Support for Multiple Cause Diagnosis with Bayesian Networks

Randy M. Jagt

THESIS
Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Applied Mathematics

September 2002

Department of Mediamatics, Faculty of Information Technology and Systems,
Delft University of Technology, the Netherlands
Information Sciences Department, University of Pittsburgh, USA

Graduation Committee:
Dr. Drs. L.J.M. Rothkrantz
Prof. Dr. J.M. Aarts
Dr. Ir. Marek J. Druzdzel
Prof. Dr. H. Koppelaar

Technische Universiteit Delft

ABSTRACT

Although a Bayesian network is widely accepted as a sound and intuitive formalism for reasoning under uncertainty in artificial intelligence, their use in diagnostic expert systems has been limited. The primary goal within these diagnostic systems is to determine the most probable cause given a set of evidence and to suggest what additional information is best to collect. The framework of a Bayesian network supports this goal by providing various reasoning algorithms for the calculations of the effect of new information. However, for the support of practical models the networks are often accompanied by restrictions. One such restriction is that only one cause can be present since the support for multiple causes becomes computationally challenging. Another restriction is the limited support for user interaction. In most systems the user has nothing to say about which causes are investigated, instead the system always investigates all the causes.

In this thesis I aim to improve the functionality of Bayesian networks by providing approximation approaches that support the diagnosis of multiple causes. At the same time I try to improve the interactivity with the user by supporting the ability to pursue and differentiate between any possible set of causes. The foundation of the approximation approaches is the relation between the probability of causes separately and the probability of a combination of those. The ability to pursue and differentiate between any possible set of causes is a generalization of current possibilities to perform diagnosis, e.g., the pursuit of one or all possible causes. I believe that these improvements will have a positive effect on the user acceptance of Bayesian networks in modelling complex diagnostic systems.

CONTENTS

ACKNOWLEDGMENTS

The fantastic time I enjoyed at the University of Pittsburgh, could not have been possible without the support of all the people involved in my graduation project and my stay in Pittsburgh.

First and foremost, I would like to thank my advisor at the University of Pittsburgh, Dr. Ir. Marek J. Druzdzel. Not only did he provide me with new insights about life, but he also let me discover the thrill and excitement that research holds.

Next, I would like to thank my fellow colleagues from the Decision Systems Laboratory for their continuous support and motivation to perform research. In particular, Tsai-Ching Lu for being a close friend and always available for lunches, endless talks, and discussions relating my research.

My admiration and thanks go to my advisor from the Delft University of Technology, Drs. Dr. L.J.M. Rothkrantz for his continuous effort and guidance to let me and many other students experience what the world offers.

Last but certainly not least, I want to express my gratitude to Eline, to hold on to her love, no matter how far away I was.

GRANTS

This thesis work is supported by grants from the following Dutch institutions,

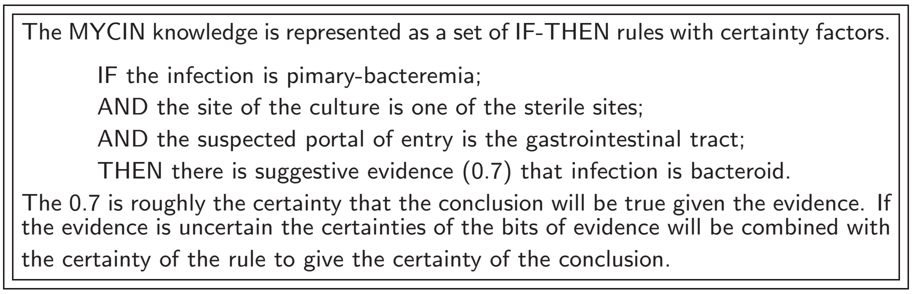
Universiteitsfonds Delft
Koninklijk Instituut Van Ingenieurs, Studie- en Reisfonds
Delft University of Technology, College van Bestuur Fonds
Department of Information Technology and Systems, ITS Fonds
Stichting Fundatie van de Vrijvrouwe van Renswoude

1. INTRODUCTION

The purpose of this thesis is to describe the research I carried out in the Decision Systems Laboratory (DSL) at the School of Information Science of the University of Pittsburgh. In short, the main objective of this research was to improve the use of Bayesian networks in diagnostic expert systems. The introduced improvements are approximation approaches for the support of multiple causes and the ability to pursue and differentiate between any possible set of causes.

1.1 Motivation

Diagnosis is generally considered as the process of determining the cause of a malfunction by means of collecting information. This essential task is faced in various domains such as medicine, business, and engineering. Consider, for example clinicians who determine the disease of a patient, business consultants who analyze what is wrong within a company, or technicians who perform tests to see which part of a machine is malfunctioning. With the goal of assisting a user in the diagnostic process, a lot of research has been done into the development of diagnostic expert systems. In general, expert systems are described as reasoning systems based on the techniques of artificial intelligence and decision theory, which perform at a level comparable to or better than a human expert within a certain domain [Horvitz *et al.*, 1988]. An example of a successful expert system is the MYCIN system, developed to aid physicians in the diagnosis of bacterial infections. Essentially, the MYCIN system uses a rule based structure with certainty factors to model the uncertainty. A short example of one of those rules is shown in 1.1. Although the diagnostic expert systems have been modelled in various ways, they generally support two tasks: determine (on the basis of gathered evidence) the most likely cause, and suggest what additional information to collect.



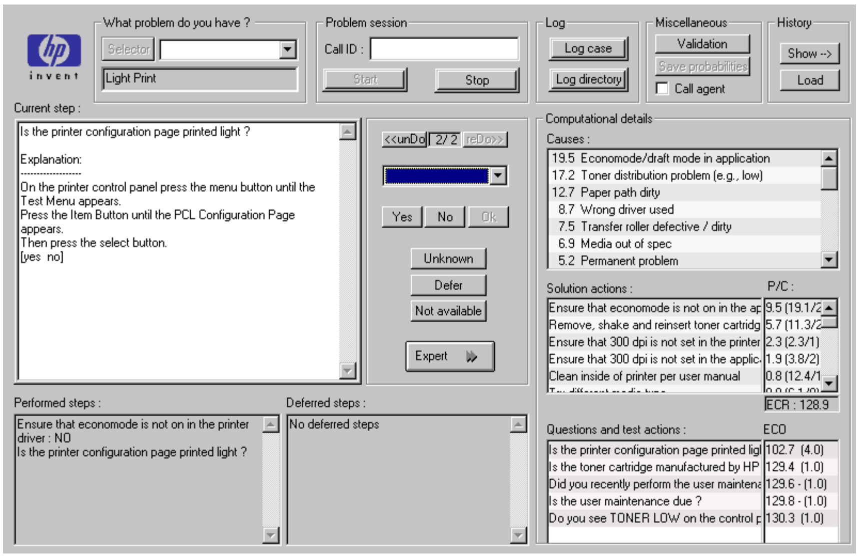
The MYCIN knowledge is represented as a set of IF-THEN rules with certainty factors.

IF the infection is pimary-bacteremia;
AND the site of the culture is one of the sterile sites;
AND the suspected portal of entry is the gastrointestinal tract;
THEN there is suggestive evidence (0.7) that infection is bacteroid.

The 0.7 is roughly the certainty that the conclusion will be true given the evidence. If the evidence is uncertain the certainties of the bits of evidence will be combined with the certainty of the rule to give the certainty of the conclusion.

Fig. 1.1: Example of the representation of one of MYCIN rules

In 1959, Ledley and Lusted [1959] discussed the underlying reasoning of a professional clinician and identified three relevant mathematical disciplines, *symbolic*, *logic*, and *probability* to model the diagnostic process. From these disciplines, probability theory with its Bayes theorem was considered the main approach for its good ability to model uncertainty. Its use resulted in various diagnostic expert systems, e.g., the diagnosis of heart disease [Gorry and Barnett, 1968] and acute abdominal pain [de Dombal *et al.*, 1972]. Although some of these systems were quite successful, interest in this approach stagnated in the late 1970s and shifted to the two other disciplines. A possible reason for this loss of interest in these systems is their limited possibilities for handling the complexity associated with the representation and the computation of the probabilistic schemes.



What problem do you have ? Light Print

Call ID :

□ Call agent

Current step :

Is the printer configuration page printed light ?

Explanation:

On the printer control panel press the menu button until the Test Menu appears.
Press the Item Button until the PCL Configuration Page appears.
Then press the select button.
[yes no]

Performed steps :

Ensure that economode is not on in the printer driver : NO
Is the printer configuration page printed light ?

Deferred steps :

No deferred steps

Computational details

Causes :

19.5	Economode/draft mode in application
17.2	Toner distribution problem (e.g., low)
12.7	Paper path dirty
8.7	Wrong driver used
7.5	Transfer roller defective / dirty
6.9	Media out of spec
5.2	Permanent problem

Solution actions :	P/C :
Ensure that economode is not on in the ap	9.5 (19.1/2
Remove, shake and reinsert toner cartridg	5.7 (11.3/2
Ensure that 300 dpi is not set in the printer	2.3 (2.3/1)
Ensure that 300 dpi is not set in the applic	1.9 (3.8/2)
Clean inside of printer per user manual	0.8 (12.4/1

ECR : 128.9

Questions and test actions :	ECO
Is the printer configuration page printed ligl	102.7 (4.0)
Is the toner cartridge manufactured by HP	129.4 (1.0)
Did you recently perform the user maintena	129.6 - (1.0)
Is the user maintenance due ?	129.8 - (1.0)
Do you see TONER LOW on the control p	130.3 (1.0)

Fig. 1.2: The interface of the printer troubleshooting system, SACSO, see [Jensen *et al.*, 2001] for more information

The development of probabilistic graphical models such as Bayesian networks [Pearl, 1988] and closely related influence diagrams [Howard and Matheson, 1981] renewed the interest in the use of the probabilistic discipline and resulted in the development of new diagnostic expert systems, e.g., diagnosis of liver disorders [Oniśko *et al.*, 1997], lymph node diseases [Heckerman *et al.*, 1992], and printer troubleshooting (SACSO) [Jensen *et al.*, 2001]. In Figure 1.2 the interface is shown of the SACSO system, in which the user is able to diagnose trouble with a printer. The strength of Bayesian networks is that they provide the user with an intuitive and mathematically sound tool to model complex relations between uncertain variables.

As an example a small Bayesian network is shown in Figure 1.3, with the probabilistic relations between the variables *Smoking*?, *LungCancer*?, and *Bronchitis*?. From the network may be concluded that *LungCancer*? and *Bronchitis*? have no probabilistic influence on each other, but the knowledge whether a person smokes will have an impact on the probabilities of both the variables.

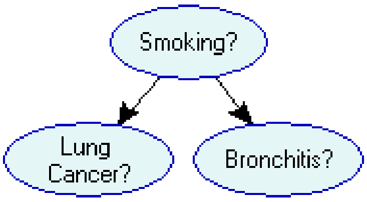


Fig. 1.3: A typical Bayesian network that shows the relation between the uncertain variables, *Smoking*?, *LungCancer*?, and *Bronchitis*?

The process of reasoning is supported by various efficient algorithms which determine the effect of instantiating variables. Within diagnostic systems these reasoning algorithms are applied to find the most likely cause of a malfunction. Another important task of diagnostic systems is to determine which (additional) information to collect in order to become more certain about the true cause. The concept of *value of information* [Howard, 1966] captures this task for Bayesian networks. Developed in decision theory, this concept provides techniques to evaluate beforehand whether or not to collect new information based on its informativeness and cost.

It seems that Bayesian networks provide sufficient tools to model diagnostic expert systems. However, most networks and systems are often accompanied by restrictions. A major restriction is that all the possible causes are mutually exclusive, i.e., only one cause is possible in the system. Take, for example someone who is ill and the cause may be either a fever or pneumonia but not both. It is obvious that systems with this restriction will have trouble modelling real world applications. The reason for this restriction is that the support of multiple cause diagnosis results both in computational as well as presentational problems when the number of possible causes is large. Another restriction of the networks is the limited support for user interaction. A typical situation is that the user already has an idea of which causes are the most likely and wants to pursue these instead of all the causes. Most existing systems do not support this approach or limit the user to pursue only one cause. An example of such a system is the GeNIe_DIAG application developed at the DSL and described in Section 3.5. Without this support users may be reluctant to accept the system as an assistant. Since these restrictions have a negative influence on using Bayesian networks as a modelling tool for real practical situations, I believe it is important to find solutions for them.

1.2 Objective

The main objective of this thesis is to improve the functionality of Bayesian networks by providing approximation approaches for the diagnosis of multiple causes and the ability to pursue and differentiate between any possible set of causes. In order to accomplish this objective, I set myself with the following goals:

- Study about Bayesian networks and their use in diagnostic systems;
- Analysis of the problem with supporting multiple causes;
- Investigation of possible approximation approaches;
- Design and implementation of multiple cause module;
- Testing of the module whether it delivers qualitative support.

Accompanied with the development of the multiple cause module is that it has to be able to handle complex systems.

1.3 Overview

The remainder of this thesis is structured as follows. Chapter 2 will provide a short introduction to Bayesian networks. In Chapter 3 I will describe how these networks may be used for the process of diagnosis. Chapters 4 and 5 will address the problems associated with multiple cause diagnosis and propose approximation approaches to solve these problems. The design and implementation of the proposed approximations into a module will be described in Chapter 6. The quality of the module will be tested in Chapter 7. Finally, I will present my conclusions and outline the direction of future work.

2. BAYESIAN NETWORKS

This chapter presents a brief introduction into Bayesian networks and describes the necessary concepts for this report. I assume that the reader is familiar with the essentials of graph and probability theory. If not, I refer the reader to Jensen *et al.* [2001] for more information about Bayesian networks and probability theory in general.

A Bayesian network [Pearl, 1988] (also known as a belief network or probabilistic network) is a formalism for reasoning under uncertainty. Decision support based on probabilistic reasoning was developed in the late 1970's and gained popularity when efficient algorithms for inference were introduced in Bayesian networks [Lauritzen and Spiegelhalter, 1988]. Thanks to an intuitive graphical interface and a sound probabilistic framework, the Bayesian network has become a popular approach to model various expert systems, e.g., medical, image interpretation, troubleshooting, and information processing.

In detail, a Bayesian network is an acyclic directed graph that represents a factorization of the joint probability distribution over a set of random variables.

The graphical structure of the network is the qualitative part of a Bayesian network and embodies a set of nodes representing the random variables and a set of arrows representing direct dependencies between connected variables. Absence of an arrow between variables implies that these variables are (conditionally) independent. The parents of a variable are the variables which are connected with an arrow with its direction going into this variable.

The joint probability distribution is the quantitative part of a Bayesian network and embodies the conditional probability distribution defined with each variable. This distribution characterizes the influence of the values of the predecessors (parents) on the probabilities of the values of the variable itself. When a variable has no parents, the probability distribution is the prior probability distribution. In practice, these distributions are derived from frequency data or elicited from an expert judgment.

Given a joint probability distribution over a set of random variables, many different graphs exist which factorize the same joint probability distribution. A factorization that is especially desired is the graph that reflects the causal structure of the problem. This graph, also known as a causal graph, normally reflects an expert's understanding of the domain and facilitates a user's insight during the operational stage.

Example 1. Consider the Bayesian network in Figure 2.1, which represents a fictitious Asia example from Spiegelhalter and Knill-Jones [1984]. This network is based on the 'knowledge' that dyspnea (DY), i.e., shortness-of-breath, may be due to tuberculosis (TC), lung cancer (LC), or bronchitis (BC). A recent visit to Asia (VA) increases the probability of tuberculosis, while smoking (SM) is known to be a risk factor for both lung cancer and bronchitis. Neither the result of a single chest X-ray (XR) nor the presence or absence of dyspnea, discriminates between lung cancer and tuberculosis.

Each of the variables is associated with a probability distribution. So has the variable SM the marginal probability distribution of Table 2.1. And, since the variable SM is the parent of the variable LC, this variable has a conditional probability distribution of LC conditioned on SM, see Table 2.2. □

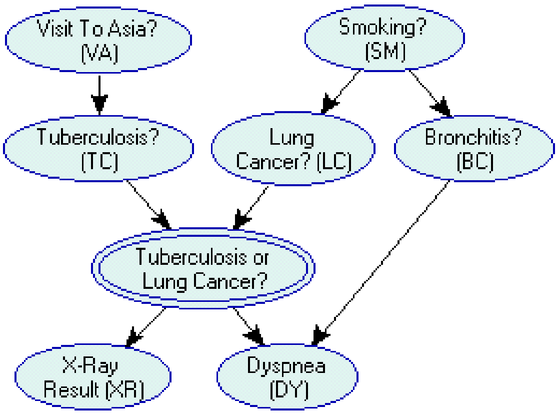


Fig. 2.1: The Bayesian network representing the fictitious Asia example from Spiegelhalter and Knill-Jones [1984]

Tab. 2.1: Prior probability table of the variable SM

$\Pr(SM)$	
$SM\_nonsmoker$	0.75
SM_smoker	0.25

Tab. 2.2: Conditional probability table of the variable LC conditioned on the variable SM

$\Pr(LC\|SM)$	$SM\_nonsmoker$	SM_smoker
LC_absent	0.75	0.55
$LC\_present$	0.25	0.45

Various efficient algorithms [Lauritzen and Spiegelhalter, 1988, Pearl, 1988, Huang and Darwiche, 1994] exist for reasoning with Bayesian networks, e.g., determining the impact of processing evidence into the network. Although the calculation of probabilistic inference is NP-hard [Cooper, 1990], the algorithms provide reasonable computing times for networks consisting of tens or even hundreds of nodes.

Before I present the definition of a Bayesian network and Bayes rule, I introduce some necessary notations. Consider a finite set of discrete random variables $\mathcal{V}$, where each variable $X \in \mathcal{V}$ is denoted as a capital letter, e.g., X, Y, Z. Each state of a variable is denoted as a lowercase letter, e.g., x, y, z. The set of all states within a variable X, is denoted as D_X. The probability distribution over a random variable X is denoted as $\Pr(X)$ and the probability of a state $x \in D_X$ as $\Pr(X = x)$ or in shorter form $\Pr(x)$. The negation of a state x is denoted as $\overline{x}$ and represents all the states apart from the state x in the variable. The probability of the negation, $\Pr(\overline{x})$ is always equal to $1 - \Pr(x)$

A combination of states of multiple variables is denoted as a *scenario*. The set of all the scenarios from a set of variables $\mathcal{V}$, is denoted as $D_{\mathcal{V}}$, and each scenario as $s \in D_{\mathcal{V}}$. In case of one variable, the set of scenarios and the set of states of the variable are identical. In Table 2.2 from Example 1 the variables LC and SM yield the four scenarios displayed in Table 2.3. The probability of a scenario is defined by the joint probability over the states in the scenario. The probability distribution over a set of variables is denoted as $\Pr(\mathcal{V})$ and the probability of a scenario $s \in D_{\mathcal{V}}$ as $\Pr(\mathcal{V} = s)$ or in shorter form $\Pr(s)$. The set of parents of a variable X is denoted as Π_X.

Tab. 2.3: Four possible scenarios of the variables SM and LC

SM_nonsmoker & *LC_absent*	*SM_nonsmoker* & *LC_present*
SM_smoker & *LC_absent*	*SM_smoker* & *LC_present*

The foundation of the Bayesian network is the Bayes theorem,

$$\Pr\left(B|\,A\right) = \frac{\Pr\left(A|\,B\right)\Pr\left(B\right)}{\Pr\left(A\right)}.$$

named after Reverent Thomas Bayes (1702-1761). The initial probability $\Pr(A)$ is called the prior probability, and the updated probability $\Pr(A|B)$ the posterior probability. An interpretation of the posterior probability is the probability of A with the knowledge of the state of variable B. When the knowledge of a variables has an effect on the probability of another variable these variables are called dependent. If variables are independent of each other, the posterior probability and the prior probability are equal, $\Pr(A|B) = \Pr(A)$.

Definition 2.1 (Bayesian network). *A Bayesian network, $BN = \langle G, \Theta \rangle$ is an acyclic directed graph, $G = \langle \mathcal{V}, \mathcal{A} \rangle$, where the arrows $\mathcal{A}$ denote a probabilistic relation between the vertices and each vertex, $V \in \mathcal{V}$ represents a discrete random variable. Associated with the vertexes is a $\theta_{V \in \mathcal{V}} : D_V \times D_{\Pi_V} \to [0,1]$ function with the condition that for each combination of $\pi_V \in \Pi_V$, there holds:*

$$\sum_{d_V \in D_V} \theta_V \left(d_V, \pi_V\right) = 1.$$

The probability distribution of each variable is embodied by the joint probability distribution encoded in a Bayesian network. Suppose for example two variables, A and B, with the joint probability distribution $\Pr(A, B)$. With marginalization, the probability distribution of A is calculated by taking the sum over the joint probability of A with all the states of B.

$$\Pr\left(A\right) = \sum_{b_i \in D_B} \Pr\left(A, b_i\right)$$

In order to determine and present the joint probability, the following theorem better known as the chain rule may be applied.

Theorem 2.1 (chain rule). *Let BN be a Bayesian network over a finite set of discrete random variables $\mathcal{V} = \{V_1, ..., V_n\}$. The joint probability distribution $\Pr\left(\mathcal{V}\right)$ is then,*

$$\Pr\left(\mathcal{V}\right) = \prod_{i=1}^{n} \Pr\left(V_i \middle| \Pi_{V_i}\right).$$

When variables are instantiated (=set to a state) I refer to these variables as evidence. A possible effect of entering evidence is a change in the dependency relations between variables, i.e., different variables may become independent of or dependent on each other. When two sets of variables become independent of each other given the instantiation of a third set, this is identified as conditional independence.

Definition 2.2 (conditional independence). *Let $\mathcal{V}$ be a finite set of discrete random variables and let $\Pr\left(\mathcal{V}\right)$ denote the joint probability distribution over the variables. Suppose three disjoint subsets of variables, $\mathcal{X}, \mathcal{Y}, \mathcal{Z} \subset \mathcal{V}$. The sets $\mathcal{X}$ and $\mathcal{Y}$ are conditionally independent given $\mathcal{Z}$, if for all $s_x \in D_{\mathcal{X}}$, $s_y \in D_{\mathcal{Y}}$, and $s_z \in D_{\mathcal{Z}}$, there holds:*

$$\Pr\left(s_x \middle| s_y, s_z\right) = \Pr\left(s_x \middle| s_z\right).$$

By combining conditional independence with the chain rule I am able to present the joint probability even more compacter, see Example 2.

Example 2. Consider the fragment of the Asia network, see Figure 2.2, with the variables, SM, LC and BC. Whether a person has lung cancer or not is conditionally independent of whether the person has bronchitis or not, when there is evidence that the person is a smoker.

$$\Pr\left(LC|\, BC, SM = smoker\right) = \Pr\left(LC|\, SM = smoker\right)$$

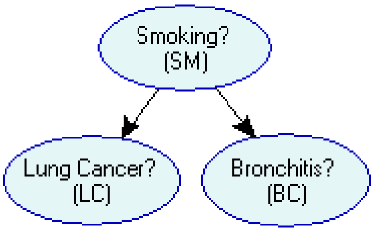


Fig. 2.2: Part of the fictitious Asia example to represent conditional independence

The benefit of conditional independence is noticeable with the determination of the joint probability. For instance, the joint probability of these three variables LC, SM, BC is according to the chain rule from Theorem 2.1:

$$\Pr(LC, SM, BC) = \Pr(LC|SM, BC) \cdot \Pr(SM, BC)$$
$$\Pr(LC, SM, BC) = \Pr(LC|SM, BC) \cdot \Pr(SM|BC) \cdot \Pr(BC).$$

Combining the joint probability with the conditional independence between the variables LC and BC given SM, the joint probability is rewritten to:

$$\Pr(LC, SM, BC) = \Pr(LC|SM) \cdot \Pr(SM|BC) \cdot \Pr(BC).$$

□

A method to determine graphically if variables are conditionally independent given other evidence is by observing whether the variables are d-separated.

