

# Visualizing Inference in Bayesian Networks

J.R. Koiter

A thesis submitted to Delft University of Technology  
in partial fulfillment of the requirements for the  
degree of Master of Science

*Faculty of Electrical Engineering, Mathematics, and  
Computer Science, Department of Man-Machine Interaction*

June 16, 2006



Graduation Committee:

*Dr. Drs. L.J.M. Rothkrantz*

*Dr. Ir. C.A.P.G. van der Mast*

*Ir. H.J.A.M. Geers*

*Dr. Ir. M.J. Druzdzal (University of Pittsburgh)*



# Abstract

Inference in Bayesian networks is used to calculate the posterior probability distributions of unobserved variables in a network. These posterior probability distributions are used to draw conclusions and are the basis for decisions, in the domain of a particular model. Inference is a complex process and can be difficult to understand for even the most experienced Bayesian network users. In this thesis, we propose a technique to visualize important aspects of a Bayesian network, in order to make the process of inference more insightful. This technique consists of augmenting the visual representation of a Bayesian network with extra information. The only function of arcs in a Bayesian network is to indicate the relationships among the variables. We have used the arcs in a Bayesian network to show additional information: (1) the thickness of an arc is automatically adjusted to represent the strength of influence between two directly connected nodes and (2) the color of an arc is automatically adjusted to indicate the sign of influence between two directly connected nodes. Our technique does this in a novel, dynamic way, which is context-specific and takes into account any indirect influences. We have implemented our technique and integrated it into a software package called GeNIe, which can be used for developing Bayesian networks and is developed at the Decision Systems Laboratory of the University of Pittsburgh. A qualitative empirical evaluation showed that our technique and implementation are easy to use and understand and give a user more insight into a particular Bayesian network.



# Acknowledgements

This thesis is the result of my graduation project, done at the Decision Systems Laboratory of the University of Pittsburgh, from October 2005 to May 2006. I had a wonderful time and want to thank several people who helped me to make this thesis to what you have in your hands now.

Most importantly, I want to thank Dr. Ir. Marek J. Druzdzel. Marek always had helpful advice whenever I was stuck with my research or whenever I needed another point of view. Considering that I can be quite stubborn, this was not always an easy task. So, Marek, thank you very much for all your input and your confidence in me, I hope that you are just as satisfied with the final result as I am.

I also would like to thank Dr. Drs. L.J.M. Rothkrantz, who offered me the possibility to go to Pittsburgh and who helped me with everything related to my graduation, including helpful comments when I was finalizing my thesis.

Next, my parents, Rob and Anja Koiter, and my sister, Dagmar. Without you I would not be where I am today. Thank you for all your support, especially during my months abroad and the sometimes stressful time that followed, in which I was completing this thesis.

Furthermore, I would like to mention the following people, who all helped me with my thesis in one way or another: Pieter Kraaijeveld, Mark Voortman, Tomasz Sowinski, Changhe Yuan, Joris Hulst, Paul Maaskant, Martijn de Jongh, Divyasheel Sharma, Agnieszka Oniśko, Adam Zagorecki, Tomek Loboda and Auke Bajema.

Finally, my stay at the University of Pittsburgh would not have been possible without the financial support of: Fundatie van de Vrijvrouwe van Renswoude, Stimuleringsfonds voor Internationale Universitaire Samenwerkingsrelaties (STIR), Faculteitsfonds and Universiteitenfonds Delft.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Context . . . . .	1
1.2	Objectives . . . . .	4
1.3	Outline of this thesis . . . . .	5
<b>2</b>	<b>Background</b>	<b>7</b>
2.1	Bayes' Rule . . . . .	7
2.2	Bayesian networks . . . . .	7
2.3	Inference . . . . .	10
2.4	Influence diagrams . . . . .	10
2.5	Types of explanations in Bayesian networks . . . . .	12
2.6	Representing explanations . . . . .	12
2.7	GeNIe and SMILE . . . . .	13
<b>3</b>	<b>Previous work</b>	<b>15</b>
3.1	Abduction . . . . .	15
3.2	Scenario based explanations . . . . .	15
3.3	INSITE . . . . .	16
3.4	BANTER . . . . .	20
3.5	Weight of evidence . . . . .	21
3.6	Elvira . . . . .	24
3.7	BayesiaLab . . . . .	32
3.8	Discussion . . . . .	38
<b>4</b>	<b>General design</b>	<b>43</b>
4.1	Ancestors and descendants . . . . .	43
4.2	Paths of influence . . . . .	43
4.3	Markov blanket . . . . .	45
4.4	Relevance of findings . . . . .	45
4.5	Multiple cases . . . . .	45
4.6	Thickness of arcs . . . . .	46
4.7	Color of arcs . . . . .	46
4.8	Discussion . . . . .	46
<b>5</b>	<b>Thickness of arcs</b>	<b>47</b>
5.1	Ordinal nodes . . . . .	47
5.2	Strategy . . . . .	47
5.3	Thickness in influence diagrams . . . . .	52

5.4	Difference between distributions . . . . .	58
5.5	Distance measures . . . . .	60
<b>6</b>	<b>Color of arcs</b>	<b>71</b>
6.1	Overview . . . . .	71
6.2	Static coloring . . . . .	71
6.3	Dynamic coloring . . . . .	73
<b>7</b>	<b>Implementation</b>	<b>77</b>
7.1	Implementation in SMILE . . . . .	77
7.2	Integration into GeNIe . . . . .	79
7.3	Examples . . . . .	81
<b>8</b>	<b>Empirical evaluation</b>	<b>87</b>
8.1	Quantitative evaluation . . . . .	87
8.2	Qualitative evaluation . . . . .	87
<b>9</b>	<b>Conclusions and future work</b>	<b>93</b>
9.1	Conclusions . . . . .	93
9.2	Future work . . . . .	94
<b>A</b>	<b>Qualitative Evaluation</b>	<b>95</b>
A.1	Introduction . . . . .	95
A.2	Evaluation . . . . .	99
<b>B</b>	<b>Manual</b>	<b>101</b>
<b>C</b>	<b>Paper version</b>	<b>107</b>
C.1	Introduction . . . . .	107
C.2	Bayesian networks . . . . .	108
C.3	Thickness of arcs . . . . .	109
C.4	Color of arcs . . . . .	110
C.5	Implementation . . . . .	112
C.6	Empirical evaluation . . . . .	113
C.7	Concluding remarks . . . . .	114



# Chapter 1

## Introduction

This thesis is the result of my research at the Decision Systems Laboratory of the School of Information Sciences at the University of Pittsburgh, USA. During my stay, I have turned my attention to finding ways to make the workings of a Bayesian network more clear to a user.

### 1.1 Context

A Bayesian network can be seen as some form of an expert system. Belonging to the field of artificial intelligence, an expert system is a computer program that holds knowledge in some domain and is able to use this knowledge to perform tasks that a human expert normally would perform. An expert system could, if it is good enough, replace such a human expert, but in most cases the expert system is there to assist a human expert, not to replace him or her. There are many reasons why an expert system does not yet replace a human expert. One of the reasons is that in many cases the expert system is just not up to the job yet, the human expert is still the one who can do it best. But there is also the fact that many people, especially experts in a certain domain, are often very sceptical about expert systems. Some people look at expert systems as some kind of “black boxes”, of which the how and why of certain conclusions are unclear. But also for people who know a lot about Bayesian networks, it can still be that there is behaviour in a network that cannot be easily explained. In this thesis, we are going to investigate ways of reducing this unclarity in Bayesian networks.

A Bayesian network [27] consists of two parts: a qualitative part and a quantitative part. The qualitative part is a directed, acyclic graph in which the nodes are random variables and the arcs represent probabilistic dependencies among the nodes. The arcs can model causal relationships, but this is not necessary. An example of a Bayesian network is shown in Figure 1.1. It shows a simple network consisting of two nodes: “Smoking” and “Lung cancer”. The directed arc from “Smoking” to “Lung cancer” indicates that they are directly dependent on each other. If there would be no arc between the two nodes, they would be independent. This arc can be interpreted in a causal way, because it can be argued that smoking causes, or is a cause of, lung cancer.

The quantitative part consists of conditional probability tables and prior

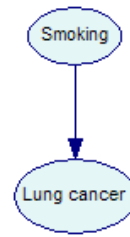


Figure 1.1: A simple Bayesian network.

yes	0.4
no	0.6

(a) The prior probability of node “Smoking”.

Smoking	yes	no
present	0.2	0.001
absent	0.8	0.999

(b) The conditional probability table of node “Lung cancer”.

Figure 1.2: The quantitative part of the Bayesian network of Figure 1.1

probabilities. A node has a prior probability if that node has no parents. When a node has one or more parents, it has a conditional probability table. Figure 1.2(a) shows the prior probability of the node “Smoking”, and Figure 1.2(b) shows the conditional probability table of the node “Lung cancer”. All the numbers are fictional. In Figure 1.2(a) we can see that, a priori, the chance that someone is a smoker is 0.4, which equals 40 percent, and that the chance that someone is not a smoker is 60 percent. In Figure 1.2(b) we see that there are two probability distributions in the conditional probability table of the node “Lung cancer”, one for each possible state of its parent, “Smoking”. If it is known that the person in question is not a smoker, then the chance of that person having lung cancer is only 0.1 percent. But when the person does smoke, then the chance of lung cancer is 20 percent.

The structure and parameters of a Bayesian network can be elicited from an expert in the domain. This means that an expert has to specify what the relations among the various variables are, e.g., which variables depend on each other and which do not. When that is done, the expert has to specify all the probabilities that are needed in the model. This can be labour intensive, but it is also possible to learn these parameters from data. So if a lot of data is present for the domain that is being modeled, that data can be used to automatically fill all the conditional probability tables and prior probabilities. Such data files can also be used to learn the structure of a network. So if all the circumstances are perfect, a Bayesian network can be learned automatically from a data set. But in practice there are often problems, for example that of an incomplete data set.

Using just the six numbers specified in Figure 1.2, the full joint probability distribution of this particular model can be reconstructed. A Bayesian network exploits the independencies among the variables in a domain, and uses these to encode the full joint probability distribution into prior probabilities and conditional probability tables. With a full joint probability distribution, any query in the domain can be answered. Bayesian networks can be used to calculate the impact of observing values of a subset of all the variables in a network on the

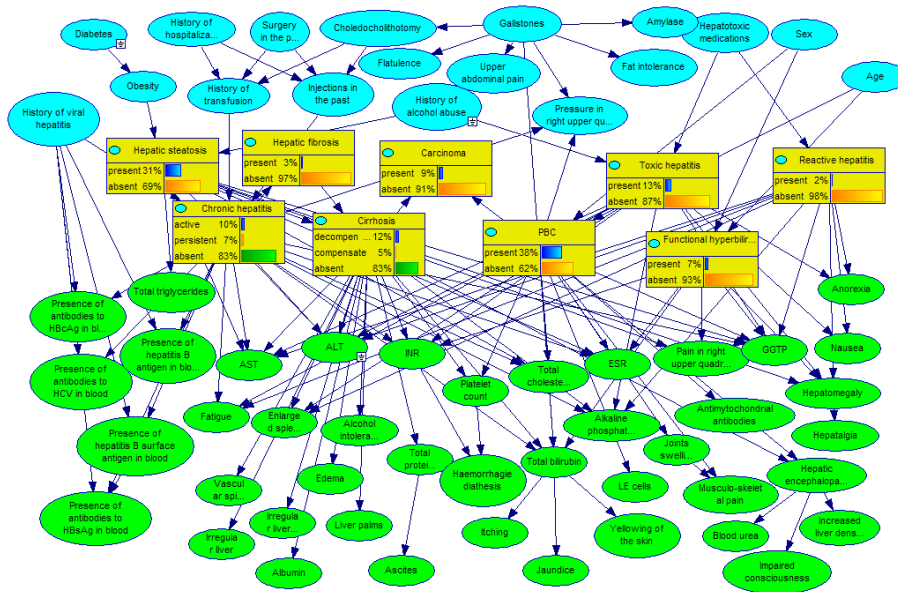


Figure 1.3: A more complicated Bayesian network.

remaining variables. For example, one could ask the question what the probability of a person being a smoker would be, if we know for a fact that that person does not have lung cancer. Or, more general, observing a set of symptoms, captured as variables in a medical diagnostic model, allows for computing the probabilities of diseases captured in that model. This process is known as inference [26], but is also called Bayesian updating, belief updating or reasoning.

The result of inference is what is interesting to a user of a Bayesian network. Decisions can be made and conclusions can be drawn by looking at the probabilities of certain variables of interest after observing one or more other variables. When the model is very small and simple, like the example of Figure 1.1, it is not that hard to understand the result of inference. But if a more complicated model is used, like that of Figure 1.3, this can be more problematic. In Figure 1.3, a Bayesian network is shown that models liver disorders [25]. The eight square nodes represent the eight possible disorders. In this situation, the variables “Diabetes”, “History of alcohol abuse” and “ALT” are observed, and for the nodes representing the disorders the current probability distribution is shown by the graphical bars. We can see that there is a fairly high probability that “PBC” or “Hepatic steatosis” is present. But why is this? Why do the three observations make the probabilities of those two disorders being present higher? The techniques and methods that try to answer such questions are called *explanations*.

There are various ways to explain a Bayesian network, many of which will be discussed in Chapter 2 and Chapter 3. To give an impression, an explanation of the situation in Figure 1.3 could try to show the user, either graphically or verbally, if and how the three observations impact a certain node of interest, e.g., “PBC” or “Hepatic steatosis”. Or something simpler, one could say that the different colors of the nodes in Figure 1.3 is also an explanation, because

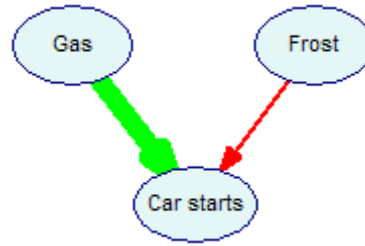


Figure 1.4: An example Bayesian network showing thickness and colors of arcs.

that shows the user a clear division between the nodes. In this case there are three classes: disorders (colored yellow), context variables (colored blue) and symptoms and test results (colored green). Another example of an explanation is shown in Figure 1.4. The simple example network consists of three nodes: “Gas”, “Frost” and “Car starts”. Both “Gas” and “Frost” influence the probability that a car will start. The thickness of the arcs indicates the amount of influence. Here we can see that having gas (or not) has a larger influence on the car starting than that frost has. Also, “Gas” has a positive influence (the arrow is colored green) on “Car starts”, i.e., having gas increases the probability of the car starting, and “Frost” has a negative influence (the arrow is colored red) on “Car starts”. Which is what we expect because if it is very cold a car is somewhat less likely to start. This makes the structure of the model much easier to understand and requires little effort from the user.

## 1.2 Objectives

In the past some attention has been given to explanations in Bayesian networks, as we will see in Chapter 3. Yet there is no common accepted way to “explain” a Bayesian network. The goal of this thesis is to create an explanation facility that is intuitive and easy to understand for a user. We can divide the goal of this thesis into the following subgoals:

- Summarize earlier work on this topic
- Propose a method to make a Bayesian network more insightful
- Implement our method
- Evaluate the performance of our method

Because we are going to try and explain something to somebody, we need to define our user. In order for our method to be insightful we have to make it in such a way that it will fit the prior knowledge of our intended users. We are going to target our method towards people who have a fairly good understanding of what Bayesian networks are, for example researchers that build Bayesian networks to aid in their research. The techniques that we are going to develop must be able to help such a user with building models and exploring them.

## 1.3 Outline of this thesis

This thesis is structured as follows. Chapter 2 introduces Bayesian networks in a more formal way, and outlines the field of explanations in Bayesian networks. Chapter 3 summarizes earlier work in this field. Chapter 4 contains our ideas and motivations. In Chapter 5 one of our two ideas, the thickness of arcs, is treated. Chapter 6 treats the other idea, color of arcs. Chapter 7 treats our implementation of the designs of Chapter 5 and 6. Chapter 8 contains the evaluation of the implementation of our methods. Finally, Chapter 9 holds our conclusions and future work propositions.



## Chapter 2

# Background

This chapter contains background information. It introduces Bayesian networks and a few other important concepts.

### 2.1 Bayes' Rule

In probability theory, where conditional probabilities are concerned, Bayes' rule [1] plays a central role. It can be formulated as follows:

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}.$$

In words this means that the posterior probability in a hypothesis  $Y$  after observation of some evidence  $X$  is equal to the likelihood of observing  $X$  given  $Y$ , times the prior probability of  $Y$ , divided by the prior probability of  $X$ .

If we assign a disease to  $Y$  and a symptom of that disease to  $X$ , we can see the importance of this theorem. It is often much easier to specify the probability of the symptom  $X$  given the disease  $Y$ , then it is to specify the probability of the disease  $Y$  given the symptom  $X$ , although that probability is very interesting to know. Bayes' rule makes this possible.

### 2.2 Bayesian networks

A Bayesian network is a probabilistic graphical network. It represents variables in a certain domain and visualizes the probabilistic relationships between them. These relationships can also be thought of as causal relationships. The formal definition of a Bayesian network is as follows [28]:

1. A set of random variables makes up the nodes of the network. Variables may be discrete or continuous.
2. A set of directed links or arrows connects pairs of nodes. If there is an arrow from node  $X$  to node  $Y$ ,  $X$  is said to be a parent of  $Y$ .
3. Each node  $X_i$  has a conditional probability distribution  $P(X_i|Parents(X_i))$  that quantifies the effect of the parents on the node.

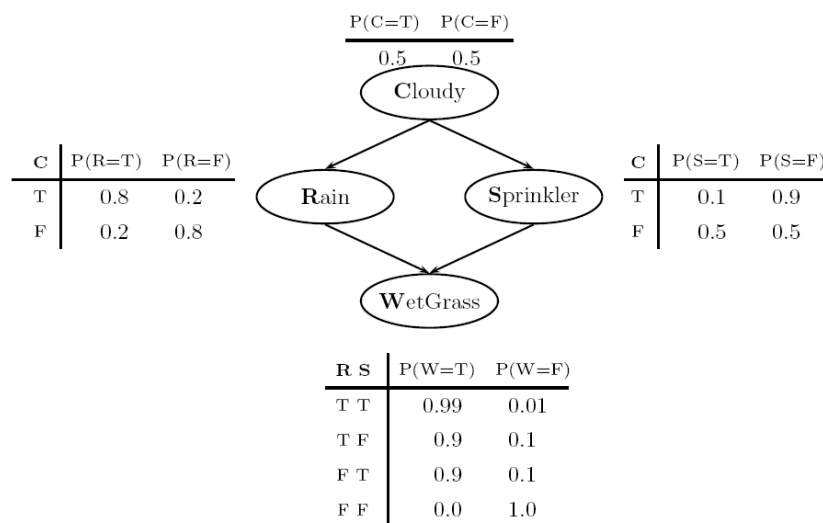


Figure 2.1: An example Bayesian network.

4. The graph has no directed cycles (and hence is a directed, acyclic graph, or DAG).

A Bayesian network defines a complete joint probability distribution over  $X$  given by:

$$P(X_1, \dots, X_i) = \prod_{i=1}^n P(X_i | \text{Parents}(X_i)).$$

To further illustrate these concepts we will introduce an example network in Figure 2.1 [28].

It shows a Bayesian network with four nodes and a conditional probability table for each node. It models the following situation: Whether it is cloudy or not influences the chance that it rains and the chance that the sprinkler will be on. If it is cloudy the sprinkler will most likely not be on. The wetness of the grass is influenced by both the rain and the sprinkler. If there is rain and the sprinkler is on, the probability of the grass being wet is the highest, i.e., 0.99. If there is no rain and the sprinkler is off, it is certain that the grass is not wet, the probability is 1.0.

An arrow between two nodes indicates that the two nodes are dependent, meaning that they influence each other. If there is no arc present between two nodes, then they have no influence on each other, at least not directly. Also, if we see that, for example, the grass is wet, then we have *observed* the variable (or node) *WetGrass*, in which case it has become *evidence*, an *observation* or a *finding*. These three terms can be used interchangeably.

This network can be used to infer probabilities like that of the sky being cloudy when we know that the grass is wet but the sprinkler is off, or the probability of the sprinkler being on when we know the grass is wet and there is no rain. We will explain why this is the case.

Every query about the domain, including the ones just posed, are specified by the full joint probability distribution  $P(\text{Cloudy}, \text{Rain}, \text{Sprinkler}, \text{WetGrass})$ .



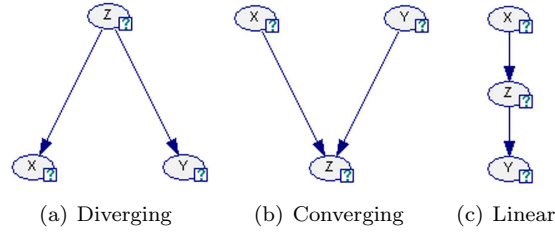


Figure 2.2: Connection types.

It consists, in this case, of  $2^4 = 16$  entries, the probability of every possible combination of variables is specified. The Bayesian network of Figure 2.1 represents the exact same distribution, but only has nine probabilities specified in its conditional probability tables. There are eighteen numbers present, but all variables are binary and therefore the probability of one state is one minus the probability of the other state. So only nine numbers are needed. This reduction is an important advantage of Bayesian networks and it is caused by the (conditional) independence assumptions made by the network. The larger the network or domain, the bigger the savings.

There are three kinds of connection types in a Bayesian network, expressing different kinds of independence, as shown in Figure 2.2.

Let  $X$ ,  $Y$  and  $Z$  be variables. If  $X$  and  $Y$  are independent, the following probabilistic expression is valid:  $P(X, Y) = P(X)P(Y)$ . Now if  $X$  is conditionally independent of  $Y$  given  $Z$ , we can write:  $P(X, Y|Z) = P(X|Z)P(Y|Z)$ . This is known as a *diverging* connection, as shown in Figure 2.2(a). We can now decompose the joint probability distribution:  $P(X, Y, Z) = P(X, Y|Z)P(Z) = P(X|Z)P(Y|Z)P(Z)$ .

