer
6. X-Ray Result
7. Bronchitis
8. Dyspnea

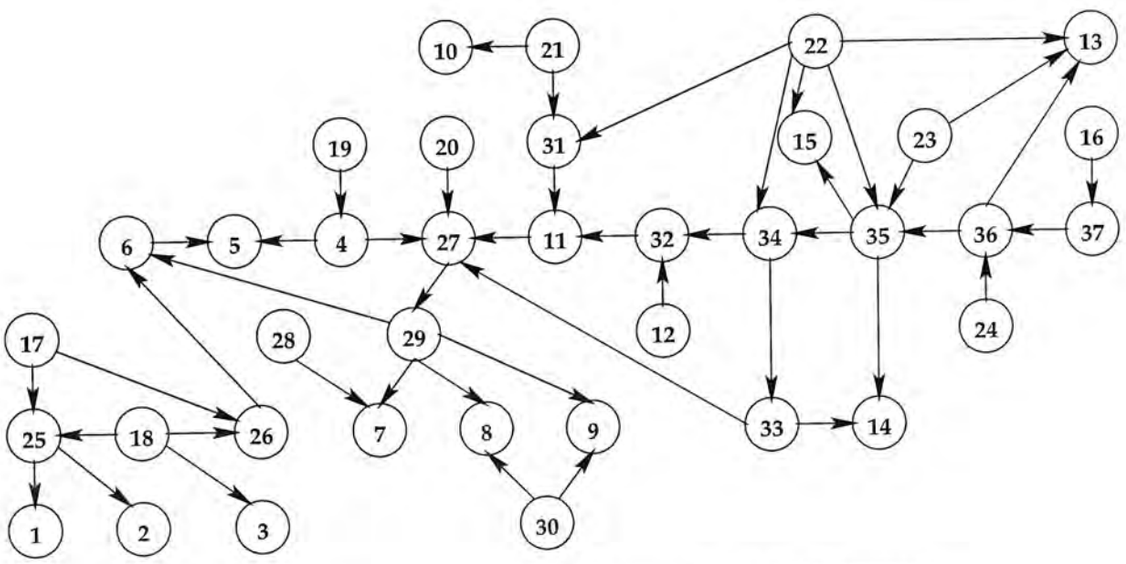
Figure 5.4: The ASIA network [43].

obtained.

5.4.1 Experimental Methodology

In our experiments, we used seven data sets generated from the well-known benchmarks of Bayesian networks including the ALARM, the PRINTD, and the ASIA networks.

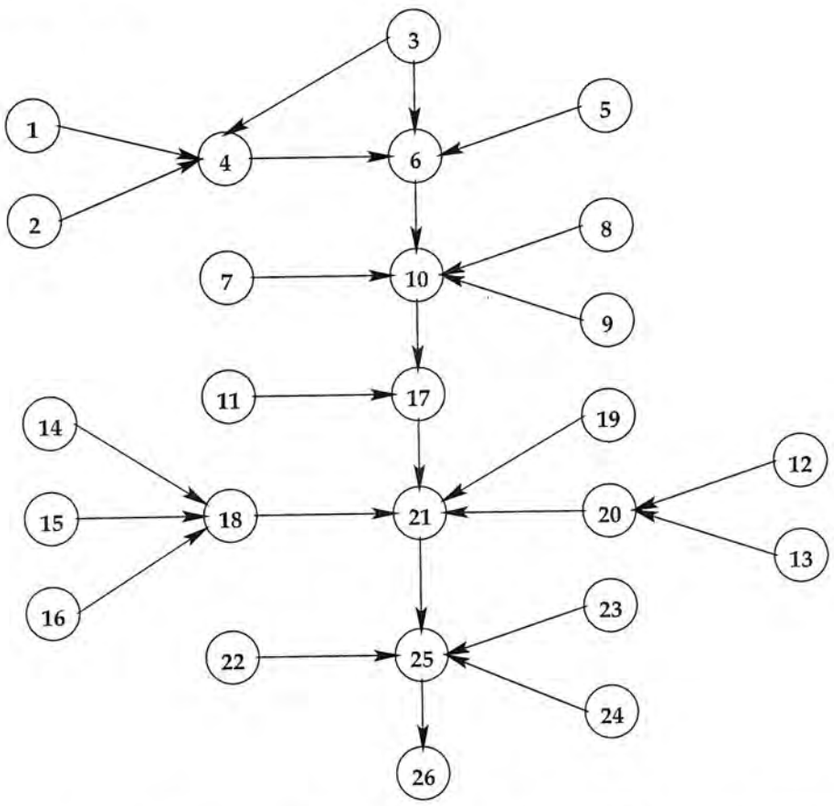
Alarm1000, alarm2000, alarm5000, alarm10000, and alarm-O were created from the ALARM network. These data sets were obtained from two different sources. One of them (alarm-O) containing 10,000 cases was obtained from Bayesian Network PowerConstructor [10]. The others were used for evaluating MDLEP [64]. The four data sets are of different sizes and contain 1,000, 2,000, 5,000, and 10,000 cases respectively. The structure of ALARM network is shown in Fig.5.5. Originally, the ALARM network is used in the medical domain for potential anesthesia diagnosis in the operating room [3]. Since it has 37 nodes and 46 directed edges, it is a complex network which is widely used for evaluating the performance of a Bayesian network learning algorithm. Examples include the K2 algorithm [16], the CB algorithm [62], the BENEDICT algorithm [1], and MDLEP [64].



1. central venous pressure
2. pulmonary capillary wedge pressure
3. history of left ventricular failure
4. total peripheral resistance
5. blood pressure
6. cardiac output
7. heart rate obtained from blood pressure
8. heart rate obtained from electrocardiogram
9. heart rate obtained from oximeter
10. pulmonary artery pressure
11. arterial-blood oxygen saturation
12. fraction of oxygen in inspired gas
13. ventilation pressure
14. carbon-dioxide content of expired gas
15. minute volume, measured
16. minute volume, calculated
17. hypovolemia
18. left-ventricular failure
19. anaphylaxis
20. insufficient anesthesia or analgesia
21. pulmonary embolus
22. intubation status
23. kinked ventilation tube
24. disconnected ventilation tube
25. left-ventricular end - diastolic volume
26. stroke volume
27. catecholamine level
28. error in heart rate reading due to low cardiac output
29. true heart rate
30. error in heart rate reading due to electrocautery device
31. shunt
32. pulmonary-artery oxygen saturation
33. arterial carbon-dioxide content
34. alveolar ventilation
35. pulmonary ventilation
36. ventilation measured at endotracheal tube
37. minute ventilation measured at the ventilator

Figure 5.5: The ALARM network [43].