Definition 2.3 (d-separation). *Let BN be a Bayesian network over a finite set of discrete random variables $\mathcal{V}$ and let $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$ stand for any three disjoint subsets of variables of $\mathcal{V}$. $\mathcal{Z}$ is said to d-separate $\mathcal{X}$ from $\mathcal{Y}$, if along every path (sequence of connected variables) between a variable in $\mathcal{X}$ and a variable in $\mathcal{Y}$, there is a variable W satisfying one of the following two conditions: (1) W has converging arrows and none of W or its descendants are in $\mathcal{Z}$, or (2) W does not have converging arrows and W is in $\mathcal{Z}$.*

The sound mathematical framework and the support for conditional independence and d-separation make a BN a powerful tool for modelling probability relations between random variables.

3. DIAGNOSTIC PROBABILITY NETWORKS

Before I discuss how a Bayesian network may be applied in the support of diagnosis, I will analyze the different tasks associated with this process. Based on this analysis I will introduce a structure for the Bayesian network that distinguishes the variables necessary for diagnosis and supports the essential tasks of diagnosis.

3.1 Process of Diagnosis

Diagnosis is best known as the process of identifying the disease or disorder of a patient or a machine by considering its history of symptoms and other signs [Stensmo and Terrence, 1994]. When diagnosis is performed to determine the trouble or faults in a machine, the diagnostic process is also referred to as troubleshooting. In general, two kinds of tasks are involved in the diagnostic process [Gorry and Barnett, 1968].

The first task is to determine the true cause or when multiple causes may occur simultaneously the combination of true causes. A cause represents the presence or absence of a disease, fault, or any other discomfort. In an expert system this task is usually recognized as 'reasoning under uncertainty'.

The second task is to reduce the uncertainty about the true cause by obtaining more information about the state of the world. Possible information sources are symptoms, results of tests, or historic data. Using this information several assumptions are made. First, the information is perfect, i.e., there is no possibility that the information is either wrong or incomplete. Second, the information is non-intervening, i.e., the information will not change the world. Third, the information never increases uncertainty, e.g., more hints during an exam will not make you more uncertain about the answer. The last assumption makes it seem worthwhile to get all available information. Unfortunately, information is seldom cost-free, e.g., time to fill in a questionnaire, or the money paid for a CTI scan. Therefore, a decision has to be made which information to collect. Since information gathering has an effect on the uncertainty in the system, each gathering should actually be seen as a step in a sequence of diagnostic steps.

By displaying the diagnostic process as a sequence of steps and outlining the two tasks this results in the process shown in Figure 3.1. As shown the sequence continues until either the uncertainty about the cause is gone or there are no more tests available.

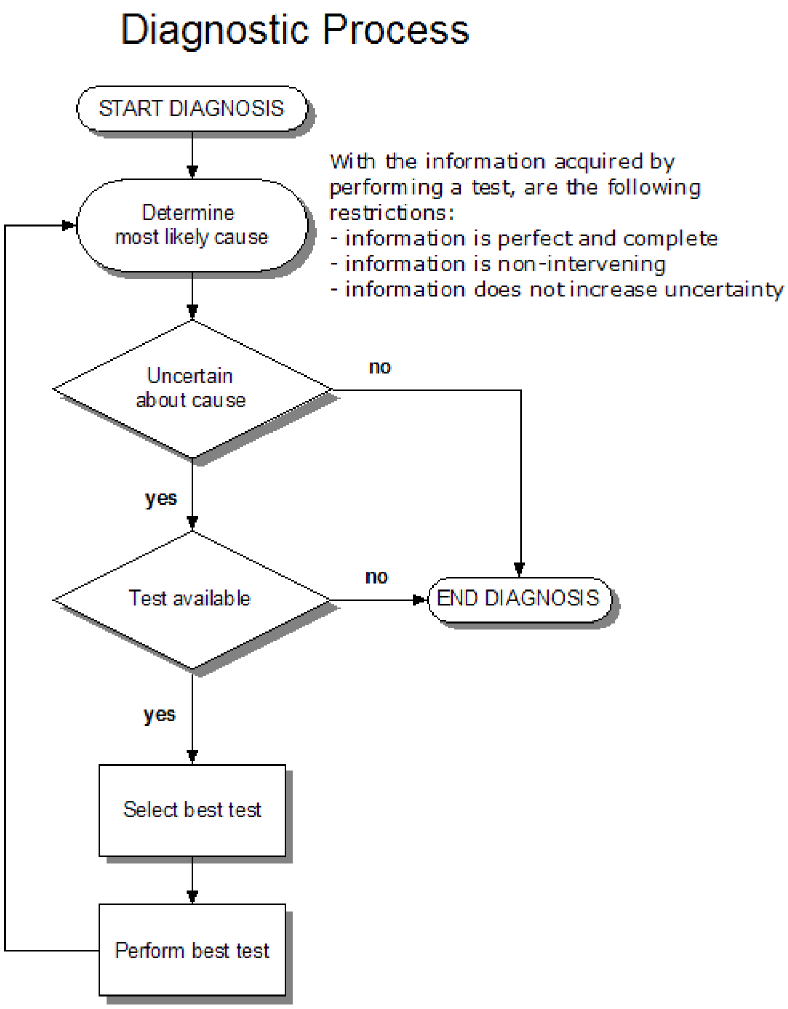

Diagnostic Process
START DIAGNOSIS
Determine most likely cause
With the information acquired by performing a test, are the following restrictions:
- information is perfect and complete
- information is non-intervening
- information does not increase uncertainty
Uncertain about cause
no
yes
Test available
no
END DIAGNOSIS
yes
Select best test
Perform best test

Fig. 3.1: The diagnostic process

3.2 Diagnostic Probability Structure

In order to let Bayesian networks (BNs) support the diagnostic process I associate the BNs with a structure where the variables necessary for the diagnostic process are distinguished. A BN associated with this structure will be denoted as a diagnostic probability network (DPN). The support of determining the most likely cause or the first task in the diagnostic process may be performed with one of the available reasoning algorithms for BNs. The support of the second task or the determination which information to acquire is provided by applying the concept of *value of information*. This concept is described in the next section.

The structure associated with a BN is described in the following definition.

Definition 3.1 (diagnostic probability network). *Let B be a BN with a set of random variables $\mathcal{V}$. A diagnostic probability network (DPN) is defined as a BN with at least one hypothesis variable and at least one test variable.*

- *Let H be a hypothesis variable and $\mathcal{H}$ a set of all hypothesis variables also referred to as hypothesis set. A state of a hypothesis variable $h \in H$ is denoted as a hypothesis state. Each hypothesis state may represent a possible disease, fault in a system, or any other discomfort.*
- *Let T be a test variable and $\mathcal{T}$ a set of all test variables, also referred to as test set. A state of a test variable $t \in T$ is denoted as a test state. Each test state may represent an observation, physical sign, indicant, symptom or laboratory result. Each test is associated with a cost value $Cost\,(T)$, if a test variable has no cost, the cost value is set to 0.*

Note that it it not necessary that every variable in a BN is either a hypothesis or a test variable. Variables which are neither hypothesis nor test variables are referred to as auxiliary variables. The following example clarifies this structure by transforming the BN of Example 1 into a DPN.

Example 3. Within the BN of Example 1, tuberculosis, lung cancer, and bronchitis are supposed to be the possible ailments. Therefore, the set $\mathcal{H} = \{TC, LC, BC\}$ is distinguished as hypothesis set. As possible tests the set $\mathcal{T} = \{VA, SM, XR, DY\}$ is distinguished. If I assume that each hypothesis variable has only two states, *absent* or *present*, the possible scenarios of the hypothesis set $\mathcal{H}$ are listed in Table 3.1. The goal of performing diagnosis with this network is to investigate which scenario belongs to the patient. □

Tab. 3.1: The possible scenarios of the hypothesis set $\mathcal{H} = \{TC, LC, BC\}$

TC_absent LC_absent BC_absent	TC_absent $LC\_present$ BC_absent	TC_absent LC_absent $BC\_present$	TC_absent $LC\_present$ $BC\_present$
$TC\_present$ LC_absent BC_absent	$TC\_present$ $LC\_present$ BC_absent	$TC\_present$ LC_absent $BC\_present$	$TC\_present$ $LC\_present$ $BC\_present$

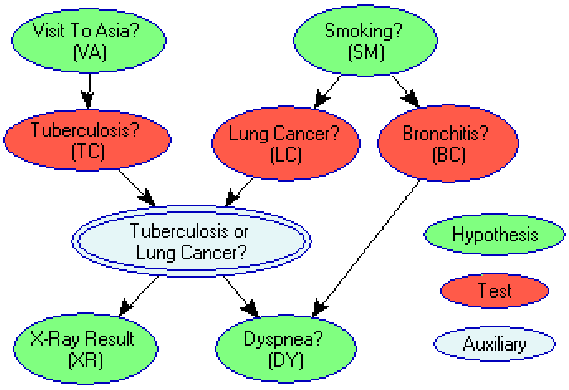


Fig. 3.2: The diagnostic probability network of the Asia example

During the process of diagnosis each test of the set of test variables may be instantiated with evidence, i.e., observation of a test. This instantiation uses the assumption that the information is perfect.

Notice that when the number of variables increases, the number of scenarios grows exponentially. This exponential growth causes the system to become too complex both to assess and to compute. Consider, for example a system with n hypothesis variables where each variable has 2 states, the number of possible scenarios is then 2^n. An often applied solution and escape from handling multiple causes, is the *naive Bayes structure*, also known as idiot's Bayes. This structure allows only one hypothesis variable and assumes conditional independency between each test variable. In Figure 3.3 the fictitious Asia example is transformed to a naive Bayes structure. With this naive Bayes structure there may only be one disease present. So within Figure 3.3 the variable *Diseases* contains the states *TC_present*, *LC_present*, *BC_present*, and *all_absent*.

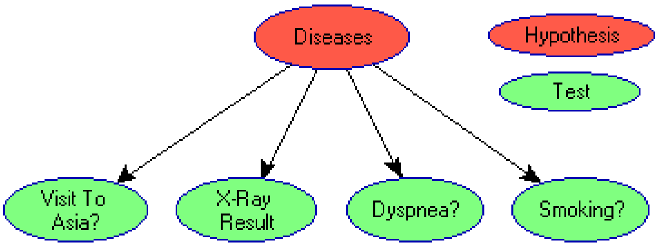


Fig. 3.3: Asian example transformed to a naive Bayes structure

The great simplicity of this structure both to assess and to compute made this approach quite popular to model diagnostic expert systems. However, the approach also received a lot of criticism because of its apparent mismatch with the real world.

3.3 Value of Information

Value of information (VOI) is a central part of performing diagnosis since it determines which test to perform, i.e., which piece of information to acquire. Evaluating prior to acquisition of information is a necessity when the user has limited sources such as time and or numerous tests to choose from. VOI determines the best test by providing *test selection measures* or *value functions* which assign a ranking to each test. Before I present possible test selection measures I first formulate the area of VOI. An overview of the complete procedure of VOI is shown in Figure 3.4. The formalization of this procedure is in a similar notation as Jensen [1996].

Consider a DPN with a set of hypothesis variables $\mathcal{H}$, a set of test variables $\mathcal{T}$, and a value function $V\left(\Pr\left(\mathcal{H}\right)\right) : [0;1] \rightarrow \mathbb{R}$. Since the outcome of a test is unknown, the *expected value* (EV) of performing a test $T \in \mathcal{T}$ is used:

$$EV\left(T\right) = \sum_{t \in T} V\left(\Pr\left(\mathcal{H} \middle| \, t\right)\right) \Pr\left(t\right).$$

The *expected benefit* (EB) of performing a test, is then defined as the difference between the expected value of performing a test and the value without performing a test:

$$\begin{gathered} EB\left(T\right) = EV\left(T\right) - V\left(\Pr\left(\mathcal{H}\right)\right) = \\ \sum_{t \in T} V\left(\Pr\left(\mathcal{H} \middle| \, t\right)\right) \Pr\left(t\right) - V\left(\Pr\left(\mathcal{H}\right)\right). \end{gathered}$$

For assigning a ranking to each test based on the benefit of the test and the cost of the test T, the *test strength* (TS) is used:

$$TS\left(\mathcal{H}, T\right) = \frac{EB\left(T\right)}{V\left(\Pr\left(\mathcal{H}\right)\right)} - K \, Cost\left(T\right).$$

The coefficient K is necessary for combining the different scaled variables cost and expected benefit. Since there is no standard formulation for the value of K, I let this variable be set by the user. This user has to determine how much the cost of a test weighs in combination with the benefit of a test.

A proper analysis of which information to acquire should contain all the possible combinations of tests, but since this is computationally intractable, I restrict myself to the myopic approximation. This approximation assumes that only one information source is acquired so the effect of performing combinations of tests is not considered.

```
Procedure: TestSelection($\mathcal{H}, \mathcal{T}$)

Input: set of hypothesis variables and set of test variables
Output: list with ranked tests ready for test selection

determine $V(\Pr(\mathcal{H}))$
create empty list to store: test strengths $list\_T$
for each $T_i \in \mathcal{T}$
    for each $t_j \in T_i$
        instantiate the test state $t_j$
        determine $V(\Pr(\mathcal{H}|\, t_j))$
    end for each
    determine $EV(T_i)$
    determine $EB(T_i)$
    determine $TS(\mathcal{H}, T_i)$
    add test strength to $list\_T$
end for each
present $list\_T$ for test selection
```

Fig. 3.4: Value of information procedure for creating the list with test strengths

3.4 Test Selection Measures

In general, any possible function may be used as a test selection measure. However, not all functions are equally useful. The following theorem shows that linear functions are useless, since they always return an expected benefit of zero [Jensen, 1996].

Theorem 3.1 (zero benefit). *Let $\mathcal{H}$ be a set of hypothesis variables, let T be a test variable, and let $V : [0,1]^n \to \mathbb{R}$ be a value function. When the value function is of a linear form $V(\Pr(\mathcal{H})) = \sum_{s \in D_{\mathcal{H}}} a_s \Pr(s)$, the expected benefit of performing the test T is zero $EB(T) = 0$, or,*

$$\sum_{t \in T} V(\Pr(\mathcal{H}|\, t)) \Pr(t) = V(\Pr(\mathcal{H})).$$

Proof.

$$\sum_{t \in T} \Pr(t) V\left(\Pr\left(\mathcal{H} | t\right)\right) = \sum_{t \in T} \Pr(t) \sum_{s \in D_{\mathcal{H}}} a_s \Pr\left(s | t\right) = \sum_{t \in T} \sum_{s \in D_{\mathcal{H}}} a_s \Pr\left(s, t\right)$$
$$= \sum_{s \in D_{\mathcal{H}}} \sum_{t \in T} a_s \Pr\left(s, t\right) = \sum_{s \in D_{\mathcal{H}}} a_s \Pr(s) = V\left(\Pr\left(\mathcal{H}\right)\right)$$

□

Apart from uselessness of linear functions, there is also a preference for convex value functions over non-convex value functions. The reason for this is explained in the following theorem [Jensen, 1996].

Theorem 3.2 (positive benefit). *Let $\mathcal{H}$ be a set of hypothesis variables, let T be a test variable, and let $V : [0,1]^n \rightarrow \mathbb{R}$ be a value function. When the value function is convex the expected benefit of performing a test is never negative.*

$$\sum_{t \in T} V\left(\Pr\left(\mathcal{H} | t\right)\right) \Pr\left(t\right) \geq V\left(\Pr\left(\mathcal{H}\right)\right)$$

Proof. *(with Jensen's inequality, see Appendix A)*

$$\sum_{t \in T} V\left(\Pr\left(\mathcal{H} | t\right)\right) \Pr\left(t\right) \geq V\left(\sum_{t \in T} \Pr\left(\mathcal{H} | t\right) \Pr\left(t\right)\right) = V\left(\sum_{t \in T} \Pr\left(\mathcal{H}, t\right)\right) = V\left(\Pr\left(\mathcal{H}\right)\right)$$

□

In other words, convex value functions always return a positive value to collecting information. This corresponds to the assumption I made that acquiring information never increases uncertainty.

According to the goal of value of information, value functions are desired which determine which test is the most *informative* and brings the closest to a proper diagnosis. Functions with this objective are known as quasi-utility based functions [Good and Card, 1971]. These functions assign high values to tests which reduce the uncertainty between the scenarios of a hypothesis set and have their minimum when the uncertainty is maximal. Below, I discuss the two most commonly used quasi-utility functions, *entropy* and *weight of evidence*. For more value functions I refer the reader to [Ben-Bassat, 1978, Glasziou and Hilden, 1989, Jensen, 1996].

Entropy

A well known measurement for determining the uncertainty of a distribution is the entropy function [Shannon, 1948]:

Definition 3.2 (entropy). *Let $\mathcal{H}$ be a set of hypothesis variables, and let $s \in D_{\mathcal{H}}$ be a scenario of the domain of $\mathcal{H}$. The entropy function $ENT\left(\Pr\left(\mathcal{H}\right)\right)$ is then,*

$$ENT\left(\Pr\left(\mathcal{H}\right)\right) \equiv - \sum_{s \in D_{\mathcal{H}}} \Pr\left(s\right) \log_2\left(\Pr\left(s\right)\right).$$

As I want the value function to be convex and increase with preference, I use the negative entropy function as the entropy based value function $V_{ENT}(\Pr(\mathcal{H})) = -ENT(\Pr(\mathcal{H}))$. For the simple case of one hypothesis variable with two states and probabilities, $\Pr(s)$ and $\Pr(\overline{s}) = 1-\Pr(s)$, this entropy based value function reduces to the following form:

$$V_{ENT}(\Pr(\mathcal{H})) = \Pr(s)\log_2(\Pr(s)) + (1-\Pr(s))\log_2(1-\Pr(s)).$$

and is plotted in Figure 3.5 as a function of $\Pr(s)$.

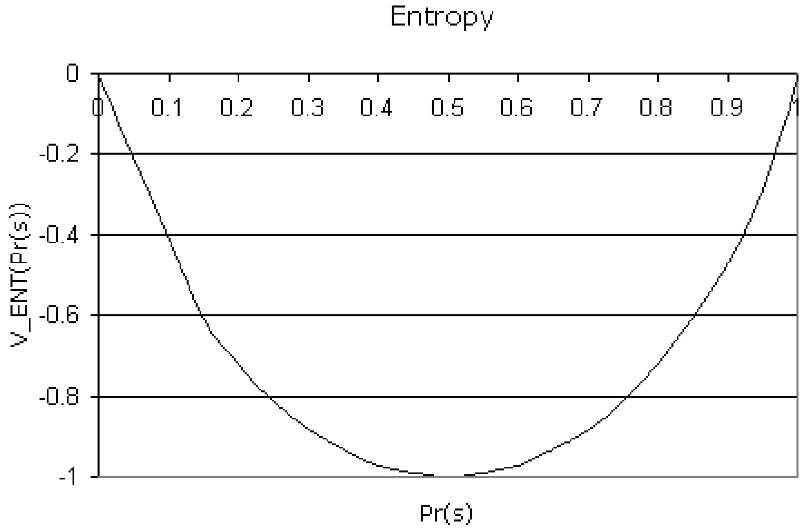


Fig. 3.5: Entropy based value function over two scenarios with probabilities $\Pr(s)$ and $1-\Pr(s)$

Theorem 3.3. *Associated with the entropy based value function are the following properties.*

1. *When each scenario from a set of hypothesis variables $s \in D_{\mathcal{H}}$ has the same probability $\Pr(s) = \frac{1}{n}$, the $V_{ENT}(\Pr(\mathcal{H}))$ function will have its minimum.*
2. *The $V_{ENT}(\Pr(\mathcal{H}))$ function is a monotonic decreasing function of the number of scenarios n, when each scenario $s \in D_{\mathcal{H}}$ has the same probabilities.*
3. *The composition law: if a set of hypothesis variables is broken down into two successive choices, the original $V_{ENT}(\Pr(\mathcal{H}))$ is equal to the weighted sum of the individual values of $V_{ENT}(\Pr(\mathcal{H}))$.*
4. *The entropy function is convex.*

Proof. *See Appendix A.* □

A possible interpretation of the entropy-based value function is a measure of the scattering of the probability distribution over the scenarios. The function has its minimum when the probability distribution is uniform, i.e., every scenario has the same probability. In general, this situation is considered as complete uncertainty since each scenario is equally likely. In the limit, when one scenario has a probability of 1 and the other scenarios have probabilities 0, the entropy based value function is maximal and equal to 0. So the more scattered the probability distribution is, the higher the value and the less the uncertainty.

Weight of Evidence

The weight of evidence function was introduced by Good and Card [1971], with the objective of reducing the uncertainty between a scenario and its negation, by observing the ratio between them.

Definition 3.3 (weight of evidence). *Let $\mathcal{H}$ be a set of hypothesis variables, and let $s \in D_{\mathcal{H}}$ be a scenario of the domain of $\mathcal{H}$. The weight of evidence function $WOE\left(\Pr\left(\mathcal{H}\right)\right)$ is then,*

$$\begin{aligned} WOE\left(\Pr\left(\mathcal{H}\right)\right) &= \log \Pr\left(s\right) - \log \Pr\left(\overline{s}\right) \\ &= \log \Pr\left(s\right) - \log\left(1 - \Pr\left(s\right)\right) \\ &= \log \frac{\Pr\left(s\right)}{\left(1 - \Pr\left(s\right)\right)}. \end{aligned}$$

In Figure 3.6, I show the weight of evidence function for a scenario with probability $\Pr\left(s\right) \in (0,1)$.