A *converging* node, as shown in Figure 2.2(b), expresses the fact that  $X$  and  $Y$  are marginally independent, but conditionally dependent given  $Z$ . This allows us to factorize the joint probability distribution again:  $P(X, Y, Z) = P(Z|X, Y)P(X)P(Y)$ .

A third and last connection type is *linear*, see Figure 2.2(c). This models the situation that, for instance,  $X$  is the fact that someone left the headlights of his or her car on overnight. This is known to cause an empty battery the following morning,  $Z$ , which in turn causes the symptom of the car not starting,  $Y$ . Now, if through some test it is diagnosed that the battery is empty, then finding out that the car does not start will have no further influence on the probability that all this is caused by leaving the headlights on overnight. The joint probability distribution is factorized as follows:  $P(X, Y, Z) = P(Y|Z)P(Z|X)P(X)$ .

This allows us to introduce the notion of *d-separation* in a Bayesian network. Two nodes are said to be *d-separated* given specific evidence or observations, if they are independent given these specific observations. In other words, if two nodes are d-separated, they do not influence each other. If that is the case, there is no *active path* between the two nodes. A path is active if, looking at figure 2.2:

1. The connection is *diverging*, and  $Z$  has *not* been observed.
2. The connection is *converging*, and  $Z$ , or one of its descendants, has been observed.

3. The connection is *linear*, and  $Z$  has *not* been observed.

Now, if we return to our example, it can be seen that the joint probability distribution  $P(\textit{Cloudy}, \textit{Rain}, \textit{Sprinkler}, \textit{WetGrass})$  can be decomposed into:

$$P(\textit{Cloudy})P(\textit{Rain}|\textit{Cloudy})P(\textit{Sprinkler}|\textit{Cloudy})P(\textit{WetGrass}|\textit{Rain}, \textit{Sprinkler}),$$

all of which are given in the conditional probability tables. Using this we can infer any probability in the domain. This process is called *inference*.

## 2.3 Inference

The basic task of a Bayesian network is to compute the posterior probability distributions for a set of query variables, given an observation of a set of evidence variables. This process is known as inference [26], but is also called Bayesian updating, belief updating or reasoning. There are two ways to approach this, either exact or approximate. Both approaches are worst-case NP-hard [5]. An exact method obviously gives an exact result, while an approximate method tries to approach the correct outcome as close as possible. Exact inference is only possible for a restricted class of networks. That is networks that belong to the class of singly connected networks, also known as polytrees. A network belongs to this class if the underlying undirected graph has either zero or one path between any two nodes. The underlying undirected graph is the graph one gets when ignoring the direction of the edges. When the network is multiply connected it is possible to use clustering techniques to convert it to a singly connected one, after which exact inference can be performed. In practice, however, the networks are sometimes of such a size that exact inference and/or clustering becomes infeasible. That is why approximate methods exist. There are many different approximate solutions possible, which one is the best depends on the network at hand. A few approximate algorithms are Probabilistic Logic Sampling [15], Likelihood Sampling [30, 9], Backward Sampling [10], Adaptive Importance Sampling [2], and Approximate Posterior Importance Sampling [34].

## 2.4 Influence diagrams

An influence diagram [16] is a Bayesian network augmented with decision and value nodes. It models a certain decision problem and the goal is to choose the decision alternative for which the expected gain (or utility) is the highest. An example of an influence diagram is shown in Figure 2.3.

Besides the two oval nodes, which are normal chance nodes like in a Bayesian network, there are three differently shaped nodes: two rectangle shaped and one diamond shaped. The rectangle shaped nodes are *decision* nodes and they represent variables that are under the control of the decision maker and model the decision alternatives available to the decision maker. Each decision node usually has multiple decision alternatives. For example, the node “Investment Decision” of Figure 2.3 has two: *invest* and *donotinvest*. If a certain decision is made, i.e., a decision node becomes observed, it will impact the value of all its children. Each child has a probability distribution specified for each possible

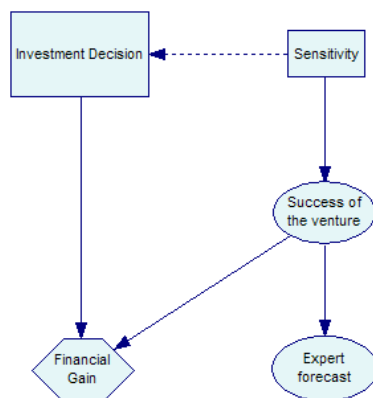


Figure 2.3: An influence diagram.

Probability distributions for different policies:

Sensitivity	Low	Nominal	High
Success	0.1	0.2	0.35
Failure	0.9	0.8	0.65

Figure 2.4: Probability distributions for node *Success of the venture*.

decision, or an expected gain if the node is a value node. For example, Figure 2.4 shows the conditional probability table of node “Success of the venture”, which is conditioned on the decision node “Sensitivity”.

The diamond shaped node, “Financial Gain”, is a *value* node. A value node represents *utility*, i.e., a measure of desirability of the outcomes of the decision process. It is quantified by the utility of each of the possible combinations of outcomes of the parent nodes. These utilities are subjective and they can be any number. The goal of the whole decision making process is to maximize this number. The definition of the node “Financial Gain” is shown in Figure 2.5.

In an influence diagram, an arrow between two decision nodes has a special meaning. Such an arrow indicates the order in which the decisions are made. The decision at the tail of the arrow is expected to be taken before the decision at the tip of the arrow will be made. In Figure 2.3 there is such an arrow between node “Sensitivity” and “Investment Decision”, it is a dashed arrow.

The influence diagram of Figure 2.4 models the following situation: We are able to invest in a certain venture. If we invest and the venture is a success, our revenues (or financial gain) will be the highest. If we invest and the venture fails, we lose all our money. The ultimate investment decision is modeled by the decision node “Investment Decision”. But before we make our decision, we have to make a decision for the node “Sensitivity”. This node models our uncertainty about the probability that the venture will be a success, which is modeled by the node “Success of the venture”. The decision node “Sensitivity”

Expected utilities for different policies:

Sensitivity	Low		Nominal		High	
	Invest	DonotInvest	Invest	DonotInvest	Invest	DonotInvest
Investment De...	-3500	500	-2000	500	250	500
Exp. utility						

Figure 2.5: Expected utilities for value node *Financial Gain*.

has three decision alternatives, each of which results in a different probability distribution in the node “Success of the venture”. Also, before we make our decision about whether to invest or not, we can consult an expert that predicts the successfulness of the venture. Finally, the value node “Financial Gain” models the final outcome of our investment, i.e., how much money we make or lose. In this case the utility can be interpreted as money, but in general utility has no units.

## 2.5 Types of explanations in Bayesian networks

Literature on the topic of explanations in Bayesian networks distinguishes three kinds of explanations. [23]. The first one is called *abduction*. Abduction is the process of determining the most probable values of the unobserved variables in a Bayesian network. Such a configuration is usually referred to as an MPE (Most Probable Explanation) and can contain every unobserved variable, in which case it is called *total* abduction, or it can contain only a subset of the unobserved variables, in which case it is called *partial* abduction. Abduction involves maximizing the probability of a set of unobserved variables given one or more findings. It is also possible to generate a set of MPE’s, for example the five configurations with the five highest probabilities.

The other two kinds of explanations are *static* and *dynamic* explanations. A *static* explanation only considers the information that is contained in the Bayesian network model, i.e., without any reasoning being done. Another way of putting it is that it offers explanations of the assumptions of the model. A static explanation could, for example, make the independence statements contained in a model explicit, or it could describe the prior probability of variables.

A *dynamic* explanation, on the other hand, is an explanation of the reasoning process in a Bayesian network. So, given one or more findings and a variable of interest, a *dynamic* explanation tries to give the user insight into the process that caused the variable of interest to be affected in the way that it has. More specifically, it tries to explain the changes in the posterior probability of the variable of interest with respect to the findings. This type of explanation can be viewed as trying to answer the questions: “What were the most influential findings?” and “Why is a certain finding influential?”. A finding is influential when it affects the posterior probability of the variable of interest in either a positive or a negative way.

Within *dynamic* explanations another distinction can be made. There is a difference between *micro* and *macro* explanations [29]. A *micro* explanation tries to justify the variations of the probability distribution of a certain node, while a *macro* explanation tries to make the main lines of reasoning from findings to variable of interest clear to the user and therefore considers a bigger part of the model.

## 2.6 Representing explanations

An explanation should be presented in a way that is effective, convenient, as well as easily accessible. A distinction that can be made in this respect is that between *verbal* and *graphical* explanations.

A *verbal* explanation could be, for example: “Variable A is dependent on variable B, but given variable C they are independent”, or “State zero is somewhat more likely than state one”.

A *graphical* explanation uses graphical means to communicate an explanation. The most obvious and basic explanation of this type is the visualization of the network structure. If the user has enough knowledge about Bayesian networks, he can deduce the dependencies and independencies between the variables in the modeled domain from this view. Another example is to display the probabilities of the various states of a variable using graphical bars that range from zero to one hundred percent. Some of the works reviewed in Chapter 3 use graphical explanations.

## 2.7 GeNIe and SMILE

At the Decision Systems Laboratory of the University of Pittsburgh, the two main software packages that are developed are called GeNIe and SMILE. SMILE is the engine and GeNIe is the graphical user interface on top of SMILE. Most of the results of the research done at the Decision Systems Laboratory ultimately find their way into GeNIe and SMILE. Because of their versatility and reliability, GeNIe and SMILE have also been embraced by a number of government, military and commercial users.

### 2.7.1 SMILE

SMILE (Structural Modeling, Inference, and Learning Engine) is a fully platform independent library of functions implementing graphical probabilistic and decision-theoretic models, such as Bayesian networks, influence diagrams, and structural equation models. Its individual functions, defined in the SMILE Applications Programmer Interface (API), allow to create, edit, save, and load graphical models, and use them for probabilistic reasoning and decision making under uncertainty. SMILE is implemented in C++ in a platform independent fashion. Individual functions of SMILE are accessible from C++ or (as functions) from the C programming language. As most implementations of programming languages define a C interface, this makes SMILE accessible from practically any language on any system. SMILE can be embedded in programs that use graphical probabilistic models as their reasoning engines.

### 2.7.2 GeNIe

GeNIe is essentially a graphical front end for SMILE. It is a development environment for building graphical decision-theoretic models. GeNIe’s name and its uncommon capitalization originates from the name Graphical Network Interface, given to its predecessor, the original simple interface to SMILE. GeNIe is implemented in Visual C++ and draws heavily on the MFC (Microsoft Foundation Classes). This makes it not easily portable, GeNIe is only available for the Windows operating system. GeNIe allows for building models of any size and complexity, limited only by the capacity of the operating memory of the computer it is running on. GeNIe is a developer environment. Models developed using GeNIe can be embedded into any application and used on any computing

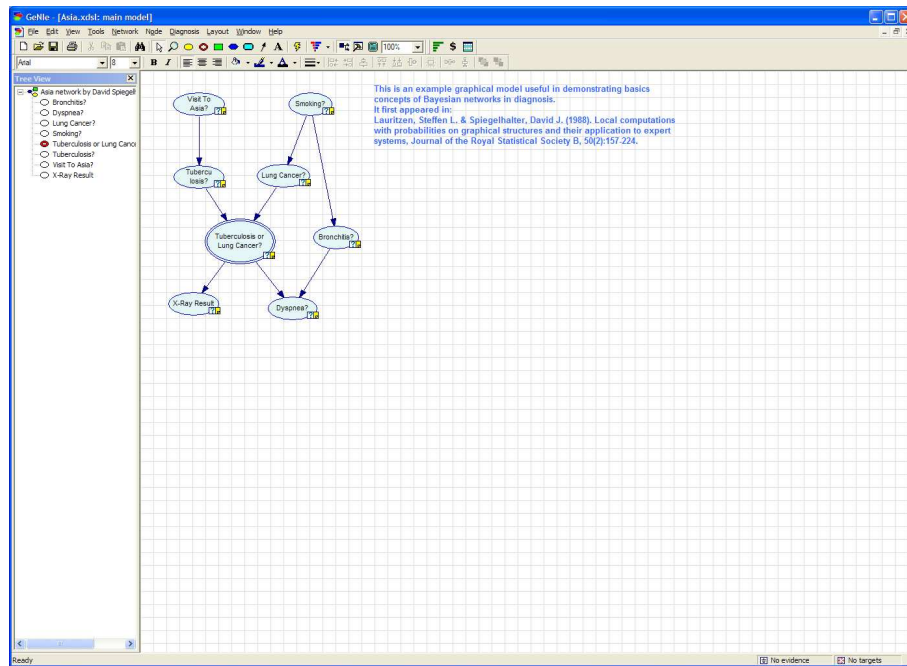


Figure 2.6: A screenshot of GeNIe displaying a Bayesian network.

platform, using SMILE, which is available for almost every platform thinkable. Figure 2.6 shows a screenshot of the GeNIe program.

## Chapter 3

# Previous work

This chapter reviews previous works in the field of explanations of Bayesian networks.

### 3.1 Abduction

As said before, abduction is the process of finding the most probable assignment of (a subset of) the unobserved variables, given some observed variables. This most probable explanation (MPE) can be formulated as finding  $\max(P(w|e))$  where  $w$  is an assignment to (a subset of) the unobserved variables  $W$ ,  $e$  is the observed evidence of the variables in  $E$  and  $E \cap W = \emptyset$ , i.e.,  $E$  and  $W$  are disjoint. For example, if we consider the simple network in Figure 3.1 [28], when we observe that the grass is wet, one possible explanation is: *Cloudy = False*, *Sprinkler = True*, *Rain = False* and *WetGrass = True*. The probability of that explanation is:  $0.5 \times 0.5 \times 0.8 \times 0.9 = 0.18$ . Note that this is *an* explanation, it does not have to be the most probable one. There are many algorithms to find the most probable explanation, for example [35, 12].

### 3.2 Scenario based explanations

An approach very closely related to abduction is one called *scenario based explanations* [7]. A *scenario* is basically a form of partial abduction, with the added semantics that the variables create a kind of causal story. An example could be: “Not Cloudy, therefore Sprinkler and no Rain, therefore WetGrass”. In order to create such a scenario the variables that are to be included into the scenario have to be identified first [8, 6]. One could ofcourse include all unobserved variables, but even for a Bayesian network of moderate size this would become incomprehensible. Therefore only those variables that are in some way related to the variable of interest should be included. For instance, variables that are independent of the variable of interest given the observed evidence can be excluded. Another criterion is that only those variables need to be included that are computationally relevant to the variable of interest. A variable is computationally relevant if its conditional probability distribution is needed to compute the posterior probability distribution of the variable of interest. This type of

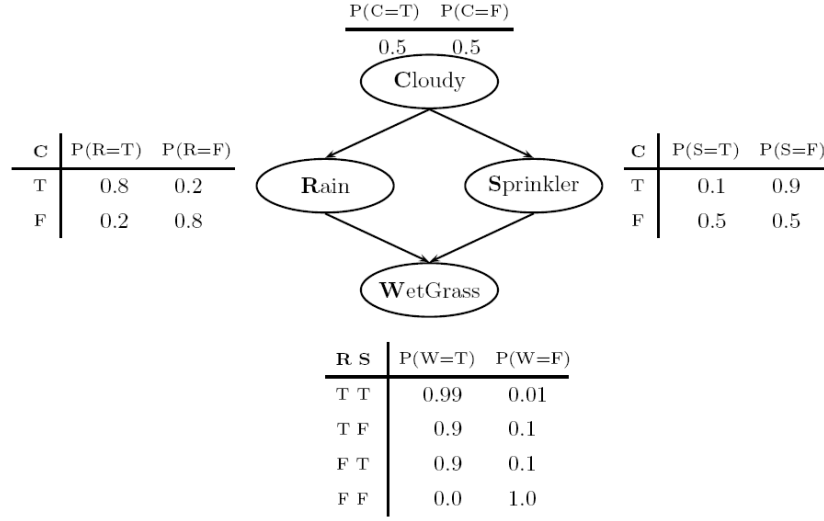


Figure 3.1: An example Bayesian network.

explanation is a *dynamic macro* explanation, because it considers not just one variable and tries to explain the reasoning process.

### 3.3 INSITE

In his PhD thesis, H.J. Suermondt proposes his methodology for explanation of Bayesian networks, which he calls *INSITE* [31], which stands for Insight about Network Structure and Inference Through Explanation. INSITE has two main features:

1. It determines which findings influence the posterior probability of a variable of interest.
2. It determines the paths through which the findings flow, the so called *chains of reasoning*.

#### 3.3.1 Influence of findings

To find out which findings influence the variable of interest a cost function is used. This cost function takes two probability distributions and assigns a score according to the difference between the two:  $H(P(D|E); P(D))$ . In this case the cost function  $H$  calculates the loss, or cost, of using  $P(D)$  when our best knowledge is  $P(D|E)$ . The cost function  $H$  can be any of many existing cost functions. Suermondt chose to use *cross-entropy*,

$$\sum_i \left[ p(d_i|e) \log \left[ \frac{p(d_i|e)}{p(d_i)} \right] \right].$$

Cross-entropy assigns a very high penalty to incorrect statements of certainty. So if, for example, a probability of a state in the distribution  $P(D|E)$  is, let



Table 3.1: Implications of the joint cost of omission.

$H^-(F)$	$H^-(\neg F)$	Conclusion
insignificant	insignificant	Both $F$ and $\neg F$ are sufficient to produce the inference result.
insignificant	significant	$F$ is not explanatory; we can disregard $F$ .
significant	insignificant	$F$ is explanatory and sufficient.
significant	significant	Some evidence in $F$ and some in $\neg F$ is necessary to explain the inference result.

us say, 0.6 and this same state has probability 0.99 in  $P(D)$ , then the outcome of the cross-entropy cost function will be very high. Besides that other advantages of cross-entropy are that it is easy to compute and that it can be used in combination with a significance threshold in order to support multiple levels of detail.

To determine whether a single finding influences the posterior probability of a variable of interest, the cost of omission of a finding  $E_i$  is defined as:

$$H^-(E_i) = H(P(D|E); P(D|E \setminus E_i)),$$

where  $\setminus$  is the set-difference operator. So if  $H^-(E_i)$  is significant it can be concluded that  $E_i$  has an important role in the inference result. If  $H^-(E_i)$  is found to be insignificant a definite conclusion cannot be drawn,  $E_i$  can either be consistent with the rest of the findings or it can be of limited importance. For multiple findings, the joint cost of omission of a set of findings  $F \subseteq E$  is defined as:

$$H^-(F) = H(P(D|E); P(D|E \setminus F)).$$

The same significance criteria hold for this case as for the case with a single finding. Note that to completely analyse all the evidence all possible subsets of  $E$  have to be instantiated and used to calculate  $H$ . This results in a complexity that is exponential in the number of findings.

Suermondt continues with defining an *explanatory* set of findings and a *sufficient* set of findings:

A set of findings  $F \subseteq E$  is *explanatory* if and only if  $H^-(F) > \theta$ ,

where  $\theta$  is a certain threshold. If a set of findings is explanatory it contains some findings that are relevant to  $D$ .

A set of findings  $F \subseteq E$  is *sufficient* if and only if  $H^-(\neg F) \leq \theta$ ,

where  $\neg F = E \setminus F$  and therefore  $H^-(\neg F) = H(P(D|E); P(D|F))$ . The term *sufficient* refers to the property that  $F$  by itself is sufficient to obtain the inference result. Table 3.1 summarizes the implications of  $H^-(F)$  and  $H^-(\neg F)$ .

The final aspect of this analysis concerns determining data conflicts. A data conflict is essentially the situation where a combination of findings point in a certain direction, but some of those findings point in a different direction. So a few of all the findings contradict the overall inference result. INSITE can detect data conflicts using the previously defined cost of omission, and a concept called the *direction of change*. The direction of change is defined as follows: For a node

$D$  with possible values  $d_1 \dots d_n$ , the direction of a change from  $P(D)$  to  $P'(D)$  is a vector  $Dir(P(D); P'(D)) = (dir_1, \dots, dir_n)$  in which  $dir_i$  is equal to the sign of  $p'(d_i) - p(d_i)$ . So the possible values of  $dir_i$  are “+”, “-” and “0”. Combinations for which the direction of change is the same are “+” and “+”, “+” and “0”, “-” and “-”, “-” and “0”, and finally “0” and “0”.

In case of a no-conflict scenario, the cost of omission of the complete set of findings,  $H^-(E)$ , is greater than the cost of omission of a single finding,  $H^-(E_i)$ . This is based on the difference between  $P(D|E)$  and  $P(D|E \setminus E_i)$ . To give a numerical example of this let us say that  $p(d|E)$  equals 0.2. If we leave out one finding, i.e.,  $p(d|E \setminus E_i)$ , the probability of  $d$  increases to 0.4. And if we finally leave out all findings, i.e.,  $p(d)$ , the probability increases further to 0.6. In this case no data conflict is present.

When a data conflict is present though, there are two possible situations. In the first one a certain finding conflicts with and dominates the remaining findings. To illustrate this with an example, let us again say that  $p(d|E)$  equals 0.2. If we omit one finding, i.e.,  $p(d|E \setminus E_i)$ , the probability of  $d$  increases to 0.6. And if we finally leave out all findings again, i.e.,  $p(d)$ , the probability drops to 0.4. So leaving out just  $E_i$  causes a greater increase in probability of  $d$  than leaving out all findings. In this case it is expected that  $H^-(E_i)$  is greater than  $H^-(E)$ .

In the second situation a finding conflicts with the remaining findings, but it does not dominate them. Instead it is the other way around, this time the conflicting finding is dominated by the remainder of the findings. Again an example, let us say that  $p(d|E)$  equals 0.4. If we leave out one finding, i.e.,  $p(d|E \setminus E_i)$ , the probability of  $d$  drops to 0.2. And if we finally leave out all findings again, i.e.,  $p(d)$ , the probability increases to 0.6. In this case we see that the directions of change are different, i.e.,  $Dir(P(D|E); P(D))$  is different from  $Dir(P(D|E); P(D|E \setminus E_i))$ .