One data set is generated from the ASIA network with 1,000 cases. Its structure is shown in Figure 5.4. ASIA network is a relatively simple structure that contains eight nodes and eight edges. The network is also known as the "chest-clinic" network which describes a fictitious medical example on whether a patient has tuberculosis, lung cancer, or bronchitis, related to the attributes (X-ray, dyspnea, visit-to-Asia, and smoking) of the patient [29, 40].



1. Spool Process OK
2. Local Disk Space Adequate
3. Application Output OK
4. Spooled Data OK
5. Print Spooling On
6. GDI Data Input OK
7. Correct Driver
8. Uncorrupted Driver
9. Correct Driver Settings
10. GDI Data Output OK
11. Correct Printer Selected
12. Correct Local Port
13. Local Cable Connected
14. Network Up
15. Correct Printer Path
16. Network Cable Connected
17. Print Data OK
18. Network Path OK
19. Network/Local Printing
20. Local Path OK
21. PC to Printer Transport OK
22. Printer On and Online
23. Paper Loaded
24. Printer Memory Adequate
25. Printer Data OK
26. Printer Output OK

Figure 5.6: The PRINTD network [43].

Another data set with 5,000 cases is generated from the PRINTD network. The PRINTD network is primarily constructed for troubleshooting printer problems in the $Windows^{TM}$ operating system [21]. The structure of the network is shown in Figure 5.6. It has 26 nodes and 26 edges.

Since both algorithm are stochastic in nature, we conducted 40 trials for each experiment. The programs were executed on the same Nix dual Intel Xeon 2.2GHz Linux machine. For both algorithms, we set $\triangle_\alpha$ to be 0.02 and the initial population size to 50. The $\triangle_\alpha$ refer to the change of α value for creating a new network in HEP. For A-HEP, the maximum and the minimum population sizes were 100 and 3 respectively. The *far-factor* and *cutoff-distance* were set to be 0.8 and 1. The maximum number of generations is 5000. The results are summarized in Fig. 5.4.1, Table 5.1 and 5.2.

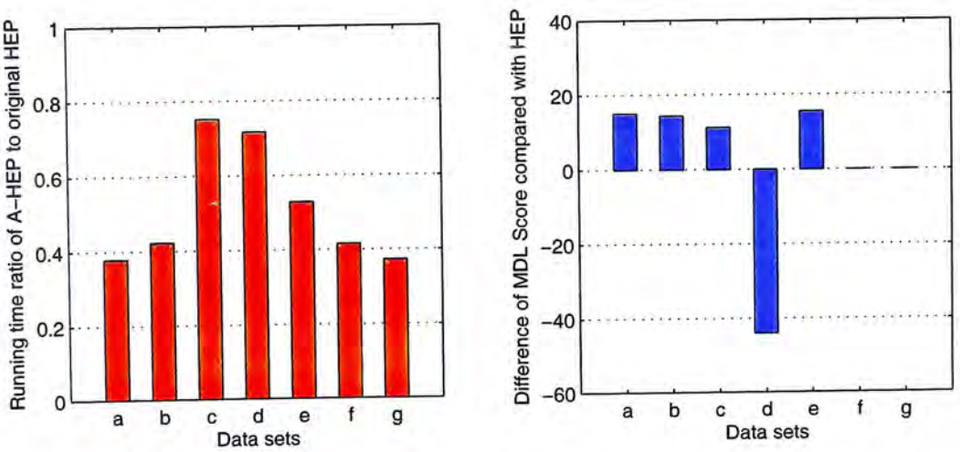


Figure 5.7: Data set (a)alarm-1000, (b)alarm-2000, (c)alarm-5000, (d)alarm-10000, (e)alarm-O (original), (f)asia-1000 and (g)printd-5000; Left: Comparison on average running time in each data set; Right: Comparison on the average fitness (MDL score) of final Bayesian network obtained

5.4.2 Comparison on Running Time

In our experiments, the running time of A-HEP is 20-60% less than the original HEP. This improvement is due to the adaptive population size concept. Once the population converges into a certain degree of similarity, redundant individuals are removed from the population. Therefore, the computation effort for later generations are greatly reduced and the running time is shortened. Table 5.1 shows the average running time for each data set. Average improvement of A-HEP is around 1.96 times faster.

This speed-up is particularly significant for small data sets, such as asia1000 and alarm1000. At the later stage of evolution, the individuals in the population are very similar or even exactly the same. The decreasing routine removes these individuals from the population dynamically, and increases the efficiency of the algorithm. By adjusting the far-factor and population size limits, the running time can be further reduced. The tuning of parameters will be described later.

Data set	Average Running Time(second)		
	HEP	A-HEP	Ratio (A-HEP/HEP)
alarm1000	53.08 ±1.14	20.10 ±4.67	0.38
alarm2000	56.05 ±0.51	23.70 ±6.01	0.42
alarm5000	63.38 ±0.9	47.68 ±11.07	0.75
alarm10000	75.97 ±1.84	54.50 ±5.92	0.72
alarm-O	102.65 ±25.25	54.33 ±11.18	0.53
asia1000	8.18 ±0.38	3.40 ±0.50	0.42
printd5000	38.33 ±0.62	14.30 ±1.34	0.37
Average	-	-	0.51

Table 5.1: Performance comparison between HEP and A-HEP on running time

5.4.3 Comparison on Fitness of Final Network

Although A-HEP can learn the Bayesian network structure in a shorter time, the quality of the final networks are also important. Therefore, the average fitness (MDL score) of the best network obtained by each algorithm are compared. Table 5.2 shows the MDL scores of the final networks obtained by both algorithms.