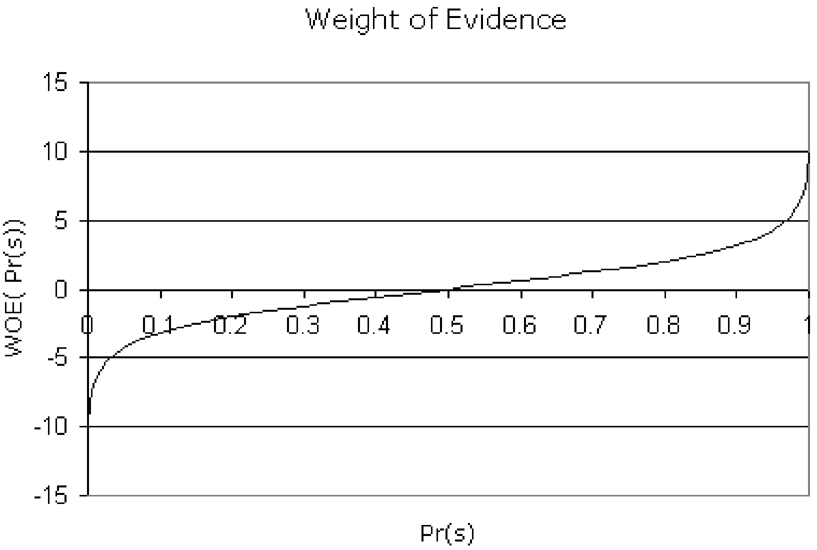


Fig. 3.6: Weight of evidence function for a scenario with probability $\Pr\left(s\right) \in (0,1)$

Theorem 3.4. *Associated with the weight of evidence function are the following properties.*

1. *When a scenario from a set of hypothesis variables* $s \in D_{\mathcal{H}}$ *and its negation have the same probability* $\Pr(s) = \Pr(\overline{s}) = \frac{1}{2}$*, the WOE function will be zero.*

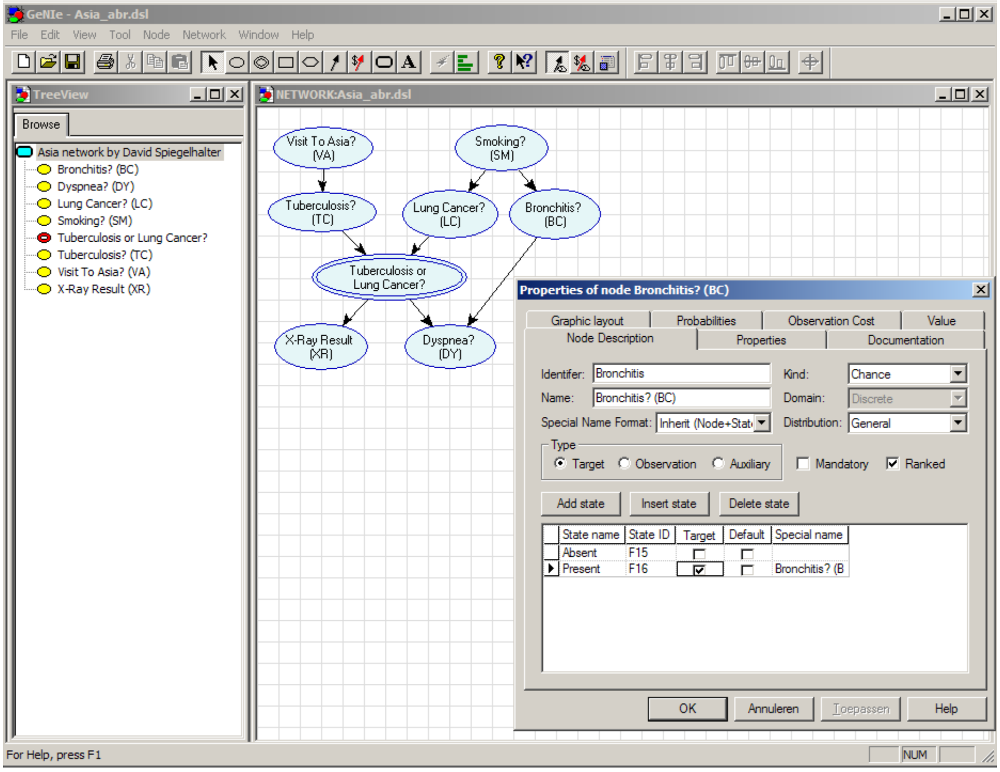
2. *The WOE function is convex, for a scenario from a set of hypothesis variables* $s \in D_{\mathcal{H}}$ *with probability* $\Pr(s) > 1/2$*.*

Proof. *See Appendix A.* □

3.5 Single Cause Diagnostic Application

Below I present an example of how the diagnostic probability network and the area of value of information may be used in a diagnostic application. This application, which I refer to as GeNIe_DIAG, is an existing part of SMILE, an inference engine, and GeNIe, a development environment for reasoning in graphical probabilistic models, both developed at the Decision Systems Laboratory in Pittsburgh. The purpose of the application is to support the user in the process of diagnosis by allowing the user to direct the diagnosis process and to suggest which test is the best to perform. This support is presented by providing the user with the option of selecting a hypothesis state from a list of preselected hypothesis states as the state he or she wants to pursue. The application then determines a ranking for each test depending on how good this test reduces the uncertainty of the selected hypothesis state. To illustrate the application I use the DPN of the Asia network, see Example 3.

Before the application may be started, a diagnostic probability network must be available. Furthermore, it expects a selection of hypothesis states in which the user is interested and wants the option to pursue. These states are denoted as target states, or targets, and may be defined in the properties of the variable, together with the defining of the type of the variable. As shown in Figure 3.7, the variable *BC* is set as a hypothesis variable and the state *present* as a target state. Note that within the application the hypothesis variables are referred to as target variables. Furthermore, the application assumes that a hypothesis variable has at least one target state, because if not, it makes no sense defining this variable as a hypothesis variable.



Identifier: Bronchitis

Name: Bronchitis? (BC)

Special Name Format: Inherit (Node+Stat

Kind: Chance

Domain: Discrete

Distribution: General

Type: ● Target ○ Observation ○ Auxiliary

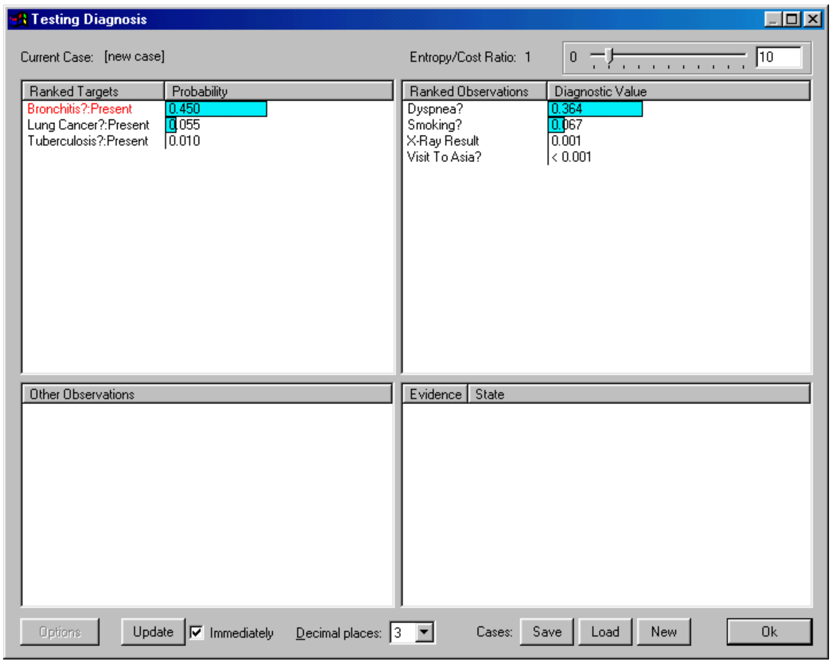
□ Mandatory ☑ Ranked

State name	State ID	Target	Default	Special name
Absent	F15	□	□	
Present	F16	☑	□	Bronchitis? (B

Fig. 3.7: The setting of the variable *BC* to a hypothesis/target variable and the state *present* as a target state

When at least one hypothesis variable together with a target state and at least one test variable are defined, the diagnostic application may be activated. This will pop up the screen as in Figure 3.8. On the left of this screen there is a list of all the target states, and on the right a list of the available tests where each test has a ranking. This ranking represents how good this test is in reducing the uncertainty of the selected target from the left list. The determination of the ranking is done by applying the concept of value of information in combination with the entropy based value function. Whenever another target is selected and pursued, the rankings in the test-list are recalculated so the user is able to see for each target which test is the best to perform.

Within the diagnosis screen the user may perform any test from the list of ranked tests. By selecting and assigning a test to a state will have an impact on the probabilities of the targets and the probability distributions of the test variables. Therefore, the probabilities in the diagnosis screen are adapted and the application recalculates the rankings of the remaining tests. In Figure 3.9, the effect of instantiating the test *Dyspnea*? with the state *present* is shown. Because of this instantiation the target *Bronchitis present* increases to the value 0.834. Furthermore, the test rankings change and present the $X - RayResult$ as the best test. The user is now able to select and perform another test. This process may be continued until the user reaches a proper diagnosis or no more tests are available.



Current Case: [new case]

Entropy/Cost Ratio: 1 (0 – 10)

Ranked Targets	Probability
Bronchitis?:Present	0.450
Lung Cancer?:Present	0.055
Tuberculosis?:Present	0.010

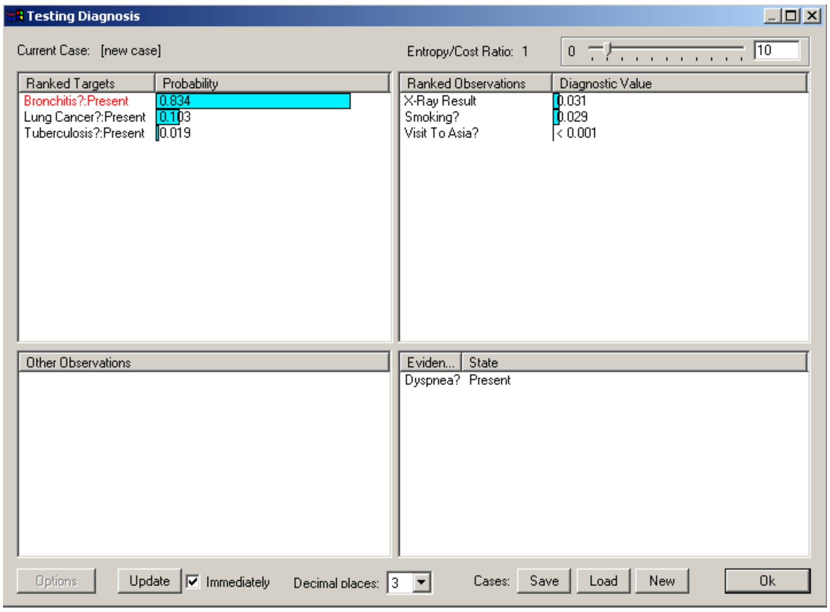
Ranked Observations	Diagnostic Value
Dyspnea?	0.364
Smoking?	0.067
X-Ray Result	0.001
Visit To Asia?	< 0.001

Other Observations

Evidence	State

☑ Immediately Decimal places: 3

Fig. 3.8: The diagnostic screen with pursuing the target *Bronchitis present*



Current Case: [new case]

Entropy/Cost Ratio: 1 (0 – 10)

Ranked Targets	Probability
Bronchitis?:Present	0.834
Lung Cancer?:Present	0.103
Tuberculosis?:Present	0.019

Ranked Observations	Diagnostic Value
X-Ray Result	0.031
Smoking?	0.029
Visit To Asia?	< 0.001

Other Observations

Eviden...	State
Dyspnea?	Present

☑ Immediately Decimal places: 3

Fig. 3.9: The diagnostic screen with the instantiation of the test *Dyspnea*

The Entropy/Cost Ratio, on top of Figure 3.9, represents the coefficient K that combines the cost and the expected benefit of the test, see Section 3.3. This variable may be adjusted at any time during the process of diagnosis. The cost of the test is defined in the properties of the variable, see the option Observation Cost in Figure 3.7.

What characterizes this application is the support for interactivity with the user during the process of diagnosis. The user has complete control to direct the diagnosis, both in investigating hypothesis states as in performing tests. The system only assists the user by determining a ranking for the test and showing the impact of performing a test. Unfortunately, the major disadvantage of this application is the restriction of pursuing only one state instead of multiple states. If all the hypothesis states are mutually exclusive as in the naive Bayes structure this application would be logical and even useful. However, if multiple causes are possible this application totally ignores the other causes and only focusses on proving the presence or absence of the selected cause.

4. MARGINAL PROBABILITY APPROACH

In Section 3.2 and 3.5 it appeared that the support for multiple hypothesis variables may become too difficult to realize. Below I formalize this problem and address it by investigating the relation between marginal and joint probability distributions. Based on this research I propose and evaluate an approximation approach which should solve the problem and still present a valuable diagnosis. Another approximation approach is presented in the next chapter.

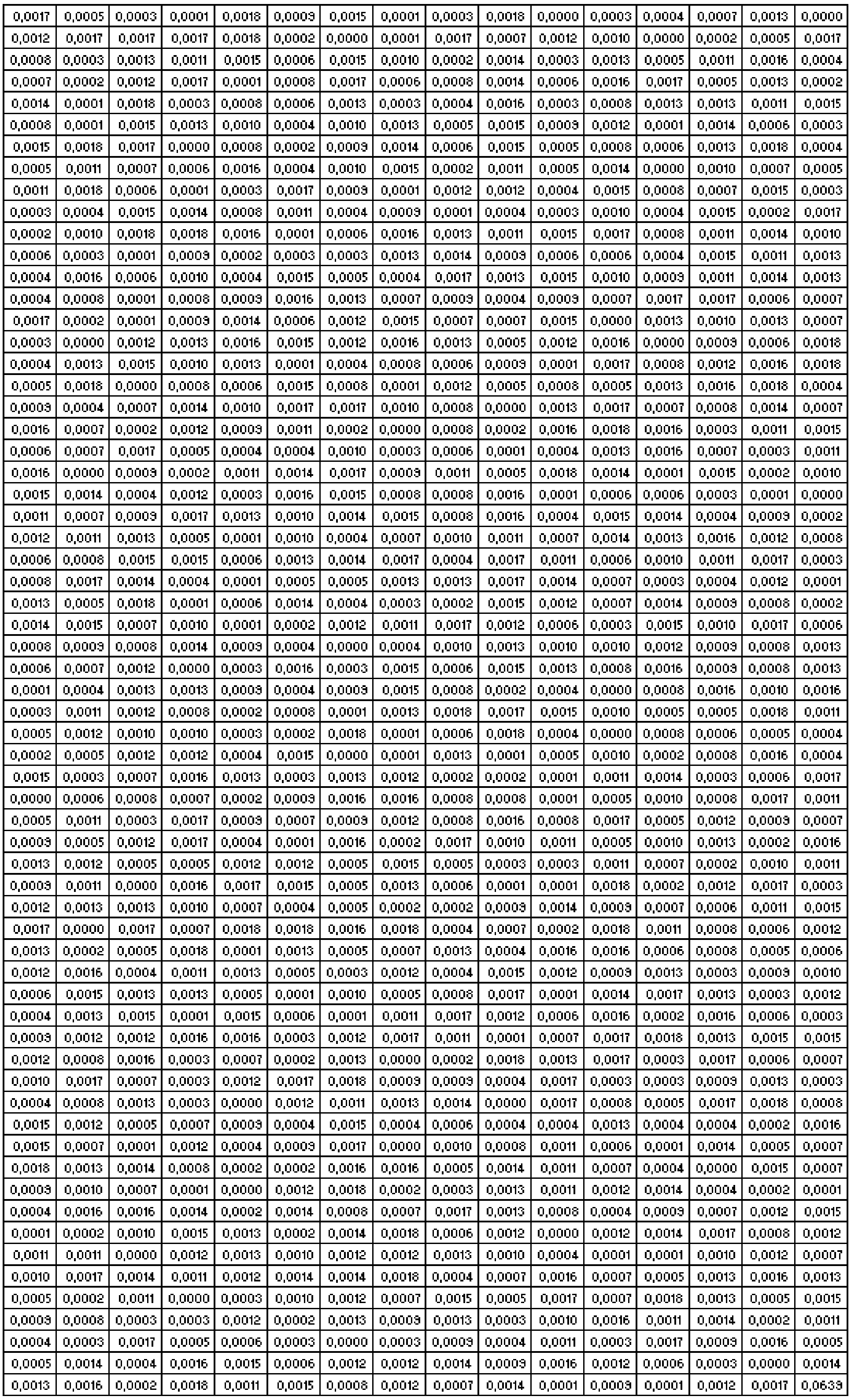
4.1 Problem Analysis

The diagnostic probabilistic network (DPN) provides support for multiple causes by allowing a set of multiple hypothesis variables. As described in Section 3.1 the process of diagnosis is to determine the most likely scenario by collecting more information. Which information to collect, is determined by the value functions which calculate the effect of a test on the probability of the scenarios. Although this support is complete and exact, the exponential growth of the number of scenarios[1] causes both presentational and computational problems.

The major presentational problem is the complexity for a user to grasp the exponential number of possible scenarios. Consider, for example a system with 10 hypothesis variables where each variable has 2 states and the user wishes to keep track of the effect of performing a test on all scenarios. This means that the user would have to observe the change in probabilities of $2^{10} = 1024$ scenarios. To give an idea of the difficulty of presenting and working with such a large number, I present a random probability distribution of 1024 scenarios, see Figure 4.1. Suppose now that only one probability changes because of performing a test and the user has to notice the effect of this change.

Apart from the trouble in presenting the exponential number of scenarios, the number also causes computational problems when applying the value functions. Since the value functions depend on the probabilities of the hypothesis scenarios, it is necessary that the entire joint probability distribution over the hypothesis set is calculated. Although the chain rule, see Theorem 2.1, is available to efficiently calculate this distribution, the space needed for storing all the probabilities becomes too large. Furthermore, little effort has so far been performed to develop an efficient algorithm for determining the joint probability distribution over a set of variables [Xu, 1995, Duncan, 2001].

[1] The number of scenarios over a set of variables $\mathcal{X} = \{X_1, ..., X_n\}$, is computed by multiplication of the number of states at each variable n_X: $\prod_{X \in \mathcal{X}} (n_X)$.



0,0017	0,0005	0,0003	0,0001	0,0018	0,0009	0,0015	0,0001	0,0003	0,0018	0,0000	0,0003	0,0004	0,0007	0,0013	0,0000
0,0012	0,0017	0,0017	0,0017	0,0018	0,0002	0,0000	0,0001	0,0017	0,0007	0,0012	0,0010	0,0000	0,0002	0,0005	0,0017
0,0008	0,0003	0,0013	0,0011	0,0015	0,0006	0,0015	0,0010	0,0002	0,0014	0,0003	0,0013	0,0005	0,0011	0,0016	0,0004
0,0007	0,0002	0,0012	0,0017	0,0001	0,0008	0,0017	0,0006	0,0008	0,0014	0,0006	0,0016	0,0017	0,0005	0,0013	0,0002
0,0014	0,0001	0,0018	0,0003	0,0008	0,0006	0,0013	0,0003	0,0004	0,0016	0,0003	0,0008	0,0013	0,0013	0,0011	0,0015
0,0008	0,0001	0,0015	0,0013	0,0010	0,0004	0,0010	0,0013	0,0005	0,0015	0,0009	0,0012	0,0001	0,0014	0,0006	0,0003
0,0015	0,0018	0,0017	0,0000	0,0008	0,0002	0,0009	0,0014	0,0006	0,0015	0,0005	0,0008	0,0006	0,0013	0,0018	0,0004
0,0005	0,0011	0,0007	0,0006	0,0016	0,0004	0,0010	0,0015	0,0002	0,0011	0,0005	0,0014	0,0000	0,0010	0,0007	0,0005
0,0011	0,0018	0,0006	0,0001	0,0003	0,0017	0,0009	0,0001	0,0012	0,0012	0,0004	0,0015	0,0008	0,0007	0,0015	0,0003
0,0003	0,0004	0,0015	0,0014	0,0008	0,0011	0,0004	0,0009	0,0001	0,0004	0,0003	0,0010	0,0004	0,0015	0,0002	0,0017
0,0002	0,0010	0,0018	0,0018	0,0016	0,0001	0,0006	0,0016	0,0013	0,0011	0,0015	0,0017	0,0008	0,0011	0,0014	0,0010
0,0006	0,0003	0,0001	0,0009	0,0002	0,0003	0,0003	0,0013	0,0014	0,0009	0,0006	0,0006	0,0004	0,0015	0,0011	0,0013
0,0004	0,0016	0,0006	0,0010	0,0004	0,0015	0,0005	0,0004	0,0017	0,0013	0,0015	0,0010	0,0009	0,0011	0,0014	0,0013
0,0004	0,0008	0,0001	0,0008	0,0009	0,0016	0,0013	0,0007	0,0009	0,0004	0,0009	0,0007	0,0017	0,0017	0,0006	0,0007
0,0017	0,0002	0,0001	0,0009	0,0014	0,0006	0,0012	0,0015	0,0007	0,0007	0,0015	0,0000	0,0013	0,0010	0,0013	0,0007
0,0003	0,0000	0,0012	0,0013	0,0016	0,0015	0,0012	0,0016	0,0013	0,0005	0,0012	0,0016	0,0000	0,0009	0,0006	0,0018
0,0004	0,0013	0,0015	0,0010	0,0013	0,0001	0,0004	0,0008	0,0006	0,0009	0,0001	0,0017	0,0008	0,0012	0,0016	0,0018
0,0005	0,0018	0,0000	0,0008	0,0006	0,0015	0,0008	0,0001	0,0012	0,0005	0,0008	0,0005	0,0013	0,0016	0,0018	0,0004
0,0009	0,0004	0,0007	0,0014	0,0010	0,0017	0,0017	0,0010	0,0008	0,0000	0,0013	0,0017	0,0007	0,0008	0,0014	0,0007
0,0016	0,0007	0,0002	0,0012	0,0009	0,0011	0,0002	0,0000	0,0008	0,0002	0,0016	0,0018	0,0016	0,0003	0,0011	0,0015
0,0006	0,0007	0,0017	0,0005	0,0004	0,0004	0,0010	0,0003	0,0006	0,0001	0,0004	0,0013	0,0016	0,0007	0,0003	0,0011
0,0016	0,0000	0,0009	0,0002	0,0011	0,0014	0,0017	0,0009	0,0011	0,0005	0,0018	0,0014	0,0001	0,0015	0,0002	0,0010
0,0015	0,0014	0,0004	0,0012	0,0003	0,0016	0,0015	0,0008	0,0008	0,0016	0,0001	0,0006	0,0006	0,0003	0,0001	0,0000
0,0011	0,0007	0,0009	0,0017	0,0013	0,0010	0,0014	0,0015	0,0008	0,0016	0,0004	0,0015	0,0014	0,0004	0,0009	0,0002
0,0012	0,0011	0,0013	0,0005	0,0001	0,0010	0,0004	0,0007	0,0010	0,0011	0,0007	0,0014	0,0013	0,0016	0,0012	0,0008
0,0006	0,0008	0,0015	0,0015	0,0006	0,0013	0,0014	0,0017	0,0004	0,0017	0,0011	0,0006	0,0010	0,0011	0,0017	0,0003
0,0008	0,0017	0,0014	0,0004	0,0001	0,0005	0,0005	0,0013	0,0013	0,0017	0,0014	0,0007	0,0003	0,0004	0,0012	0,0001
0,0013	0,0005	0,0018	0,0001	0,0006	0,0014	0,0004	0,0003	0,0002	0,0015	0,0012	0,0007	0,0014	0,0009	0,0008	0,0002
0,0014	0,0015	0,0007	0,0010	0,0001	0,0002	0,0012	0,0011	0,0017	0,0012	0,0006	0,0003	0,0015	0,0010	0,0017	0,0006
0,0008	0,0009	0,0008	0,0014	0,0009	0,0004	0,0000	0,0004	0,0010	0,0013	0,0010	0,0010	0,0012	0,0009	0,0008	0,0013
0,0006	0,0007	0,0012	0,0000	0,0003	0,0016	0,0003	0,0015	0,0006	0,0015	0,0013	0,0008	0,0016	0,0009	0,0008	0,0013
0,0001	0,0004	0,0013	0,0013	0,0009	0,0004	0,0009	0,0015	0,0008	0,0002	0,0004	0,0000	0,0008	0,0016	0,0010	0,0016
0,0003	0,0011	0,0012	0,0008	0,0002	0,0008	0,0001	0,0013	0,0018	0,0017	0,0015	0,0010	0,0005	0,0005	0,0018	0,0011
0,0005	0,0012	0,0010	0,0010	0,0003	0,0002	0,0018	0,0001	0,0006	0,0018	0,0004	0,0000	0,0008	0,0006	0,0005	0,0004
0,0002	0,0005	0,0012	0,0012	0,0004	0,0015	0,0000	0,0001	0,0013	0,0001	0,0005	0,0010	0,0002	0,0008	0,0016	0,0004
0,0015	0,0003	0,0007	0,0016	0,0013	0,0003	0,0013	0,0012	0,0002	0,0002	0,0001	0,0011	0,0014	0,0003	0,0006	0,0017
0,0000	0,0006	0,0008	0,0007	0,0002	0,0009	0,0016	0,0016	0,0008	0,0008	0,0001	0,0005	0,0010	0,0008	0,0017	0,0011
0,0005	0,0011	0,0003	0,0017	0,0009	0,0007	0,0009	0,0012	0,0008	0,0016	0,0008	0,0017	0,0005	0,0012	0,0009	0,0007
0,0009	0,0005	0,0012	0,0017	0,0004	0,0001	0,0016	0,0002	0,0017	0,0010	0,0011	0,0005	0,0010	0,0013	0,0002	0,0016
0,0013	0,0012	0,0005	0,0005	0,0012	0,0012	0,0005	0,0015	0,0005	0,0003	0,0003	0,0011	0,0007	0,0002	0,0010	0,0011
0,0009	0,0011	0,0000	0,0016	0,0017	0,0015	0,0005	0,0013	0,0006	0,0001	0,0001	0,0018	0,0002	0,0012	0,0017	0,0003
0,0012	0,0013	0,0013	0,0010	0,0007	0,0004	0,0005	0,0002	0,0002	0,0009	0,0014	0,0009	0,0007	0,0006	0,0011	0,0015
0,0017	0,0000	0,0017	0,0007	0,0018	0,0018	0,0016	0,0018	0,0004	0,0007	0,0002	0,0018	0,0011	0,0008	0,0006	0,0012
0,0013	0,0002	0,0005	0,0018	0,0001	0,0013	0,0005	0,0007	0,0013	0,0004	0,0016	0,0016	0,0006	0,0008	0,0005	0,0006
0,0012	0,0016	0,0004	0,0011	0,0013	0,0005	0,0003	0,0012	0,0004	0,0015	0,0012	0,0009	0,0013	0,0003	0,0009	0,0010
0,0006	0,0015	0,0013	0,0013	0,0005	0,0001	0,0010	0,0005	0,0008	0,0017	0,0001	0,0014	0,0017	0,0013	0,0003	0,0012
0,0004	0,0013	0,0015	0,0001	0,0015	0,0006	0,0001	0,0011	0,0017	0,0012	0,0006	0,0016	0,0002	0,0016	0,0006	0,0003
0,0009	0,0012	0,0012	0,0016	0,0016	0,0003	0,0012	0,0017	0,0011	0,0001	0,0007	0,0017	0,0018	0,0013	0,0015	0,0015
0,0012	0,0008	0,0016	0,0003	0,0007	0,0002	0,0013	0,0000	0,0002	0,0018	0,0013	0,0017	0,0003	0,0017	0,0006	0,0007
0,0010	0,0017	0,0007	0,0003	0,0012	0,0017	0,0018	0,0009	0,0009	0,0004	0,0017	0,0003	0,0003	0,0009	0,0013	0,0003
0,0004	0,0008	0,0013	0,0003	0,0000	0,0012	0,0011	0,0013	0,0014	0,0000	0,0017	0,0008	0,0005	0,0017	0,0018	0,0008
0,0015	0,0012	0,0005	0,0007	0,0009	0,0004	0,0015	0,0004	0,0006	0,0004	0,0004	0,0013	0,0004	0,0004	0,0002	0,0016
0,0015	0,0007	0,0001	0,0012	0,0004	0,0009	0,0017	0,0000	0,0010	0,0008	0,0011	0,0006	0,0001	0,0014	0,0005	0,0007
0,0018	0,0013	0,0014	0,0008	0,0002	0,0002	0,0016	0,0016	0,0005	0,0014	0,0011	0,0007	0,0004	0,0000	0,0015	0,0007
0,0009	0,0010	0,0007	0,0001	0,0000	0,0012	0,0018	0,0002	0,0003	0,0013	0,0011	0,0012	0,0014	0,0004	0,0002	0,0001
0,0004	0,0016	0,0016	0,0014	0,0002	0,0014	0,0008	0,0007	0,0017	0,0013	0,0008	0,0004	0,0009	0,0007	0,0012	0,0015
0,0001	0,0002	0,0010	0,0015	0,0013	0,0002	0,0014	0,0018	0,0006	0,0012	0,0000	0,0012	0,0014	0,0017	0,0008	0,0012
0,0011	0,0011	0,0000	0,0012	0,0013	0,0010	0,0012	0,0012	0,0013	0,0010	0,0004	0,0001	0,0001	0,0010	0,0012	0,0007
0,0010	0,0017	0,0014	0,0011	0,0012	0,0014	0,0014	0,0018	0,0004	0,0007	0,0016	0,0007	0,0005	0,0013	0,0016	0,0013
0,0005	0,0002	0,0011	0,0000	0,0003	0,0010	0,0012	0,0007	0,0015	0,0005	0,0017	0,0007	0,0018	0,0013	0,0005	0,0015
0,0009	0,0008	0,0003	0,0003	0,0012	0,0002	0,0013	0,0009	0,0013	0,0003	0,0010	0,0016	0,0011	0,0014	0,0002	0,0011
0,0004	0,0003	0,0017	0,0005	0,0006	0,0003	0,0000	0,0003	0,0009	0,0004	0,0011	0,0003	0,0017	0,0009	0,0016	0,0005
0,0005	0,0014	0,0004	0,0016	0,0015	0,0006	0,0012	0,0012	0,0014	0,0009	0,0016	0,0012	0,0006	0,0003	0,0000	0,0014
0,0013	0,0016	0,0002	0,0018	0,0011	0,0015	0,0008	0,0012	0,0007	0,0014	0,0001	0,0009	0,0001	0,0012	0,0017	0,0639