So, to summarize, if there is no conflict then the following holds:

$$H^-(E_i) \leq H^-(E)$$

and

$$Dir(P(D|E); P(D|E \setminus E_i)) = Dir(P(D|E); P(D)).$$

If the following is true:

$$Dir(P(D|E); P(D|E \setminus E_i)) \neq Dir(P(D|E); P(D)),$$

then finding  $E_i$  is in conflict with the other findings, but it does not dominate them.

If, finally, the following is true:

$$H^-(E_i) > H^-(E),$$

then finding  $E_i$  is in conflict with the other findings and it dominates them.

### 3.3.2 Chains of reasoning

A *chain of reasoning* is a path through the network from a finding  $E_i$  to a variable of interest  $D$ . In order to detect the nodes that are part of such a path Suermondt defines two types of nodes: A *nuisance* node and a *proctored* node.

A *nuisance* node is a node that is computationally related to  $D$  given  $E$ , but that is not part of any *direct chain* from any  $E_i$  to  $D$ . A *direct chain* is a sequence of nodes where each node is connected to the next one through an arc and where each node is computationally related to  $D$  given  $E$ . A nuisance node does not need to be included in an explanation because it is considered to be a side effect.

A *proctored* node is a node that is adjacent to two of its parents within a direct chain and is an evidence node or has at least one successor that is an evidence node. A proctored node is special because, when there is just one direct chain from a finding to the variable of interest, it is the only node that does not d-separate that chain. Because of this the evidence sort of bypasses the proctored node, only the two adjacent parents that are part of the direct chain have to be considered.

In a multiply connected network there can be more than one direct chain for the same finding and variable of interest, as opposed to singly connected networks, where there can only be one direct chain. These chains can overlap as well. To organize multiple chains Suermondt identifies so called *knots*. He defines a *knot* as follows: A knot in a set  $S$  of direct chains from finding  $E_i$  to node  $D$  given a set of evidence  $E$  is a node  $K_j$  such that  $K_j$  is in every chain in  $S$ , and  $K_j \cup (E \setminus E_i)$  d-separates  $E_i$  from  $D$ . A knot is useful because any change in its marginal distribution fully explains the effect of finding  $E_i$  on  $D$ . Knots are used to avoid discussing some subchains multiple times. Knots are identified and the explanation is structured so that every subchain is treated separately, i.e., from the finding to the first knot, between knots, and from the last knot to the variable of interest. It could be that multiple paths exist between two knots. In that case, to further organize things, Suermondt uses a heuristic that treats certain nodes like knots, that are actually not really knots in the sense that they do not d-separate the part of the network that is being considered.

The INSITE method continues with determining which chains are relevant to the inference result on a numerical basis. Each node in a chain that d-separates the finding from the variable of interest (which is every non-proctored node) is inspected. If there is no significant change between the prior and posterior probability according to the cost function, then that node is “blocking” the chain and the chain can be discarded.

To determine whether a chain has a positive or negative influence the probability distribution of  $D$  is studied when a chain is temporarily severed. The arc to remove is selected in such a way that the least number of chains are severed at once, ideally only one chain is severed when an arc is removed. Unless the arc was redundant, removing an arc results in a new independence assumption in the model. This can lead to changes in the prior probabilities of the nodes in the network. Any significant changes in the prior probability of the variable of interest  $D$  should be taken into account when drawing conclusions. Table 3.2 summarizes this.

To find out if a certain chain that contains a certain arc contributes or conflicts with the overall inference result, an approach like that of finding influential findings in Section 3.3.1 is used.

The final part of the INSITE method involves determining the effect of an arc within a chain. This is done to analyse the flow of evidence along the chain. The arc between every pair of nodes along the chain is temporarily severed and the implications are studied. This study focuses on both of the nodes that

Table 3.2: Implications of the removal of an arc.

$H(P(D); P'(D))$	$H(P(D E); P'(D E))$	Conclusion about arc $A$
insignificant	insignificant	Arc $A$ is not important to the inference result.
insignificant	significant	Arc $A$ is relevant to the inference result.
significant	insignificant	Arc $A$ influences the prior probability of $D$ , but this effect is overshadowed by the transmission of evidence from $E$ to $D$ . Arc $A$ is not relevant to the inference result.
significant	significant	Removal of arc $A$ changes the prior probability of $D$ . Arc $A$ is important but it cannot be determined from these results whether arc $A$ is necessary for evidence transmission from $E$ to $D$ .

are connected by the severed arc separately. If we have two nodes,  $A$  and  $B$ , and an arc  $AB$  from  $A$  to  $B$ , the cost and direction of change in variable  $B$  is determined to explain the flow of evidence along the chain. From the point of view of  $A$  though, the cost and direction of change is used to determine whether there is evidence other than  $E_i$ , or another chain of reasoning, that is affecting nodes  $A$  and  $B$ . If this is the case then the explanation takes this into account by noting that the change in probability of  $A$  is caused by findings other than  $E_i$ .

### 3.4 BANTER

BANTER [13] is a generic Bayesian network-based tutoring shell designed to be used in a medical domain. For any network where the variables can be grouped into hypothesis, observations and diagnostic procedures it can perform various tasks:

- compute the posterior probability of a hypothesis
- determine the best diagnostic procedure to affirm or exclude a hypothesis
- quiz the user in the selection of optimal diagnostic procedures
- explain the system's reasoning

The explanation of reasoning is done by a method that is based on Suermondt's INSITE. It also has the two aspects: identifying influential evidence as well as finding the paths through which evidence flows.

### 3.4.1 Influence of findings

To determine the influence of a finding on a hypothesis  $H$ , BANTER uses the measure  $I(a_i; b_j) = \log \frac{P(a_i|b_j)}{P(a_i)}$ , where  $a_i$  and  $b_j$  are states of random variables  $A$  and  $B$ , respectively. It expresses the information provided by the event  $A = a_i$  by the event  $B = b_j$ . A large value means that  $b_j$  strongly increases the probability of  $a_i$  while a large negative value means that it strongly decreases the probability. This is used to define one of the two equations that BANTER uses:

$$\text{influence}(H; E; E_i) = \sum_{h_j \in H} I(h_j; E) \cdot I(h_j; E_i).$$

This determines whether the probability shift produced by  $E_i$  is in the same direction as that of all the findings combined,  $E$ .

The second equation is used when a certain  $E_i$  shifts some states of the hypothesis  $H$  in one direction, and some states in the other. The finding is then said to have a *mixed* influence and that can be quantified, without regard to direction, using:

$$\text{impact}(H; E_i) = \sum_{h_j \in H} |I(h_j; E_i)|.$$

Using these two measures all the findings are analysed and separated into findings that agree with the overall inference result, findings that disagree and findings that have a mixed influence. Findings in any one of the categories are used in the verbal explanation facility of BANTER when they are of strong influence.

### 3.4.2 Paths of influence

Using the set of important findings that is identified in Section 3.4.1, BANTER finds the influential paths from findings to the hypothesis  $H$ . This is done by a depth-first search starting from each important finding, using an algorithm that identifies nodes that are part of a path using a chart based on d-separation. To limit the number of paths that are generated a maximum length of a path is set. If there are more than five paths<sup>1</sup> the number of paths is reduced by computing the strength of a path and choosing the five strongest ones. A path is only as strong as its weakest link. For every node  $N$  along a path  $\text{impact}(N; E_i)$  is computed and the strength of the path is the minimum value found. The information obtained this way is used to generate a verbal explanation of the path. An example of such an explanation could be: “Disease A causes Symptom B, which is detected by Test C”.

## 3.5 Weight of evidence

Madigan, et al. [24] propose an explanation method for Bayesian networks that is also able to find the most influential findings, as well as the paths through which they flow. Their method is based on Good’s *weight of evidence* [11]:

$$W(H : E_i) = \log \frac{P(E_i|H)}{P(E_i|\neg H)},$$

<sup>1</sup>The number five was chosen arbitrarily.

where  $E_i$  is a finding and  $H$  is a hypothesis, or variable of interest. The weight of evidence expresses numerically how much a finding  $E_i$  supports  $H$ . As can be seen  $H$  has a positive and negative state, which indicates a binary variable. For non-binary variables Madigan, et al. flag one state as the positive one, and all the others as negative.

They continue with defining the *potential weight of evidence*  $W(H : e_i)$ , where  $e_i$  is an outcome of a certain evidence variable, or test,  $E$ . This can be calculated for each possible outcome. It expresses the weight of evidence that would be provided for  $H$  if the outcome was known to be  $e_i$ .

The final definition is the *expected weight of evidence*. This is provided by a test  $E$  for a hypothesis  $H$  and is the average weight of evidence of the possible test outcomes when  $H$  is true:

$$EW(H : E) = \sum_{i=1}^n W(H : e_i) P(e_i|H).$$

This measure expresses the information content of a future finding.

Suppose we have a chain of three binary nodes,  $A$ ,  $B$  and  $C$ , where  $A$  is a parent of  $B$  and  $B$  is a parent of  $C$ . To visualize the flow of evidence along a certain path each node in that path is visited and three criteria are considered:

1.  $\text{sign } W(C = 1 : A = 1) = \text{sign } W(C = 1 : B = 1) \times \text{sign } W(B = 1 : A = 1)$

So, if  $A = 1$  is of positive influence on  $B = 1$  and  $B = 1$  is of positive influence on  $C = 1$ , then  $A = 1$  is of positive influence on  $C = 1$ .

2.  $|W(C = 1 : A = 1)| \leq |W(B = 1 : A = 1)|$

So, the influence that  $A$  can have on  $C$  is limited by the influence that  $A$  has on  $B$ . In other words,  $B$  acts as a gateway for the flow of evidence.

3.  $|W(C = 1 : A = 1)| \leq |W_{rel:A=1}(C = 1 : B)|$

Here  $|W_{rel:A=1}(C = 1 : B)|$  is the *relevant outgoing weight of evidence* which is defined as:

$$W_{rel:A=1}(C = 1 : B) = \begin{cases} W(C = 1 : B = 1) & \text{if } W(B = 1 : A = 1) > 0 \\ W(C = 1 : B = 0) & \text{if } W(B = 1 : A = 1) \leq 0 \end{cases}$$

So if  $A = 1$  is of positive influence on  $B = 1$ , then  $W(C = 1 : B = 1)$  is the relevant outgoing weight, otherwise it is  $W(C = 1 : B = 0)$ .

This all means that the total weight of evidence of  $W(C = 1 : A = 1)$  is constrained by both  $W(B = 1 : A = 1)$  and by the relevant outgoing weight.

### 3.5.1 Presentation

Madigan, et al. have implemented this method into a software package called GRAPHICAL-BELIEF<sup>2</sup>. It can be used to create Bayesian networks which are displayed in a graphical way, i.e., with drawn nodes and arcs. Its most interesting features are *evidence balance sheets* and visualizing *flows of evidence*.

<sup>2</sup>GRAPHICAL-BELIEF is not available anymore.

Indicant	State	WOE	Target Probability
Initial			0.72
Pain	<input checked="" type="checkbox"/>		0.71
Previous-Surgery	<input checked="" type="checkbox"/>		0.63
Size	<input checked="" type="checkbox"/>		0.65

Figure 3.2: An evidence balance sheet.

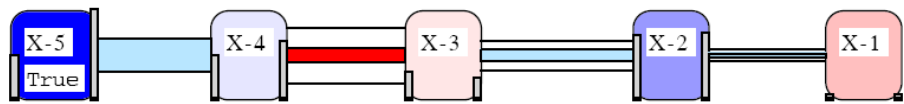


Figure 3.3: An evidence chain.

3.5.2 Evidence balance sheets

An evidence balance sheet visualizes the contribution of each finding to the variable of interest by displaying the corresponding weights of evidence. In order to do so each finding is added to the pool of evidence sequentially, one after another. Each time the weight of evidence is calculated. Figure 3.2 is an example of such an evidence balance sheet.

3.5.3 Flows of evidence

If the Bayesian network to be explained is a polytree, there exists exactly one path between a finding and a variable of interest. For each node along the path the actual weight of evidence and the potential weights of evidence are calculated and visualized. An example of this is shown in Figure 3.3. The width of the connection between two nodes represents the potential weight of evidence available, if the node's value was known with certainty. The width of the interior connection represents the actual weight of evidence. The difference in width between the two connections is an indication of how much a node could influence the next one compared to how much it influences it in this particular situation. The color blue represents a positive influence while the color red indicates a negative influence.

An alternative display uses circles to dynamically show the flow along a path. Figure 3.4 is an example. The diameter of the circles corresponds with the width of the inner connection in the previous figure.

3.5.4 Beyond polytrees

The above method is only applicable to singly connected Bayesian networks. In multiply connected networks there can be inconsistencies, because there is not just one unique path from finding a to a variable of interest. A certain

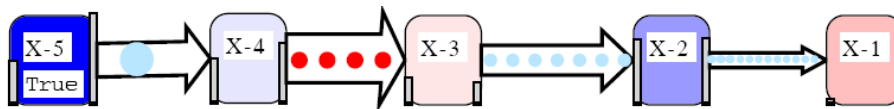


Figure 3.4: An evidence chain with circles.

finding can have a certain weight of evidence with respect to a certain variable of interest, but along each path leading from that finding to the variable of interest there is a connection between two variables for which all potential weights of evidence are zero. This would indicate that the influence along that path is zero, but that is not the case. The weight of evidence criterion cannot be used in such situations.

To relax the requirement of singly connected networks, Madigan, et al. show that their method can also be used on *Berge networks*. A Berge network is an undirected graph for which two conditions hold: (1) the graph is *chordal* and (2) the maximum clique intersection is of size one. A graph is chordal when all cycles of length greater than three have a connection between two intermediate nodes in the cycle. For these Berge networks there exists an algorithm that finds a unique path between two nodes, by collapsing the model onto that path. It finds the relevant variables for an explanation.

Still, for some non-Berge networks, there is no automated way to create a unique path between two nodes. In such a situation, Madigan, et al. require the user to manually combine nodes until it is transformed into a Berge network.

In GRAPHICAL-BELIEF there are five levels of explanations:

1. **Node coloring**

Nodes are colored according to their probability or weight of evidence.

2. **Node marginals**

Plots two vertical bars at the side of each node of which the lengths correspond to the prior and posterior probabilities of the node.

3. **Evidence balance sheet**

This has been discussed earlier in this section. It shows which how much each finding influences the variable of interest.

4. **Relevant potential weights of evidence**

This consists of showing graphically the potential weights of evidence by adjusting the width of the edges, as shown in Figure 3.3.

5. **Support for binary variables**

To support binary variables the user can mark one state of a non-binary variable as the “postive” state. Madigan, et al. view this feature as a level of explanation.

## 3.6 Elvira

Elvira is a software package that allows for creating Bayesian networks and performing inference [3, 21, 22, 23]. It is created by the Elvira Consortium, which is a joint operation between various Spanish universities. The project was



started in 1997. In Elvira, a lot of attention has been given to explanations. Its capabilities in this area are presented in the following sections. To demonstrate certain features we will make use of the Hepar II network [25]. This is a network modeling liver disorders. The network consists of 9 disorder nodes, 18 risk factor nodes and 44 symptom, sign and laboratory test result nodes. Figure 3.5 shows a screenshot of Elvira with the Hepar II network loaded.

### 3.6.1 Static explanations in Elvira

Elvira offers various static explanations.

#### Verbal explanation of the complete network

Elvira is capable of generating a verbal explanation of a complete network. This explanation relies on the user to classify the nodes of the network into various categories like diseases, symptoms, signs and tests. Part of the explanation of the Hepar II network is the following:

The disease / anomaly PBC can be produced by the next RISK FACTORS:

Age, Sex, Pressure in right upper quadrant,  
It may have the following DISEASES / ANOMALIES:  
Carcinoma,

SYMPTOMS:

Hepatic encephalopathy, Musculo-skeletal pain,

SIGNS:

Itching, Jaundice, Yellowing of the skin, Impaired consciousness,  
Increased liver density, Joints swelling, Haemorrhagic diathesis,

The following tests help to discard or confirm its presence:

ESR, Antimythochondrial antibodies, Total bilirubin, Total  
cholesterol, Blood urea, GGTP, LE cells, Platelet count

#### Verbal explanation of a link

Figure 3.6 shows the dialog holding a verbal explanation of a link. The explanation of the link between the risk factor *Obesity* and the disease *Hepatic steatosis* is shown. From top to bottom this dialog first states the source and destination nodes of the link. After that the kind of relation is shown, which has to be specified when defining the model. The final piece of information consists of likelihood ratios. The first textbox<sup>3</sup> mentions the ratio between the two states of the node Obesity when the value of the node Hepatic steatosis is “present”. In this case it is greater than one, which means that the state “present” of Obesity has a stronger influence on the “present” state of Hepatic steatosis than the “absent” state has. The second textbox presents the same information, but for the other state of Hepatic steatosis, which is “absent”.

<sup>3</sup>Elvira is a Spanish project and some text has not been translated into English.

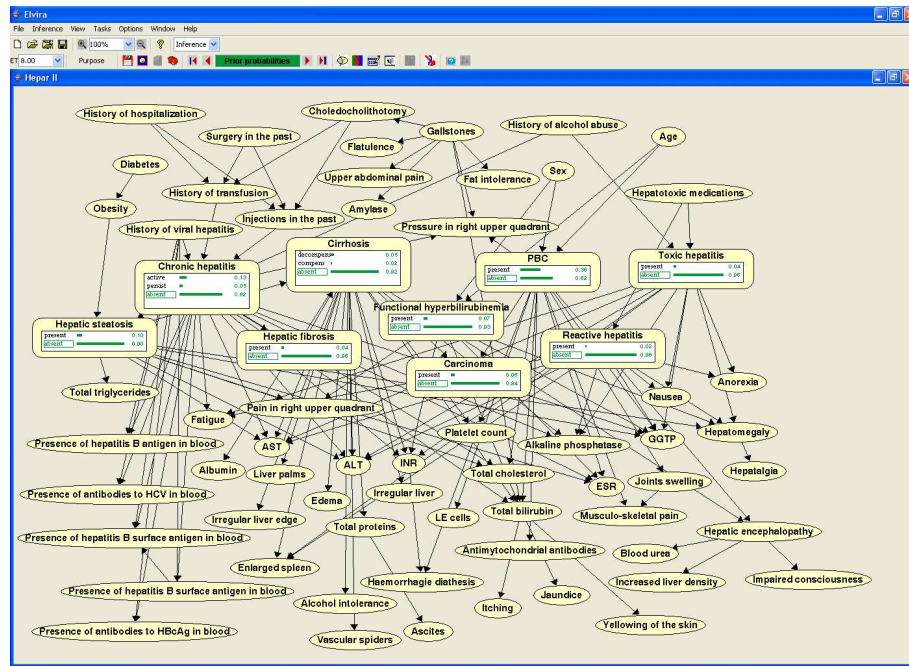


Figure 3.5: Elvira displaying the Hepar II network.

**Explicación del enlace**

Source: Obesity      Destina...: Hepatic steatosis

Kind of Relation: Favorece

Likelihood Ratio

"present": La R. V. de los estados de Obesity es: MAYOR QUE 1, por lo que el estado "present" lo explica mejor que el e En concreto, 2.44 veces mejor

"absent": La R. V. de los estados de Obesity es: MENOR QUE 1, lo que implica que el estado "absent" lo explica mejor que el es En concreto, 0.86 veces mejor

View Link Properties      Close

Figure 3.6: Verbal explanation of the link connecting Obesity to Hepatic steatosis.

### Coloring of links

Elvira can color the links in a network according to the influence of the corresponding parent and child nodes. Figure 3.7 shows an example. Links that are colored red indicate a *positive* influence, links that are colored blue indicate a *negative* influence, links that are colored black have no or *null* influence and links that are colored purple have an ambiguous or *unknown* influence. Furthermore, the width of the link indicates the relative strength of the influence. For this to work the nodes have to be ordinal. So the various states of all the nodes must be ordered in the same way, for example from good to bad, from large to small, from high to low or from desirable to not desirable. This is a necessity to determine whether an influence is positive or negative. An influence is said to be positive if, given two variables  $A$  and  $C$  where  $A$  is a parent of  $C$  and  $B$  represents the set of other parents of  $C$ , higher values of  $A$  lead to higher values of  $C$ , for every possible configuration of  $B$ . More specifically, an influence is positive if:

$$\forall a_i \forall a_j \forall b, a_i > a_j \Rightarrow Dist(C|a_i, b) > Dist(C|a_j, b),$$

where

$$\begin{aligned} Dist(C|a_i, b) > Dist(C|a_j, b) &\iff \\ &\{[\forall c, P(C \geq c|a_i, b) \geq P(C \geq c|a_j, b)] \\ &\wedge [\exists c, P(C \geq c|a_i, b) > P(C \geq c|a_j, b)]\} \end{aligned}$$

and

$$P(C \geq c|a_i, b) = \sum_{k \geq i} P(C = a_k|b).$$

What this means is that the probability distribution of  $C$  given  $a_i$  is higher than given  $a_j$  if the cumulative probability is greater or equal for every value of  $C$  and that there exists at least one value of  $C$  for which it is greater.

In causal networks most of the links are generally positive.

### Thickness of links

Elvira can also adjust the thickness of links in a network proportional to the amount of influence the corresponding parent node has on the child node. i.e., the influence in the direction of the link. An example can be seen in Figure 3.7. The magnitude of the influence transmitted by a link from node A to node B is defined as:

$$max_k [max_i [P(B \geq b_k|a_i) - P(B \geq b_k|a_0)]] .$$

### Importance factors

Each node in a network can be given an *importance factor*. This is a number ranging from zero to ten and controls whether a node is expanded or not in inference mode. When a node is expanded it is drawn as a rectangle showing information about its probability and when a node is not expanded only its

name is shown in an oval shape. In Figure 3.5 the node “PBC” is expanded, while the node “Pressure in right upper quadrant” is not. In inference mode a global importance threshold can be set. If a node’s importance factor is greater or equal to that threshold it is expanded.