In real trials, both algorithms can obtain the individual with minimum MDL score in most cases. The differences of MDL score on Table 5.2 is mainly due to obtaining sub-optimal structures in particular trials. However, those MDL score differences are insignificant. A-HEP can even obtain fitter Bayesian network in more cases than HEP in alarm-10000 data set. Therefore, we can conclude that A-HEP has comparable BN structure learning performance as HEP. With shorter running time and comparable quality of final network structure obtained, A-HEP is more efficient than the state-of-the-art Bayesian network learning algorithm - HEP.

Data set	Average MDL Score of final network		
	HEP	A-HEP	Difference (A-HEP - HEP)
alarm1000	17862.48 ±19.68	17877.48 ±28.31	15.01 (0.08%)
alarm2000	33773.05 ±3.14	33787.45 ±56.24	14.39 (0.04%)
alarm5000	81004.00 ±0.0	81015.22 ±68.18	11.22 (0.01%)
alarm10000	158517.54 ±247.02	158473.53 ±90.45	-44.01 (-0.28%)
alarm-O	138549.48 ±385.83	138564.95 ±405.60	15.48 (≈0%)
asia1000	3398.66 ±0.16	3398.60 ±0.00	-0.06 (≈0%)
printd5000	106542.00 ±0.00	106542.00 ±0.00	0.00 (0%)
Average	-	-	1.72

Table 5.2: Performance comparison between HEP and A-HEP on fitness of final network

5.4.4 Comparison on Similarity to the Original Network

Besides the MDL Score (Fitness of final network) and running time, there are other metrics to evaluate the performance of A-HEP and original HEP. Among them, the structural difference between the learned structure and the original structure is the more important. A good Bayesian network learning algorithm should minimize this difference and obtain an accurate network structure. Table 5.3 shows the performance comparison between A-HEP and HEP. Two approaches for calculating the structural difference are used. The first one is simply counting the number of edge difference between the final solution and the original network. The second one is counting the number of edge difference between final solution and the equivalent class of original network, i.e. a set of networks representing the same data distribution.

The smaller the value, the better the performance. Here we can see that A-HEP performs better in large data set alarm-O. That means the final network structure found is more similar to the original one, than the one obtained by original HEP. On the other hand, the HEP performs better in other smaller data sets. However, the differences are insignificant. Therefore, we can conclude that the A-HEP have comparable performance to the original HEP.

	Structural Difference		Structural Difference (Eq.class)	
Data set	HEP	A-HEP	HEP	A-HEP
alarm1000	10.75	12.25	14.25	14.70
alarm2000	7.18	7.73	7.21	7.55
alarm5000	7.10	7.70	6.00	6.25
alarm10000	4.59	5.53	4.49	5.55
alarm-O	9.28	7.55	8.83	6.73
asia1000	2.75	3.00	2.75	3.00
printd5000	0.00	0.00	0.00	0.00
Average	5.95	6.25	6.22	6.25

Table 5.3: Performance comparison between HEP and A-HEP on the similarity to the original network

5.4.5 Parameter Study

In the previous section, A-HEP is shown to be efficient for learning Bayesian network structures. In our algorithm, we have a set of parameters affecting the performance and population size variation. Here, we have a more comprehensive study on these parameters.

The important core of A-HEP are the increasing routine and decreasing routine. Some of the following parameters are taken into account in both routines, including limits on maximum and minimum population size, initial population size and the far-factor used for adding new individuals into population. The following experiments use the same data sets and methodology as the previous section, but parameter of different values are studied.

Maximum population size

In the increasing routine of A-HEP, a random number is generated to compare against the the ratio of current population size to the maximum population size limit. It controls the degree of expansion of population size in early stage of evolution. The maximum population size is set to 100 (2×50, the initial population size) in the previous experiments.

	Running Time Ratio				Diff. of Fitness of Final Network			
Max. pop.size	75	**100**	125	150	75	**100**	125	150
Data set								
alarm-1000	0.36	0.38	0.42	0.49	20.21	15.01	10.86	21.77
alarm-2000	0.42	0.42	0.45	0.57	11.64	14.39	-1.91	-0.89
alarm-5000	0.78	0.75	0.99	1.20	13.31	11.22	0.00	0.00
alarm-10000	0.69	0.72	0.76	1.40	-21.14	-44.01	-37.44	-41.34
alarm-O	0.54	0.53	0.55	0.56	-0.95	15.48	41.55	106.77
asia-1000	0.38	0.42	0.46	0.48	-0.06	-0.06	-0.06	-0.06
printd-5000	0.36	0.37	0.37	0.38	0.00	0.00	0.00	0.00
Average	0.50	0.51	0.57	0.73	3.29	1.72	1.86	12.32

Table 5.4: Parameter study on maximum population size

Table 5.4 shows that the running time increases with the maximum pop-

ulation size limit. Obviously, the larger the limit, the more new individuals can be added in the early stage of evolution. Therefore, more computation time is required for each generation. However, simply choose a smaller limit, like 75, we may not get the optimal solution. It is because the expansion of population is not able to meet sufficient diversity, so that it is trapped in local optimal solution. From the analysis of above experiments, the maximum population size limit should be set to two fold of the initial population size (50).

Minimum population size

Similarly, there is a random number which is compared against the ratio of current population size to the minimum population size limit in the decreasing routine of A-HEP. It controls the degree of contraction of population size in the later stage of evolution.

	Running Time Ratio				Diff. of Fitness of Final Network			
Min. pop.size	1	**3**	5	10	1	**3**	5	10
Data set								
alarm-1000	0.38	0.38	0.39	0.48	15.01	15.01	13.89	13.71
alarm-2000	0.42	0.42	0.44	0.53	14.39	14.39	14.39	14.39
alarm-5000	0.76	0.75	0.75	0.78	11.22	11.22	11.22	11.22
alarm-10000	0.72	0.72	0.72	0.75	-44.01	-44.01	-44.86	-48.96
alarm-O	0.53	0.53	0.53	0.54	15.48	15.48	19.00	12.77
asia-1000	0.41	0.42	0.42	0.47	-0.06	-0.06	-0.06	-0.06
printd-5000	0.35	0.37	0.41	0.53	0.00	0.00	0.00	0.00
Average	0.51	0.51	0.52	0.58	1.72	1.72	1.94	0.44

Table 5.5: Parameter study on minimum population size

Table 5.5 shows that the running time increases with the minimum population size limit. In the later stage of evolution, the population usually shrinks to the minimum size limit. Therefore, more computation time is required for each generation where the limit is larger. On the contrary, a large limit can prevent early convergence and improve the solution quality.