Fig. 4.1: A random probability distribution of 1024 scenarios

On the other hand, several efficient reasoning algorithms are available that determine the effect of new evidence on marginal probabilities [Lauritzen and Spiegelhalter, 1988, Pearl, 1988, Huang and Darwiche, 1994].

To solve the two problems described above, I propose two approximation approaches. Each of these approaches provides a solution to the computationally complexity as well as the presentational complexity. The approaches are described in detail in the current and next chapter, but for better understanding I provide a short summary of each approach.

Marginal Probability Approach
The first approach, referred to as the marginal probability approach, uses the relation between the marginal and joint probability distribution to justify the use of marginal probabilities. This approach saves a lot of computational effort, since the joint probability distribution is no longer calculated. Furthermore, the approach allows the presentation of the hypothesis states instead of the enormous number of hypothesis scenarios.

Joint Probability Approach
The second approach, referred to as the joint probability approach, is less radical in the sense that it uses an approximation for the joint probability. The applied approximation is the use of a marginal based or *copula* function to create the joint probability distribution and the use of a differential technique to allow the display of marginal probabilities.

Along with solving these general problems, these approaches are designed to continue the interactivity support provided by the diagnostic application from Section 3.5. In particular, allowing the user to direct the process of diagnosis. Within multiple cause diagnosis this support translates to providing the user with the ability to select and pursue any set of scenarios.

4.2 Relation between Marginal and Joint Probability

It is obvious that a strong relation exists between the marginal probabilities of states and the joint probabilities over the states. Below, I formalize this relation by deriving a lower and upper bound on the joint probability based on the marginal probabilities. The quality of this relation is determined by investigating the difference between the bounds under a growing number of variables. The derived bounds are not new but are better known as the Fréchet-Hoeffding bounds [Fréchet, 1957]. Since these bounds are originally derived for continuous probability distribution functions, I derive them for the discrete case.

Theorem 4.1 (upper bound joint probability). *Let $\mathcal{X}$ be a set of random variables $\{X_1, ..., X_n\}$, where each variable is associated with a probability distribution. The joint probability over a possible combination of states (scenario) $(x_1, ..., x_n)$ is smaller than or equal to the minimum of the marginal probabilities of the states in the combination, i.e.,*

$$\Pr(x_1, ..., x_n) \leq \min_i \Pr(x_i) \quad \forall X_i \in \mathcal{X}.$$

Proof. *I consider a set with two discrete binary random variables A and B, with each a probability distribution* $\Pr(a_1)$ & $\Pr(a_2)$ *and* $\Pr(b_1)$ & $\Pr(b_2)$. *From the Venn diagram in Figure 4.2, it is easy to see that* $\Pr(a_1, b_1) \leq \Pr(a_1)$ *and* $\Pr(a_1, b_1) \leq \Pr(b_1)$ *or in other words the joint probability of* (a_1, b_1) *is always smaller than or equal to the minimum of the probabilities of* a_1 *and* b_1. *For more than two variables or for variables with more states, the proof is similar.*

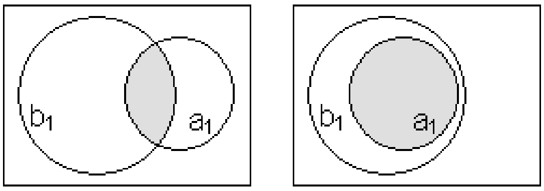


Fig. 4.2: Venn diagram of the probability distributions of A and B where the joint probability $\Pr(a_1, b_1)$ (arced area) is always smaller than the minimum of probability of a_1 and b_1

□

From Theorem 4.1, I learn that when at least one of the marginal probabilities of the states in the scenario is low (close to 0), it is already impossible that the probability of the scenario is high (close to 1). In other words, a high joint probability of a scenario is only possible when all the marginal probabilities of the states in the scenario are high. However, high marginal probabilities do not automatically imply a high joint probability. To ensure this I need a high lower bound. That a lower bound for a scenario exists is revealed in the following example.

Example 4. Suppose two random variables A and B, with states a_1, a_2 and b_1, b_2. Associated with the states are the following probabilities, $\Pr(a_1) = 0.9$, $\Pr(a_2) = 0.1$ and $\Pr(b_1) = 0.8$, $\Pr(b_2) = 0.2$.

From the joint probability table, see Table 4.1, it is easy to see that $\Pr(a_1, b_1)$ have to be at least 0.7, since $\Pr(a_1, b_1)$ and $\Pr(a_1, b_2)$ have to add up to 0.9 and $\Pr(a_1, b_2)$ is at most 0.2. □

Tab. 4.1: Joint probability table with the marginal probabilities of the states at the right and below the table

	b_1	b_2	
a_1	≥ 0.7	≤ 0.2	0.9
a_2	≤ 0.1	≤ 0.1	0.1
	0.8	0.2	

Before I present the theorem and the formula for the lower bound, I derive the lower bound formula for the scenario (a_1, b_1) from Example 4. The first step is to use marginalization to write the relation between marginal probability and the probability of a scenario.

$$\Pr(a_1, b_1) + \Pr(a_1, b_2) = \Pr(a_1) \tag{4.1}$$

Since I am interested in a formula with only the probability of the first scenario and the marginal probabilities, I replace the second scenario (a_1, b_2), by a marginal probability. The only suitable marginal probability is that of $\Pr(b_2)$, since $\Pr(b_2) = \Pr(a_1, b_2) + \Pr(a_2, b_2)$. Because I am only interested in marginal probabilities I use the approximation $\Pr(b_2) \geq \Pr(a_1, b_2)$. Replacing $\Pr(a_1, b_2)$ in Equation 4.1 with the approximation $\Pr(b_2)$ gives:

$$\begin{aligned}&\Pr(a_1, b_1) + \Pr(b_2) \geq \Pr(a_1)\\&\Pr(a_1, b_1) \geq \Pr(a_1) - \Pr(b_2).\end{aligned}$$

I see that this formula agrees with the lower bound I found in Example 4:

$$\Pr(a_1) - \Pr(b_2) \Longleftrightarrow 0.9 - 0.2 = 0.7.$$

In short, the lower bound for the probability of a particular scenario is derived by replacing the probability of each scenario, except the lower bound-scenario, with an approximation of marginal probabilities. Because scenarios are unique, I am able to say that every scenario, apart from the lower bound-scenario, contains at least one state that is not in the lower bound scenario. Therefore, to replace all the probabilities of scenarios and in particular the scenarios where only one state is different, I need all the marginal probabilities of all the states not in the lower bound scenario. Below I formalize this process in the proof of the lower bound joint probability theorem.

Theorem 4.2 (lower bound joint probability). *Let $\mathcal{X}$ be a set of random variables $\{X_1, ..., X_n\}$, where each variable X_i is associated with a probability distribution. The lower bound for the joint probability of a possible combination of states (scenario) $(x_1, ..., x_n)$ is,*

$$\Pr(x_1, ..., x_n) \geq \max\left\{0; 1 - n + \sum_{i=1}^{n} \Pr(x_i)\right\}.$$

Proof. *I consider a set of random variables $\mathcal{X} = \{X_1, ..., X_n\}$, where each variable X_i is associated with a probability distribution $\Pr(X_i)$. Each variable X_i, consists of states which are denoted as $x_i^k \in D_{X_i}$. The index k represents the index of the state in the variable X_i and since every variable may consist of a different number of states each variable has a different letter that indexes the states. Without loss of generality, I prove the lower bound for the scenario $\left(x_1^1, x_2^1, ..., x_n^1\right)$.*

With marginalization, I write the probability of x_1^1 as,

$$\Pr\left(x_1^1\right) = \sum_{x_2^k} ... \sum_{x_n^m} \Pr\left(x_1^1, x_2^k, ..., x_n^m\right) \tag{4.2}$$

$$\Pr\left(x_1^1\right) = \Pr\left(x_1^1, x_2^1, ..., x_n^1\right) + \sum_{x_2^{k\neq 1}} ... \sum_{x_n^{m\neq 1}} \Pr\left(x_1^1, x_2^k, ..., x_n^m\right) \tag{4.3}$$

$$\Pr\left(x_1^1, x_2^1, ..., x_n^1\right) = \Pr\left(x_1^1\right) - \sum_{x_2^{k\neq 1}} ... \sum_{x_n^{m\neq 1}} \Pr\left(x_1^1, x_2^k, ..., x_n^m\right). \tag{4.4}$$

The goal is to rewrite Equation 4.4 with marginal probabilities. Since each scenario in the sum has at least one state that is not in the scenario $\left(x_1^1, x_2^1, ..., x_n^1\right)$, *I need all the marginal probabilities of the states which are not in the scenario* $\left(x_1^1, x_2^1, ..., x_n^1\right)$*. During replacement I omit the scenarios associated with the marginal probabilities so each marginal probability is greater than the scenarios it replaces. The sum from Equation 4.4 is then,*

$$\sum_{x_2^{k\neq 1}} ... \sum_{x_n^{m\neq 1}} \Pr\left(x_1^1, x_2^k, ..., x_n^m\right) \leq \sum_{x_2^{k\neq 1}} \Pr\left(x_2^k\right) + ... + \sum_{x_n^{m\neq 1}} \Pr\left(x_2^m\right) \tag{4.5}$$

$$\sum_{x_2^{k\neq 1}} ... \sum_{x_n^{m\neq 1}} \Pr\left(x_1^1, x_2^k, ..., x_n^m\right) \leq \left(1 - \Pr\left(x_2^1\right)\right) + ... + \left(1 - \Pr\left(x_n^1\right)\right) \tag{4.6}$$

$$\sum_{x_2^{k\neq 1}} ... \sum_{x_n^{m\neq 1}} \Pr\left(x_1^1, x_2^k, ..., x_n^m\right) \leq n - 1 - \sum_{i=2}^{n} \Pr\left(x_i^1\right). \tag{4.7}$$

Combining this Equation 4.4 with Equation 4.7, I get the desired lower bound.

$$\Pr\left(x_1^1, x_2^1, ..., x_n^1\right) \geq 1 - n + \Pr\left(x_i^1\right) + \sum_{i=2}^{n} \Pr\left(x_i^1\right) \tag{4.8}$$

$$\Pr\left(x_1^1, x_2^1, ..., x_n^1\right) \geq 1 - n + \sum_{i=1}^{n} \Pr\left(x_i^1\right) \tag{4.9}$$

Since the proof is valid for any possible scenario, I use the following notation, $\Pr\left(x_1, ..., x_n\right) \geq 1 - n + \sum_{i=1}^{n} \Pr\left(x_i^1\right)$*. When* $1 - n + \sum_{i=1}^{n} \Pr\left(x_i^1\right) < 0$*, the lower bound is equal to zero.* □

Since the objective of performing diagnosis is to reduce the uncertainty in a system, I am interested in scenarios that have probabilities close to zero and one. The following corollary states that these interesting probabilities are only possible if the marginal probabilities are close to zero and one.

Corollary 4.1 (marginal strength). *Suppose a set of random variables* $\mathcal{X} = \{X_1, ..., X_n\}$*, where each variable is associated with a probability distribution. The probability of a scenario s from the domain* $D_{\mathcal{X}}$ *is only close to* 0 *or* 1*, if and only if, all the marginal probabilities of the variables in the set* $\mathcal{X}$ *are maximal or close to* 1*.*

This corollary provides the right to change the goal of reducing the uncertainty between scenarios to the goal of reducing uncertainty of marginal probabilities. The benefits of this approach are noticeable in the number of computations, since it is no longer necessary to determine the computationally expensive joint probability distribution. But also in the perception of the user that may look at the states and their marginal probabilities and interpret them directly instead of needing to look at the numerous scenarios and their probabilities. Consider, for example a number of states with each a high marginal probability and all the other states with a low marginal probability. The user may deduce that the combinations of states with the low probabilities states have a low probability. And the scenario that consists of states with high marginal probabilities has a high probability.

Although Corollary 4.1 is true for any possible set of states, it is stronger for a small set of states than for a large set of states. The reason for this is the strong dependency of the lower bound with the number of states. Consider, for example the determination of the lower bound for a scenario where all the marginal probabilities of the states are 0.9. In case of a scenario of two states, the lower bound will be $0.9 - 0.1 = 0.8$. However, if the scenario consists of 10 states, then the lower bound will be $0.9 - (9 \cdot 0.1) = 0$, i.e., no meaningful statement about the lower bound can be made.

This behavior is in sharp contrast with the upper bound, which is independent of the number of variables. So the upper bound is always as low as the lowest marginal probability of the states in a scenario. In order to get more insight into this behavior I investigate the maximal distance between the upper and lower bound. As stated in the following theorem, this distance depends on the number of variables and will grow when the number of variables grows.

Theorem 4.3 (maximum distance between bounds). *Let $\mathcal{X} = \{X_1, ..., X_n\}$ be a set of random variables and $s = (x_1, ..., x_n)$ a scenario where $s \in D_{\mathcal{X}}$. The distance between the upper and lower bound, from respectively Theorem 4.1 and 4.2, for the probability of the scenario s is at most $1 - \frac{1}{n}$.*

Proof. *See Appendix A.* □

In Figure 4.3, the distance is shown between upper and lower bound for all the possible probabilities of a scenario of two states (a, b). According to Theorem 4.3, the distance between the upper and lower bound is maximal $1 - \frac{1}{2} = \frac{1}{2}$. Obvious is that the minimal distance between the bounds is zero and is reached when all marginal probabilities are equal to zero or one. That the maximal distance for the scenario of two states is reached when both states have a probability of $\frac{1}{2}$ is not a coincidence. The following theorem states that the maximal distance occurs when each marginal probability is equal to $1 - \frac{1}{n}$.

Theorem 4.4 (maximum distance condition). *The distance between the upper and lower bound of the probability of a scenario $s = (x_1, ..., x_n)$ in a set of random variables $\mathcal{X} = \{X_1, ..., X_n\}$ is maximal and equal to $1 - \frac{1}{n}$, if and only if, every marginal probability of the states in the scenario s is equal to $1 - \frac{1}{n}$.*

Proof. *See Appendix A.* □

This theorem provides me with the information on how to attend to the increasing distance consequent to an increasing number of variables. When the marginal probabilities are equal to $1 - \frac{1}{n}$, they are less interesting, since such a distribution makes me most uncertain of the true joint probability.

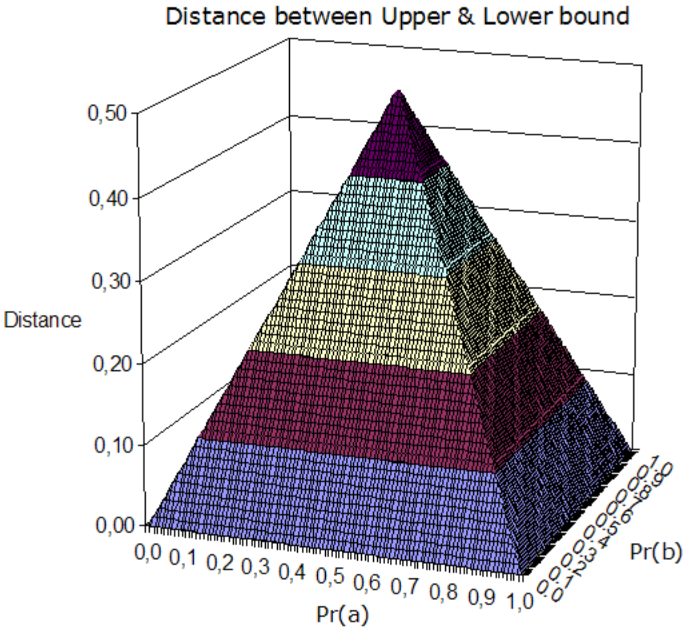


Fig. 4.3: Distance between upper and lower bound for all the possible probabilities of the scenario (a, b)

4.3 Marginal Probability Approach

The research described above formalizes the relation between the marginal and joint probability. The bounds that the marginal probabilities impose on the joint probability allows me to say that small and large joint probabilities are only possible when the marginal probabilities are either small or large. Justified by this knowledge, I propose a marginal probability approach that basically uses the marginal probabilities of the states in the scenarios instead of the joint probability over the states.

This approach refrains from working with the enormous number of scenarios but instead uses the limited number of states. The advantages of this approach are that it immediately solves the complexity aspect of presenting multiple cause diagnosis but also the computational problems. So it is no longer necessary to display all the scenarios of a set of hypothesis variables but only the hypothesis states. For this approach I designed new test selection measures, which also focus on reducing the uncertainty of the hypothesis set, but use marginal probabilities instead of joint probabilities. These new test selection measures, or marginal based test selection measures, are derived in the next section. Another advantage of this approach is that it continues the support to direct the process of diagnosis. During the selection of multiple states of a list of hypothesis states, the user is actually selecting scenarios.

That the approach is based on a large approximation becomes clear when observing the growing distance between the bounds, see Theorem 4.3. From this distance I may conclude that a given set of states with marginal probabilities, may generate any probability within these bounds for the combination of these states. The point where the distance is maximal is the worst situation, since I know the least about the value of the joint probability. According to Theorem 4.4 this maximal uncertainty occurs when all the marginal probabilities are equal to $1 - \frac{1}{n}$. To account for this problem, the new test selection measures should account for this negative effect by not rewarding marginal probabilities equal or close to $1 - \frac{1}{n}$.

How this approach is implemented and how the user is able to use it, is explained in Chapter 6.

4.4 Marginal Based Test Selection Measures

If I translate the marginal probability approach into the design of a marginal based test selection measure, I want a function that assigns high values to marginal probabilities close to 0 and 1 and has its minimum at $1 - \frac{1}{n}$. Together with these restrictions I prefer a convex function, so tests which provide information will always get a positive value.

Since the approach provides the user with the ability to select and pursue states and intermediately scenarios, I propose to refer to these states as target or focus states. These states are a selection of the hypothesis states of the different hypothesis variables. The set of targets is denoted as $\mathcal{F}$, each target state as f, and the number of targets in a set $\mathcal{F}$ as $n_{\mathcal{F}}$. The marginal based functions are applied over the probabilities of the target states.

Within the restrictions I created two functions, one without the support for the maximal distance and one with it. Both the functions have been scaled so that they return a ranking between zero and one. The following definition describes the function without the maximal distance support and has its minimum when all the probabilities of the targets are equal to 0.5.