### 3.6.2 Dynamic explanations in Elvira

Elvira is able to handle multiple sets of findings. Such a set of findings is called a *case*. If a node is expanded there are different colored bars to represent the probabilities, one for each case. Two cases can be seen in Figure 3.8, the green one is the prior case in which no findings are present and the red one is the case containing the findings “*Sex=female*”, “*Yellowing of the skin=present*” and “*Musculo-skeletal pain=absent*”. These nodes are colored grey to indicate that they have been observed. The following sections elaborate on Elvira’s dynamic explanation capabilities.

#### Explanation of a node

Figure 3.9 shows a verbal explanation of the node “PBC”. The dynamic part consists of the two textboxes displaying the probability ratios. The topmost textbox displays the ratio between the prior probabilities while the bottom textbox displays the ratio between the posterior probabilities.

Elvira is also capable of coloring all the nodes in the network according to the change they suffered. An example is shown in Figure 3.10. Nodes that are colored red have had a positive change, those colored blue have had a negative change, purple nodes have changed in an undefined way and yellow nodes have had no change at all. The intensity of the color indicates the magnitude of the change. The changes can be relative to the prior probabilities of the nodes or to the probabilities of the nodes according to another case.

#### Explanation of a case

For explaining a particular case Elvira offers the dialog box as shown in Figure 3.11. This dialog shows the name of the case, the findings associated with it and the probability of this set of findings occurring together. After this the user can select a target variable which is in this case “PBC”. When a variable is selected, the prior and posterior probabilities of that node are shown, together with a third measure which is the result of a limited sensitivity analyses:

$$P(target|findings) = \log \frac{P(target|findings)}{P(target)}.$$

Furthermore there are two buttons, one labeled ‘Why?’ and the other labeled ‘How?’. The following two sections will elaborate on this.

#### Summary of findings

The ‘Why?’ button mentioned in the previous section produces a summary of the influence of the various findings when they are added to the set of findings sequentially. A screenshot can be seen in Figure 3.12. The way to interpret this is as follows: there are only findings listed in the “positive” box, so every finding

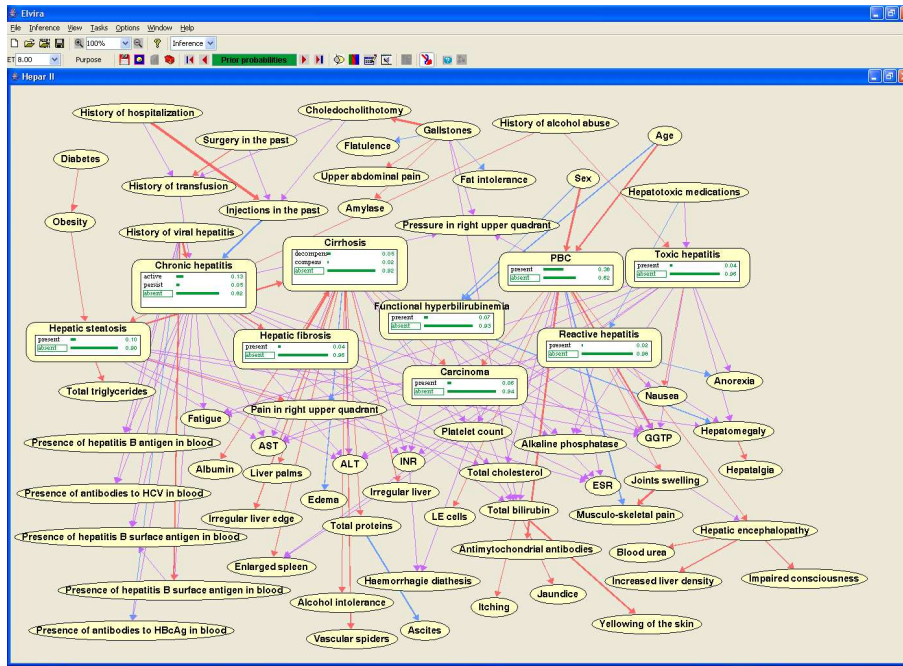


Figure 3.7: Elvira displaying colored links.

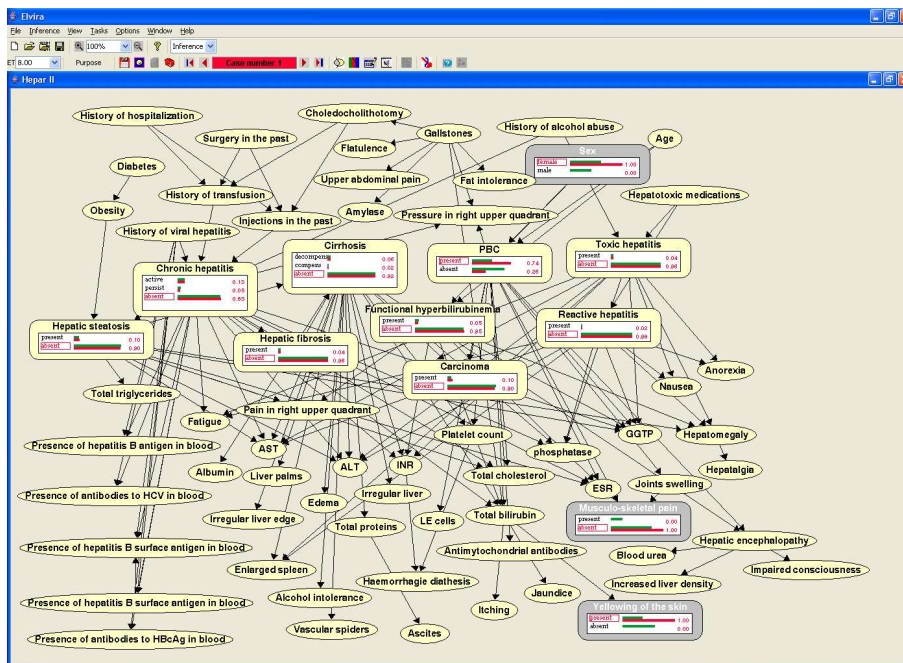


Figure 3.8: Elvira displaying a case.

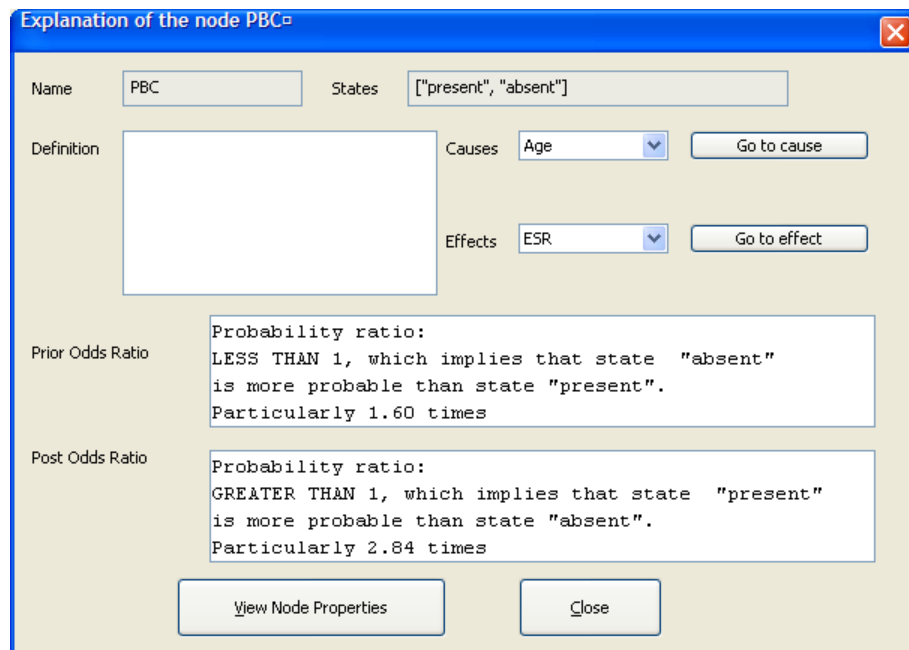


Figure 3.9: Elvira displaying a verbal explanation of a node.

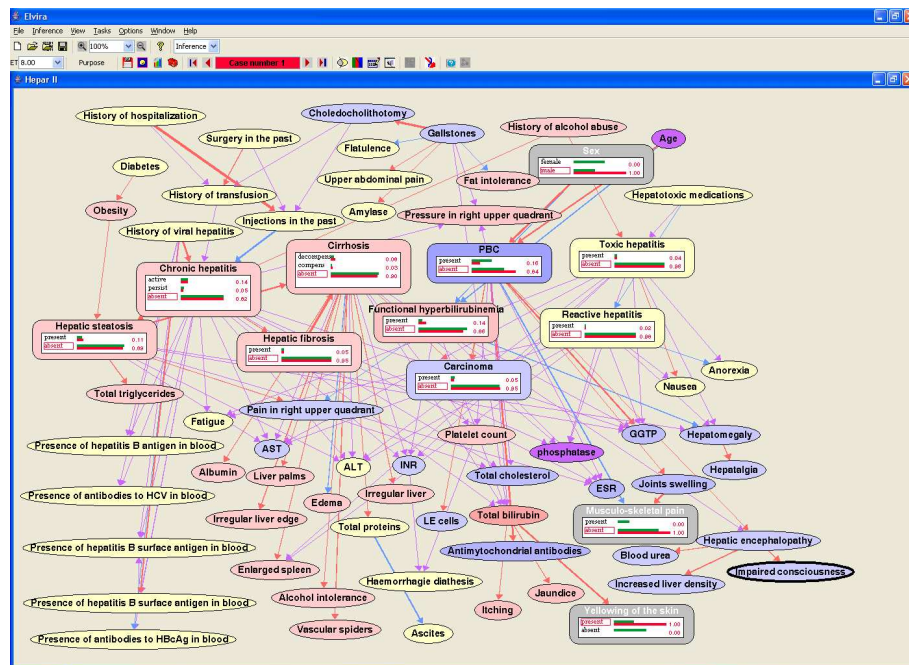
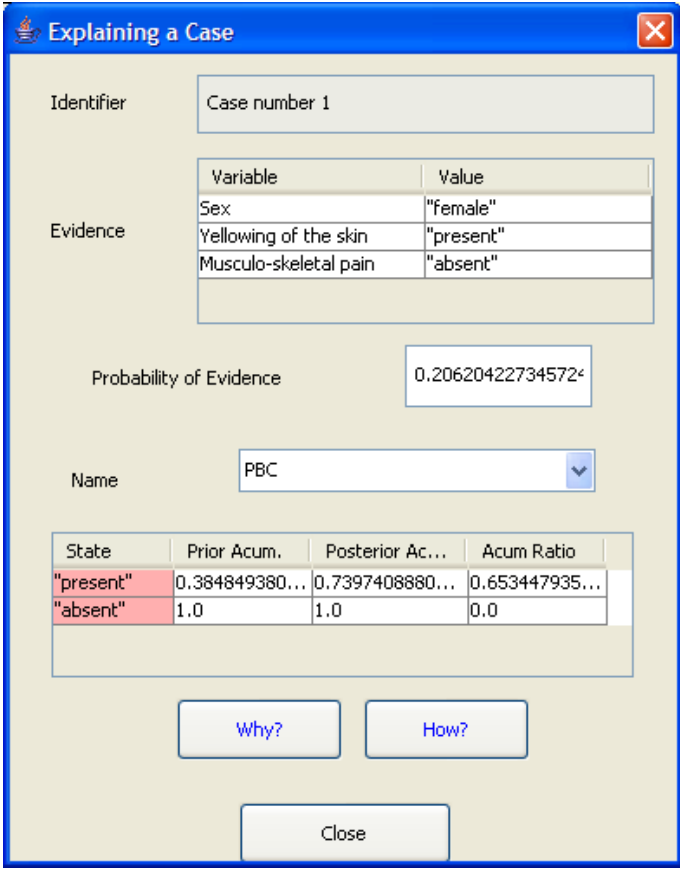


Figure 3.10: Elvira displaying colored nodes.



The dialog box titled "Explaining a Case" contains the following elements:

- Identifier:** A text field containing "Case number 1".
- Evidence:** A table with two columns: "Variable" and "Value".

Variable	Value
Sex	"female"
Yellowing of the skin	"present"
Musculo-skeletal pain	"absent"
- Probability of Evidence:** A text field containing the value 0.206204227345724.
- Name:** A dropdown menu currently showing "PBC".
- Table:** A table with four columns: "State", "Prior Acum.", "Posterior Ac...", and "Acum Ratio".

State	Prior Acum.	Posterior Ac...	Acum Ratio
"present"	0.384849380...	0.7397408880...	0.653447935...
"absent"	1.0	1.0	0.0
- Buttons:** "Why?", "How?", and "Close".

Figure 3.11: Elvira displaying an explanation of two cases.

has a positive influence on the target variable “PBC”. The value that is mentioned after each finding is the difference between (1) the posterior probability of “PBC” when the set of findings consists of that particular finding and each finding listed above it and (2) the (posterior) probability of “PBC” when the set of findings consists of *only* the findings listed above that particular finding. So the value indicates how much influence that particular finding has on the target variable in comparison with the other findings.

### Chains of reasoning

Elvira uses the INSITE method of Suermondt [31] as discussed in Section 3.3 to generate *chains of reasoning*. These can be accessed by pressing the ‘How?’ button shown in Figure 3.11. The result can be seen in Figure 3.13. Only the paths from the findings to the target variable “PBC” are shown, the rest of the network is made invisible. Nodes are colored according to their change in probability. Again red for positive changes, blue for negative ones, purple for undefined changes and yellow if the probability has remained the same. Given a variable  $V$  with values  $v_1, \dots, v_s$  and one or more findings  $e$  a change is positive if:

$$\forall k \in \{1 \dots s\} P(C \geq v_k | e) > P(C \geq v_k)$$

and

$$\exists l \in \{1 \dots s\} P(C \geq v_l) < P(C \geq v_l | e)$$

Links between variables are colored in the same way as described in Section 3.6.1.

## 3.7 BayesiaLab

BayesiaLab<sup>4</sup> is a commercial software package for the creation and evaluation of Bayesian networks. A trial edition can be downloaded and its explanation capabilities are reviewed in the next sections. The current version at the time of writing is version 4.0. Figure 3.14 shows BayesiaLab displaying the Hepar II network, with the node “PBC” set as the target variable and containing the findings “*Sex=female*”, “*Yellowing of the skin=present*” and “*Musculo-skeletal pain=absent*”.

### 3.7.1 Arc analysis

BayesiaLab can adjust the thickness of the arcs proportional to the strength of the probabilistic relations that they represent in the global probability law. This function can be used to locate the most influential arcs in a model. Figure 3.15 is an example. The thickness of an arc is based on the Kullback-Leibler divergence [20] between the joint probability distribution *with* and *without* the arc. This is a measure of dependence of two variables. If we consider the simple two node network of Figure 3.16, the importance of the arc is determined by comparing  $P(LC, S) = P(LC|S) \cdot P(S)$  with  $P'(LC, S) = P(LC) \cdot P(S)$ , where

<sup>4</sup><http://www.bayesia.com/>



**Análisis de la evidencia en PBC**

**Clasificación de los hallazgos según la influencia que ejercen**

<i>Positiva</i>	<i>Negativa</i>
Yellowing of the skin= "present" --> 0.14,  Sex= "female" --> 0.18,  Musculo-skeletal pain= "absent" --> 0.02	[]

<i>Nula</i>	<i>Desconocida</i>
[]	[]

OK

Figure 3.12: Elvira displaying influence of findings.

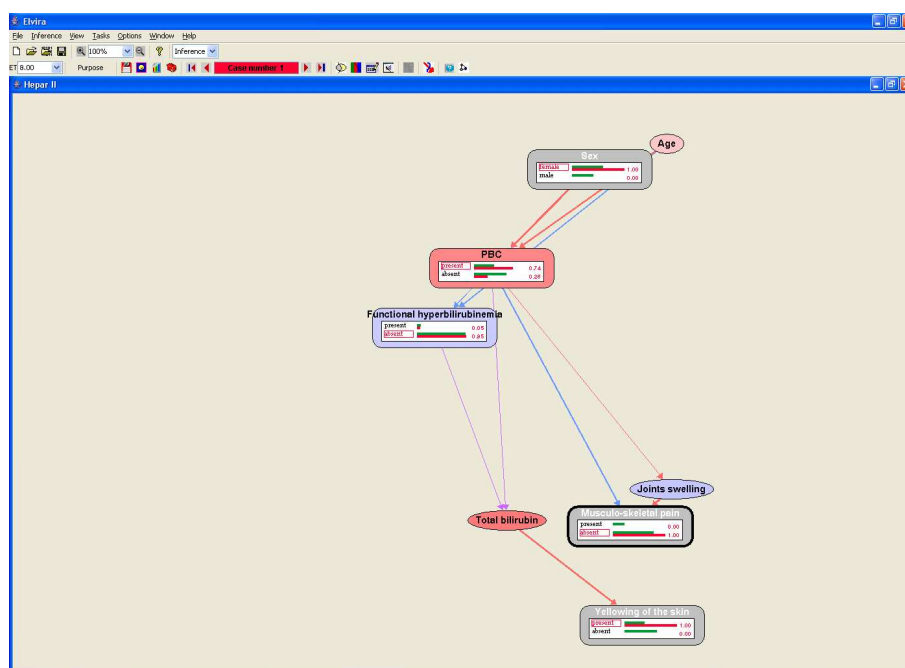


Figure 3.13: Elvira displaying chains of reasoning.

$P'(LC, S)$  is the joint probability without the arc and  $P(LC) = \sum_S P(LC, S)$ . In this example  $P(LC, S)$  becomes

$$P(LC, S)$$

	s	$\neg s$
lc	0.04	0.0008
$\neg lc$	0.16	0.7992

and  $P'(LC, S)$  becomes

$$P'(LC, S)$$

	s	$\neg s$
lc	0.008	0.032
$\neg lc$	0.192	0.768

The Kullback-Leibler divergence between these two distributions is used as an indication of the importance of the arc. In a network, the arc with the highest importance is given the thickest arc possible and the thickness of other arcs is determined relative to the thickest one. This means that the thickness of the arcs can only be used to draw conclusions within one single network, not between different networks. The Kullback-Leibler divergence of the thickest arc in one network can be very different from the arrow with that same thickness in a different network.

### 3.7.2 Target node analysis

BayesiaLab can perform a *target node analysis*, which allows the user to see how much each node contributes to the current probability distribution of the target node. Within each related node a rectangle is drawn, of which the brightness indicates the amount of influence. The brighter the higher the influence. Figure 3.17 is an example.

### 3.7.3 Target state analysis

BayesiaLab can also perform a *target state analysis*, which allows the user to see two things: (1) the type of influence a node has by looking at the symbol inside the node, which summarizes the evolution of the conditional probability of the target state of the target node with respect to each state of the node and (2) the relative contribution of each node on the target state by looking at the brightness. Figure 3.18 is an example.

### 3.7.4 Target analysis report

BayesiaLab can generate a *target analysis report*. An example is shown<sup>5</sup> in Figure 3.19. From top to bottom, it starts with showing the set of findings that was present when the report was generated. After that it mentions the posterior probabilities of the target node, in this case “PBC”. Next follows a list showing all the nodes related to the target variable together with the mutual information and the relative contribution they have with respect to the target variable. The report ends with various statistics for every related node for each state of the target node.

<sup>5</sup>The report has been shortened to fit the page.

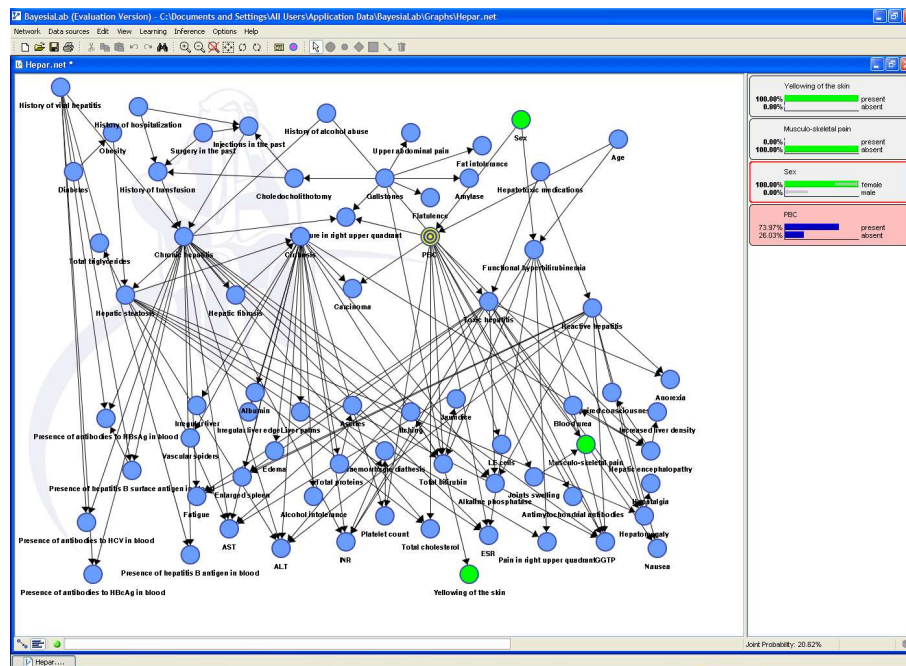


Figure 3.14: BayesiaLab displaying the Hepar II network.