Therefore, a balancing exercise must be made between these two factors. In our experiments, the minimum population size is set to 3.

Initial population size

The main difference between A-HEP and original HEP is that the former can change the population size adaptively according to the dissimilarity between individuals to increase efficiency. Original HEP has a fixed population size throughout the evolution. For fair comparison, the initial population size is set to 50 for both algorithms in the previous experiments. Since A-HEP is able to vary population adaptively, this section attempts to investigate the effect of initial population size to A-HEP. The original initial population size, 50, is still used for HEP algorithm as the reference.

	Running Time Ratio					Diff. of Fitness of Final Network				
Initial pop.	10	20	30	40	**50**	10	20	30	40	**50**
Data set										
alarm-1000	0.31	0.32	0.36	0.37	0.38	58.85	34.82	39.49	24.93	15.01
alarm-2000	0.34	0.34	0.39	0.41	0.42	53.51	23.64	5.07	14.91	14.39
alarm-5000	0.59	0.65	0.70	0.72	0.75	58.80	42.84	6.64	4.46	11.22
alarm-10000	0.59	0.63	0.66	0.68	0.72	114.31	13.11	6.39	-52.14	-44.01
alarm-O	0.41	0.45	0.45	0.46	0.53	142.42	263.30	93.05	34.52	15.48
asia-1000	0.38	0.39	0.39	0.41	0.42	0.01	-0.04	-0.06	-0.06	-0.06
printd-5000	0.32	0.33	0.35	0.35	0.37	0.00	0.00	0.00	0.00	0.00
Average	0.42	0.45	0.47	0.49	0.51	61.13	53.95	21.51	3.80	1.72

Table 5.6: Parameter study on initial population size

Table 5.6 shows that the running time increases with the initial population size as the running time depends highly on population size. However, the quality of Bayesian network learned is not satisfactory for small initial population size. The possible reason may be insufficient divergence in the population throughout the evolution. The search space is not explored extensively and the algorithm is trapped in local optimum. On the other hand, the increasing routine is not adaptive enough for exploring the search space.

There are rooms for improvement in this situation. With the correct use of initial population size, A-HEP can still work efficiently.

Far-factor

In the increase routine of A-HEP, the new offspring is added if it is very different from the the current population. If its average dissimilarity to the individuals of current population is larger than the threshold, the *far-factor*, then the new offspring is added into the population regardless its fitness. This approach can increase the diversity of the population in the early stage of evolution.

	Running Time Ratio					Diff. of Fitness of Final Network				
far-factor	0.6	0.7	**0.8**	0.9	1.0	0.6	0.7	**0.8**	0.9	1.0
Data set										
alarm-1000	0.39	0.39	0.38	0.40	0.39	16.37	16.35	15.01	14.21	12.10
alarm-2000	0.42	0.42	0.42	0.45	0.45	11.13	11.64	14.39	-1.00	-1.26
alarm-5000	0.78	0.77	0.75	0.77	0.76	0.21	5.04	11.22	15.83	5.04
alarm-10000	0.77	0.69	0.72	0.72	0.73	-49.11	-34.84	-44.01	-81.79	-76.31
alarm-O	0.51	0.51	0.53	0.50	0.49	130.23	68.32	15.48	82.20	81.45
asia-1000	0.57	0.49	0.42	0.35	0.28	-0.06	-0.06	- 0.06	-0.06	-0.06
printd-5000	0.38	0.37	0.37	0.37	0.37	0.00	0.00	0.00	0.00	0.00
Average	0.54	0.52	0.51	0.51	0.50	15.54	9.49	1.72	4.20	2.99

Table 5.7: Parameter study on far-factor

As shown in Table 5.7, the running time is not greatly affected by the value of far-factor. The insignificant drop is caused by the decrease of average population size in the early stage of evolution. The larger the far-factor, the more difficult it is for the routine to add new individuals. For the quality solution, a correct value of far-factor should be used in order to get an optimal Bayesian network structure. When a smaller value is used, it is easier to add new offspring. The population size reaches a larger value or even attends maximum in the early stage of evolution, and those potential good individuals cannot be added to the population in the later stage of evolution.

When a larger value of far-factor is used, the ease of adding new offsprings decreases. The population does not able to get sufficient diversity, so that it is trapped in local optimum. Therefore, a suitable value of far-factor is important for finding a better solution.

5.5 Applications on Medical Domain

As discussed in chapter 2, applying various data mining techniques on medical domain to discover knowledge in database and/or to develop a decision support system has become a trend. Bayesian networks have an important role to play. This section investigates the use of our proposed A-HEP on medical domain.

5.5.1 Discussion

In recent years, a number of researches on knowledge discovery have been done in medical domain. Among the knowledge discovery models, Bayesian network (BN) is a popular choice as it can represent the causal relationship of different events or attributes with probability. It have emerged as some of the most successful tools for medical diagnostics and many have been deployed in real medical environments or implemented in off-the-shelf diagnostic software [55]. BN is widely used because of its ability to encode the probabilistic relationships among variables, and efficiency and flexibility in inference.

In Hong Kong, researchers have applied evolutionary algorithms to discover knowledge from medical databases successfully. They used MDLEP and genetic algorithm to learn the Bayesian network structure for the fracture database and Scoliosis database. Genetic algorithm has been used on the learning of discretization policies on variables. [64][51].

Our proposed algorithm A-HEP is an optimized version of HEP, in the mean time the HEP is an extension of MDLEP. The common characteristic

Name	Type	Explanation
Sex	Nominal	Sex
Age	Numeric	Age(between 0 to 16 years old)
Admday	Date	Admission day(between 1984 to 1996)
Stay	Numeric	Length of staying in hospital(in days)
Diagnosis	Nominal	Diagnosis of fracture based on the fracture location
Operation	Nominal	Operation
Surgeon	Nominal	Surgeon (null if no operation)
Side	Nominal	Side of fracture("Left", "Right", "Both" or "Missing")

Table 5.8: Attributes in the Fracture Database

of these algorithms is using evolutionary algorithms as the search-and-score approach for learning Bayesian network structure. Therefore, applying A-HEP on medical applications is expected to be feasible and effective.