Definition 4.1 (Marginal Strength1). *Let $\mathcal{F}$ be a set of target states where each target state f represents a hypothesis state which the user wishes to pursue. The marginal strength1 function $MS1\left(\Pr\left(\mathcal{F}\right)\right)$ is then,*

$$MS1\left(\Pr\left(\mathcal{F}\right)\right) \equiv \left(\frac{\sum_{f \in \mathcal{F}} \left(f - 0.5\right)^2}{\left(\frac{1}{2}\right)^2} - n_{\mathcal{F}} \right) * \frac{1}{n_{\mathcal{F}}}.$$

In Figure 4.4, the function is displayed for a set of two targets. It can be clearly be seen that the minimum of the function is reached when each of the states is equal to 0.5. That the function is convex is ensured by the summation of convex functions.

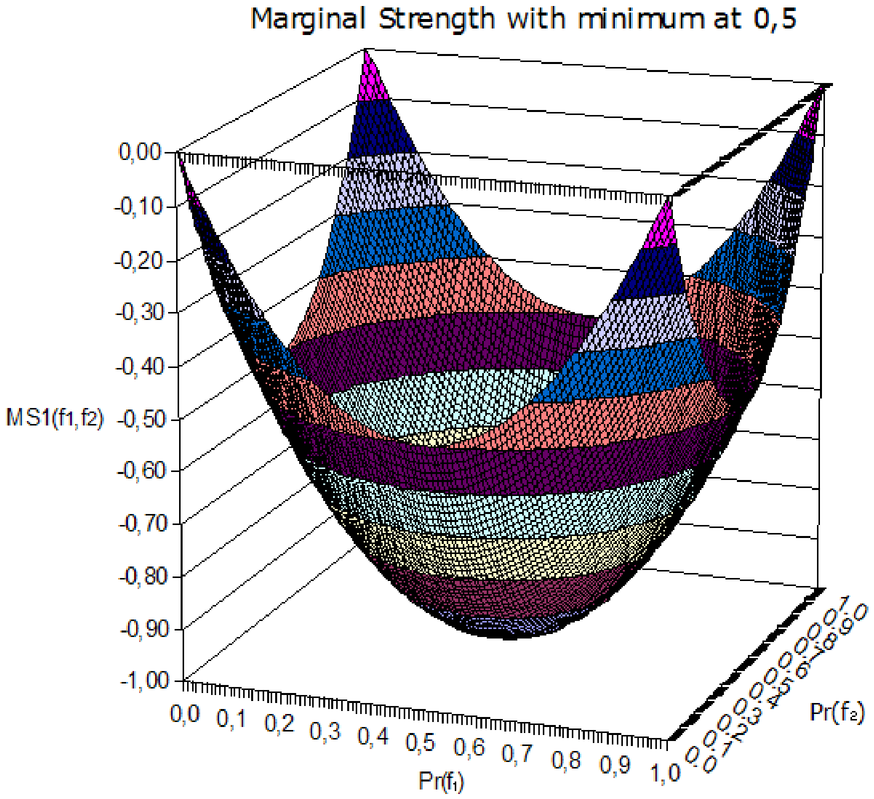


Fig. 4.4: The MS1 function over two random targets f_1 and f_2

The second function is a combination of two functions into one function, which is continuous on the area $[0, 1]$.

Definition 4.2 (Marginal Strength2). *Let $\mathcal{F}$ be a set of target states where each target state f represents a hypothesis state which the user wishes to pursue. The marginal strength function2 $MS2\left(\Pr\left(\mathcal{F}\right)\right)$ is then,*

$$MS2\left(\Pr\left(\mathcal{F}\right)\right) \equiv \begin{cases} \left(\frac{\sum_{f\in\mathcal{F}}\left(f-\left(1-\frac{1}{n_{\mathcal{F}}}\right)\right)^2}{\left(1-\frac{1}{n_{\mathcal{F}}}\right)^2} - n_{\mathcal{F}}\right) * \frac{1}{n_{\mathcal{F}}} & : \quad 0 \leq f \leq 1-\frac{1}{n_{\mathcal{F}}}. \\ \left(\frac{\sum_{f\in\mathcal{F}}\left(\left(f-\left(1-\frac{1}{n_{\mathcal{F}}}\right)\right)*\left(n_{\mathcal{F}}-1\right)\right)^2}{\left(1-\frac{1}{n_{\mathcal{F}}}\right)^2} - n_{\mathcal{F}}\right) * \frac{1}{n_{\mathcal{F}}} & : \quad 1-\frac{1}{n_{\mathcal{F}}} < f \leq 1. \end{cases}$$

In Figure 4.5, the change of the minimum depending on the number of targets $n = n_{\mathcal{F}}$ is shown. Within this figure it is assumed that each target f has an equal probability.

Whether the second function performs better than the first function is determined with tests in Chapter 7.

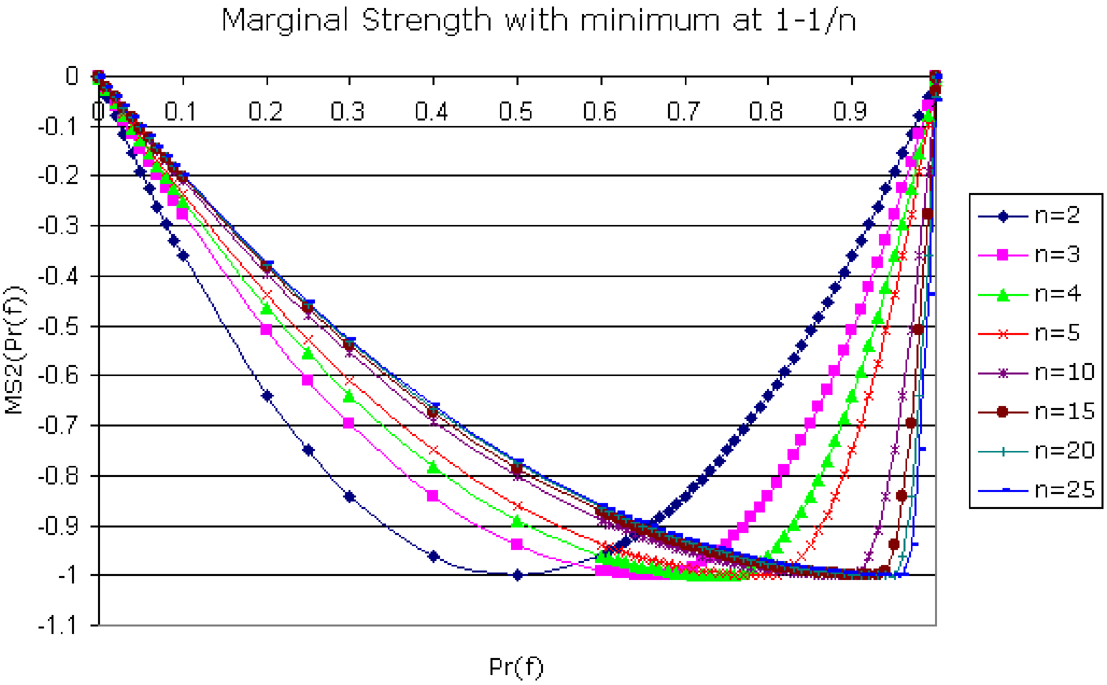


Fig. 4.5: The Marginal Strength2 function for a different number of targets $n = n_{\mathcal{F}}$

5. JOINT PROBABILITY APPROACH

The major advantage of using marginal based test selection measures for the process of diagnosis, is the speed and reduction in complexity. However, it is undeniable that the approach is based on a rough approximation. Therefore, I propose another approach which comes a lot closer to using the true joint probability. Although this approach is far less radical, it is far more expensive to calculate. Especially in large networks (networks with more than 20 or 30 nodes and a large number of dependency relations) this approach may cost an extreme amount of time.

Basically this approach is separated into two areas, the area of *copulas*, and the area of *differential diagnosis*. The first area provides marginal based approximations for the joint probability distribution. The second area basically allows the user to pursue and differentiate between any possible set of scenarios. This area is necessary for presenting the enormous number of scenarios to a user and still providing the user with the ability to pursue any set of scenarios.

5.1 Area of Copulas

In essence, copulas are functions that join or "couple" multivariate probability distributions to their one-dimensional marginal probability distribution [Nelsen, 1998]. Since the process of multiple cause diagnosis uses the joint probability distribution, this area may be applied to find a qualitative approximation for the joint probability. In doing so, I start with a general introduction to copulas, whereafter I investigate how the area of copulas may be applied. For this introduction I follow the notation used in Nelsen [1998], where it is assumed that the probability distributions are continuous and described by a cumulative distribution function. The cumulative distribution function or distribution function over a random variable X is a function $F_X : \mathbb{R} \to [0,1]$ defined as $F_X(x) = (X \leq x)$.

Before defining copulas I first need to introduce some additional notations. Let $\mathbb{R}$ denote the ordinary real line $(-\infty, \infty)$, and $\overline{\mathbb{R}}$ denote the extended real line $[-\infty, \infty]$. For any positive integer n, let $\overline{\mathbb{R}}^n$ denote the extended n-space $\overline{\mathbb{R}} \times \overline{\mathbb{R}} \times \cdots \times \overline{\mathbb{R}}$. The vector notation is used for the points in $\overline{\mathbb{R}}^n$, e.g., $\mathbf{a} = (a_1, a_2, ..., a_n)$. The notation $\mathbf{a} \leq \mathbf{b}$ is used when $a_k \leq b_k$ for all k. $[\mathbf{a}, \mathbf{b}]$ denotes the n-box $B = [a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_n, b_n]$, the Cartesian product of n closed intervals. The vertices of a n-box B are the points $\mathbf{c} = (c_1, c_2, ..., c_n)$ where each c_k is equal to either a_k or b_k. A n-place real function H is a function whose domain, $DomH$, is a subset of $\overline{\mathbb{R}}^n$ and whose range, $RanH$, is a subset of $\mathbb{R}$.

Since the definition of a n-dimensional copula depends on the positiveness of the volume under a n-place real function I start with the definition of a H-volume.

Definition 5.1 (H-volume). *Let $S_1, S_2, \ldots, S_n$ be nonempty subsets of $\overline{\mathbb{R}}$, and let H be a n-place real function such that $DomH = S_1 \times S_2 \times \cdots \times S_n$. Let $B = [\mathbf{a}, \mathbf{b}]$ be a n-box all of whose vertices are in $DomH$. Then the $H-volume$ of B is given by*

$$V_H\left(B\right) = \sum sgn(\mathbf{c})H(\mathbf{c}), \tag{5.1}$$

where the sum is taken over all vertices $\mathbf{c}$ of B; and $sgn(\mathbf{c})$ is given by

$$sgn\left(\mathbf{c}\right) = \begin{cases} 1, & \textit{if } c_k = a_k \textit{ for an even number of } k\textit{'s.} \\ -1, & \textit{if } c_k = a_k \textit{ for an odd number of } k\textit{'s.} \end{cases} \tag{5.2}$$

For the two dimensional case $n = 2$, where H be a function such that $DomH = S_1 \times S_2$. Let $B = [a_1, a_2] \times [b_1, b_2]$ be a rectangle all of whose vertices are in $DomH$. Then the H-volume of B is given by:

$$V_H(B) = H\left(x_2, y_2\right) - H\left(x_2, y_1\right) - H\left(x_1, y_2\right) + H\left(x_1, y_1\right). \tag{5.3}$$

Definition 5.2 (n-dimensional copula). *A n-dimensional copula is a function $C : [0,1]^n \rightarrow [0,1]$ with the following properties:*

1. *For every $\boldsymbol{a} \in [0,1]^n$,*

   $C\left(\boldsymbol{a}\right) = 0$ if at least one coordinate of $\boldsymbol{a}$ is 0.

2. *For every $\boldsymbol{a} \in [0,1]^n$,*

   if all coordinates of $\boldsymbol{a}$ are 1 except a_k, then $C\left(\boldsymbol{a}\right) = a_k$.

3. *For every $\boldsymbol{a}$ and $\boldsymbol{b}$ in $[0,1]^n$ such that $\boldsymbol{a} \leq \boldsymbol{b}$,*

   $V_C\left([\boldsymbol{a}, \boldsymbol{b}]\right) \geq 0$.

The following theorem introduces the connection between the copula functions and their margins.

Theorem 5.1 (Sklar's theorem in n-dimensions). *Let H be a n-dimensional distribution function with margins $F_1, F_2, \cdots, F_n$. Then there exists a n-copula C such that for all $\mathbf{x}$ in $\overline{\mathbb{R}}^n$,*

$$H\left(x_1, x_2, \cdots, x_n\right) = C\left(F_1\left(x_1\right), F_2\left(x_2\right), \ldots, F_n\left(x_n\right)\right).$$

If $F_1, F_2, \ldots, F_n$ are all continuous, then C is unique: otherwise, C is uniquely determined on $RanF_1 \times RanF_2 \times \cdots RanF_n$. Conversely, if C is a n-copula and $F_1, F_2, \ldots, F_n$ are distribution functions, then the function H defined above, is a n-dimensional distribution function with margins $F_1, F_2, \ldots, F_n$.
Proof. *See Nelsen [1998].* □

The following theorem presents the upper and lower bound of the Copula functions. Notice that they are equal to the ones I derived in Section 4.2.

Theorem 5.2 (copula bounds). *If C is any n-copula, then for every $\mathbf{u}$ in DomC there holds,*

$$M^n(\mathbf{u}) = \min(u_1, u_2, \ldots, u_n) \leq C(\mathbf{u}) \leq \max(u_1 + u_2 + \cdots, u_n - n + 1, 0) = W^n(\mathbf{u})$$

Proof. *See Nelsen [1998].* □

Within the area of Copulas many families of possible copula functions have been created, see [Joe, 1997] for an overview. As an example I present the *Cuadras-Augé family of copulas*, where $\theta \in [-1, 1]$ is a constant, representing the dependence between the variables:

$$C_\theta(\mathbf{u}) = [M^n(\mathbf{u})]^\theta [\Pi^n(\mathbf{u})]^{1-\theta}.$$

The value of θ may be determined with Spearman's correlation coefficient, see [Dall'Aglio *et al.*, 1991].

This short introduction provides a general idea of what copula functions are. However, when applying copulas to Bayesian networks various problems are encountered. One primary problem is that the theory of copulas is based on continuous (mostly) bivariate probability distributions, where as Bayesian networks generally work with discrete multivariate probability distributions. A possible solution is to translate the discrete probability distribution of every variable to a continuous distribution, apply a selected copula function to calculate the continuous joint probability distribution, and finally translate this distribution back to a discrete joint probability distribution. It is obvious that this approach is computationally quite expensive, and it also implies the use of several approximations necessary for the translations.

A much more appealing solution is using copula functions that work with discrete probability distributions and calculate a discrete joint probability distribution. Unfortunately, I have found only one copula function that supports this,

$$C_{prod}(\mathbf{u}) \equiv \prod^n(\mathbf{u}) = u_1 u_2 \cdots u_n.$$

Within copulas this function is better known as the product copula. In probability theory this function represents the joint probability when all the variables are independent of each other. To my knowledge this copula function is the only function that may be used for continuous as well as for discrete probabilities.

By using this function I am able to implement the copula approach in the test selection part of the diagnostic process. Before the test selection measures from Section 3.4 are applied, the product copula is used to determine the joint probability distribution. More information about the implementation of this approach is given in Section 6.3.

5.2 Differential Diagnosis

A way in which human diagnosticians cope with the complexity of diagnosis, is by counter opposing competitive hypothesizes and seeking evidence that differentiates between them. This approach, which I refer to as differential diagnosis, is considered standard in medical science, but only applied in a simplified form in current diagnostic expert systems. In this section I generalize this approach and adapt the test selection measures from Chapter 3.3, for the support of this approach.

Given a diagnostic probabilistic network (DPN) with a set $\mathcal{H}$ of hypothesis nodes, and $D_{\mathcal{H}}$ the set of all possible scenarios, I define the concept of differential diagnosis as the ability to differentiate between any possible partition P. A partition P of a set Δ is a set of non-empty subsets of Δ which are mutually disjoint and whose union is Δ. How the partitions are used in a DPN, is defined in the following definition.

Definition 5.3 (differential diagnosis partition). *A partition from a set of hypothesis nodes $\mathcal{H}$ consists of one or more scenarios from the domain $D_{\mathcal{H}}$. The probability of a partition P is defined as the sum of the probabilities of the scenarios $s \in P$, $\Pr(P) = \sum_{s \in P} \Pr(s)$. Within a DPN I disallow a partition that contains all the hypothesis scenarios.*

Below I present two possible partition selections within the diagnostic probabilistic network of the Asia example, see Example 3.2.

Example 5. Suppose differential diagnosis is performed on the hypothesis set $\mathcal{H} = \{TC, LC, BC\}$ in the DPN from Example 3 with all the possible scenarios in Table 3.1. Now assume a doctor wants to investigate whether a person has only one disease present. In this case four partitions are formed, see Table 5, three partitions with one disease present and other diseases absent and one partition which contains all other scenarios.

Tab. 5.1: Partitions with one disease present and the rest

P_1	P_2	P_3	P_4
TC_present *LC_absent* *BC_absent*	*TC_absent* *LC_present* *BC_absent*	*TC_absent* *LC_absent* *BC_present*	all other scenarios

Another interesting case is to pursue the scenario that all diseases are present. The partitions, see Table 5, are then, one partition with all diseases present and one partition with all other scenarios. □

Tab. 5.2: Partitions with all diseases present and the rest

P_1	P_2
TC_present *LC_present* *BC_present*	all other scenarios

For the partitions I want tests that distinguish between the probabilities of the partitions i.e., reduce the uncertainty between the partitions. Since the test selection measures from Section 3.4 share this same goal but for scenarios, I adapted these functions to work with partitions instead of scenarios.

Definition 5.4 (differential entropy). *Let $\mathcal{H}$ be a set of hypothesis variables, and let $\mathcal{P}$ be a set of n partitions that cover all the scenarios of $s \in D_{\mathcal{H}}$. The differential entropy function* $\textit{Diff\_ENT}\left(\Pr\left(P \in \mathcal{P}\right)\right) : [0;1]^n \to \mathbb{R}$ *is then,*

$$\textit{Diff\_ENT}\left(\Pr\left(\mathcal{P}\right)\right) \equiv \sum_{P \in \mathcal{P}} \Pr\left(P\right) \log_2\left(\Pr\left(P\right)\right)$$

with $log_2(0) = 0$.

This measure will calculate the entropy over any number of partitions $\{P_1, ..., P_n\}$. It is easy to see that all the properties specified for the entropy function, defined in Theorem 3.3 also hold for this function.

Definition 5.5 (differential weight of evidence). *Let $\mathcal{H}$ be a set of hypothesis variables, and let $\mathcal{P}$ be a set of n partitions that cover all the scenarios of $s \in D_{\mathcal{H}}$. The differential weight of evidence function* $\textit{Diff\_WOE}\left(\Pr\left(P \in \mathcal{P}\right)\right) : [0;1]^n \to \mathbb{R}$ *is then,*

$$\begin{aligned}\textit{Diff\_WOE}\left(\Pr\left(\mathcal{P}\right)\right) &\equiv \log \Pr\left(P_1\right) - \log \Pr\left(\overline{P_1}\right) \\ &= \log \Pr\left(P_1\right) - \log\left(1 - \Pr\left(P_1\right)\right) \\ &= \log \frac{\Pr\left(P_1\right)}{\left(1 - \Pr\left(P_1\right)\right)}\end{aligned}$$

Also here, this function restricts itself to comparing only one partition with the rest. It is easy to see that this function has the same properties as the ones specified for regular weight of evidence, defined in Theorem 3.4.

Although the benefit of this approach is the improved ability to control and direct the process of diagnosis it also increases presentational problems. So it is practically impossible in complex networks to let a user decide which partitions to create, since the number of partitions grows exponentially and even faster than the number of scenarios. Consider, for example the 10 hypothesis variables where each hypothesis variable has 2 states and the number of scenarios is then 1024. The number of possible partitions[1] is then 115974. Because of this exponentially fast growing number, it is essential to use a technique that makes the selection of partitions manageable for the user.

The technique I propose, is to let the user choose hypothesis states and select one out of three interesting methods to form partitions. This technique fits perfectly in the single fault diagnosis application from Section 3.5, where states are denoted as targets. These targets represent the states in which the user is

[1] The number of possible partitions is also known as the sum of Stirling numbers of the second kind $S(n,k)$ minus 1 since I do not allow the partition that contains all the scenarios; $\sum_{k=1}^{n} S\left(n,k\right) = \sum_{k=1}^{n} \frac{1}{k!} \sum_{i=0}^{k-1} \left(-1\right)^i \left(\begin{array}{c} k \\ i \end{array}\right) \left(k-i\right)^n - 1$, where n is the number of scenarios

interested and wishes to have the option of pursuing. After selecting a number of targets the user may choose one out of the three partition-distributions.

Partitions with one or more targets This distribution consists of partitions, where each partition contains a scenario with at least one target present and additional one partition for the rest of the scenarios.

Partitions with all the targets This distribution consists of partitions, where each partition contains a scenario with all the targets present and additional one partition for the rest of the scenarios.

Partitions with only one target This distribution consists of partitions, where each partition contains a scenario with at most one target present and additional one partition for the rest of the scenarios.

Suppose in Example 5 the presence of the diseases, tuberculosis, lung cancer and bronchitis are denoted as targets. Then Table 5 shows the first partition-distribution where all the targets are present and one partition for the rest. Furthermore, Table 5 represents the second partition-distribution where at least one target is present and one partition for the rest. If the third partition-distribution in this example is used, then each partition consists of one scenario.

With this technique I am able to provide the desired ability to pursue and differentiate between any set of scenarios and also provide the user with a workable interface. The details of the implementation of this technique in combination with the marginal based copula function is described in Section 6.3.

6. MULTIPLE CAUSE MODULE

This chapter describes the multiple cause module (MCM) I developed for the support of multiple cause diagnosis. For this support I implemented the approximation approaches derived in the previous chapters. The MCM has been made a part of the diagnostic module already implemented in GeNIe and SMILE. This existing module provides support for the diagnosis of a single cause as described in Section 3.5. With some small modifications the interface used for single cause diagnosis is also used for the MCM. The interface then provides the user with the ability to select any number of targets and investigates these. Additional to the two approximation approaches, marginal and joint probability approach, I also implemented the use of the true joint probability in combination with the entropy value function. Before I describe the details of the implementations I provide some information about GeNIe and SMILE.

6.1 GeNIe & SMILE

GeNIe is a versatile and user-friendly development environment for building graphical decision models developed at the Decision Systems Laboratory. Its name with its uncommon capitalization originates from the name Graphical Network Interface and has been developed at the Decision Systems Laboratory. This original simple interface was designed for SMILE (Structural Modelling, Reasoning, and Learning Engine), a library of C++ classes implementing graphical probabilistic and decision-theoretic models. GeNIe may be seen as an outer shell to SMILE. Furthermore, GeNIe is implemented in Visual C++ and draws heavily on the Microsoft Foundation Classes.