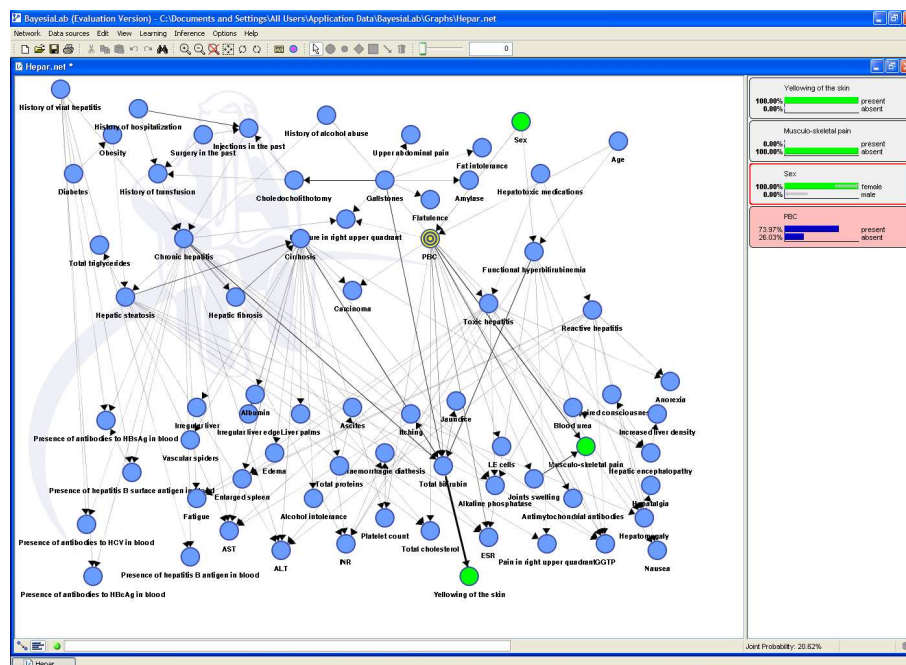


Figure 3.15: BayesiaLab after arc analysis.

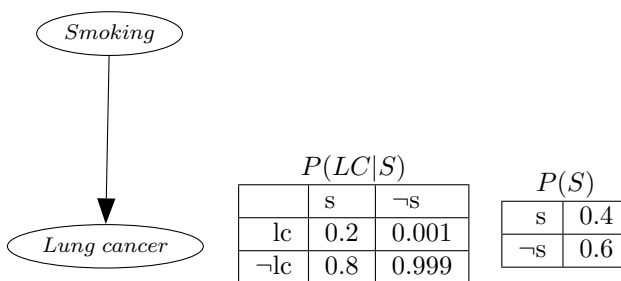


Figure 3.16: *Smoking* (S) is said to influence the risk of *Lung cancer* (LC).

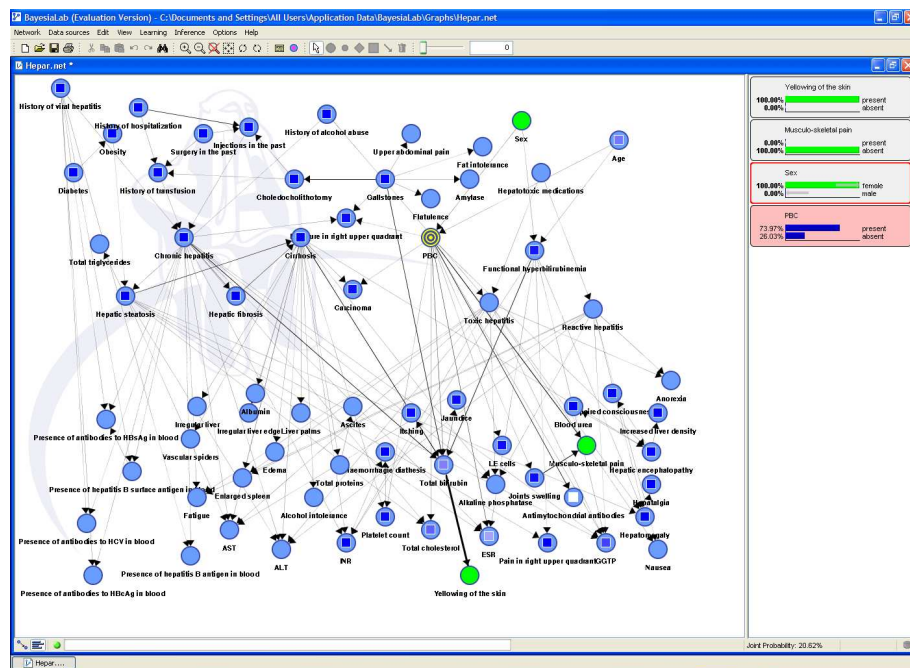


Figure 3.17: BayesiaLab after target node analysis.

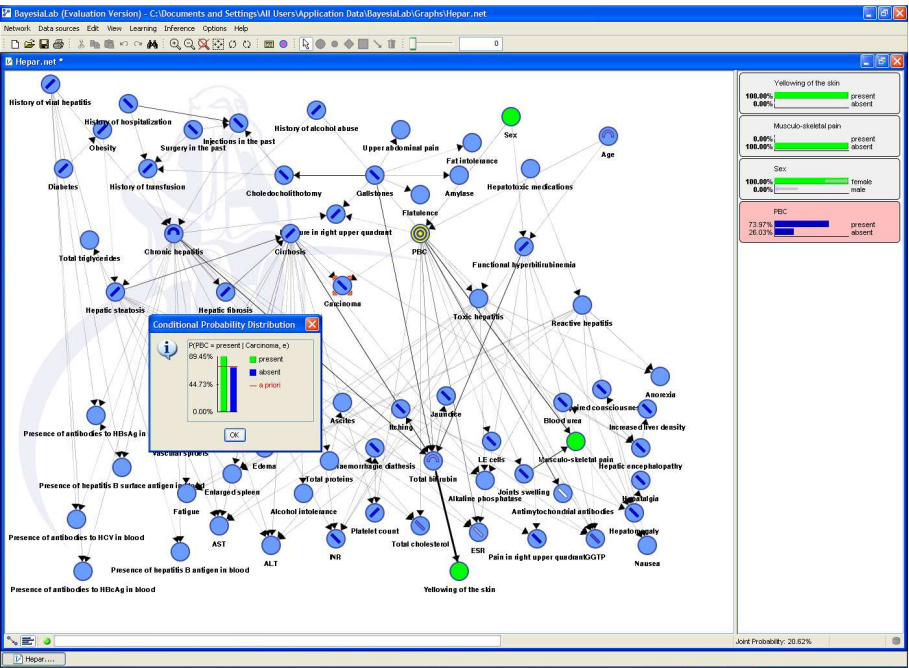


Figure 3.18: BayesiaLab after target state analysis.

Analysis Context

Sex	female
Yellowing of the skin	present
Musculo-skeletal pain	absent

Marginal Probabilities

present 73.97%  
absent 26.03%

Node significance with respect to the information gain brought by the node

Node	Mutual information	Relative significance
Antimythochondrial antibodies	0.2287	1.0000
ESR	0.1472	0.6437
⋮		
History of transfusion	0.0000	0.0000
Surgery in the past	0.0000	0.0000

Node significance with respect to the information gain brought by the node to the knowledge of the target value

PBC = present (73.97%)											
Node	Binary mutual information	Binary relative significance	Modal Value	A priori modal value	Variation	Maximal Positive Variation	Maximal Negative Variation				
Antimythochondrial antibodies	0.2287	1.0000	present	56.79%	absent	57.68%	present	14.47%	absent	14.47%	
ESR	0.1472	0.6437	a200_50	40.81%	a14_0	51.19%	a200_50	9.22%	a14_0	12.52%	
⋮											
History of transfusion	0.0000	0.0000	absent	83.77%	absent	83.77%	0.0001	absent	0.01%	present	0.01%
Surgery in the past	0.0000	0.0000	absent	57.63%	absent	57.64%	-0.0002	present	0.01%	absent	0.01%
PBC = absent (26.03%)											
Node	Binary mutual information	Binary relative significance	Modal Value	A priori modal value	Variation	Maximal Positive Variation	Maximal Negative Variation				
Antimythochondrial antibodies	0.2287	1.0000	absent	98.81%	absent	57.68%	0.7765	absent	41.12%	present	41.12%
ESR	0.1472	0.6437	a14_0	86.79%	a14_0	51.19%	0.7616	a14_0	35.60%	a200_50	26.19%
⋮											
History of transfusion	0.0000	0.0000	absent	83.75%	absent	83.77%	-0.0003	present	0.02%	absent	0.02%
Surgery in the past	0.0000	0.0000	absent	57.66%	absent	57.64%	0.0006	absent	0.03%	present	0.03%

Figure 3.19: Target analysis report for node “PBC”.

### 3.7.5 Evidence analysis report

For an instantiated target variable BayesiaLab can produce an *Evidence analysis report*. An example is shown in Figure 3.20. After listing the current findings a value is presented that indicated whether or not there is contradicting evidence. This value is defined as  $\log_2 \frac{p(e_1) \dots p(e_i)}{p(e_1, \dots, e_i)}$ , so the logarithm of the product of the marginal probabilities of each of the findings, divided by the joint probability of all the findings. When this number is negative the joint probability is greater than the product of the marginal ones which means that the findings support the same conclusion and do not contradict.

Finally, there is a listing of all the findings and how they influence the instantiated target variable. There are three possibilities. A finding either has a positive influence on it, a negative influence or it has no influence, in which case it is neutral.

### 3.7.6 Relationship analysis report

A *relationship analysis report* computes the Kullback-Leibler divergence for each link in the network. This measure indicates the difference between the probability distribution of the network with the arc *present* and that of the network with the arc *not present*, as explained earlier in Section 3.7.1.

### 3.7.7 Influence paths

According to the d-separation criterion, BayesiaLab is able to display the paths from a particular node to the target node. It generates a listing of all the paths together with their lengths and it is able to visualize one path at a time in the graph by coloring the links belonging to the path purple. An example is shown in Figure 3.22.

## 3.8 Discussion

We have reviewed various theories and programs that focus on making the reasoning process more clear to a user.

The usefulness of abduction, as discussed in Section 3.1, is questionable. The probability of the most probable configuration of the unobserved variables is in many cases very low, it merely shows what the most likely states of all the variables are.

The scenario based explanations of Section 3.2 are more interesting. Such an explanation, presented in natural language, is easy to read and explains why a certain variable is in a certain state. But to get the most out of such an explanation, the user still has to know *why* a certain conclusion is drawn, for example by looking at the network and following the various steps in the explanation.

The INSITE method, discussed in Section 3.3, tries to explain which observations have influenced a certain target variable and to what extent. It also determines the paths through which the relevant findings influence the target variable, the so called “chains of reasoning”. The result is a very clear and insightful explanation of why a certain target variable has been influenced in a certain way. A user can see which observations have the largest influence on

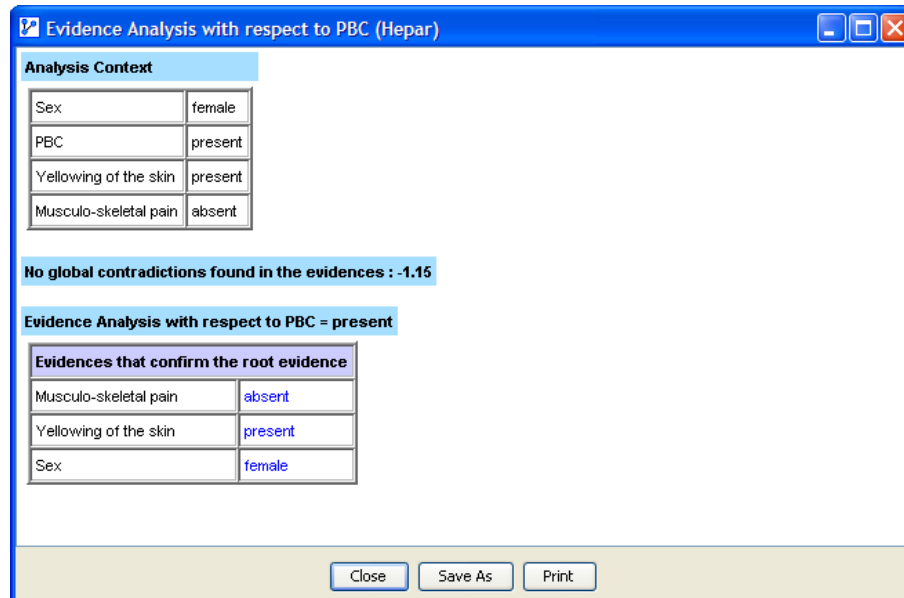


Figure 3.20: Evidence analysis report with respect to node “PBC”.

Analysis Context			
Sex	female		
Yellowing of the skin	present		
Musculo-skeletal pain	absent		

Relationship Analysis			
Parent	Child	Kullback-Leibler divergence	Relative Weight
Total bilirubin	Yellowing of the skin	0.178212	1.0000
Joints swelling	Musculo-skeletal pain	0.060640	0.3403
PBC	Musculo-skeletal pain	0.060640	0.3403
Gallstones	Cholecholethotomy	0.055398	0.3109
Chronic hepatitis	Total bilirubin	0.053129	0.2981
Cirrhosis	Total bilirubin	0.052587	0.2951
Functional hyperbilirubinemia	Total bilirubin	0.051745	0.2904
Gallstones	Total bilirubin	0.049398	0.2772
⋮			
Reactive hepatitis	Enlarged spleen	0.000023	0.0001
History of viral hepatitis	Presence of antibodies to HCV in blood	0.000015	0.0001

Figure 3.21: Relationship analysis report.

the target variable, and the paths through which they reach the target variable. Only the most relevant observations are included in the explanations. Observations that have little or no impact on the posterior probability distribution of the target variable are discarded.

The tutoring shell BANTER of Section 3.4 is based on the INSITE method. It was made for use in the medical domain, to assist in diagnosis. It can list the most influential findings and the paths they flow through. The system only generates verbal explanations, which is somewhat primitive and limiting.

The work of Madigan, discussed in Section 3.5, essentially does the same as the INSITE method, but uses a different approach. Its two aspects are also finding the most influential findings, and determining the paths through which they flow. The generated “flows of evidence” are visualized in such a way that easily can be seen where a potential “bottleneck” of evidence transmission is located. A big disadvantage, though, is the fact that it only works on polytrees or Berge networks. This limitation immediately excludes many practical networks, unless such a network is somehow transformed to a Berge network, but this is not a trivial process and therefore not desirable.

The software package Elvira (Section 3.6) incorporates many forms of explanations, both verbal and graphical. The verbal explanations come down to descriptions about the various nodes and their type. For this to work all nodes must be classified. The classification, in conjunction with the network structure, is used to build up a verbal description of the network, in a causal way. Other verbal explanations use likelihood ratios, saying that one state is, for example, “3.77” times more likely than some other state. The graphical capabilities are probably the best part of Elvira. Colors are used throughout the program to indicate the direction of change of probability distributions of nodes. Also, Elvira is able to show the “chains of reasoning” by using the INSITE method, along with coloring of nodes and links to indicate the changes in probability. Overall, Elvira delivers quite nice explanations, with the graphical part being more useful than the verbal part. The program has quite a few bugs that result in strange behaviour, though, but that does not take away anything from the good explanations.

Finally, BayesiaLab, discussed in Section 3.7, also features quite a few options that help a user understand what is going on in a model. The most interesting parts make use of a target node, which the user has to set. Various statistics can then be generated. BayesiaLab makes good use of the graphical representation of the network by augmenting it with various symbols to signify changes and characteristics and by adjusting the thickness of the arcs to indicate the contribution of that arc to the current situation of the network. Besides this BayesiaLab can generate various statistical reports. These reports can be useful, but the user has got to have very good knowledge of the workings of a Bayesian network in order to interpret all the figures in the correct way. The graphical explanations of BayesiaLab are more accessible, but still have a slight learning curve because the symbols used are sometimes not that intuitive, especially not at first sight.



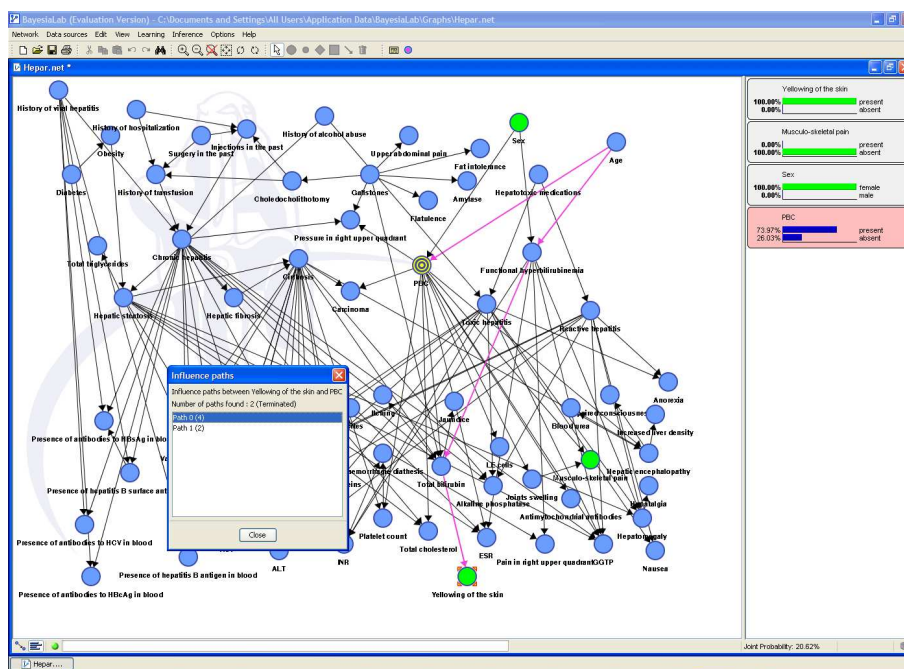


Figure 3.22: BayesiaLab showing an influence path from “Yellowing of the skin” to target “PBC”.



## Chapter 4

# General design

After reviewing previous works and discussing the characteristics, the first thing we decided was to focus on graphical explanations and not so much on verbal ones. This is because we believe that a graphical explanation has a lot more explanatory power. A good visualization can deliver information much faster and in a more convenient way than a verbal explanation can. Reading takes time, while the same knowledge can often be deduced by looking at a visualization for only a few seconds. We want to create something that is easy for a user to use and interpret. If the techniques demand very little from the user, but deliver valuable extra information about the network, we believe that the usefulness will be maximized. In the following sections the ideas we found to be interesting are introduced.

### 4.1 Ancestors and descendants

A simple but potentially very useful feature would be to easily see, for a certain node, what its parents and children are. In a complicated network it is not always clear which nodes are connected to a certain node, especially not at a glance. More general, we think it would be useful to graphically point out the ancestors and/or descendants of a node. The user might be able to say how many generations of ancestors or descendants he or she wishes to see. For example, the parents are the ancestors of generation one and the children are the descendants of also generation one. A rough impression of what this could look like in GeNIe is shown in Figure 4.1.

### 4.2 Paths of influence

Inspired by the feature of BayesiaLab discussed in Section 3.7.7, we are interested in showing the user the paths of influence from a chosen variable to a certain target variable. The paths along which a certain variable impacts another variable are determined using the *d-separation* criterion, as introduced in Section 2.2. Paths can be opened up and blocked by the observed evidence. The approach of BayesiaLab visualizes one path at a time, but we think that this is useless most of the time, because the number of paths can easily be very large. To have thousands of paths is not an exception. Therefore, we propose

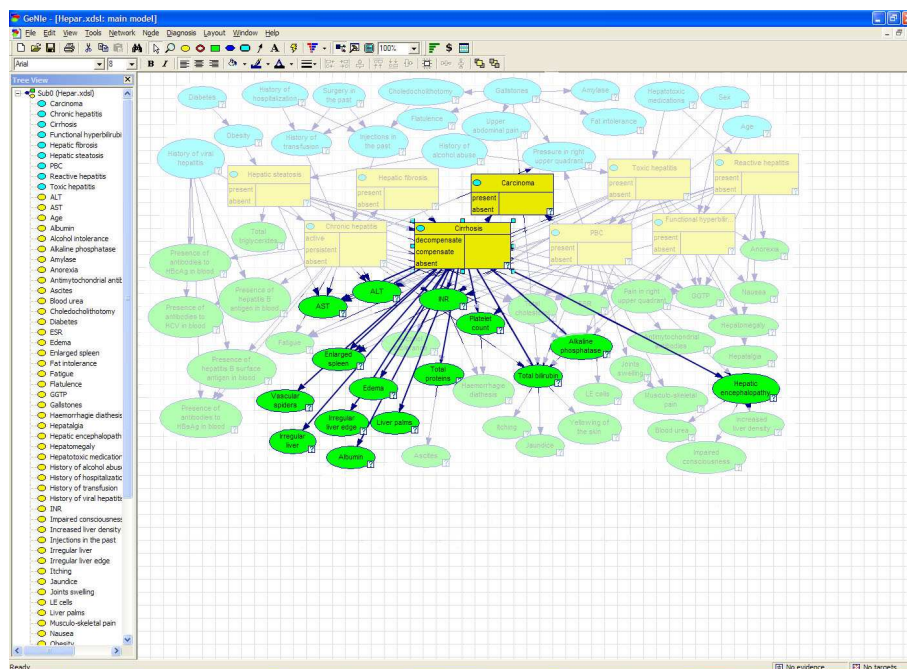
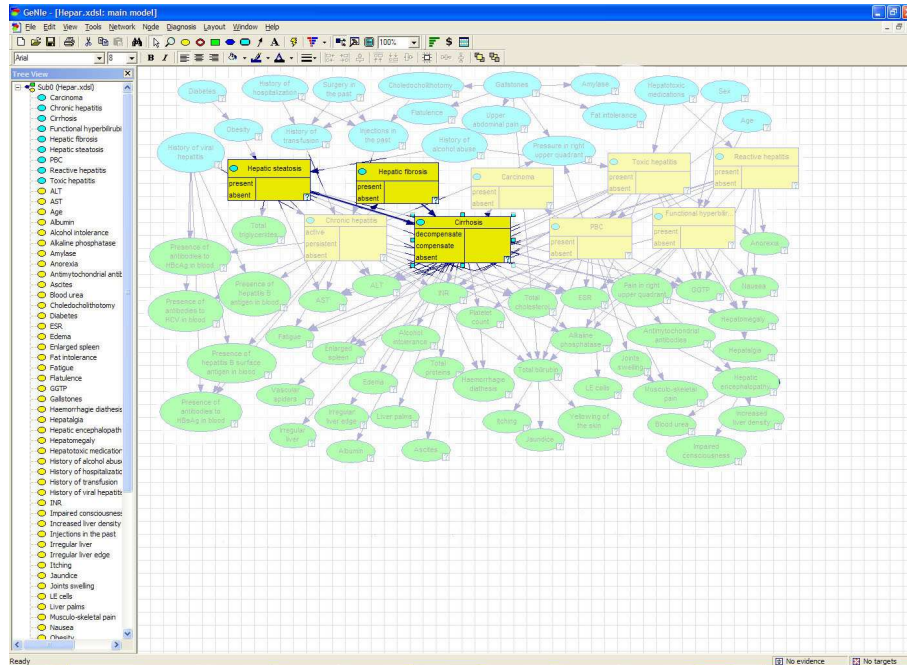


Figure 4.1: Rough impression of showing ancestors and descendants.

to visualize the subgraph that contains all the nodes and arcs that are on some path from the variable of interest to the target variable. This gives less precise information about the individual paths, but if there are many paths it becomes undoable for a user to inspect each path separately. A user can instead look at the complete subgraph containing all paths.