5.5.2 An Example

Since most medical data sets contain patients' private personal data and clinical records, there are not many medical data sets available in public archives. Those data sets are sometimes treated as intellectual properties of research projects as well. With the help of authors of MDLEP, we can obtain the Fracture database used in their paper [64][51].

Problem Definition

Fracture is a medical database obtained from Orthopaedic Department of Prince of Wales Hospital of Hong Kong. It consists of children with limb fractures admitted to the hospital in the period 1984-1996. The data can provide information for the analysis of fracture pattern. Fracture database has over 6500 records and eight attributes, which are listed in Table 5.8.

In those papers, authors proposed to learn discretization policy using genetic algorithms. Since MDLEP, HEP and A-HEP also use MDL metric as the fitness of the network structure, the GA discretization policy obtained from the papers can still be used in our example. Table 5.9 shows the dis-

Age : [0-4] [5-9] [10-12] [13-16]
Year : [1984-1987] [1988-1991] [1992-1996]
Stay : [0-3] [4-12] [13-1081]

Table 5.9: Discretization Policy of the Fracture Database

cretization range for different attributes. For the *Admday* attribute, *Day* and *Month* values are discretized into one range, which means that only one value is considered. As a typical medical data set, there are some missing values for each attributes. For simplicity, we remove those records containing missing values in any attribute from our testing data. The size of the Fracture data set becomes 5294 records.

In the following experiments, A-HEP is compared to HEP and MDLEP on the Bayesian network structure learning performance on the Fracture data set. The result of MDLEP is directly obtained from the paper, while A-HEP and HEP are run to get the result. Since both algorithm are stochastic in nature, we conducted 40 trials for each experiment. The programs were executed on the same Nix dual Intel Xeon 2.2GHz Linux machine. For both algorithms, we set $\triangle_\alpha$ to be 0.02 and the initial population size to 50. For A-HEP, the maximum and the minimum population sizes were 100 and 3 respectively. The *far-factor* and *cutoff-distance* were set to be 0.8 and 1. The maximum number of generations is 5000.

Experimental Results

The Bayesian network structure obtained by MDLEP stated in the papers is shown in Fig. 5.8. In our experiments, the structure obtained from reduced Fracture data set by A-HEP and HEP are exactly the same. It is shown in Fig. 5.9. Comparing these two structures, an edge is removed between attribute *Operation* and *Year*, and the edge between *Operation* and *Diagnosis* is reversed. The cause of these changes may be due to the removal of records with missing values in the Fracture data set. However, incomplete

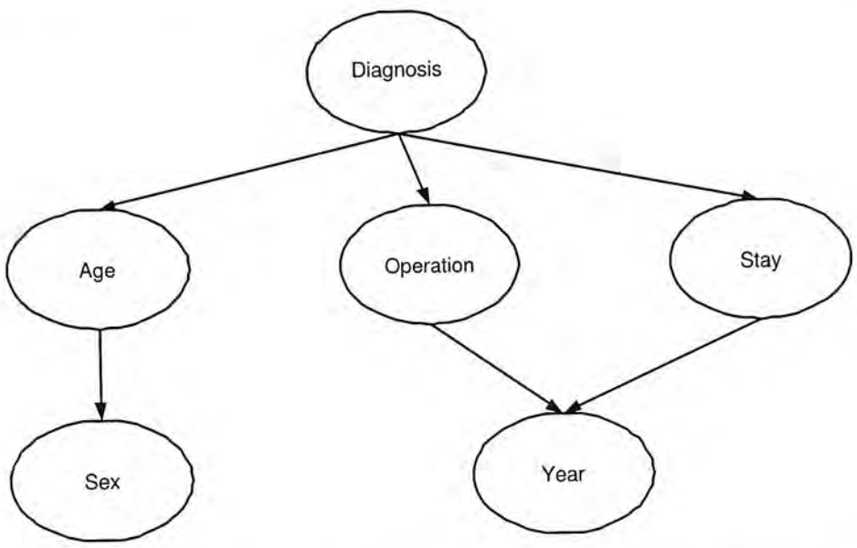


Figure 5.8: Bayesian network obtained by MDLEP from Fracture data set

data set is a common problem in medical data mining. It is still a popular research topic in this field [55].

From the network structure constructed by A-HEP, the following relationship are observed.

- The value of *Operation* affects the values of *Diagnosis*. This may be odd in some sense. As described in previous paragraph, this phenomenon is resulted from data preprocessing. However, it still shows that these two attributes are interrelated. Different fractures are treated with different operations.

- The value of *Diagnosis* affects the values of *Stay*. Different fractures require different time of recovery.

- The value of *Diagnosis* affects the values of *Age*. Some fractures are more frequently occur in particular age groups.

- The value of *Age* affects the value of *Sex*. It is observed that the young patients are more likely to be female, and older patients are more likely

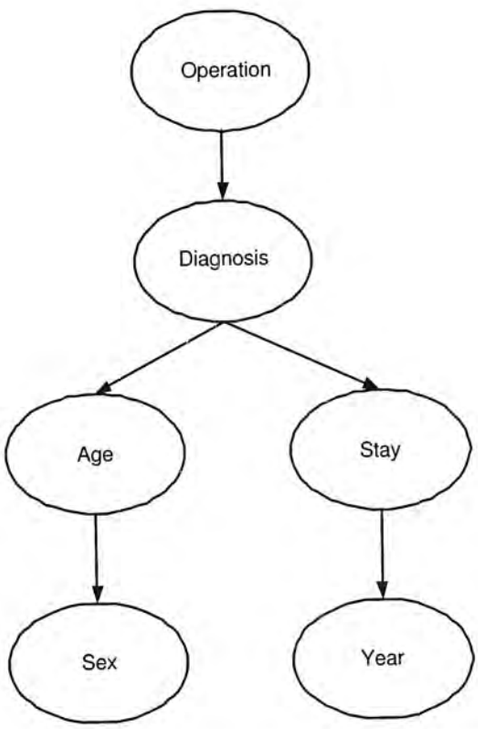


Figure 5.9: Bayesian network obtained by HEP and A-HEP from Fracture data set

to be male.