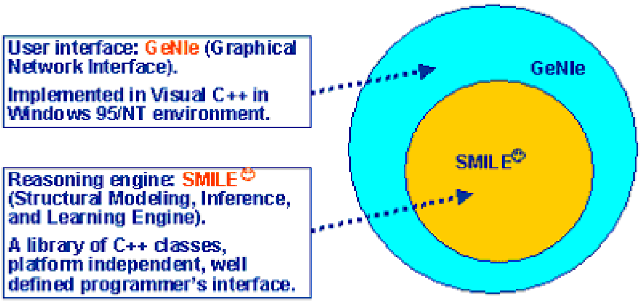
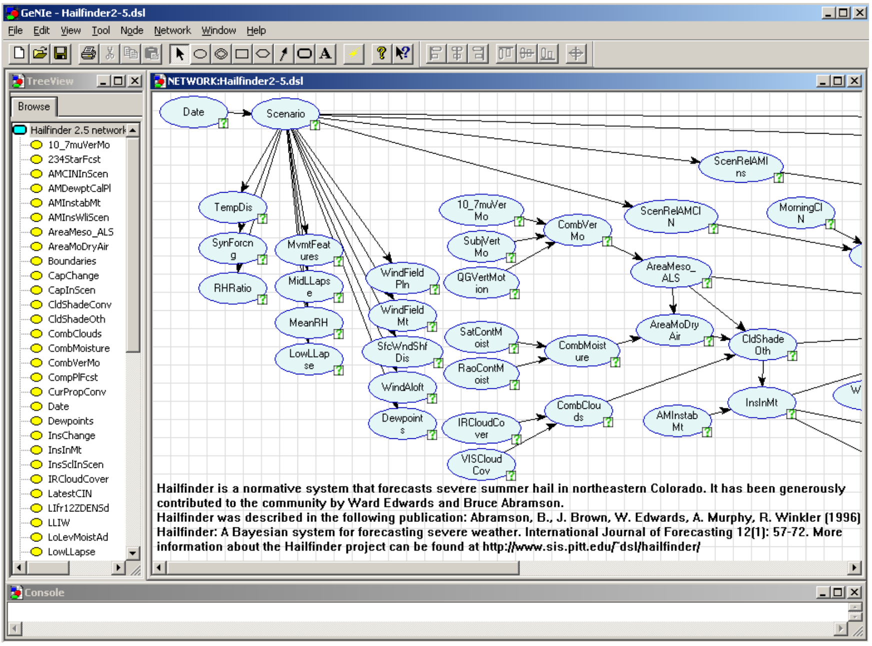

User interface: GeNIe (Graphical Network Interface).
Implemented in Visual C++ in Windows 95/NT environment.
Reasoning engine: SMILE (Structural Modeling, Inference, and Learning Engine).
A library of C++ classes, platform independent, well defined programmer's interface.
GeNIe
SMILE

Fig. 6.1: The architecture of GeNIe and SMILE

SMILE is a fully platform independent library of C++ classes implementing graphical probabilistic and decision-theoretic models, such as Bayesian networks, influence diagrams, and structural equation models. Its individual classes, defined in SMILE Applications Programmer Interface, allow to create, edit, save, and load graphical models, and use them for probabilistic reasoning and decision making under uncertainty. These classes are accessible from C++ or (as functions) from C programming languages. As most implementations of programming languages define a C interface, this make SMILE accessible from practically any language on any system. Also SMILE may be embedded in programs that use graphical probabilistic models as their reasoning engines. Furthermore, models developed in SMILE can be equipped with a user interface that suits the user of the resulting application most. Additional to the SMILE platform is the development of SmileX, an ActiveX Windows component that allows SMILE to be accessed from any Windows programming environment, including World Wide Web pages.

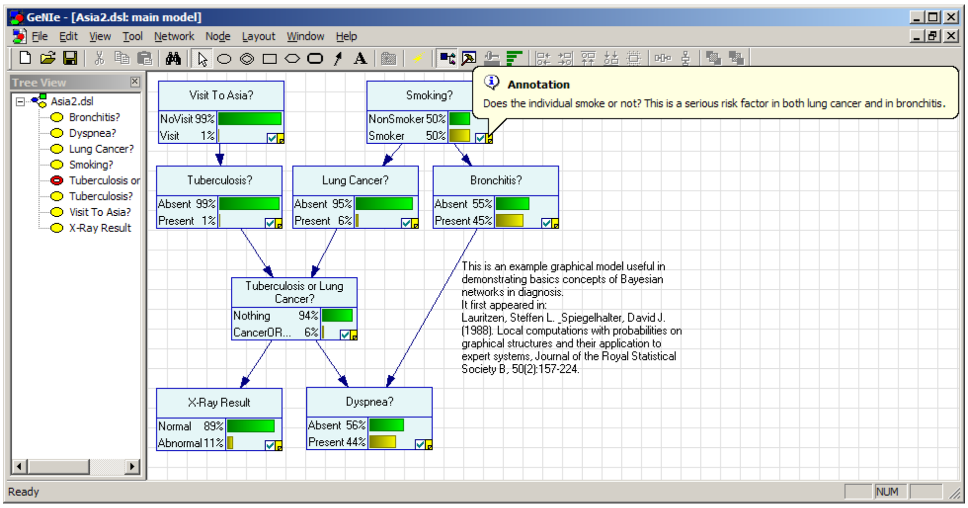


Hailfinder is a normative system that forecasts severe summer hail in northeastern Colorado. It has been generously contributed to the community by Ward Edwards and Bruce Abramson.
Hailfinder was described in the following publication: Abramson, B., J. Brown, W. Edwards, A. Murphy, R. Winkler (1996) Hailfinder: A Bayesian system for forecasting severe weather. International Journal of Forecasting 12(1): 57-72. More information about the Hailfinder project can be found at http://www.sis.pitt.edu/~dsl/hailfinder/

Fig. 6.2: A schematic view of GeNIe with the Hailfinder network

Some of the applications, built using GeNIe or SMILE, are: battle damage assessment (Rockwell International and U.S. Air Force Rome Laboratory), group decision support models for regional conflict detection (Decision Support Department, U.S. Naval War College) intelligent tutoring systems (Learning and Development Research Center, University of Pittsburgh), medical therapy planning (National University of Singapore), medical diagnosis (Medical Informatics Training Program, University of Pittsburgh; Technical University of Bialystok, Poland). GeNIe and SMILE have been also used in teaching statistics and decision-theoretic methods at several universities, even the Technical University of Delft.

Currently the Decision Science Laboratory is in its final stage of developing the second version of GeNIe, *GeNIe2*. This new version is characterized by its improved functionality to handle decision-theoretic models, e.g., Bayesian networks. A big improvement is found in the presentational and clarity aspect. An example of this clarity aspect is the ability to annotate any part of the network: variables, states and even arrows. The improved presentational aspect is found in the option of showing a bar chart of the probability distribution of a variable, see Figure 6.3. Within this bar chart the user is able to set a variable to a desired state and immediately notice the effect on the probabilities of the other variables.



Annotation

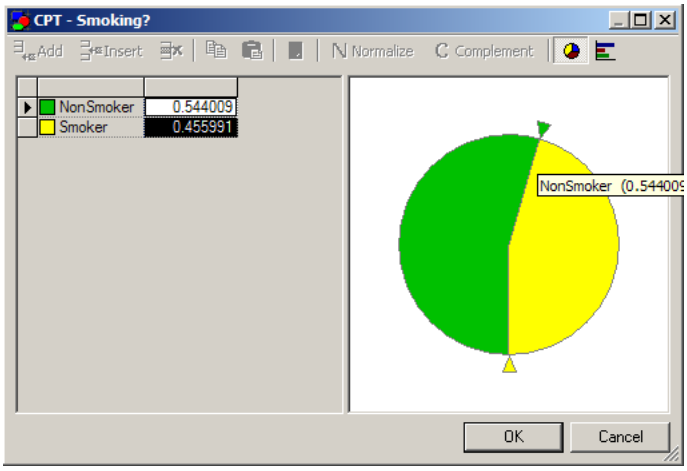
Does the individual smoke or not? This is a serious risk factor in both lung cancer and in bronchitis.

This is an example graphical model useful in demonstrating basics concepts of Bayesian networks in diagnosis.
It first appeared in:
Lauritzen, Steffen L. _Spiegelhalter, David J. (1988). Local computations with probabilities on graphical structures and their application to expert systems, Journal of the Royal Statistical Society B, 50(2):157-224.

Fig. 6.3: The GeNIe2 environment with the Asia network displayed in bar charts

A similar improvement is found in the assigning of the probability tables. As shown in Figure 6.4, the user may now assign a probability distribution with the help of a pie chart or bar chart.

The part in SMILE that contains the necessary functions and algorithms for the diagnostic application described in Section 3.5, is known as the diagnostic module of SMILE. This module acts as an “extra” layer over the SMILE library. The benefit of this design is that the module can make use of any function or class defined in the rest of SMILE. The three important classes of the diagnostic module are *DSL_extraDefinition*, *DSL_diagNetwork* and *DSL_fastEntropyAssessor*.



🟩 NonSmoker	0.544009
🟨 Smoker	0.455991

Fig. 6.4: The assigning of the probability distribution of the variable *Smoking* with help of a pie chart

DSL_extraDefinition
This class supplies the functions which define the necessary variables within a diagnosis session. So all the nodes in the available network are divided into three types: target, observation, and auxiliary. Each node may only be one of the three types. The target and observation nodes represent the hypothesis and test variables as defined in Section 3.2. From each of the target nodes a number of states are denoted as target states.

DSL_diagNetwork
The necessary functions to perform diagnosis are provided by this class. Whenever the user selects a target state to pursue, functions in this class are called to determine the ranking of each test and returns the results. The actual determination of the rankings is however not done in this class but is performed by functions from the following class.

DSL_fastEntropyAssessor
The actual process of value of information, see Figure 3.4, is arranged by the functions in this class. For the single cause diagnosis this class contains the function to determine the expected benefit and test strength of each available test. The value function which is used for the support of single cause diagnosis is given by the entropy value function from Section 3.4.

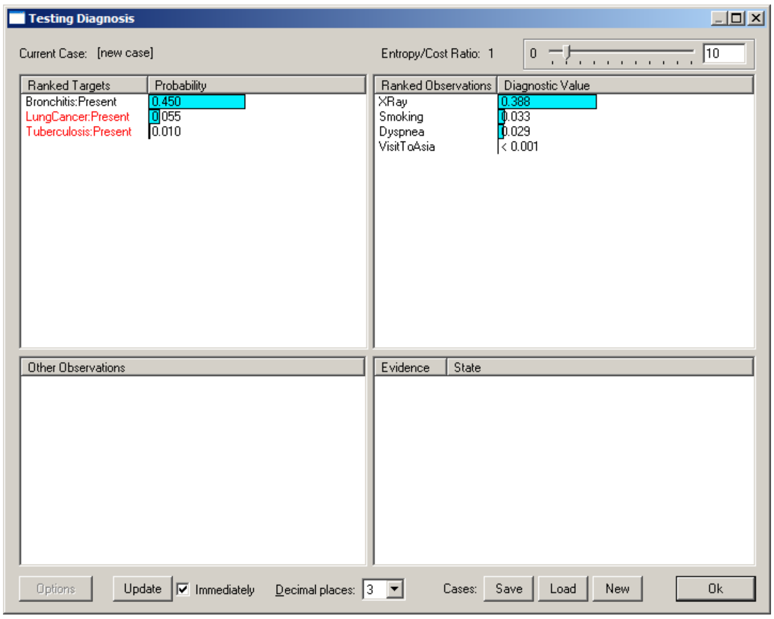
Together with the interactive interface described in Section 3.5, the diagnostic module already provides a strong support to perform diagnosis with Bayesian networks. However, to complete this support it is essential to implement the support of multiple causes.

6.2 Description of the Multiple Cause Application

The multiple cause application assumes the same preparation as the single cause application from Section 3.5. In other words, at least one hypothesis variable along with a target state and at least one test variable must be available.

As is shown in Figure 6.5, the interface of the diagnostic application is practically the same as the interface of the single diagnostic application. At the left of the screen the targets are displayed and on the right the available tests. The difference is that the user is able to select any number of targets and start the ranking of the available tests. How the tests are ranked depends on which approach is used, marginal or joint probability approximation. The result of the rankings is displayed with the list of tests.

In Figure 6.5 the two targets *LungCancer* and *Tuberculosis present* are pursued. The ranking of the tests is determined by the marginal probability approach in combination with the MS1 function. According to the test-list the test $X-RayResult$ is the best test to perform.



Current Case: [new case]

Entropy/Cost Ratio: 1

Ranked Targets	Probability
Bronchitis:Present	0.450
LungCancer:Present	0.055
Tuberculosis:Present	0.010

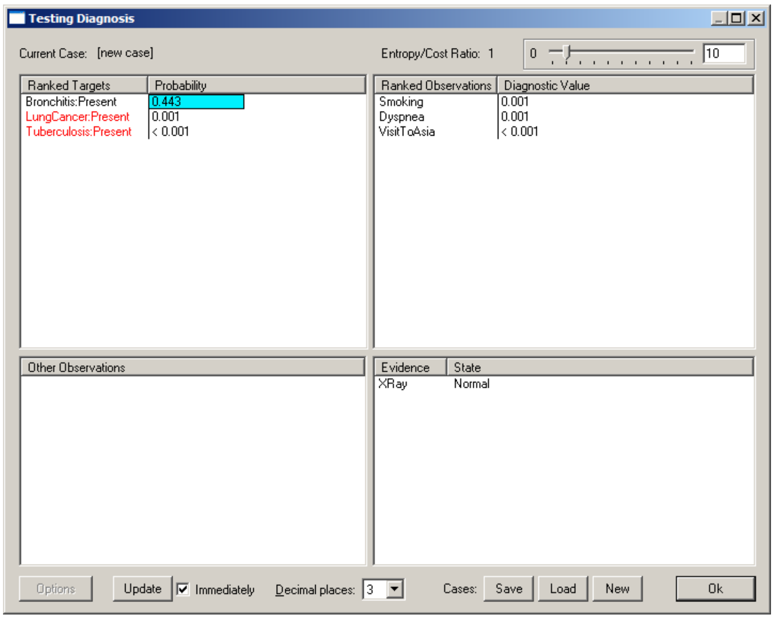
Ranked Observations	Diagnostic Value
XRay	0.388
Smoking	0.033
Dyspnea	0.029
VisitToAsia	< 0.001

Other Observations

Evidence	State

Fig. 6.5: The diagnostic screen with pursuing the two targets *LungCancer* and *Tuberculosis present* from the Asia DPN

The rankings in the test-list show which test is best in reducing the uncertainty between possible scenarios of the selected targets. The instantiation of a test will have an impact on the probabilities of the targets and also the probability distributions of the test variables. Therefore, the values in the diagnosis screen will adapt and the multiple cause application will recalculate the ranking of the remaining tests for the selection of targets. Note that after instantiation of a test the selection of targets remains the same. In Figure 6.6 the effect of instantiating the test $X - RayResult$ with the state $Normal$ is shown. The effect is that both $LungCancer$ and $Tuberculosis$ $present$ get a low probability. Furthermore, no test provide itself with a high ranking to become even more certain about the targets. A logical step would be to stop investigating the target states $LungCancer$ and $Tuberculosis$ $present$ but instead investigate the presence of the target $Dyspnea$.



Current Case: [new case]

Entropy/Cost Ratio: 1 10

Ranked Targets	Probability
Bronchitis:Present	0.443
LungCancer:Present	0.001
Tuberculosis:Present	< 0.001

Ranked Observations	Diagnostic Value
Smoking	0.001
Dyspnea	0.001
VisitToAsia	< 0.001

Other Observations

Evidence	State
XRay	Normal

☑ Immediately

Decimal places: 3

Fig. 6.6: The diagnostic screen with the instantiation of the test $X - RayResult$ to the state $Normal$.

The assigning of the ranking for each test may be done by applying either the marginal probability approach or the joint probability approach. How these are implemented and the available options are described in the following sections. Which approach and which underlying function is used, may be set in the code.

6.3 Implementation of the Available Approaches

The implementation of the support for the multiple cause application follows the value of information procedure displayed in Figure 3.4. Within this procedure the two approaches, marginal and joint probability approach, are used whenever the value of the test selection measure is determined. As shown in Figure 3.4, this is when $V\left(\Pr\left(\mathcal{H}\right)\right)$ and for every test outcome $V\left(\Pr\left(\mathcal{H}|\, t_j\right)\right)$. The rest of the test selection procedure uses the result of the approaches to determine a ranking for each test. Additional to the two approaches I also implemented the calculational of the full true joint probability in combination with the differential entropy function.

Each of the approaches is implemented in the class *DSL_fastEntropyAssessor*. Since the class *DSL_diagNetwork* asks the functions from *DSL_fastEntropyAssessor* class, I also adapted this class so it selects the right approaches. In the class *DSL_extraDefinition* little has changed since it already provides all the support necessary for multiple cause diagnosis. Finally, some changes were made to the single cause interface from Section 3.5. The changes adjusted the interface so it supports the ability to select and pursue multiple causes. Below, I provide the details concerning the implementation of the different approaches.

The marginal probability approach

This approach was the easiest to implement, since it directly uses the marginal probabilities of the set of selected targets and not the entire joint probability distribution. With the variable *marginalFunction* a choice may be made which of the two marginal based functions, MS1 and MS2 is applied. The general procedure of the marginal probability approach is shown in Figure 6.7.

```
Procedure: MarginalValue($\mathcal{F}$, marginalFunction)

Input: $\mathcal{F}$ = set of selected target states, marginalFunction= choice of marginal function
Output: value $V(\Pr(\mathcal{F}))$

collect the set of selected targets $\mathcal{F}$
set variable result to zero
for each $f_i \in \mathcal{F}$
    if marginalFunction=MS1
        determine $MS1(\Pr(f_i))$
    else marginalFunction=MS2
        determine $MS2(\Pr(f_i))$
    end if
    add determined value to result
end for each
return variable result
```

Fig. 6.7: Marginal based approach procedure for the calculation of the test selection measure

With the second function, MS2, I make use of an IF statement since this function is a combination of two functions. See Figure 6.8 for details about the implementation of the MS2 function. The implementation of the function MS1 is quite similar to the implementation MS2 function but without the IF statement and with the pursuedFaultCount variable set to the value 2.

```
double DSL_fastEntropyAssessor::CalculateMarginalStrength2 (const DSL_intArray pursued-
Faults, DSL_network &thisNet)
{
This value function assigns probabilities close to zero and one a value close to zero and has its
minimums = -1 at the value 0.5
    int pursuedFaultCount=pursuedFaults.NumItems();
    double marginalStrength=0.0;
    double purFaultCountInv = 1/pursuedFaultCount;
    double purFaultCountInvMinOne = 1-purFaultCountInv;
    for (int a=0; a<pursuedFaultCount; a++)
    {
        Determine the node and the state of the pursuedFault
        int afault = pursuedFaults[a];
        const DIAG_faultyState &fs = theNetwork→GetFaults()[afault];
        int theFaultNode = fs.node;
        int theFaultOutcome = fs.state;
        Get the marginal of this pursuedfault
        DSL_Dmatrix theFaultPriors;
        thisNet.GetNode(theFaultNode)→Value()→GetValue(&theFaultPriors);
        double faultMarginal = (*theFaultPriors)[theFaultOutcome];
        Calculation of the measure
        if (faultMarginal<=purFaultCountInvMinOne)
            marginalStrength+=pow((faultMarginal-purFaultCountInvMinOne),2);
        else
            marginalStrength+=pow(((faultMarginal-purFaultCountInvMinOne) ×
(pursuedFaultCount-1)),2);
    }
    Normalizing the Marginal Strength to (0,-1) area
    marginalStrength/=(pow(purFaultCountInvMinOne,2)*pursuedFaultCount);
    marginalStrength-=1;
    return(marginalStrength);
}
```

Fig. 6.8: Implementation of the MS2 value function

The joint probability approach

This approach was somewhat more difficult to implement, since it first required all the possible scenarios with the pursued target set. The function then determines if the scenarios belong to the selected partition-distribution. If yes the differential value function is calculated over it. For the implementation I choose the differential entropy function, see Section 5.4. An overview of the process of using the joint probability approach is shown in Figure, 6.9.

```
Procedure: $JointValue(\mathcal{F}, \Theta)$

Input: $\mathcal{F}$ = set of selected target states, $\Theta$ = choice of differential diagnosis distribution
Output: value $V(\Pr(\mathcal{F}))$

collect the set of selected targets $\mathcal{F}$
collect the possible scenarios of the targets $D_{\mathcal{F}} = \{s_1, s_2, \ldots, s_n\}$
set variable $result$ and $sum\_probability$ to zero
for each $s_i \in D_{\mathcal{F}}$
    if $s_i$ belongs to the $\Theta$ partition-distribution
        determine probability of this scenario $\Pr(s_i)$ with product copula function
        determine Diff_$ENT(\Pr(s_i))$
    end if
    add Diff_$ENT(\Pr(s_i))$ to $result$
    add $\Pr(s_i)$ to $sum\_probability$
end for each
determine Diff_$ENT(1 - sum\_probability)$
add Diff_$ENT(\Pr(s_i))$ to $result$
return $result$
```

Fig. 6.9: Procedure of the joint probability based approach

The variable Θ defines which partition-distribution is taken over the set of possible target scenarios. According to Section 5.2 there are three possible states of Θ, at least one target state, only one target state, or all target states in the scenario. The check whether a scenario is in the partition-distribution is done by the function *CheckDiffDiag*. When this function indicates the scenario as part of the partition-distribution, then the differential entropy function is calculated over this scenario, see Figure 6.10.

```
double DSL_fastEntropyAssessor::CalculateJointEntropy_Independence(const DSL_intArray
& pursuedFaultsNodes, const DSL_intArray & pursuedFaultsNumStates, const int_intVectors
& pursuedFaultsMatrix, DSL_network &theNet) {
double sumScenarioProb=0.0, jointEntropy=0.0;
int numberFaultsNodes = pursuedFaultsNodes.NumItems();
DSL_intArray coordinates;
int resultNext=DSL_OKAY;
while (resultNext==DSL_OKAY)
{
    int check=CheckDiffDiag(pursuedFaultsMatrix, pursuedFaultsNumStates, coordinates);
    if (check==DSL_TRUE)
    {
        double scenarioProb=1.0;
        for (int b=0; b<numberFaultsNodes;b++)
        {
            int theFaultNode = pursuedFaultsNodes[b];
            int theFaultState = coordinates[b];
            DSL_Dmatrix *theFaultProbs;
            theNet.GetNode(theFaultNode)→Value()→GetValue(&theFaultProbs);
            double faultMarginal = (*theFaultProbs)[theFaultState];
            scenarioProb*=faultMarginal;
        }
        if (scenarioProb== 0.0 || scenarioProb== 1.0)
            jointEntropy += 0.0;
        else
            jointEntropy += -scenarioProb * Log2(scenarioProb);
        sumScenarioProb+=scenarioProb;
    }
    resultNext=NextScenarioCoordinates(coordinates, pursuedFaultsNumStates);
}
double restScenariosProb=1-sumScenarioProb;
if (restScenariosProb== 0.0 || restScenariosProb== 1.0)
    jointEntropy += 0.0;
else
    jointEntropy += -restScenariosProb * Log2(restScenariosProb);
return(jointEntropy);
}
```

Fig. 6.10: Implementation of the calculating of the differential entropy value function

The true joint probability

Additional to the marginal and joint probability approach, I also implemented the determination of the true joint probability distribution. Thanks to this implementation I am able to compare the use of the approximation approaches with the use of the theoretical approach. For the calculations of this distribution the chain rule of Theorem 2.1 is applied. Since I want to provide support for pursuing and differentiating between any set of scenarios, I used the differential entropy function to determine the test rankings.