We can use the INSITE method, see Section 3.3, to find the most influential paths and display those, possibly individually, to a user. This makes more sense, because there are usually only a few paths that influence a target variable significantly.

### 4.3 Markov blanket

The Markov blanket of a node in a Bayesian network is the set of nodes composed of the parents of that node, its children and the children of its parents. The Markov blanket is an interesting concept, because it “shields off” the node from the rest of the network. Given a node  $A$ , its Markov blanket  $Markov(A)$ , the node is independent from the rest of the network:

$$P(A|Markov(A), B) = P(A|Markov(A)) , \quad (4.1)$$

where  $B$  is some set of other nodes in the network. This means that the nodes that are part of the Markov blanket of a certain node, are the only nodes needed to predict the behaviour of that node. We can visualize the Markov blanket the same way as we did the ancestors and descendants of Section 4.1, i.e., greying out everything but the Markov blanket of the desired node.

### 4.4 Relevance of findings

Both the INSISTE method (Section 3.3) and the method of Madigan (Section 3.5) are able to show which findings are most significant for a certain target variable, and are able to indicate if those findings conflict or agree with the overall inference result. We think this is an interesting statistic to be able to show to a user. How we can show this in the most effective way has to be researched.

### 4.5 Multiple cases

A case is a combination of evidence and inference results. In the GeNIe program, introduced in Section 2.7, being able to display multiple cases at once allows for easy comparison. At this time, only one set of observations and inference results can be shown. If another piece of evidence is observed, the previous inference results are lost. An intuitive way to display and work with more than one case has to be developed. We can also generate interesting statistics for the user, such as the amount of change between the active cases for a certain variable. Ways to visualize this have to be explored.

## 4.6 Thickness of arcs

In many networks it can be of value to know to what extent two directly connected nodes can influence each other. This information can be visualized by automatically varying the thickness of the arc connecting the nodes. It is quite intuitive to draw a thicker arc when the influence is strong. Two reviewed earlier works, Elvira (Section 3.6) and BayesiaLab (Section 3.7), use the thickness of the arcs to visualize influence. Elvira uses the conditional probability tables in the definition of the model to determine the thicknesses of the arcs. This gives a good view of how two directly connected nodes interact, but it does not take into account any observations or indirect influences, the information is completely local. BayesiaLab uses the thickness of the arcs to visualize the current role of that arc in the current situation of the network. What we are going to do is use the thickness of the arcs to visualize the *potential* influence that two directly connected nodes have on each other, i.e., what the effect of observing one node would have on the other. We refer to this new method as a *dynamic* method, as opposed to the previously mentioned static one. This method does take into account any observations and indirect influences. This part is discussed in detail in Chapter 5.

## 4.7 Color of arcs

We are going to visualize the sign of influence between nodes by varying the color of the arcs. When we consider two directly connected nodes, then, if the probability distribution of one of the nodes somehow changes, for example by observing that node, the probability distribution of the other node can also change because of that. If that change in distribution is always in the same direction, we can visualize that by giving the arc a certain color. The arc then indicates that a change in a certain direction of the probability distribution of one node, will always cause a change in a certain direction of the probability distribution of the other node. This can also be done in a static way, and in our new dynamic way, just like the thickness of arcs. This part is discussed in detail in Chapter 6.

## 4.8 Discussion

The best possibilities, we think, lie in somehow augmenting the visual representation of a Bayesian network with extra information. Ultimately, we have decided to focus on the arcs in a network and to focus on the ideas of Section 4.6 and 4.7. We think that using the arcs in a Bayesian network to represent additional information is intuitive and easy to understand and interpret. Besides this, we think that we can contribute the most to this field of research with our dynamic approach. Chapter 5 treats the thickness of arcs, and Chapter 6 treats the color of arcs.

## Chapter 5

# Thickness of arcs

This chapter deals with the thickness of arcs. First, we will introduce ordinal nodes. Second, the design and strategy for a regular Bayesian network will be detailed. Third, the method will be extended to influence diagrams. After that we will explore and discuss various ways to measure differences in discrete probability distributions.

### 5.1 Ordinal nodes

In order to realize our goals we have to introduce the concept of *ordinal nodes*. In a Bayesian network, the various states that a node has are not required to be ordered in any way. But if we do have nodes with ordered states, which is often the case in a diagnostic network, we can take this extra information into account when determining the amount of influence between the two nodes connected by the arc in question, and to generate an explanation indicating if the influence is positive or negative. See for example Figure 5.1. It shows a network consisting of two nodes, *Excercise* and *Body type*. They are drawn as rectangles, but they are regular chance nodes, not decision nodes. Both have three states and are ordinal. In this case the states of both of the nodes are in increasing order. In each of the subfigures *Excercise* is observed. In Figure 5.1(a) the value *none* is observed, in Figure 5.1(b) *some* is observed and in Figure 5.1(c) *often* is observed. When a higher value of *Excercise* is observed the probabilities of the higher values of *Body type* also increase. In this case we would say that *Excercise* positively influences *Body type*.

### 5.2 Strategy

The information that we want to provide for a user by varying the thickness of arcs is the amount of influence one node has on the other. The approach by BayesiaLab, as described in Section 3.7.1, uses the joint probability distribution, while Elvira (see Section 3.6.1) determines the influence by looking at the conditional probability tables and determining the influence a parent node has on a child node. This approach to determine the influence of a parent node on a child node is static. This means that the calculations do not take into account any current observations. But it could be that, with certain observations, the

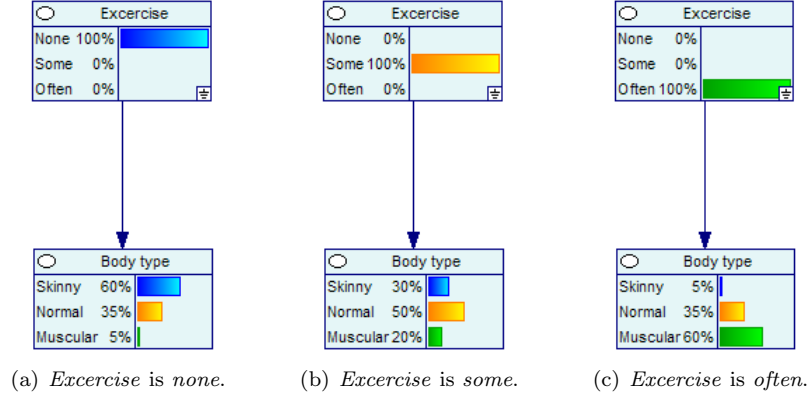


Figure 5.1: Example of ordinal nodes.

influence of a certain parent node on a child node is significantly different from the observation-free situation, in which case the static information would be incorrect. The static information can be used to get a global impression of the interactions between the nodes, but it is not tailored to a certain situation.

Besides that, while it is true that a parent influences its child(ren) if it is observed, a child, when observed, can also influence the probability distribution of its parent(s). These two influences can be quite different from each other. To give an example, if a laptop is dropped from a high building, we almost know for certain that it will end up getting smashed into many pieces. But if we find a laptop that is smashed into many pieces, we cannot be just as sure about what caused this. It could have been dropped from a high building, but it could just as well have been run over by a car, or maybe someone got angry and stamped on it. So while the probability of “dropped from a high building” will increase when finding a smashed laptop, the probability will not increase as much the other way around, i.e., that of “laptop will get smashed” when we drop it from a high building.

Therefore, we are going to do this differently. Many networks contain one or more *target* or *hypothesis* nodes. See for example the Hepar II network [25] shown in Figure 5.2, modeling various liver disorders. The yellow colored nodes represent the diseases. What we would be interested in most is the influence that the other nodes have on these disease nodes. Such nodes are often called *target* nodes. When an arrow connects a target node with a non-target node, we will determine the influence the non-target node has on the target node, regardless of the direction of the arrow. When two non-target or two target nodes are connected by an arrow, we will, by default, use the average of the influences in both directions. In total there are four situations possible, shown in Figure 5.3. In the first situation, shown in Figure 5.3(a), we will visualize the influence *A* has on *B*. In the second situation, that of Figure 5.3(b), we will consider the influence that *B* has on *A*. In the last two situations, depicted in Figures 5.3(c) and 5.3(d), we will consider the influence in both directions, and average them. The user will have the ability to override any of these default actions by specifying in which direction the influence for a particular arrow should be calculated.



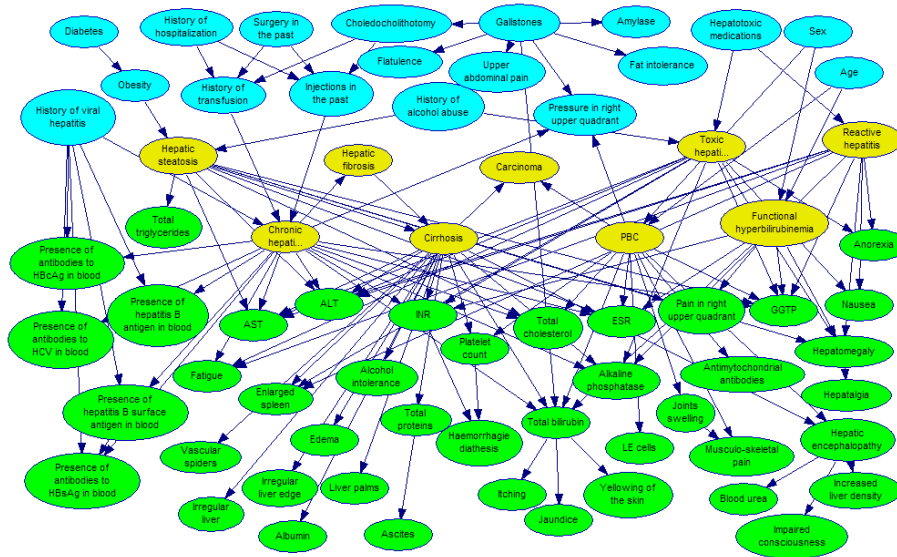


Figure 5.2: The Hepar II network.

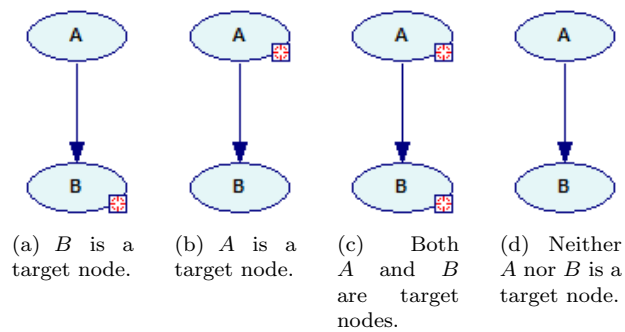


Figure 5.3: Four different situations.

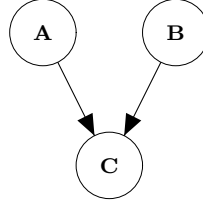


Figure 5.4: A small network.

Table 5.1: A conditional probability table for the network of Figure 5.4. Finding  $A$  additionally to  $B$ , or vice versa, will have little effect on  $C$ .

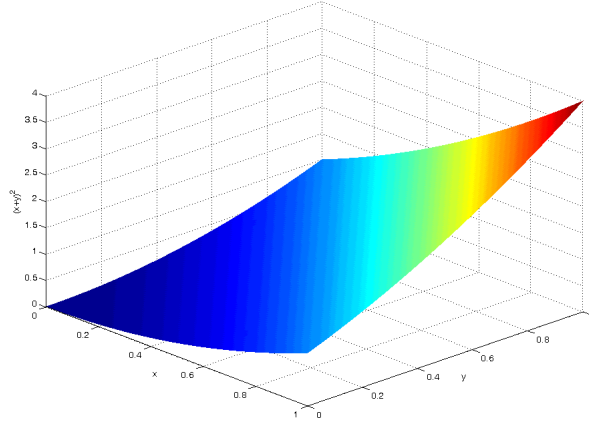
	a		$\neg a$	
	b	$\neg b$	b	$\neg b$
c	0.99	0.98	0.98	0.01
$\neg c$	0.01	0.02	0.02	0.99

Furthermore, the approach that we are proposing here is a dynamic one. It considers the network in its current state, including any observations. It essentially indicates how much potential influence a node has on a direct successor or predecessor, so the influence that a node could have if it was observed next. This can give a lot more insight in a situation. For example, consider Figure 5.4, which shows a simple network consisting of two parent nodes,  $A$  and  $B$ , with a common child node,  $C$ , and  $C$  is a target node. With no observations, let us say the potential influence that  $A$  has on  $C$  is bigger than the potential influence that  $B$  has on  $C$ . If we observe  $B$ , this could have such an impact on the posterior probability distribution of  $C$  that observing  $A$  additionally to  $B$  would mean little or no difference. This means that the potential influence of  $A$  on  $C$  has diminished. The conditional probability table of Table 5.1 demonstrates this situation. All the nodes have two states: true and false. If  $A$  is true and  $B$  is false, then additionally finding that  $B$  is true does not have a big impact on the probability that  $C$  is true. In our approach, when either  $A$  or  $B$  is observed, the thickness of the arc will be recalculated to reflect the new situation.

Another advantage of our approach as opposed to the static one is the fact that in some situations it does not need to account for the *synergy* between the different parents of a node with more than one parent, simply because it is not there anymore in those situations. The definition of synergy can be given as:

Table 5.2: Another conditional probability table for the network of Figure 5.4. The combined effect of the parents is much greater than the sum of their individual effects.

	a		$\neg a$	
	b	$\neg b$	b	$\neg b$
c	0.99	0.1	0.1	0.01
$\neg c$	0.01	0.9	0.9	0.99

Figure 5.5: Example of a synergy between  $x$  and  $y$ .

“the interaction of two or more agents or forces so that their combined effect is greater than the sum of their individual effects”. In case of a Bayesian network, this applies to the combined effect that the observation of multiple parents of one node can have on that node. The combined effect can be greater than the individual effects. Table 5.2 gives an example of this. Looking at the table we can see that when both  $A$  and  $B$  are false, the probability of  $C$  is 0.01. When either  $A$  or  $B$  is true, the probability of  $C$  increases to 0.1. But when both  $A$  and  $B$  are true, the probability of  $C$  increases much more, to 0.99. Another way to understand this is to look at Figure 5.5, which visually shows a synergy between two variables  $x$  and  $y$ , both ranging from 0 to 1, defined as  $(x + y)^2$ . When only one of the two variables approaches 1 the value is not as high as when both  $x$  and  $y$  approach 1.

This phenomenon cannot be accurately captured by varying the thickness of the arcs, which is one dimensional. In our dynamic approach, though, when all but one of the parents or children of a certain node are observed, i.e., there is no synergy anymore, we are able to accurately display the actual situation, because we are considering potential influences in the current state of the network. As soon as there is a change in the network, for example another observation is done, the thickness of an arc is recalculated if necessary.

We are going to determine the strength of the influence by looking at the posterior probability distribution of a node, for each possible state of the parent or child node, depending on the type of connection as discussed earlier in Figure 5.3. For a node with  $n$  states, this will result in  $n$  potentially different posterior probability distributions of the connected node(s). We will compute the amount of difference between these distributions and base our final determination of the thickness of the arc on either the average of all the differences, the maximum of all the differences, or the weighted average. The weighted average is defined as

$$\sum_{i=0}^n a_i \cdot D(P(A), P(B|A = a_i)) , \quad (5.1)$$

where  $A$  and  $B$  are two directly connected nodes,  $A$  has  $n$  states and  $D$  is a

function measuring the distance between two distributions.

How we will calculate the difference between two distributions will be discussed in Section 5.5.

This method implies that for each of the states of each node that is a parent or a child, the probabilities of all of its direct predecessors and successors have to be updated. If a network has  $N$  nodes and each node has  $n$  states this would require, in the worst case,  $N \cdot n$  updates of the network. This leads to a complexity of  $O(N)$ . Performing inference itself has been proven to be NP-hard [5], but with the current state and speed of Bayesian updating algorithms this procedure will, for most practical networks, be completed within seconds.

### 5.3 Thickness in influence diagrams

Up to now, we have only considered networks consisting of general chance nodes. We are going to extend our method so that it can also determine the thickness of the arcs in an influence diagram in a meaningful way. An influence diagram, as discussed in Section 2.3, is a Bayesian network augmented with decision and value nodes.

#### 5.3.1 Decision nodes

If there are one or more decision nodes in a network, they have an impact on our proposed strategy to determine the influence between two directly connected nodes in two ways: (1) we ofcourse have to define how to calculate and interpret the thickness of an arc if one or both nodes are decision nodes, but (2) we also have to update our strategy for determining the influence between other types of nodes. Let us first elaborate on the latter.

##### Impact on other nodes

A decision node, in its unobserved state, introduces multiple posterior probability distributions, or expected utilities if the node is a value node, for all of its descendants. For each possible decision each descendant has a single posterior probability distribution or expected utility. Or, in case there are multiple unobserved decision nodes, there is a posterior probability distribution or expected utility for each possible combination of decisions. One such a combination of possible decisions is also known as a *policy*. See for example the network of Figure 5.6. This is the same network as used to introduce influence diagrams in Section 2.4, please refer to that section for an explanation of the model and its nodes. Figure 5.7 shows the values of the nodes *Success of the venture* in Figure 5.7(a), *Expert forecast* in Figure 5.7(b) and value node *Financial gain* in Figure 5.7(c), after performing inference. We can see that the values of the nodes are indexed by the possible outcomes of the decision node(s) that precede them. The values of all three nodes are indexed by the possible outcomes of decision node *Sensitivity*, which has three possible outcomes: *low*, *nominal* and *high*. Node *Financial gain* is, besides by node *Sensitivity*, also indexed by decision node *Investment decision*. This clearly impacts our strategy to determine the influence between two directly connected nodes, because we have to take care that we compare the right distributions with each other. Remember that, when

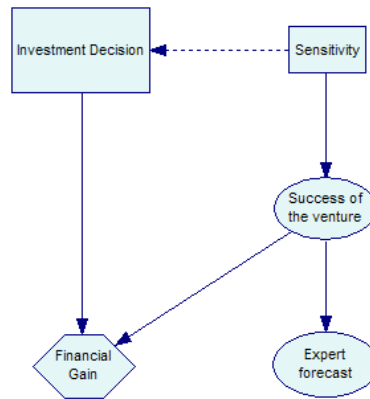


Figure 5.6: An influence diagram, none of the nodes are observed.

Probability distributions for different policies:			
Sensitivity	Low	Nominal	High
Success	0.1	0.2	0.35
Failure	0.9	0.8	0.65

(a) Probability distributions for node *Success of the venture*.

Probability distributions for different policies:			
Sensitivity	Low	Nominal	High
Good	0.13	0.16	0.205
Moderate	0.31	0.32	0.335
Poor	0.56	0.52	0.46

(b) Probability distributions for node *Expert forecast*.

Expected utilities for different policies:						
Sensitivity	Low		Nominal		High	
	Invest	DonotInvest	Invest	DonotInvest	Invest	DonotInvest
Investment De...						
Exp. utility	-3500	500	-2000	500	250	500

(c) Expected utilities for value node *Financial gain*.

Figure 5.7: Probability distributions for the network of Figure 5.6.

determining the influence a parent has on a child, we are comparing the value of the child when the parent is *unobserved*, with every other value of the child, one for each *observed* state of the parent. If we refer to this first value as the *prior* value of the child and to all the other values, when the parent is observed, as the *posteriors*, we can distinguish three possible situations, listed in Table 5.3. One situation is missing, i.e., the posterior is indexed and the prior is not indexed. This is because this situation is impossible. A node cannot have a prior value that is not indexed and have a posterior value that is indexed. If the prior value is not indexed the posterior value will also not be indexed.

The first situation shown in Table 5.3 requires no special treatment. There are no decision nodes in the network or all decision nodes have been observed, i.e., all decisions have been made.

The second situation can occur when, due to the observation of a node, the node we are considering gets d-separated from the decision node(s). This

Table 5.3: Possible situations when comparing a prior and a posterior value.

Situation	Prior	Posteriors
1	not indexed	not indexed
2	indexed	not indexed
3	indexed	indexed

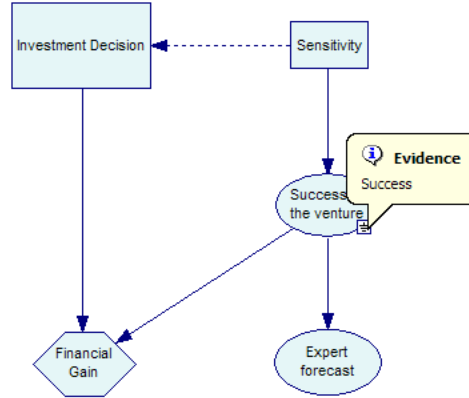
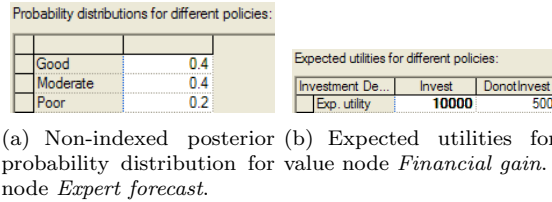
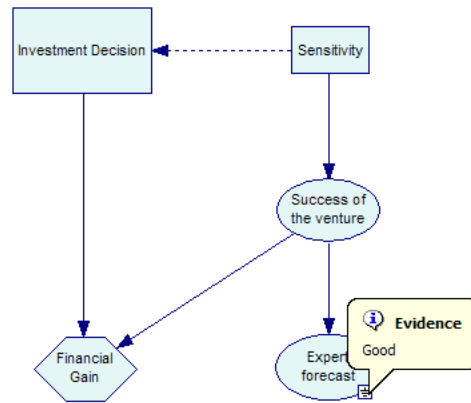
Figure 5.8: Influence diagram with node *Success of the venture* observed.