- The value of *Operation* and *Stay* affects the value of *Year*. It is observed from the database that length of stay in hospital is longer in the years 1985, 1986 and 1994.

	Fitness of Final Network (MDL Score)	Running Time(second)
HEP	49780.40	6.58
A-HEP	49780.40	5.10
Differnce/Ratio	0	77.57%

Table 5.10: Performance comparison between HEP and A-HEP on the Fracture data set

In order to prove the efficiency of proposed A-HEP, its running time is compared with that of HEP. The results are shown in Table 5.10. It is quite consistent with the previous experimental results. The A-HEP is more efficient than original HEP for learning Bayesian network structure. The improvement is around 78% of original HEP which is below the average improvement 50% of original HEP. The possible reason is the small size of the Fracture data set with small number of attribute. A-HEP has better performance in large data sets and complicate problems.

In this section, the performance of proposed A-HEP is evaluated by applying it to solve real-life medical problem. Compared to existing algorithms, it has shown that A-HEP has comparable BN learning ability but shorter running time. It will be one of the excellent algorithms for learning Bayesian network structure and mining causal information in complicated real-life problems.

5.6 Summary

In this chapter, the adaptive population size evolutionary algorithm, A-HEP, for learning Bayesian network structures has been presented. This is

an optimized version of HEP which is one of the state-of-the-art algorithms of this type. The technique is based on the concept of adjusting population size adaptively according to the dissimilarity of individuals in current population. With the use of increasing and decreasing routines, the population expands in early generations for increasing diversity, and contracts in later generations for reducing computation time. A-HEP has been experimentally tested with several data sets of different sizes and numbers of variables. The performance of it is compared with the original HEP on running time and quality of Bayesian network obtained. All experiments have demonstrated that A-HEP consistently and significantly reduces the running time with comparable performance on Bayesian network learning. This speed-up is very important as A-HEP can be used efficiently for learning Bayesian networks on many data mining problems.

At the end of this chapter, A-HEP has also demonstrated its applicability on medical data mining. Compared to original HEP, similar result can be obtained but with a great speed improvement.

In order to further optimize the learning performance of A-HEP, we are going to design a new operator for crossover between two individuals. We also want to investigate the feasibility of applying dynamic population size concept in other algorithms.

□ **End of chapter.**

Chapter 6

Conclusion

We conclude our work with a summary of our contributions and discuss some possible future research directions.

6.1 Summary

At the beginning of this thesis, we describe the trend of medical data mining in the world with the Hepatitis B virus genome project as an example. The objective of this project is to find the genetic and clinical markers for hepatocellular carcinoma (HCC) from the virus DNA sequences and clinical data. Based on this project, the use of different machine learning and data mining models and techniques are investigated. Our research work can be summarized into three major parts.

Clinical data mining is the first part of the work. It focuses on mining the inter-relationship among those clinical attributes, as well as their contributions to liver cancer. Among various machine learning models, Bayesian network is chosen because of its representation on the casual dependency with probability. Bayesian network classifiers are also popular classification models with satisfactory performance. In Chapter 3, the learning algorithms for Bayesian-augmented Naïve-Bayes classifier (BAN) and General

Bayesian Network classifier (GBN) have been proposed. In these algorithms, some modifications are made on HEP and the Markov Blanket concept is introduced. For performance evaluation, the classifiers learned by proposed algorithms have been tested by experiments on benchmark data sets, as well as a real-life clinical data set. The experimental results show that both models are satisfactory for classification. They can also discover the inter-relationship among the attributes. By observation, the BAN and GBN are especially useful for data sets with larger number of attributes. In addition, an easily-missed error on conditional probability table calculation has been reported. This error is mainly caused by the zero entry of CPT. Feasible solutions are proposed and used in our experiments.

The second part of this thesis concentrates on DNA sequence analysis. Based on our project, we develop a general framework for DNA sequence analysis aimed for classification. This framework includes modules of data preprocessing, clustering, feature selection, and classification model. Each module serves its functions in the architecture with a certain flexibility for customization. In Chapter 4, the framework is described in detail with various suggestions for each module. The importance of independent test on evaluation is also mentioned. The HBV DNA analysis work has also been carried out under this framework. It is a pioneer project in the field of Bioinformatics. In the clustering step, we have discovered several important findings such as the existence of subgroups in HBV virus. Genetic markers which can guarantee over 70% of accuracy and sensitivity have been found. These results are great contributions to the biochemistry and medical fields.

Since Bayesian network (BN) is a major data mining model we used in this thesis, further research on the optimization of one of BN learning algorithm, HEP, have been conducted. The success of the optimized algorithm is based on the concept of adjusting population size adaptively according to the dissimilarity of individuals in current population. Using an increas-

ing and an decreasing routines, the population expands in early generations for increasing diversity, and contracts in later generations for reducing computation time. Our A-HEP has been experimentally tested with several data sets of different sizes and numbers of variables. All experiments have demonstrated that A-HEP consistently and significantly reduces the running time by half on average, with comparable performance on Bayesian network learning. Moreover, We have also demonstrated its applicability of A-HEP on medical data mining with real-life data set.

All the above work demonstrate the feasibility of using computer science technology to solve medical and biochemical problems, as well as its efficiency and power.

6.2 Future Work

As part of our future work, the research work in the HBV genome project is continued. The second phase of this project concentrates on the relation between virus DNA and the drug response to the Lamivudine therapy. The proposed framework may be used with some modifications in the next phase of study.

At this stage, the clinical data and genetic markers are analyzed separately. Combining both of them may yield a better classifier of higher accuracy and sensitivity. It is a worthy direction for future research.

Most of the research results of our project have not been published yet. A special task force is recently established for the patent application of specific feature sites and methodologies we have used in this project.

□ **End of chapter.**

Bibliography

[1] S. Acid and L. M. de Campos. BENEDICT:an algorithm for learning probabilistic belief networks. In *Proceedings of the IPMU-96 Conference*, 1996.