The procedure for using the true joint probability is actually similar to the procedure of the joint probability approach. However, a large difference between the procedures is that the entire joint probability is calculated before the differential entropy function is applied. This difference makes this approach immediately far more expensive to calculate. The reason for this lies in the fact that according to the chain rule, the joint probability distribution is calculated by multiplication of conditional and marginal probabilities. This implies the instantiation of states of multiple variables. The consequence of instantiating a state is that all the pursued target variables have to be updated. For this reason I focused on reducing the number of instantiations to a minimum and create the entire joint probability distribution at once. A consequence is that for every state in every test variable this entire joint probability distribution is recalculated. It is obvious that this approach only works within reasonable small networks and only with a limited number of target nodes. Since the joint probability keeps the same structure concerning the combination of states I only once determine which scenarios are interesting according to the differential diagnosis distribution. The entire process of using the true joint probability in combination with the differential entropy function is displayed in Figure 6.11.

```
Procedure: TrueJointValue($\mathcal{F}$, $\Theta$)

Input: $\mathcal{F}$ = set of selected target states, $\Theta$ = choice of differential diagnosis distribution
Output: value $V(\Pr(\mathcal{F}))$

collect the set of selected targets $\mathcal{F}$
collect the target variables
determine entire joint probability distribution over all possible scenarios $D_{\mathcal{F}} = \{s_1, s_2, \ldots, s_n\}$ with the target variables
determine which scenarios are in the $\Theta$ partition-distribution and store these in scenario_locations
set variable result and sum_probability to zero
for each $s_i \in$ scenario_locations
    determine Diff_ENT $(\Pr(s_i))$
    add Diff_ENT $(\Pr(s_i))$ to result
    add $\Pr(s_i)$ to sum_probability
end for each
determine Diff_ENT $(1 - sum\_probability)$
add Diff_ENT $(\Pr(s_i))$ to result
return result
```

Fig. 6.11: Procedure of the true joint probability approach

7. TESTS & RESULTS

This chapter describes the two test procedures which I used to determine the quality of the support of the marginal and joint probability approach. The first procedure tests in how many cases the diagnosis sequence results in a correct diagnosis. The second procedure determines the times necessary to perform diagnosis with a number of targets. In order to show the necessity of the approximation approaches I performed the second test procedure also with the true joint probability. The test procedures have been primarily tested on the Hepar II system, a practical network for the diagnosis of multiple liver disorders.

7.1 Description of the Hepar II System

The Hepar II system is a continuation of the Hepar project, conducted in the Institute of Biocybernetics and Biomedical Engineering of the Polish Academy of Sciences in collaboration with physicians at the Medical Center of Postgraduate Education in Warsaw. The Hepar system was designed for gathering and processing the clinical data on patients with liver disorders and aimed at reducing the need for hepatic biopsy by modern computer-based diagnostic tools. An integral part of the Hepar system is its database, created in 1990 and thoroughly maintained since then at the Gastroentorogical Clinic of the Institute of Food and Feeding in Warsaw. The current database contains over 800 patient records and its size is steadily growing. Each hepatological case is described by over 200 different medical findings, such as patient self-reported data, results of physical examinations, laboratory tests, and finally a histopathologically verified diagnosis.

The structure of the Hepar II system, see Figure 7.1, is divided into three colors, the red nodes represent the hypothesis variables and the blue and green nodes the test variables. The reason for the use of two colors with the test nodes is that blue nodes are mainly patient self-reported data and risk factors while green nodes indicate symptoms, the results of physical examinations, and laboratory test. Most of the nine hypothesis variables consist of two states, present and absent of the liver disease. However, the variables $Chronic$, $Hepatitis$, and $Cirrhosis$ each contain three states where two states represent the intensity of the liver disease and one state represents the absence of the disease. The total number of possible scenarios with this hypothesis set is then 1152 scenarios.

Before I may use this network for the multiple cause diagnosis application, I have to indicate which of the hypothesis states are considered as targets. Since the network is designed to diagnose whether a patient has one or a combination of liver diseases I distinguish the presence of a liver disease as a target. This results in the following list of targets, see Table 7.1.

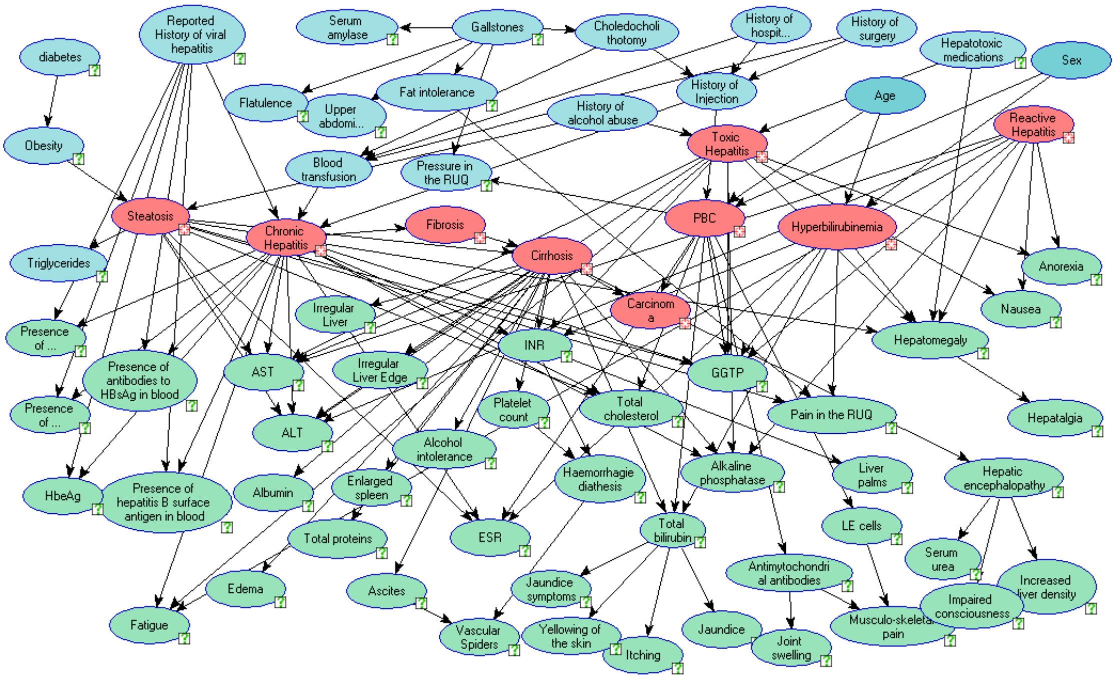


Fig. 7.1: The Hepar II network translated to a diagnostic probability network, with the hypothesis variables in red and the test variables in blue and green

Tab. 7.1: The targets of the Hepar II network

Hepatic steatosis	*present*
Chronic hepatitis	*active* *persistent*
Hepatic fibrosis	*present*
Cirrhosis	*decompensate* *compensate*
Carcinoma	*present*
PBC	*present*
Toxic hepatits	*present*
Functional hyperbilirubinemia	*present*
Reactive hepatitis	*present*

7.2 Quality and Time Procedures

The goal of testing the multiple cause diagnosis module from Section 6.2, is to determine if the application performs well in real networks and results in valuable diagnosis. In order to reach this goal I apply two test procedures on the Hepar II network. The test procedures are performed on the marginal test approach with its two marginal functions and the joint probability approach with the product copula function, the differential entropy function and the partition-distribution of at least one target. The reason to restrict the differential diagnosis to only one partition-distribution is that a different distribution is only useful and interesting if the user has an idea which combination of the causes is most likely. Unfortunately, it is hard to implement this users intuition into a computer test program.

Quality Test Procedure

The first procedure tests the quality of the rankings in the test-list, determined by the different approaches available in the application. The quality is tested by comparing the quality of a sequence of performing the best test with the quality of performing all the tests. The sequence of performing the best test is stopped when according to the test-ranking there is no interesting test left.

The quality of performing a test or sequence of tests may be measured by comparing the sensitivity and specificity of a test. Before I explain these terms I start by introducing the true and false positiveness and negatives of a diagnostic process. In most cases there holds that after performing a test the uncertainty about a diagnosis is not completely taken away. There are still some situations imaginable where the information collected by a sequence of tests, does not completely distinguish between the presence and absence of a cause. By counting in how many cases a sequence of tests was correct or not, delivers the 4 frequencies of Table 7.2. The frequencies TP, FP, FN and TN represent the observed frequencies of in how many cases the sequence of tests concluded if the cause was present or absent and if this diagnosis was correct. In short, these abbreviations stand for:

- TP: The frequency of how often the sequence of tests was correct indicating the presence of a cause;
- FP: The frequency of how often the sequence of tests was not correct indicating the presence of a cause;
- FN: The frequency of how often the sequence of tests was not correct indicating the absence of a cause;
- TN: The frequency of how often the sequence of tests was correct indicating the absence of a cause.

Tab. 7.2: The four aspects of diagnostic evaluation

		True disease state	
		D+	D-
Sequence of Tests	T+	TP	FP
	T-	FN	TN

Given these frequencies, the accuracy of the diagnosis may be determined. This is done by applying the concepts sensitivity and specificity. The sensitivity of a test is defined as the likelihood that a diseased patient has a positive test,

$$Se = \frac{TP}{TP + FN}.$$

If all patients with a disease have a positive test, i.e., not diseased patients have negative tests, then the test sensitivity is 1. A test with high sensitivity is useful to exclude a diagnosis because a highly sensitivity test will render few results that are falsely negative.

The specificity of a test is the likelihood that a healthy patient has a negative test,

$$Sp = \frac{TN}{TN + FP}.$$

If all patients with no disease have negative tests, i.e., not healthy patients have positive tests, then the test specificity is 1. A test with high specificity is useful to confirm a diagnosis, because a highly specific test will have few results that are falsely positive. The best possible test is the test with sensitivity and specificity of 1. This test is never wrong in diagnosing a disease.

The sensitivity and the specificity are then used in an ROC analysis. The Receiver Operating Characteristic (ROC) analysis was introduced in medical science in the late 1960s for the assessment of imaging devices. This analysis now belongs to the standard tools for the evaluation of clinical laboratory tests. The underlying assumption of the ROC analysis is that a diagnostic variable is to be used as a discriminator of two defined groups of responses, e.g., presence or absence of a cause. ROC analysis then assesses the diagnostic performance of the system in terms of Se and $(1 - Sp)$. This is done for each observed value of the discriminator variable (cut-off point to differentiate between the two groups of responses). The pairs Se and $(1 - Sp)$ for each of these cut-off points are then displayed as a ROC curve. The connection of the points leads to a staircase trace that originates from the upper right corner and ends at the lower left corner. The higher the curve is to the top left corner, the higher are the values of the sensitivity and the specificity or the better the quality of the diagnosis process.

The generation of the data necessary for the ROC curve is done by a test program created by Pryztula and Dash. Basically this program generates records of entire diagnosis sequences and determines in how many cases the diagnosis was correct and not correct. These records are then used for the creation of ROC curves. Unfortunately, the program is yet to be described in an upcoming paper [Pryztula and Dash], so I am not allowed to disclose any details about this program.

Time Test Procedure

The second procedure tests the time necessary for creating the rankings in the test-list. This test procedure is quite simple, but essential, since I expect it to show that the better the quality of the application, the more calculational effort is performed. The timer in the program will start whenever the approach function is called in the class *DSL_fastEntropyAssessor*. This timer will then stop and calculate the difference when the function is done calculating all the rankings. The time to get the different test-rankings on the screen is not measured since this is the same for every approach and associated function.

7.3 Test Results

The quality test procedure performed by the test program of Pryztula and Dash on the Hepar II network, generated a set of 200 test sequences. Each of these 200 records contained a measurement concerning the quality of the test sequence over all the available targets, see Table 7.1. From this measurement data the specificity and the sensitivity of each record was calculated and translated into a ROC curve.

As is shown in Figure 7.2 this procedure has been performed for the two marginal strength functions, $MS1$ and $MS2$ of the marginal probability approach. But also for the joint probability approach with its product copula and the differential entropy function, *Product Copula*. The top curve in the figure represents the diagnosis of performing all tests, *All tests*. This curve is immediately a measurement of how good the network actually is in diagnosing the targets.

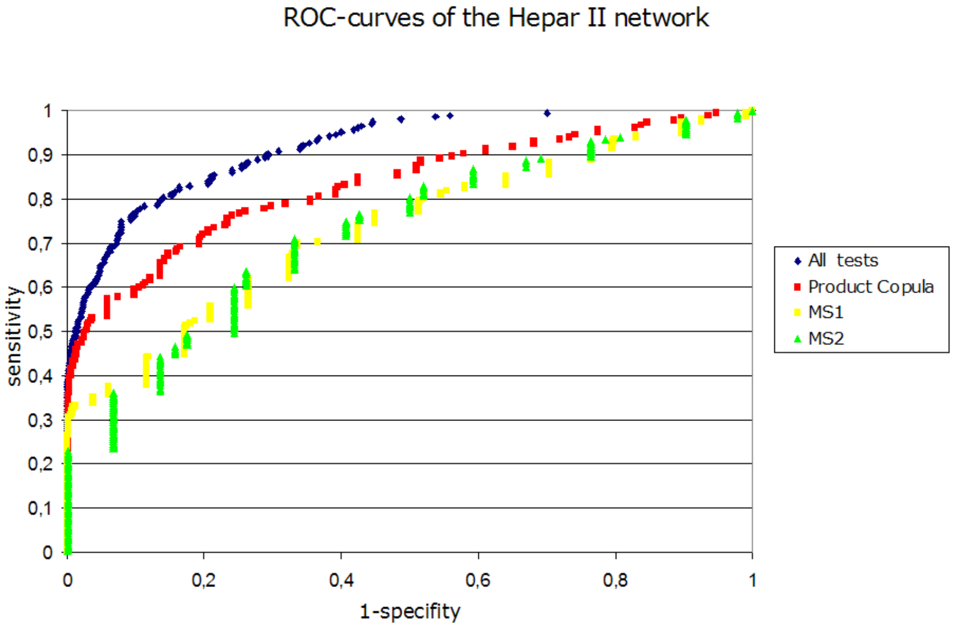


Fig. 7.2: The ROC-curves of the Hepar II network

Since the quality of a ROC curve is defined by how close it is to the top left corner, it confirms that performing all available tests results in the qualitatively best diagnosis. By comparing the other ROC curves to this curve a statement may be made about the quality of the approximation approach. Since the joint probability approach is the closest to the all tests curve this approach provides relative comparable quality. On the other hand the marginal probability approach with the two functions $MS1$ and $MS2$ is even further away from the *All tests* curve. This indicates that this approximation approach has a large effect on the quality of performing diagnosis. A peculiar thing about the ROC-curves of $MS1$ and $MS2$ is that they are almost equal to each other. Apparently the negative effect of the growing distance between the bounds is not that serious to account for.

A reason for the equality of the $MS1$ and $MS2$ ROC-curves might be the small number of targets. To investigate this, I also performed the test procedure with the marginal probability approach on a network much larger than the Hepar II network. This network has a total of 168 nodes where 47 nodes are hypothesis variables, 117 nodes are test variables and 4 nodes are auxiliary variables. From the hypothesis variables, only one state is selected as a target state, thus in total there are 47 targets to pursue. Since this network belongs to a company and is used in a professional environment, I did not receive permission to describe this network in my report. Therefore, I refer to the network as the Pitt network and will only discuss the test results of this network.

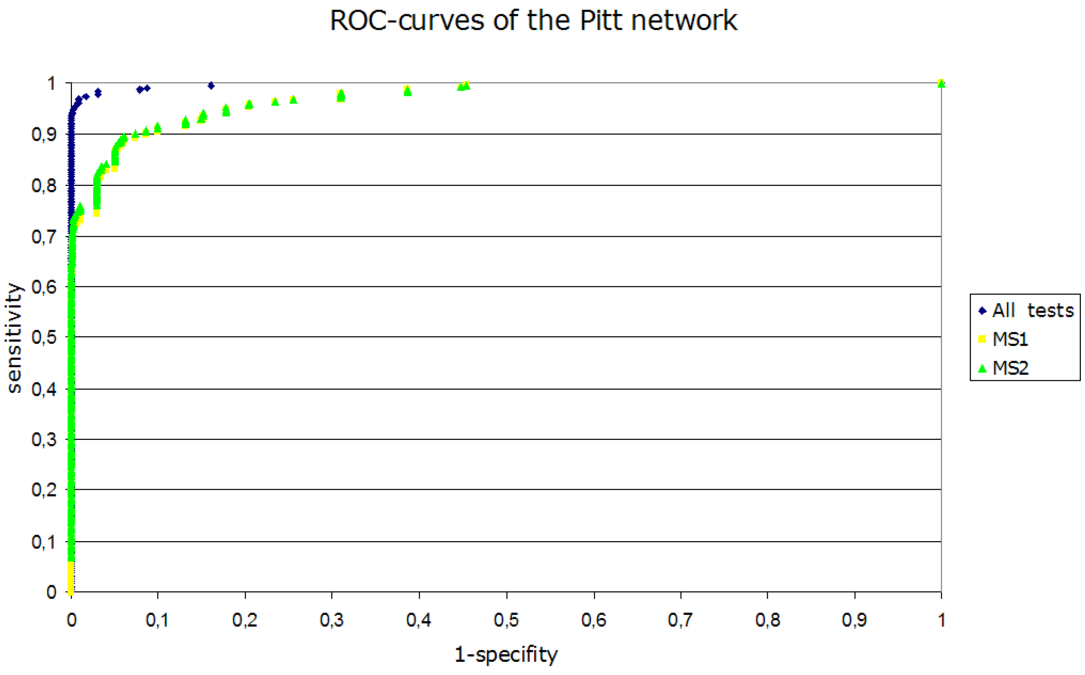


Fig. 7.3: The ROC-curves of the Pitt network

In Figure 7.3 the ROC curves of the Pitt network are displayed. By comparing these curves with the curves of the Hepar II network, it is easy to see that this network is better in performing diagnosis. The most important indication for this is that the *All tests* curve within the Pitt network is much better than the *All tests* curve of the Hepar II network. However, also with this network both the $MS1$ and $MS2$ curves have almost identical values. Apparently the increase in number of targets has little effect on which marginal based function is qualitatively better. On the other hand it is very interesting to notice that the quality of the $MS1$ and $MS2$ curves lie so much closer to the curve of *All tests*. Maybe there holds that the better the quality of the network to perform diagnosis with, the better the quality of the marginal probability approach. Unfortunately, I could not collect more networks to further test this hypothesis.

Although it is clear which approach results in qualitative better diagnosis, it is wrong to say that this approach is automatically the best approach. Therefore, the second test procedure was designed and performed. This procedure resulted in the time measurements of the Tables 7.3, 7.4, 7.5, and 7.6. In short these tables present the necessary times for the calculation of the test rankings for different numbers of pursued targets. To account for the difficulty of calculating the test rankings in complex networks the measurements have been performed with three different networks, Asia_diag, Hepar II, and Pitt.

Tab. 7.3: The calculational times in seconds for using the Marginal Strength1 function

MS1	2 targets	3 targets	5 targets	10 targets	15 targets	20 targets	25 targets
Asia_diag	0	0	***	***	***	***	***
Hepar II	1	1	1	1	***	***	***
Pitt	1	1	1	1	1	1	1

Tab. 7.4: The calculational times in seconds for using the Marginal Strength2 function

MS2	2 targets	3 targets	5 targets	10 targets	15 targets	20 targets	25 targets
Asia_diag	0	0	***	***	***	***	***
Hepar II	1	1	1	1	***	***	***
Pitt	1	1	1	1	1	1	1

The first two tables use the functions of the marginal probability approach. As expected these functions take almost no time to process the rankings in the test-list even in the large Pitt network. This is different in Table 7.5 where the calculational times for the joint probability approach are displayed. This table continues the small calculational times for the networks, Asia_diag and Hepar II, but not for the Pitt network. Up till the pursuing of 15 targets the support is still manageable with times smaller than 1 minute but for more targets these times increase rapidly. Since the Pitt network contains 47 targets a consequence is that the pursuing of all targets will certainly take several hours.

Tab. 7.5: The calculational times in seconds for using the product copula function

ProdCop	2 targets	3 targets	5 targets	10 targets	15 targets	20 targets	25 targets
Asia_diag	0	0	***	***	***	***	***
Hepar II	1	1	1	1	***	***	***
Pitt	1	1	1	2	43	1636	>60 min

The reason for using approximation approaches becomes clear by observing Table 7.6 which contains the calculational times of using the true joint probability distribution. Although the use of this distribution has no effect on the calculational times with the Asia_diag network it has on the other hand already a large effect on the calculation times with the Hepar II network. In this network the pursuit of 5 targets is still within a minute but 10 targets already takes 514 seconds or 8 minutes and 34 seconds. This effect is even worse in the Pitt network where the pursuit of 10 targets takes 2303 seconds or 38 minutes and 23 seconds. Pursuing more than 10 targets implies calculational times of longer than an hour.

Tab. 7.6: The calculational times in seconds for using the the true joint probability

True Joint	2 targets	3 targets	5 targets	10 targets	15 targets	20 targets	25 targets
Asia_diag	0	0	***	***	***	***	***
Hepar II	5	11	42	514	***	***	***
Pitt	7	14	61	2303	>60 min	>60 min	>60 min

8. CONCLUSIONS AND FUTURE RESEARCH

8.1 Conclusions

The main objective of this thesis was to improve the functionality of Bayesian networks by providing approximations for the support of multiple causes. At the end of this thesis I may conclude that this objective is reached and two approximations approaches are available to provide support for diagnosis with small and large Bayesian networks. Furthermore, both approaches provide the user with the ability to pursue and differentiate between causes.

The study of Bayesian networks at the beginning of this thesis made clear what a powerful tool these networks provide for modelling uncertainty relations. Since the objective of performing diagnosis is to reduce the uncertainty in the system, these networks are a logical choice to model diagnostic systems. In order to model these systems I introduced a structure that distinguishes the necessary variables and supports the essential tasks of diagnosis. Unfortunately this support is in practice limited to diagnosing only one cause since a combination of causes delivers both presentational as well as computational problems. Because these problems are not directly solvable I investigated the use of approximations. This investigation resulted into the development of two approximation approaches.

The marginal probability approach uses the relation between the marginal and joint probability to justify the use of marginal based test selection measures. That this approach makes use of a large approximation is noticeable in the loss of quality of performing diagnosis. Not even improving the marginal based test selection measures by taking into account the increasing uncertainty of more variables could improve this quality. However, the major benefit of this approach lies in its speed to perform diagnosis on any network. Independent of the size or complexity of the network, this approach always delivers fast results.

The joint probability approach is a less radical approximation since it tries to approximate the necessary joint probability distribution by use of a copula function. Because the joint probability distribution is accompanied with an enormous amount of scenarios, I developed the area of differential diagnosis so the user has the ability to control the process of diagnosis. This approach results in a qualitatively good support for performing diagnosis. Unfortunately, this support is also not ideal since its use causes large computational times whenever large amount of causes are pursued.

Although both approximation approaches are not optimal in providing support for multiple cause diagnosis, they are however necessary to apply. The reason for this is that the traditional approach of using the true full joint probability with a test selection measure results in presentational as well as computational problems, see Table 7.6. Therefore, I recommend the use of the approximation approaches but let the choice of which approach to apply be dependent on the network and the number of causes to pursue. Whenever a large complex network is used, the marginal probability approach is most suitable because of its speed. Within a small network or the pursuit of small amount of causes, the joint probability approach is better to use because of its better quality to perform diagnosis.

8.2 Future Research

For future research I recommend that more research is performed in using the joint probability approach. Since the area of copula is quite a large research area with various applications, I believe that it holds other options and functions than using the product copula function to provide support. As was shortly introduced in Section 5.1, the Spearman's correlation coefficient may be used to determine the correlation within a network. With this extra information a better copula function may be applied which eventually may result in better diagnosis. Since this correlation coefficient only has to be determined once in the calculational process it should not have a large negative effect on the performance. With more copula functions available, a choice may be made in combination with performance. An important aspect to take into account is that the copula functions have to work with discrete variables.