Figure 5.9: Posterior values for the network of Figure 5.8.

happens when we are determining the influence of *Success of the venture* on *Expert forecast*. Figure 5.8 shows the influence diagram with node *Success of the venture* observed. The posterior probability distribution of node *Expert forecast* for that situation is shown in Figure 5.9(a). When compared to that of Figure 5.7(b), we can see that node *Sensitivity* no longer indexes the value of node *Expert forecast*, as opposed to the case when *Success of the venture* is not observed. If this happens we are going to compare the single posterior probability distribution or the single expected utility to each possible distribution or expected utility in the prior.

The third and final situation can itself be split up in two possibilities: one where the posterior value has exactly the same indexing as the prior value, and one where the indexing is different because one or more, but not all, indexing parents have become d-separated from the node. If we are determining the influence that *Expert forecast* has on *Success on the venture*, as depicted in Figure 5.10, we can see that the value of node *Success on the venture* in this situation, given in Figure 5.11, has exactly the same policies as the one we are comparing it with, which is the one shown in Figure 5.7(a). We are going to compare each posterior probability distribution with the prior probability distribution of that exact same policy. When both the prior and posterior values are indexed, but differently, we have to loosen this approach somewhat. For instance, if we are determining the influence that *Success of the venture* has on the value node *Financial gain*, we are going to observe *Success of the venture*, as illustrated in Figure 5.8, and check the effect on *Financial gain*. The expected utilites of node *Financial gain* after observing *Success of the venture*,

Figure 5.10: Influence diagram with node *Expert forecast* observed.

Probability distributions for different policies:				
Sensitivity	Low	Nominal	High	
Success	0.307692	0.5	0.682927	
Failure	0.692308	0.5	0.317073	

Figure 5.11: Posterior probability distribution for node *Success of the venture*.

are shown in Figure 5.9(b). These will have to be compared with the expected utilities shown in Figure 5.7(c). We can see that they are both indexed, but that the posterior expected utilities are only indexed by the possible outcomes of *Investment decision*, while the prior expected utilities are also indexed by the possible outcomes of node *Sensitivity*. In this case, when the indexing parents differ, we are going to compare each posterior expected utility or probability distribution to those prior expected utilities or probability distributions that, for each indexing parent of the posterior, have those same indexing parents in the same state. For example, the expected utility of 1000 in Figure 5.9(b), belonging to the policy *Investment decision=Invest*, is compared to the expected utilities  $-3500$ ,  $-2000$  and  $250$  in Figure 5.7(c).

Table 5.4 summarizes the approaches for each of the three situations.

Table 5.4: Possible situations and approaches when comparing a prior and a posterior value.

Situation	Prior	Posteriors	Approach
1	not indexed	not indexed	Simply compare both values.
2	indexed	not indexed	Compare single expected utility or probability distribution of the posterior to each one of the prior.
3	indexed	indexed	Compare each expected utility or probability distribution of the posterior to those of the prior for which the indexing parents of the posterior match those of the prior.

Table 5.5: Possible situations and approaches when a decision node is involved.

Parent	Child	Approach
decision node	decision node	No influence calculated.
not a decision node	decision node	No influence calculated.
decision node	not a decision node	Calculate influence parent has on child, by comparing all posteriors to each other.

### If parent or child node is a decision node

If both parent and child nodes are decision nodes we are not going to calculate anything for that arc. Such an arc is merely there to indicate the order in which the decisions are made. It makes no sense to try and determine the influence of one decision node on the other, because a decision node is always observed, or decided, by the user, they are not influenced by any other node in the network. More generally, if the child node is a decision node, regardless of what type of node the parent is, we are not calculating any influence, simply because it is not there.

If the parent node is a decision node we are going to impose a restriction on the direction in which the influence is calculated. The direction will always be from the decision node to the other node, which can be any other type of node other than a decision node. The other way around would make no sense, because, like said before, a decision node is not influenced by anything. What a decision node actually does in this situation, when it is observed, is ruling out some possibilities in the current value of the child node, because making a decision reduces the number of policies in the network. Look, for example, at Figure 5.7(a), showing the possible probability distributions of node *Success of the venture*, one for each possible outcome of the connected decision node *Sensitivity*. If *Sensitivity* is observed, the effect on child node *Success of the venture* is simply that the current value changes into the probability distribution belonging to the observed state of *Sensitivity*. Therefore, because the posteriors are just subsets of the prior, the comparisons we need to make in this situation only involve the posterior values. We are going to compare all the posteriors against each other, instead of comparing each of them to the prior.

Table 5.5 summarizes the various situations where a decision node is involved.

### 5.3.2 Value nodes

A value node, as opposed to a chance node, is not defined by a probability distribution, but by expected utilities, one for each policy, for which holds:

$$expected\ utility \in \mathbb{R}. \quad (5.2)$$

This poses a problem because our strategy is based on comparing probability distributions, not a restriction-free combination of real numbers. To solve this we are going to use the method proposed in [4] to convert a value node to a chance node, which has a regular probability distribution. Let  $\Pi_v$  be the set of parents of a value node  $V$ , and let  $v(\Pi_v)$  be the value function mapping the different expected utilities to the various combinations of states of the parents. The transformation is then defined as follows:



Success of the venture	Success		Failure	
Investment Decision	Invest	DonotInvest	Invest	DonotInvest
Value	10000	500	-5000	500

Figure 5.12: Definition of node *Financial gain*.Table 5.6: Definition of value node *Financial gain* transformed to a chance node definition.

Success of the venture	Success		Failure	
Investment decision	Invest	DonotInvest	Invest	DonotInvest
true	1	0.37	0	0.37
false	0	0.63	1	0.63

$$P(V = T|\Pi_v) = \frac{v(\Pi_v) + k_2}{k_1}, \quad (5.3)$$

where

$$k_1 = \max_{\Pi_v}[v(\Pi_v)] - \min_{\Pi_v}[v(\Pi_v)] \quad (5.4)$$

and

$$k_2 = -\min_{\Pi_v}[v(\Pi_v)]. \quad (5.5)$$

This essentially performs a linear transformation of the expected utilities to the  $[0,1]$  range, assigning 0 to the lowest expected utility and 1 to the highest expected utility. The transformation implies that the result is a binary chance node with states *true* and *false*, so:

$$P(V = F) = 1 - P(V = T). \quad (5.6)$$

We can, for example, transform the definition of value node *Financial gain*, shown in Figure 5.12. This definition is indexed by the parent nodes *Success of the venture* and *Investment decision*. So these two nodes together form  $\Pi_v$ . Then  $k_2 = -(-5000) = 5000$  and  $k_1 = 10000 - (-5000) = 15000$ . If we then apply Equation 5.3 we get the probability distributions shown in Table 5.6. This way we can treat a value node the same way as we treat a regular chance node.

Finally, there is one more situation possible concerning value nodes. When both parent and child are value nodes, the child node becomes a *multi-attribute utility* (MAU) node. A MAU node has one or more utility nodes as its parents, and its definition is defined by weights. An example is shown in Figure 5.13. All three value nodes, *Income*, *Growth* and *Happiness* influence the MAU node *Total Satisfaction*. The weights are used to calculate the expected utility of the MAU node, which is the sum of the expected utilities of each of the parents multiplied by the corresponding weight. An example of the definition is shown in Figure 5.14. If we want to visualize influence using the thickness of the arcs, we will always want to do so in the direction of the MAU node, because the parents each have a certain part in the expected utility of the MAU node. There is no influence in the other direction. To calculate this we can simply use the

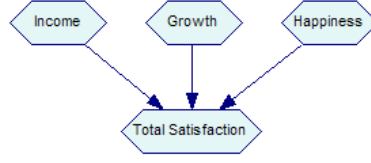


Figure 5.13: *Total Satisfaction* is an example of a *multi-attribute utility* (MAU) node.

Parents	Income	Growth	Happiness
Weights	1	2.2	5

Figure 5.14: Definition of the MAU node *Total Satisfaction* of Figure 5.13.

weights of the definition of the MAU node. We can transform a weight  $w$  to the  $[0,1]$  range in the following way:

$$w_{transformed} = abs(w) / max(abs(w_{min}), abs(w_{max})) , \quad (5.7)$$

where  $w_{min}$  is the minimum weight present in the definition, and  $w_{max}$  the maximum.

The result of that transformation for the definition of the node *Total Satisfaction* is given in Table 5.7. This result can be directly interpreted as an influence measure, so no comparisons are needed.

Table 5.8 summarizes the various situations where a value node is involved.

## 5.4 Difference between distributions

Given two discrete probability distributions, we want to know how much they differ so that we can make a good comparison between various differences and draw valid conclusions about which change is more significant than the other. In our case, there are two different situations possible, one in which the states of the distribution are ordered, and one in which they are not. If the states of the distribution are ordered, we may want to adjust our interpretation of a change of a distribution. The main thought is that, if the states are ordered from left to right, from less important to most important, the more the probability shifts from one side of the distribution to the other, the more important the change is. If we have three distributions,  $A = [1, 0, 0, 0]$ ,  $B = [0, 1, 0, 0]$  and  $C = [0, 0, 0, 1]$ , then, in the situation where the four states are not ordered in any way, the change from  $A$  to  $B$  would be considered just as big as the change from  $A$  to  $C$ . But if we know that the states are in an ascending order, then we would consider the change from  $A$  to  $B$  to be less significant than the change from  $A$  to  $C$ .

Table 5.7: Definition of MAU node *Total Satisfaction* transformed to the  $[0,1]$  range.

Parents	Income	Growth	Happiness
influence	0.2	0.44	1

Table 5.8: Possible situations and approaches when a value node is involved.

Parent	Child	Approach
value node	value node (MAU)	Use transformed definition of child.
<i>not</i> a value node	value node	Transform value to probability distribution and treat like regular chance node.
value node	<i>not</i> a value node	Impossible, a value node can only have another value node as a child.

We also need to pay attention to the way we are going to treat differences when one of the probabilities approaches either 0 or 1. Are we, for example, going to treat a difference in probability from 0.0001 to 0.01 in the same way as one from 0.71 to 0.72? We might want to consider the former a more significant increase than the latter, while in both cases the difference is about 0.01. If we imagine a weather forecaster who predicts a 71% chance of rain, while the actual chance is 72%, we would say that he did a good job. But if an expert predicts a chance of 1% of getting some serious disease, when the actual chance is just 0.01%, we would probably say that he was not very accurate. This leads us to the distinction between *absolute* and *relative* differences. An absolute difference is expressed in the same units as the two compared values. There are no units on a relative difference, they are expressed in percentages. In our example the absolute differences are 0.0099 and 0.01, respectively, while the relative differences are  $\frac{|0.71-0.72|}{0.71} \cdot 100 = 1.41\%$  and  $\frac{|0.01-0.0001|}{0.01} \cdot 100 = 99\%$ . So the absolute differences are almost equal, but the relative differences are not equal at all.

Also, we need to compare various differences to each other, which requires that our measure is *symmetric*. Symmetric means that, for a certain distance measure  $D$  and two probability distributions  $P$  and  $Q$ , the following holds:

$$D(P, Q) = D(Q, P) . \quad (5.8)$$

If we would only do comparisons from one point of view, meaning that one of the two distributions is constant during the comparisons, we could just take care that we use the same order of arguments to the distance function every time. But this is not the case. Consider, for example, the network in Figure 5.15. If we want to determine the influence in the direction of the arrows, so from parent to child, there are two influences: that of  $A$  on  $B$  and that of  $B$  on  $C$ . To determine the amount of influence we would calculate  $D(P(B), P(B|A))$  and  $D(P(C), P(C|B))$ . Now, assuming that all nodes are binary, if  $P(B) = [0.5, 0.5]$ ,  $P(B|A) = [0.9, 0.1]$ ,  $P(C) = [0.9, 0.1]$  and  $P(C|B) = [0.5, 0.5]$ , we could run into trouble when using an asymmetric measure, because essentially we are going to do the following comparison:

$$D([0.5, 0.5], [0.9, 0.1]) == D([0.9, 0.1], [0.5, 0.5]) . \quad (5.9)$$

We are required to use a symmetric measure, otherwise we could obtain two different values while we want to consider these two differences equal.

Finally, we would like to have a measure that gives us values in a certain bounded range, preferably from 0 to 1. This will provide an easy direct mapping to the thickness of an arc. We could use an unbounded measure, ofcourse, and just assign the highest difference the thickest arc, and determine the thickness

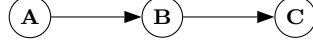


Figure 5.15: A network consisting of three nodes.

of all the other arcs relative to that thickest one. But that way we cannot, for example, compare the thickness of the arcs in two different networks with each other, because the two highest values of two networks can differ considerably and therefore a certain thickness of an arc in one network can indicate a very different amount of influence than that same thickness does in another network.

## 5.5 Distance measures

In this section we will discuss various distance measures. For this purpose we now define  $P$  and  $Q$  as two discrete probability distributions:

$$P, Q \in \left\{ (p_1, p_2, \dots, p_n) \mid p_i > 0, \sum_{i=1}^n p_i = 1 \right\}, \quad n > 1. \quad (5.10)$$

### 5.5.1 Euclidean distance

A well known measure is the *Euclidean distance*. It is defined as:

$$E(P, Q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}. \quad (5.11)$$

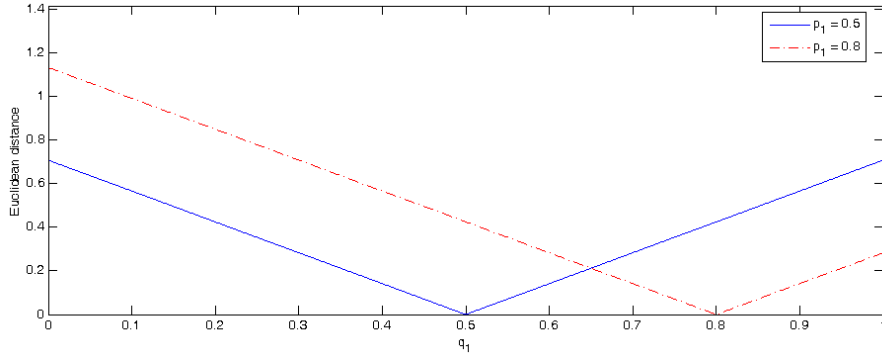
If  $P$  and  $Q$  are two points in some  $N$ -dimensional space this calculates the actual spatial distance between the two points. When used with discrete probability distributions, where the sum of all elements is always equal to one, the value of this measure ranges from 0, when there is no difference, to  $\sqrt{2}$ , the maximum difference. The Euclidean distance is a symmetric measure. Figure 5.16 shows the behaviour of the Euclidean distance in various situations. We can transform the Euclidean distance to the  $[0, 1]$  range easily:

$$E_{norm}(P, Q) = \frac{E(P, Q)}{\sqrt{2}}. \quad (5.12)$$

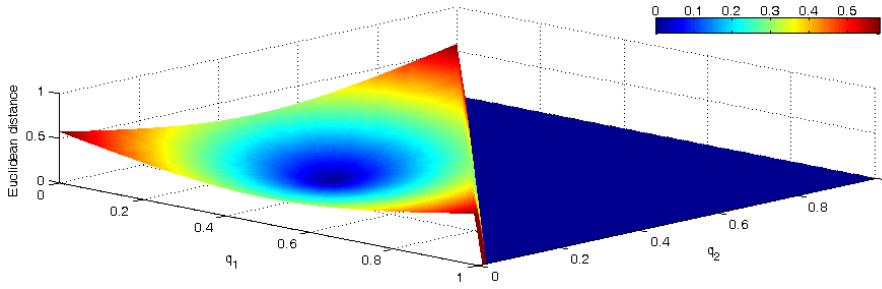
### 5.5.2 Hellinger distance

Another measure that is often used is the *Hellinger distance* [14], which is defined as:

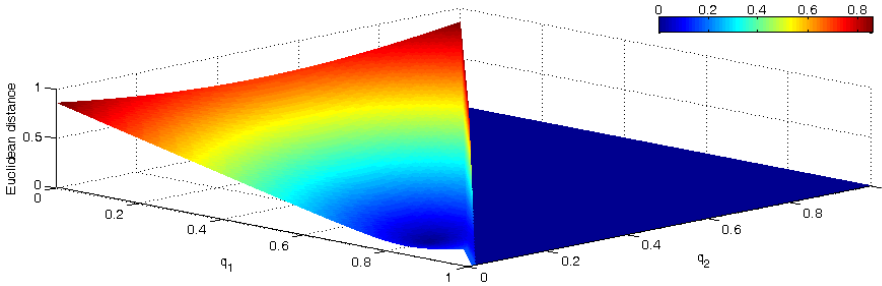
$$H(P, Q) = \sqrt{\sum_{i=1}^n (\sqrt{p_i} - \sqrt{q_i})^2}. \quad (5.13)$$



(a) Euclidean distances for the binary distribution  $P = (p_1, p_2)$  against distribution  $Q = (q_1, q_2)$  where  $p_2 = 1 - p_1$  and  $q_2 = 1 - q_1$ .



(b) Euclidean distance for the three state distribution  $P = (0.33, 0.33, 0.34)$  against distribution  $Q = (q_1, q_2, q_3)$  where  $q_3 = 1 - q_1 - q_2$ . When  $q_1 + q_2 > 1$  the probability distribution is invalid and its distance is set to 0.



(c) Euclidean distance for the three state distribution  $P = (0.8, 0.1, 0.1)$  against distribution  $Q = (q_1, q_2, q_3)$  where  $q_3 = 1 - q_1 - q_2$ . When  $q_1 + q_2 > 1$  the probability distribution is invalid and its distance is set to 0.

Figure 5.16: Euclidean distances

The value of this measure ranges from 0 to  $\sqrt{2}$ . Figure 5.17 shows the behaviour of the Hellinger distance. It can be seen that the Hellinger distance is more sensitive when approaching 0 or 1, as opposed to the Euclidean distance. The Hellinger distance is also symmetric and we can, again, easily define a linear transformation to get it to the  $[0, 1]$  range:

$$H_{norm}(P, Q) = \frac{H(P, Q)}{\sqrt{2}} . \quad (5.14)$$

### 5.5.3 Kullback-Leibler distance

The *Kullback-Leibler distance* [20], or *Kullback-Leibler divergence*, comes from the field of information theory and is given as:

$$K(P, Q) = \sum_{i=1}^n (p_i \log_2(\frac{p_i}{q_i})) . \quad (5.15)$$

It can also be written as:

$$K(P, Q) = - \sum_{i=1}^n p_i \log_2(q_i) + \sum_{i=1}^n p_i \log_2(p_i) = H(P, Q) - H(P) , \quad (5.16)$$

where  $H(P, Q)$  is the cross-entropy of  $P$  and  $Q$ , which expresses the overall difference between two distributions, and  $H(P)$  is the entropy of  $P$ , which is a measure of how much information  $P$  carries.

The value of this measure ranges from 0 to  $\infty$ . Figure 5.18 shows the behaviour of the Kullback-Leibler distance. Like the Hellinger distance, changes near 0 or 1 are treated differently than changes in other regions.

But, for our purpose, there are three problems with the Kullback-Leibler distance. First, it is not symmetric, second, its values go to infinity, and third, if a  $q_i = 0$  there is a division by zero. We will deal with these problems with the help of the next measure, the J-divergence.

### 5.5.4 J-divergence

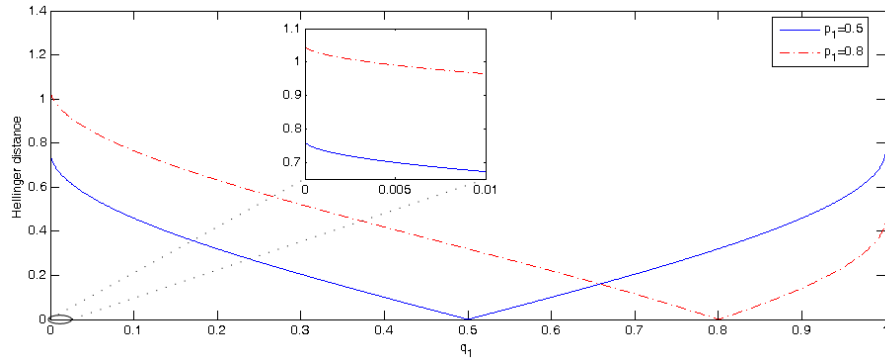
To make the Kullback-Leibler distance symmetric, we can instead choose to use the *J-divergence* [17, 18], which can be given as the average of the two possible values of the Kullback-Leibler distance:

$$J(P, Q) = \frac{K(P, Q) + K(Q, P)}{2} . \quad (5.17)$$

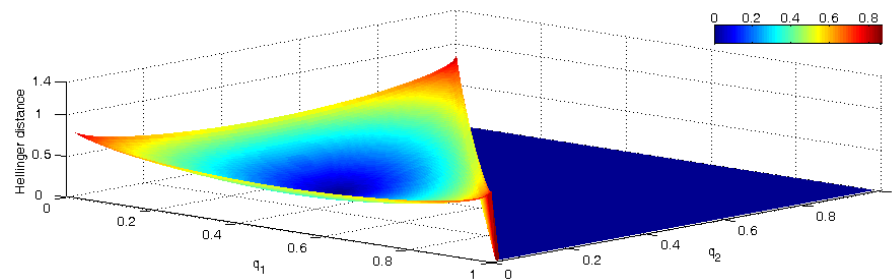
This solves the symmetry issue, but it still has values that go to infinity. To make the J-divergence range from 0 to 1 we can normalize it as follows [32]:

$$J_{norm}(P, Q) = \frac{J(P, Q)}{\sqrt{J(P, Q)^2 + \alpha}} , \quad (5.18)$$

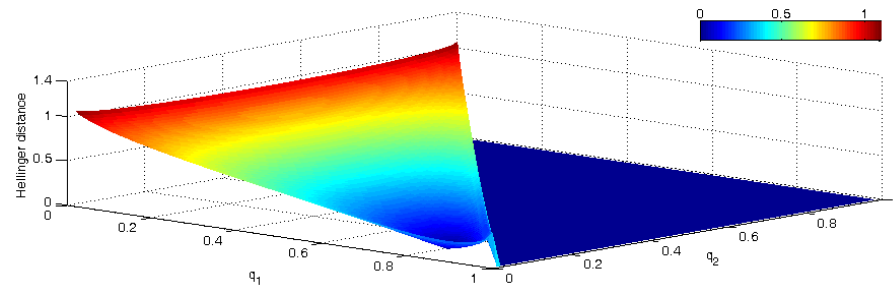
where  $\alpha \geq 0$  is a parameter controlling the “smoothness” of the normalization. Figure 5.19 shows the normalized J-divergence with  $\alpha = 10$ . If we compare this



(a) Hellinger distances for the binary distribution  $P = (p_1, p_2)$  against distribution  $Q = (q_1, q_2)$  where  $p_2 = 1 - p_1$  and  $q_2 = 1 - q_1$ .