[2] P. R. B. and D. S. A sofm approach to predicting hiv drug resistance. 2002. Proc. of Pacific Symposium on Biocomputing, PSB 2002, Kaua'i Marriott, Kaua'i, Hawaii.

[3] I. Beinlinch, H. Suermondt, R. Chavez, and G. Cooper. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *Proceedings of Second European Conference Artificial Intelligence in Medicine*, pages 247–256, 1989.

[4] P. Bertone and M. Gerstein. Integrative data mining: the new direction in bioinformatics. *IEEE Engineering Medicine Biology Magazine*, 20(4):33–40, Jul-Aug 2001.

[5] W. Buntine. A guide to the literature on learning probabilistic networks from data. *IEEE Transactions on Knowledge and Data Engineering*, 8(2):195–210, April 1996.

[6] J. Cheng, D. A.Bell, and W. Liu. An algorithm for bayesian belief network construction from data. In *Proceedings of the 6th International Workshop on Artificial Intelligence and Statistics*, 1997.

[7] J. Cheng, D. A.Bell, and W. Liu. Learning belief networks from data: an information theory based approach. In *Proceedings of the Sixth ACM International Conference on Information and Knowledge Management*, 1997.

[8] J. Cheng and R. Greiner. Comparing bayesian network classifiers. In *Proceedings of the fifteenth conference on uncertainty in artificial intelligence*, pages 101–108, 1999.

[9] J. Cheng and R. Greiner. Learning bayesian belief network classifiers: Algorithms and system. In *Proceedings of the fourteenth Canadian conference on artificial intelligence*, volume LNCS 2056, pages 141–151, 2001.

[10] J. Cheng, R. Greiner, J. Kelly, D. Bell, and W. Liu. Learning Bayesian network from data: An information-theory based approached. *Artificial Intelligence*, 137:43–90, 2002.

[11] D. M. Chickering, D. Geiger, and D. Heckerman. Learning bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:197–243, March 1995.

[12] D. M. Chickering, D. Heckerman, and C. Meek. A bayesian approach to learning bayesian networks with local structure. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, pages 80–89. Morgan Kaufmann, August 1997.

[13] L. Chrisman. A roadmap to research on bayesian networks and other decomposable probabilistic models. Technical report, School of Computer Science, CMU, May 1996.

[14] A. Ciancio, A. Smedile, and M. Rizzetto. Identification of hbv dna sequences that are predictive of response to lamivudine therapy. *Hepatology*, 39:64–73, 2004.

[15] K. J. Cios and G. W. Moore. Uniqueness of medical data mining. *Artificial Intelligence in Medicine*, 26(1-2):1–24, Sep-Oct 2002.

[16] G. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–347, 1992.

[17] G. F. Cooper. Computational complexity of probabilistic inference using bayesian belief networks. *Artificial Intelligence*, 42(2-3):393–405, March 1990.

[18] N. Friedman. Building classifiers using bayesian networks. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-1996)*, pages 1277–1284, 1996.

[19] A.-F. Gustavo and S. Luis. A temporal bayesian network for diagnosis and prediction. In *Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 13–20, San Francisco, CA, 1999. Morgan Kaufmann Publishers.

[20] D. Heckerman. A tutorial on learning Bayesian networks. Technical report, Microsoft Research, Advanced Technology Division, March 1995.

[21] D. Heckerman and M. P. Wellman. Bayesian networks. *Communications of the ACM*, 38(3):27–30, March 1995.

[22] http://jbnc.sourceforge.net/ (July 2004). jbnc, bayesian network classifier toolbox.

[23] *http* : *//www.doc.ic.ac.uk/ nd/surprise*$_{9}$6*/journal/vol*4*/cs*11*/report.html* (July 2004). Neural networks, c. stergiou and d. siganos.

[24] http://www.ebi.ac.uk/clustalw/ (July 2004). European bioinformatics institute (embl-ebi), clustalw.

[25] http://www.hepatitis-central.com/hbv/hepbfaq/short.html (July 2004). Hepatitis-central.com, introduction to hepatitis b.

[26] http://www.hepb.org/02-0059.hepb (July 2004). Hepatitis b foundation, what is hepatitis b.

[27] http://www.ics.uci.edu/ mlearn/MLRepository.html (July 2004). Uci machine learning repository.

[28] http://www.nlm.nih.gov/medlineplus/ency/imagepages/1031.htm (July 2004). A.d.a.m. inc, medlineplus medical encyclopedia: Hepatitis b virus.

[29] http://www.norsys.com/net_library.htm (July 2004). Norsys Bayes net library.

[30] http://www.rulequest.com/see5-info.html (July 2004). Rulequest research (1997-2004), data mining tools see5 and c5.0.

[31] http://www.sgi.com/tech/mlc/index.html (July 2004). Sgi mlc++.

[32] K. Huang, I. King, and M. R.Lyu. Learning maximum likelihood semi-naive bayesian network classifier. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics (SMC2002)*, 2002.

[33] M. Hunt, B. von Konsky, S. Venkatesh, and P. Petros. Bayesian networks and decision trees in the diagnosis of female urinary incontinence. In *Proceedings of the 22nd Annual EMBS International Conference*, 2000.

[34] J.C.G.Ramirez, D. Cook, L.L.Peterson, and D.M.Peterson. Temporal pattern discovery in course-of-disease data. *IEEE Engineering in Medicine and Biology Magazine*, 19(4):63–71, Jul/Aug 2000.

[35] J.F.Roddick and M.Spiliopoulou. A survey of temporal knowledge discovery paradigms and methods. *IEEE Transactions on Knowledge and Data Engineering*, 14(4), 2002.

[36] E. J.Keogh and M. J.Pazzani. Learning augmented bayesian classifiers: A comparison of distribution-based and classification-based approaches. *International Journal on Artificial Intelligence Tools*, 11(4):587–601, 2002.

[37] K.C.Tan, Q.Yu, C.M.Heng, and T.H.Lee. Evolutionary computing for knowledge discovery in medical diagnosis. *Artificial Intelligence in Medicine*, 27(2):129–154, February 2003.

[38] M. Kukar, I. Kononenko, C. Groselj, K. Kralj, and J. Fettich. Analysing and improving the diagnosis of ischaemic heart disease with machine learning. *Artificial Intelligence in Medicine*, 16(1):25–50, 1999.