Another approach which has not been discussed in this paper is the use of an efficient algorithm to determine the joint probability over an arbitrary set of variables. So far only two papers [Xu, 1995, Duncan, 2001] have appeared that discuss this approach and provide an efficient algorithm. The reason that I did not pursue this approach was that these algorithms probably only work well for determining the joint probability over small sets of variables and not with large sets in complex networks. However the befit of this approach is that it allows the use of the traditional test selection measures in combination with the area of differential diagnosis.

When the support for multiple cause diagnosis is optimized it may be interesting to use this theory for the area of value of information. As noted in Section 3.3, the general assumption with VOI is that only one information source is consulted and not a combination of sources. It is obvious that this approach may result in incorrect advice about which information to collect. Since the investigation of a combination of sources implies the calculation of the joint probability over these sources, I expect the research for the support of multiple cause diagnosis may be useful. However, combining these different supports may also result in radical performance problems. For instance, the effect of each combination of tests has to be calculated on each possible scenario of a set of hypothesis variables. Consider, for example 10 hypothesis and test variables with each 2 states, the number of possible options of the effect of tests combinations on hypothesis combinations is then $2^{10} * 2^{10} = 1024 * 1024 = 1048576$.

APPENDIX

A. DEFINITIONS & THEOREMS

This appendix contains the definitions, theorems, and proofs referred to in this report.

Definition A.1 (convex). *A function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is convex if the domain of f, $\boldsymbol{dom}_f$ is a convex set and if for all $x, y \in \boldsymbol{dom}_f$, and θ with $0 \leq \theta \leq 1$, I have*

$$f\left(\theta x + (1-\theta) y\right) \leq \theta f(x) + (1-\theta) f(y)$$

Theorem A.1 (Jensen's inequality). *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a convex function. Let $\mathbf{x}_1, ..., \mathbf{x}_n \in \mathbb{R}^n$ and let $a_1, ..., a_n \in [0,1]$, such that $\sum_{i=1}^{n} a_i = 1$. Then*

$$f\left(\sum_{i=1}^{n} a_i \mathbf{x}_i\right) \leq \sum_{i=1}^{n} a_i f(\mathbf{x}_i)$$

Proof. *The proof is by induction on n. For $n = 2$ the inequality is exactly the convex definition:*

$$f(a_1 \mathbf{x}_1 + a_2 \mathbf{x}_2) \leq a_1 f(\mathbf{x}_1) + a_2 f(\mathbf{x}_2)$$
$$f(a_1 \mathbf{x}_1 + (1-a_1) \mathbf{x}_2) \leq a_1 f(\mathbf{x}_1) + (1-a_1) f(\mathbf{x}_2)$$

Now I assume that the inequality holds for $n-1$. Then let $a' = a_i/(1-a_n)$ and assume that $a_n \neq 1$.

$$\begin{aligned} f\left(\sum_{i=1}^{n} a_i \mathbf{x}_i\right) &= f\left(a_n \mathbf{x}_n + (1-a_n) \sum_{i=1}^{n-1} a_i' \mathbf{x}_i\right) \\ &\leq a_n f(\mathbf{x}_n) + (1-a_n) f\left(\sum_{i=1}^{n-1} a_i' \mathbf{x}_i\right) \\ &\leq a_n f(\mathbf{x}_n) + (1-a_n) f \sum_{i=1}^{n-1} a_i' f(\mathbf{x}_i) \\ &= \sum_{i=1}^{n} a_i f(\mathbf{x}_i) \end{aligned}$$

□

Theorem A.2. *Associated with the entropy based value function are the following properties.*

1. *When each scenario from a set of hypothesis variables $s \in D_{\mathcal{H}}$ has the same probability $\Pr(s) = \frac{1}{n}$, the $V_{ENT}(\Pr(\mathcal{H}))$ function will have its minimum.*

2. *The* $V_{ENT}\left(\Pr\left(\mathcal{H}\right)\right)$ *function is a monotonic decreasing function of the number of scenarios* n*, when each scenario* $s \in D_{\mathcal{H}}$ *has the same probabilities.*

3. *The composition law: if a set of hypothesis variables is broken down into two successive choices, the original* $V_{ENT}\left(\Pr\left(\mathcal{H}\right)\right)$ *should be the weighted sum of the individual values of* $V_{ENT}\left(\Pr\left(\mathcal{H}\right)\right)$*.*

4. *The entropy function is convex.*

Proof.
1, 2, and 3. See [Shannon, 1948].
4. Let H *be a hypothesis variable, and let* $\Pr(H)$ *and* $\Pr(H')$ *be two distributions over* H*. I shall prove that for each* $t \in [0,1]$*,*

$$tV\left(\Pr\left(H\right)\right) + \left(1-t\right)V\left(\Pr\left(H'\right)\right) \geq V\left(t\Pr\left(H\right) + \left(1-t\right)\Pr\left(H'\right)\right).$$

First note that the function $x \log x$ *is convex for* $x > 0$ *(the second derivative is positive). So for all* $x, y > 0$*,*

$$tx\log x + \left(1-t\right)y\log y \geq \left(tx + \left(1-t\right)y\right)\log\left(tx + \left(1-t\right)y\right).$$

Then

$$\begin{aligned} & t\left(-V_{ENT}\left(\Pr\left(H\right)\right)\right) + \left(1-t\right)\left(-V_{ENT}\left(\Pr\left(H'\right)\right)\right) \\ &= \sum_{h\in H}\left[t\Pr(h)\log\Pr(h) + (1-t)\Pr(h')\log\Pr(h')\right] \\ &\geq \sum_{h\in H}\left[(t\Pr(h) + (1-t)\Pr(h'))\log\left(t\Pr(h) + (1-t)\Pr(h')\right)\right] \\ &= -V_{ENT}\left(t\Pr\left(H\right) + \left(1-t\right)\Pr\left(H'\right)\right) \end{aligned}$$

□

Theorem A.3. *Associated with the weight of evidence function are the following properties.*

1. *When a scenario from a set of hypothesis variables* $s \in D_{\mathcal{H}}$ *and its negation have the same probability* $\Pr\left(s\right) = \Pr\left(\overline{s}\right) = \frac{1}{2}$*, the* WOE *function is zero.*

2. *The* WOE *function is convex, for a scenario from a set of hypothesis variables* $s \in D_{\mathcal{H}}$ *with probability* $\Pr\left(s\right) > 1/2$

Proof. *If I take the second derivative from the Weight of Evidence function,*

$$\begin{aligned} V_{WOE}(x) &= \log x - \log 1 - x \\ \frac{d^2V_{WOE}(x)}{dx^2} &= \frac{1}{\left(1-x\right)^2} - \frac{1}{x^2}, \end{aligned}$$

I see that this derivative is only positive if $x > \frac{1}{2}$*. Since a function is only convex if the second derivative is convex this implies that the function is only convex if the probability of a scenario is greater than* $\frac{1}{2}$ □

Theorem A.4 (maximum distance between bounds). *Let $\mathcal{X} = \{X_1, ..., X_n\}$ be a set of random variables and $s = (x_1, ..., x_n)$ a scenario where $s \in D_{\mathcal{X}}$. The distance between the upper and lower bound, from respectively Theorem 4.1 and 4.2, for the probability of the scenario s is at most $1 - \frac{1}{n}$.*

Proof. *In order to prove the maximum distance between the upper and lower bound, I distinguish two situations, $1 - n + \sum_{i=1}^{n} \Pr(x_i) \leq 0$ (the lower bound is zero) and $1 - n + \sum_{i=1}^{n} \Pr(x_i) \geq 0$.*

In the first situation, the distance is equal to the Upp_bound or the minimum of the marginal probabilities. Without loss of generality, I assume that the minimal marginal probability is $\Pr(x_k)$, so the distance is equal to this probability. Since $\Pr(x_k)$ is the minimal marginal probability, the summation of n times this probability, $\sum_{i=1}^{n} \Pr(x_k) = n \cdot \Pr(x_k)$ has to be smaller than the summation of all the marginal probabilities, $\sum_{i=1}^{n} \Pr(x_i)$. With the restriction that $1 - n + \sum_{i=1}^{n} \Pr(x_i) \leq 0$ or that $\sum_{i=1}^{n} \Pr(x_i) \leq n - 1$, it is clear that $n \cdot \Pr(x_k)$ is also smaller than $n - 1$. Then, $\Pr(x_k)$ is smaller than $1 - \frac{1}{n}$ and consequently, the distance is maximal $1 - \frac{1}{n}$.

For the second situation, I assume the lower bound is greater than 0, so $\sum_{i=1}^{n} \Pr(x_i) \geq n - 1$. The distance between upper and lower bound is then $\min_i \Pr(x_i) - 1 + n - \sum_{i=1}^{n} \Pr(x_i)$. With the assumption that $\Pr(x_k)$ is the minimal marginal probability, I rewrite the distance, $-1 + n - \sum_{i=1, i \neq k}^{n} \Pr(x_i)$. In order to prove that this distance is smaller than $1 - \frac{1}{n}$, I proof it for the two cases: $\Pr(x_k) \leq \frac{n-1}{n}$ and $\Pr(x_k) \geq \frac{n-1}{n}$. Starting with the first case, I separate the sum in the restriction $\sum_{i=1}^{n} \Pr(x_i) \geq n - 1$ into two parts, $\sum_{i=1, i \neq k}^{n} \Pr(x_i) + \Pr(x_k) \geq n - 1$, so $\Pr(x_k) \geq n - 1 - \sum_{i=1, i \neq k}^{n} \Pr(x_i)$. Since $\Pr(x_k) \leq \frac{n-1}{n}$ the distance, $n - 1 - \sum_{i=1, i \neq k}^{n} \Pr(x_i)$, also has to be smaller than $1 - \frac{1}{n}$. For the second case I use that $\sum_{i=1, i \neq k}^{n} \Pr(x_i) \geq (n-1) \cdot \Pr(x_k)$, since $\forall_i \Pr(x_i) \geq \Pr(x_k)$. Combined with $\Pr(x_k) \geq \frac{n-1}{n}$, I get $\sum_{i=1, i \neq k}^{n} \Pr(x_i) \geq (n-1) - \frac{n-1}{n}$ which is the same as $-1 + n - \sum_{i=1, i \neq k}^{n} \Pr(x_i) \leq \frac{n-1}{n}$. Consequently the distance is also in this case smaller than $1 - \frac{1}{n}$. Hence the distance is maximal $1 - \frac{1}{n}$ for all possible probabilities. □

Theorem A.5 (maximum distance condition). *The distance between the upper and lower bound of the probability of a scenario $s = (x_1, ..., x_n)$ in a set of random variables $\mathcal{X} = \{X_1, ..., X_n\}$ is maximal and equal to $1 - \frac{1}{n}$, if and only if every marginal probability of the states in the scenario s is equal to $1 - \frac{1}{n}$.*

Proof. *($\Rightarrow$) I assume here that the distance is maximal and equal to $1 - \frac{1}{n}$. In order to prove that every marginal probability is equal to $1 - \frac{1}{n}$ two situations are distinguished: $1 - n + \sum_{i=1}^{n} \Pr(x_i) \leq 0$ (the lower bound is zero) and $1 - n + \sum_{i=1}^{n} \Pr(x_i) \geq 0$.*

In the first situation, the distance is equal to the Upp_bound, so the minimum of the marginal probabilities is equal to $1 - \frac{1}{n}$. Without loss of generality, I assume that the minimal marginal probability is $\Pr(x_k)$, so $\Pr(x_k) = 1 - \frac{1}{n}$. Since $\Pr(x_k) \leq \forall_i \Pr(x_i)$, the summation of n-1 times this probability $\Pr(x_k)$, has to be smaller than the summation of the marginal probabilities without the minimal marginal probability, $(n-1) \cdot \Pr(x_k) \leq \sum_{i=1, i \neq k}^{n} \Pr(x_i)$. Furthermore, by separating the sum in the constraint $\sum_{i=1}^{n} \Pr(x_i) \geq n - 1$ into two parts, $\sum_{i=1, i \neq k}^{n} \Pr(x_i) + \Pr(x_k) \geq n - 1$, so $\Pr(x_k) \geq n - 1 - \sum_{i=1, i \neq k}^{n} \Pr(x_i)$ there holds that $\sum_{i=1, i \neq k}^{n} \Pr(x_i) \leq n - 1 - \Pr(x_k)$. Consequently there holds $(n-1) \cdot \Pr(x_k) \leq \sum_{i=1, i \neq k}^{n} \Pr(x_i) \leq n - 1 - \Pr(x_k)$ and after replacing $\Pr(x_k)$ with $1 - \frac{1}{n}$, I have $n - 1 - \frac{n-1}{n} \leq \sum_{i=1, i \neq k}^{n} \Pr(x_i) \leq n - 1 - \frac{n-1}{n}$. So, $\sum_{i=1, i \neq k}^{n} \Pr(x_i) = (n-1) \cdot \Pr(x_k)$, which is only possible if $\forall_i \Pr(x_i) = \Pr(x_k) = 1 - \frac{1}{n}$.

For the second situation, I assume the lower bound is greater than 0, so $\sum_{i=1}^{n} \Pr(x_i) \geq n - 1$. The distance is then equal to $\min_i \Pr(x_i) - 1 + n - \sum_{i=1}^{n} \Pr(x_i) = 1 - \frac{1}{n}$. With the assumption that $\Pr(x_k)$ is the minimal marginal probability, I rewrite the distance, $-1 + n - \sum_{i=1, i \neq k}^{n} \Pr(x_i) = 1 - \frac{1}{n}$. Before I prove $\forall_i \Pr(x_i) = 1 - \frac{1}{n}$, I show that $\Pr(x_k) = 1 - \frac{1}{n}$ by deriving $\Pr(x_k) \leq \frac{n-1}{n}$ and $\Pr(x_k) \geq \frac{n-1}{n}$. Since $\forall_i \Pr(x_i) \geq \Pr(x_k)$, there also holds that $(n-1) \cdot \Pr(x_k) \leq \sum_{i=1, i \neq k}^{n} \Pr(x_i)$. If I combine this with the provided distance $-1 + n - \sum_{i=1, i \neq k}^{n} \Pr(x_i) = 1 - \frac{1}{n}$, I get $(n-1) \cdot \Pr(x_k) \leq (n-1) - \frac{n-1}{n}$. Since $n \geq 2$ I derive that $\Pr(x_k) \leq 1 - \frac{1}{n}$. For the second derivation I rewrite the condition for the existence of the lower bound, $\sum_{i=1}^{n} \Pr(x_i) = \sum_{i=1, i \neq k}^{n} \Pr(x_i) + \Pr(x_k) \geq n - 1$. Combining this with the provided distance, it gives $n - 1 - \frac{n-1}{n} + \Pr(x_k) \geq n - 1$ and that $\Pr(x_k) \geq 1 - \frac{1}{n}$. Consequently there has to hold that $\Pr(x_k) = 1 - \frac{1}{n}$. When multiplying $\Pr(x_k)$ with $n - 1$, I get $(n-1) \cdot \Pr(x_k)$ which is the same as $\sum_{i=1, i \neq k}^{n} \Pr(x_i)$. This is only possible if $\forall_i \Pr(x_i) = \Pr(x_k) = 1 - \frac{1}{n}$.

($\Leftarrow$)I assume here that all the marginal probabilities are equal to $1 - \frac{1}{n}$. In order to prove that the distance is also maximal and equal to $1 - \frac{1}{n}$, I distinguish two situations: $1 - n + \sum_{i=1}^{n} \Pr(x_i) \leq 0$, (the lower bound is zero) and $1 - n + \sum_{i=1}^{n} \Pr(x_i) \geq 0$.

In the first situation, the distance is equal to the upper bound, which is the minimum of the marginal probabilities. Since all the marginal probabilities are equal to $1 - \frac{1}{n}$, the upper bound is equal to $1 - \frac{1}{n}$ and also the distance is equal to $1 - \frac{1}{n}$.

In the second situation, the distance is equal to the upper bound minus the lower bound. Since all the marginal probabilities are equal to $1 - \frac{1}{n}$, the upper bound is equal to $1 - \frac{1}{n}$ and the lower bound is equal to $n - 1 - n \cdot \left(1 - \frac{1}{n}\right) = 0$. Consequently the distance is equal to $1 - \frac{1}{n}$. □

BIBLIOGRAPHY

[Andreassen *et al.*, 1987] S. Andreassen, M. Woldbye, B. Falck, and S.K. Andersen. MUNIN – A causal probabilistic network for interpretation of electromyographic findings. In J. McDermott, editor, *Proceedings of the 10th International Joint Conference on Artificial Intelligence, IJCAI–87*, pages 366–372, Los Altos, CA, 1987. Morgan Kaufmann Publishers, Inc.

[Ben-Bassat *et al.*, 1980] M. Ben-Bassat, V.K. Carlson, V.K. Puri, M.D. Davenport, J.A. Schriver, M.M. Latif, R. Smith, E.H. Lipnick, and M.H. Weil. Pattern-based interactive diagnosis of multiple disorders: The MEDAS system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2:148–160, 1980.

[Ben-Bassat, 1978] Moshe Ben-Bassat. Myopic policies in sequential classification. *IEEE Transactions on Computers*, 27:170–178, 1978.

[Buchanan and Shortliffe, 1984] G. G. Buchanan and E. H. Shortliffe. *Rule-Based Expert Systems:The MYCIN Experiments of the Stanford Heuristic Programming Project*. Addison-Wesley, Reading, MA, 1984.

[Clemen and Reilly, 1999] Robert T. Clemen and Terence Reilly. Correlations and copulas for decision and risk analysis. *Management Science*, 45:208–224, 1999.

[Clemen, 1996] Robert T. Clemen. *Making Hard Decisions: An Introduction to Decision Analysis*. Duxbury Press, An Imprint of Wadsworth Publishing Company, Belmont, California, 1996.

[Cooper, 1990] Gregory F. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42(2–3):393–405, March 1990.

[Dall'Aglio *et al.*, 1991] G. Dall'Aglio, S. Kotz, and G. Salinetti. *Advances in Probability Distributions with Given Marginals*. Kluwer, Dordrecht, Netherlands, 1991.

[de Dombal *et al.*, 1972] F.T. de Dombal, D.J. Leaper, J.R. Staniland, A.P. McCann, and Jane C. Horrocks. Computer-aided diagnosis of acute abdominal pain. *British Medical Journal*, 2:9–13, April 1972.

[de Kleer and Williams, 1987] Johan de Kleer and Brian C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32(1):97–130, April 1987.

[Duncan, 2001] Smith Duncan. The efficient propagation of arbitrary subsets of beliefs in discrete-valued bayesian networks. *AI & Statistics 2001 Conference*, 2001.

[Fréchet, 1957] M. Fréchet. Les tableaux de corrélation dont les marges et des bornes sont données. *Annales de l'Université de Lyon, Sciences Mathématiques et Astronomie*, 20:13–31, 1957.

[Glasziou and Hilden, 1989] Paul Glasziou and Jørgen Hilden. Test selection measures. *Medical Decision Making*, 9:133–141, 1989.

[Good and Card, 1971] I. Good and W. Card. The diagnostic process with special reference to errors. *Method of Information Medicine*, 10(176–188), 1971.

[Good, 1985] I. Good. Weight of evidence: A brief survey. *Bayesian Statistics*, 2:249–270, 1985.

[Gorry and Barnett, 1968] Anthony G. Gorry and Octo G. Barnett. Experience with a model of sequential diagnosis. *Computer and Biomedical Research*, 1(5):490–507, May 1968.

[Heckerman *et al.*, 1992] David E. Heckerman, Eric J. Horvitz, and B. N. Nathwani. Toward normative expert systems: Part I the pathfinder project. *Methods of Information in Medicine*, 31:90–105, 1992.

[Heckermann *et al.*, 1995] David E. Heckermann, J. Breese, and K. Rommelse. Decision-theoretic troubleshooting. *Communications of the ACM*, 38:49–56, 1995.

[Horvitz *et al.*, 1988] Eric J. Horvitz, John S. Breese, and Max Henrion. Decision analysis in expert systems and artificial intelligence. *International Journal of Approximate Reasoning*, 2:247–302, 1988.

[Howard and Matheson, 1981] Ronald A. Howard and James E. Matheson. Influence diagrams. In Ronald A. Howard and James E. Matheson, editors, *The Principles and Applications of Decision Analysis*, pages 719–762. Strategic Decisions Group, Menlo Park, CA, 1981.

[Howard, 1966] Ronald A. Howard. Information value theory. *IEEE Transactions on Systems Science and Cybernetics*, SSC-2(1):22–26, August 1966.

[Huang and Darwiche, 1994] Cecil Huang and Adnan Darwiche. Inference in Belief networks: A procedural guide. *International Journal of Approximate Reasoning*, 11:1–158, 1994.

[Jaynes, 1957] E. T. Jaynes. Information theory and statistical mechanics. *Physical Review*, 106:620–630, 1957.

[Jensen *et al.*, 2001] Finn V. Jensen, Uffe Kjærulff, Brian Kristiansen, Helge Langseth, Claus Skaanning, Jirĺ Vomlel, and Marta Vomlelová. The SACSO methodology for troubleshooting complex systems. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM)*, 2001. To Appear in a Special Issue on AI in Equipment Service.

[Jensen, 1996] Finn V. Jensen. *An Introduction to Bayesian Networks*. Springer, New York, NY, 1996.

[Joe, 1997] Harry Joe. *Multivariate Models and Dependence Concepts.* Chapman & Hall, London, 1997.

[Lauritzen and Spiegelhalter, 1988] S. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems (with discussion). *Journal of the Royal Statistical Society series B*, 50:157–224, 1988.

[Ledley and Lusted, 1959] Robert S. Ledley and Lee B. Lusted. Reasoning foundations of medical diagnosis. *Science*, 130:9–21, 1959.

[Lucas, 1996] Peter Lucas. Structures in diagnosis from theory to medical application. Master's thesis, Free University of Amsterdam, the Netherlands, 1996.

[McGehee *et al.*, 1979] Harvey A. McGehee, James Bodley, and Jeremiah A. Barondess. *Differential Diagnosis : the Interpretation of Clinical evidence.* Saunders, Philadelphia, 1979.

[Miller, 1983] Perry L. Miller. Attending: Critiquing a physician's management plan. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(5):449–461, 1983.

[Nelsen, 1998] Roger B. Nelsen. *An Introduction to Copulas.* Springer, New York, NY, 1998.

[Oniśko *et al.*, 1997] Agnieszka Oniśko, Marek J. Druzdzel, and Hanna Wasyluk. Application of Bayesian belief networks to diagnosis of liver disorders. In *Proceedings of the Third Conference on Neural Networks and Their Applications*, pages 730–736, Kule, Poland, 14–18 October 1997.

[Pearl, 1988] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1988.

[Pryztula and Dash,] K. Wojtek Pryztula and Denver Dash. Testing of diagnostic models based on bayesian networks.

[Shannon, 1948] C. E. Shannon. A mathematical theory of communication. *Bell System Tech. J.*, 27:379–423, 623–656, 1948.

[Spiegelhalter and Knill-Jones, 1984] David J. Spiegelhalter and Robin P. Knill-Jones. Statistical and knowledge-based approaches to clinical decision-support systems, with an application in gastroenterology. *Journal of the Royal Statistical Society*, 147, Part 1:35–77, 1984.

[Stensmo and Terrence, 1994] Magnus Stensmo and Sejnowski J. Terrence. A mixture model diagnosis system. 1994.

[Xu, 1995] H. Xu. Computing marginals for arbitrary subsets from marginal representation in markov trees. *Artificial Intelligence*, 74:177–189, 1995.