[39] P. Larrañaga, M. Poza, Y. Yurramendi, R. Murga, and C. Kuijpers. Structural learning of Bayesian network by genetic algorithms: A performance analysis of control parameters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9):912–926, September 1996.

[40] S. L. Lauritzen and D. J. Spieglhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of Royal Statistics Society*, 50(2):157–194, 1988.

[41] N. Lavrac. Selected techniques for data mining in medicine. *Artificial Intelligence in Medicine*, 16(1):3–23, 1999.

[42] K. Y. Lee, M. L. Wong, Y. Liang, K. S. Leung, and K. H. Lee. A-HEP: Adaptive hybrid evolutionary programming for learning bayesian networks. In M. Keijzer, editor, *Late Breaking Papers at the 2004 Genetic and Evolutionary Computation Conference*, Seattle, Washington, USA, 26 July 2004.

[43] S. Y. Lee. Learning bayesian networks using evolutionary computation and its application in classification. Master's thesis, Department of Computer Science and Engineering, The Chinese University of Hong Kong, 2001.

[44] K. S. Leung and Y. Liang. Adaptive elitist-population based genetic algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, volume LNCS 2723, pages 1160–1171, 2003.

[45] K. S. Leung, M. L. Wong, W. Lam, Z. Wang, and K. Xu. Learning nonlinear multiregression networks based on evolutionary computation. *IEEE Transactions on Systems, Man and Cybernetics Part B*, 32(5):630–644, 2002.

[46] F. Lin, C. Chiu, and S. Wu. Mining time dependency patterns in clinical pathways. In *IEEE Proceedings of the 35th Hawaii International Conference on System Sciences*, 2002.

[47] F. Lin, C. Chiu, and S. Wu. Using bayesian networks for discovering temporal-state transition patterns in hemodialysis. In *IEEE Proceedings of the 35th Hawaii International Conference on System Sciences*, 2002.

[48] C. Lu, V. Gestel, J. T. Suykens, and S. V. Huffel. Preoperative prediction of malignancy of ovarian tumors using least squares support vector machines. *Artificial Intelligence in Medicine*, 28(3):281–306, July 2003.

[49] P. Lucas. Restricted bayesian network structure learning. Technical report, Institute for Computer and Information Sciences. University of Nijmegen Toernooiveld 1, 2002.

[50] M. G. Madden. Evaluation of the performance of the markov blanket bayesian classifier algorithm. Technical Report NUIG-IT-011002, Department of Information Technology, National University of Ireland, 2002.

[51] K. S. L. Man Leung Wong, Wai Lam, P. S. Ngan, and J. Cheng. Discovering knowledge from medical databases using evolutionary algorithms. *IEEE Engineering in Medicine and Biology Magazine*, 19(4):45–55, July/Aug 2000.

[52] J. W. Myers, K. B. Laskey, and K. A. DeJong. Learning bayesian networks from incomplete data using evolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 1999.

[53] NCBI. Genebank.

[54] Y. T. Ng. Optimizing performance of classification on biological data sets using evolutionary algorithm. M.Phil. Term 2 Report, Department of Computer Science and Engineering, CUHK.

[55] D. Nikovski. Constructing bayesian networks for medical diagnosis from incomplete and partially correct statistics. *IEEE Transactions on Knowledge and Data Engineering*, 12(4):509–516, July 2000.

[56] J. Pearl. *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[57] A. R. Riccardo Bellazzi. Learning bayesian networks probabilities from longitudinal data. *IEEE Transactions on Systems, Man and Cybernetics Part A*, 28(5):629–636, Sept 1998.

[58] R.Sterritt, A.H.Marshall, C.M.Shapcott, and S.I.McClean. Exploring dynamic bayesian belief networks for intelligent fault management systems. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, pages 3646–3652, 2000.

[59] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1st edition, January 1995.

[60] A. Sboner, C. Eccher, and E. Blanzieri. A multiple classifier system for early melanoma diagnosis. *Artificial Intelligence in Medicine*, 27(1):29–44, January 2003.

[61] W. Siedlecki and J. Sklansky. A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters*, 10(5):335 – 347, November 1989.

[62] M. Singh and M. Valtorta. An algorithm for the construction of Bayesian network structures from data. In D. Heckerman and E. H. Mamdani, editors, *Proceedings of the Ninth Conference of Uncertainty in Artificial Intelligence*, pages 259–265, San Mateo, CA, 1993. Morgan Kaufmann.

[63] S. Souafi-Bensafi, M. Parizeau, F. Lebourgeois, and H. Emptoz. Bayesian networks classifiers applied to documents. In *16 th International Conference on Pattern Recognition (ICPR'02) Volume 1*, page 10483, 2002.

[64] M. L. Wong, W. Lam, and K. S. Leung. Using evolutionary programming and minimum description length principle for data mining of Bayesian networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(2):174–178, February 1999.

[65] M. L. Wong, S. Y. Lee, and K. S. Leung. Applying evolutionary algorithms to discover knowledge from medical databases. In *Proceedings of IEEE International Conference on Systems, Man, and Cybernetic*, pages 936–941, 1999.

[66] M. L. Wong, S. Y. Lee, and K. S. Leung. A hybrid approach to discover Bayesian networks from databases using evolutionary programming. In *Proceedings of the 2002 IEEE International Conference on Data Mining*, pages 498–505, 2002.

[67] X. Wu, P. J. Lucas, S. Kerr, and R. Dijkhuizen. Learning bayesian-network topologies in realistic medical domains. In *Intelligent Data Analysis in Medicine and Pharmacology IDAMAP*, 2001.

[68] J. Yang and V. Honavar. Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems*, 13(2):44–49, 1998.

[69] S. Yang and K.-C. Chang. Comparison of score metrics for bayesian network learning. In *IEEE Transactions on System, Man and Cybernetics - Part A: System and Humans*, volume 32, pages 419–428, 2002.

[70] Z. Yang, I. Lauder, and H. Lin. Molecular evolution of the hepatitis b virus genome. *Journal of Molecular Evolution*, 41(5):587–596, 1995.