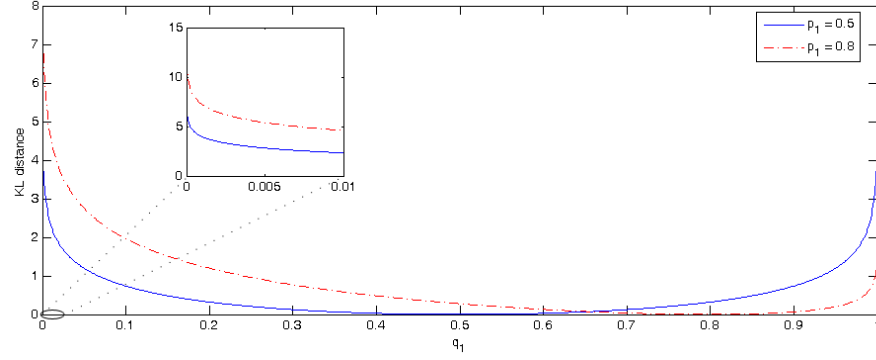


(b) Hellinger distance for the three state distribution  $P = (0.33, 0.33, 0.34)$  against distribution  $Q = (q_1, q_2, q_3)$  where  $q_3 = 1 - q_1 - q_2$ . When  $q_1 + q_2 > 1$  the probability distribution is invalid and its distance is set to 0.

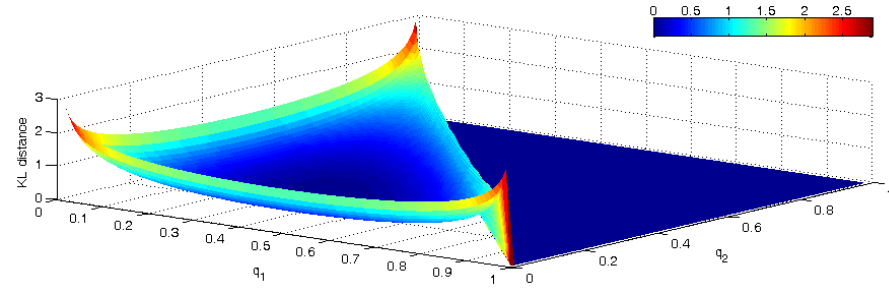


(c) Hellinger distance for the three state distribution  $P = (0.8, 0.1, 0.1)$  against distribution  $Q = (q_1, q_2, q_3)$  where  $q_3 = 1 - q_1 - q_2$ . When  $q_1 + q_2 > 1$  the probability distribution is invalid and its distance is set to 0.

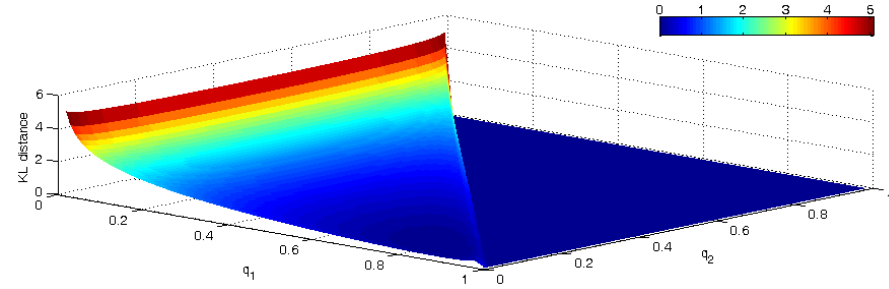
Figure 5.17: Hellinger distances



(a) Kullback-Leibler distances for the binary distribution  $P = (p_1, p_2)$  against distribution  $Q = (q_1, q_2)$  where  $p_2 = 1 - p_1$  and  $q_2 = 1 - q_1$ .



(b) Kullback-Leibler distance for the three state distribution  $P = (0.33, 0.33, 0.34)$  against distribution  $Q = (q_1, q_2, q_3)$  where  $q_3 = 1 - q_1 - q_2$ . When  $q_1 + q_2 > 1$  the probability distribution is invalid and its distance is set to 0.



(c) Kullback-Leibler distance for the three state distribution  $P = (0.8, 0.1, 0.1)$  against distribution  $Q = (q_1, q_2, q_3)$  where  $q_3 = 1 - q_1 - q_2$ . When  $q_1 + q_2 > 1$  the probability distribution is invalid and its distance is set to 0.

Figure 5.18: Kullback-Leibler distances



to Figure 5.18 we can see that we have nicely captured a very similar behaviour, but now the measure is symmetric, and the values range from 0 to 1.

To also solve the third and final problem, the possible division by zero, we can change the definition into:

$$J_{norm}(P, Q) = \begin{cases} 1 & \exists i \in (1, \dots, n), q_i = 0 \\ \frac{J(P, Q)}{\sqrt{J(P, Q)^2 + \alpha}} & \text{else} \end{cases} . \quad (5.19)$$

### 5.5.5 CDF distance

This measure is based on the method used in [19]. It is targeted towards ordinal distributions, meaning that if the states are ordered from left to right, from less important to most important, the more the probability shifts from one side of the distribution to the other, the more important the change is. It compares the cumulative distribution functions (CDF) of the two distributions to be compared. We have generalized this measure so that it can be used as a distance measure between two probability distributions. It is defined as:

$$C(P, Q) = \frac{1}{n-1} \sum_{i=1}^n |P(P \leq p_i) - P(Q \leq q_i)| . \quad (5.20)$$

The range of this measure is from 0, when there is no difference, to 1, the maximum difference. We will illustrate how this measure works by means of an example. Say we have three discrete probability distributions:  $P = [0, 0, 1, 0]$ ,  $Q = [0.1, 0, 0, 0.9]$  and  $R = [0, 0.1, 0, 0.9]$ . Each has four states, labeled, from lowest to highest, as: *none*, *low*, *medium*, *high*. The CDF for each of these distributions is shown in Figure 5.20.

If we want to compare  $P$  to  $Q$  as well as  $R$  we can combine them as shown in Figure 5.21. Intuitively, the difference between  $P$  and  $Q$  is greater than that between  $P$  and  $R$ , because in  $Q$  the probability of the first state increases when compared to  $P$ , while in  $R$  the probability of the second state increases. The probability of  $P$  is concentrated in the third state, so an increase in the first state is more important than an increase in the second state, because the second state is located directly next to the third state, while the first state is located one state away to the left. This is reflected in Figure 5.21. We are considering the size of the non-overlapping parts as the difference between the two distributions. So for Figure 5.21(a) we get, by applying equation 5.20:

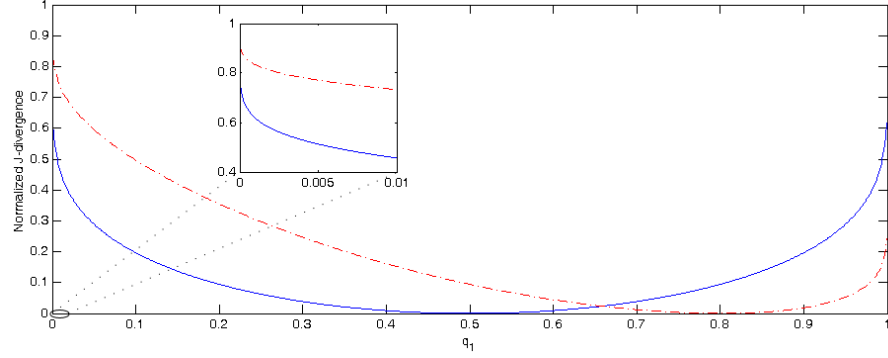
$$\frac{1}{4-1} (0.1 + 0.1 + 0.9 + 0) = \frac{11}{30} \approx 0.37 .$$

For Figure 5.21(b) it is:

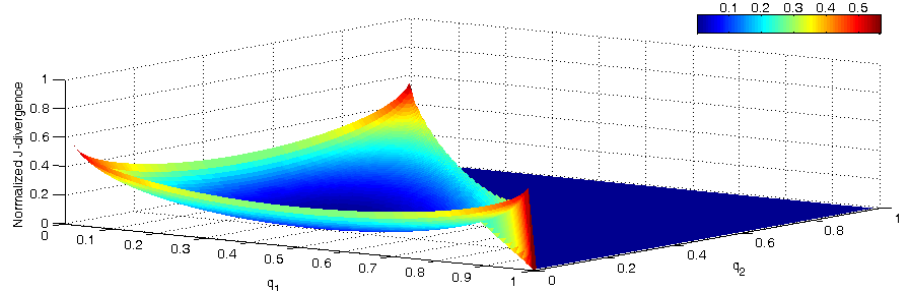
$$\frac{1}{4-1} (0 + 0.1 + 0.9 + 0) = \frac{10}{30} \approx 0.33 .$$

We see that the difference between  $P$  and  $Q$  indeed is greater than that between  $P$  and  $R$ , because  $0.37 > 0.33$ . It is only a small difference of 0.04, but that is because the difference between the two actually is pretty small.

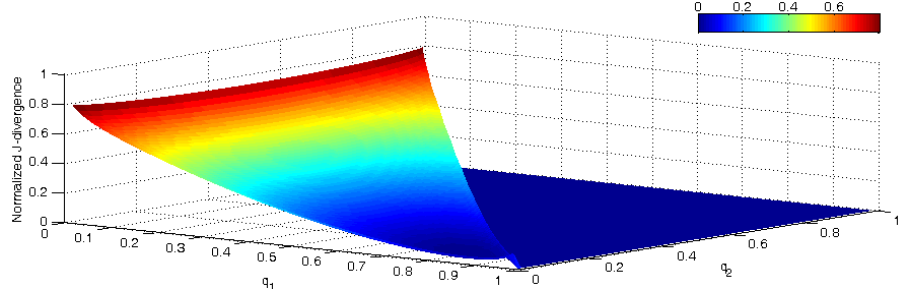
Figure 5.22 shows the behaviour of the CDF distance. If we look at the extremes of Figure 5.22(b), we can see that when  $[q_1, q_2, q_3] = [0, 1, 0]$  the difference with  $[0.33, 0.33, 0.34]$  is less than when  $[q_1, q_2, q_3] = [0, 0, 1]$  or  $[q_1, q_2, q_3] =$



(a) J-divergence normalized with  $\alpha = 10$  for the binary distribution  $P = (p_1, p_2)$  against distribution  $Q = (q_1, q_2)$  where  $p_2 = 1 - p_1$  and  $q_2 = 1 - q_1$ .

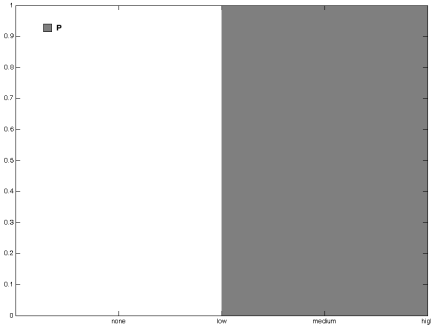


(b) J-divergence normalized with  $\alpha = 10$  for the three state distribution  $P = (0.33, 0.33, 0.34)$  against distribution  $Q = (q_1, q_2, q_3)$  where  $q_3 = 1 - q_1 - q_2$ . When  $q_1 + q_2 > 1$  the probability distribution is invalid and its distance is set to 0.

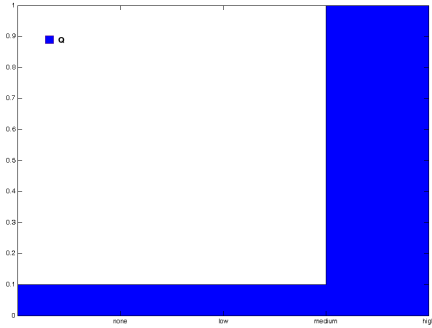


(c) J-divergence normalized with  $\alpha = 10$  for the three state distribution  $P = (0.8, 0.1, 0.1)$  against distribution  $Q = (q_1, q_2, q_3)$  where  $q_3 = 1 - q_1 - q_2$ . When  $q_1 + q_2 > 1$  the probability distribution is invalid and its distance is set to 0.

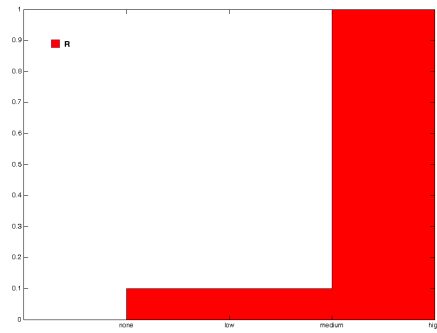
Figure 5.19: Normalized J-divergence



(a) CDF for distribution P

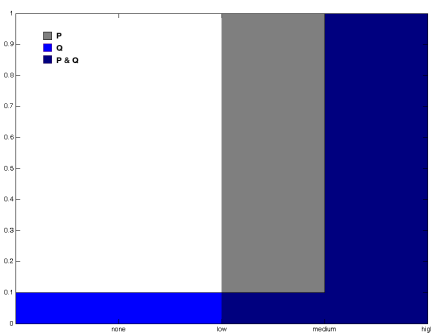


(b) CDF for distribution Q

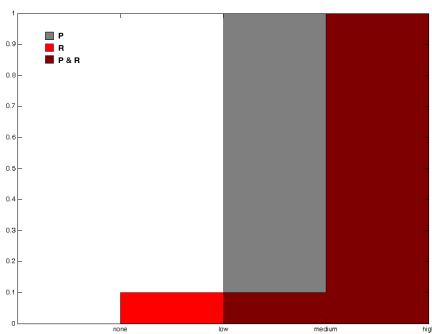


(c) CDF for distribution R

Figure 5.20: CDF for distributions P, Q and R



(a) P and Q combined



(b) P and R combined

Figure 5.21: CDF for combined distributions

Table 5.9: Summary of distance measures.

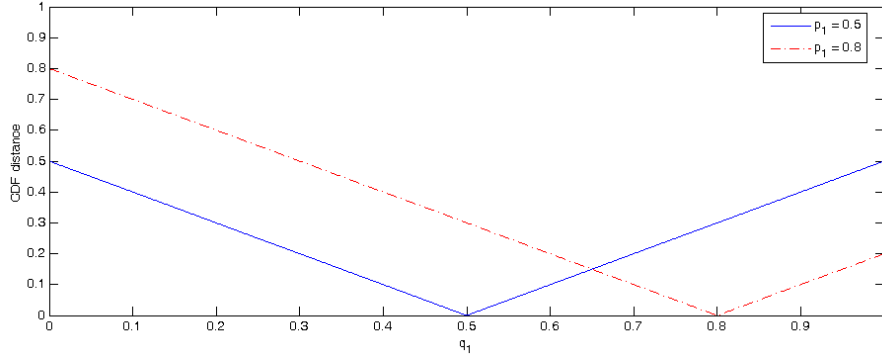
Measure	Range	Symmetric	Comments
Euclidean	$[0, \sqrt{2}]$	yes	-
Hellinger	$[0, \sqrt{2}]$	yes	More sensitive near 0 and 1.
Kullback-Leibler	$[0, \infty)$	no	More sensitive near 0 and 1. Division by zero if $q_i = 0$ .
J-divergence	$[0, \infty)$	yes	More sensitive near 0 and 1. Division by zero if $q_i = 0$ .
CDF	$[0, 1]$	yes	Targeted towards ordinal distributions.

$[1, 0, 0]$ , where the difference is maximum. This is because in the last two situations the evenly distributed probability distribution  $[0.33, 0.33, 0.34]$  gets completely concentrated in either the left-most or right-most state of  $Q$ , causing a maximum change, while in the other case all the probability has completely shifted to the center state, which is a less extreme change.

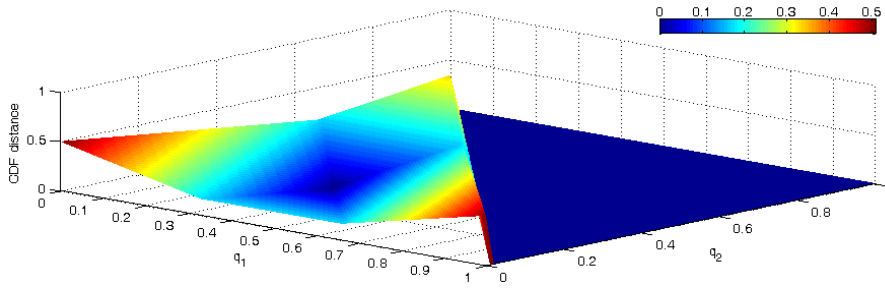
If we look at Figure 5.22(c), we can see that when  $[q_1, q_2, q_3] = [0, 0, 1]$  the difference with  $[0.8, 0.1, 0.1]$  is maximum, because the probability shifts from mostly the left states to the extreme right, you could say. We can also see that the change to  $[q_1, q_2, q_3] = [0, 1, 0]$  is considered greater than the change to  $[q_1, q_2, q_3] = [1, 0, 0]$ , which again can be explained by the shift of probability being greater in the first case.

### 5.5.6 Conclusions

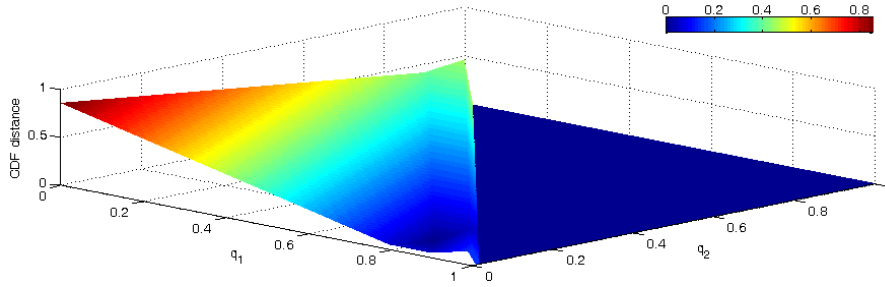
Table 5.9 summarizes the properties of the discussed distance measures. The Kullback-Leibler distance, J-divergence and Hellinger distance all are more sensitive near 0 and 1, which is nice because that captures relative differences. But only the J-divergence and the Hellinger distance are symmetric, a required property. The CDF distance is a good choice when there are ordinal nodes, because it represents the shift of probability according to the cumulative probability functions of the two distributions. A drawback is that it treats differences near 0 and 1 the same as elsewhere. Which measure will provide the best results in our situation is hard to determine, even more so because the performance of each measure can vary for networks with various characteristics. In our implementation we are going to allow for the use of the Euclidean distance, the Hellinger distance, the CDF distance and the J-divergence. Switching measures will be very easy which makes our implementation very flexible and prepared for most situations. We are not going to use the Kullback-Leibler distance, because it is not symmetric and therefore not suitable to our needs.



(a) CDF distances for the binary distribution  $P = (p_1, p_2)$  against distribution  $Q = (q_1, q_2)$  where  $p_2 = 1 - p_1$  and  $q_2 = 1 - q_1$ .



(b) CDF distances for the three state distribution  $P = (0.33, 0.33, 0.34)$  against distribution  $Q = (q_1, q_2, q_3)$  where  $q_3 = 1 - q_1 - q_2$ . When  $q_1 + q_2 > 1$  the probability distribution is invalid and its distance is set to 0.



(c) CDF distances for the three state distribution  $P = (0.8, 0.1, 0.1)$  against distribution  $Q = (q_1, q_2, q_3)$  where  $q_3 = 1 - q_1 - q_2$ . When  $q_1 + q_2 > 1$  the probability distribution is invalid and its distance is set to 0.

Figure 5.22: CDF distances



## Chapter 6

# Color of arcs

### 6.1 Overview

If a network, or a part of a network, consists of ordinal nodes (see Section 5.1), it is meaningful to determine the *sign of influence*. The sign of influence between a node  $A$  and a direct successor or predecessor  $B$  can be *positive*, meaning that higher values of  $A$  always lead to higher values of  $B$ , *negative*, meaning that higher values of  $A$  always lead to lower values of  $B$ , *null*, meaning that higher values of  $A$  always lead to values of  $B$  that are neither higher nor lower, or *ambiguous*, meaning that the influence is neither *positive*, *negative* nor *null*. In this context, 'always' refers to the fact that  $A$  can have more parents than just  $B$ , in which case the stated relations must hold for every possible configuration of the other parents of  $A$ , i.e., always. These definitions stem from the field of *qualitative probabilistic networks* [33], in which the relations between nodes in a network are not defined by conditional probability tables, but by the signs of influences among nodes.

Up to now the approach to determining the sign of influence has been based on calculations using the conditional probability tables, which results in a *static* and local method. We will implement this feature but we will also extend it so that it is *context-specific* and non-local, i.e., taking into account any observed variables and indirect influences. We will discuss both in the following two sections.

### 6.2 Static coloring

The static determination of the sign of influence uses the conditional probability tables of a Bayesian network. As mentioned earlier, there can be *positive*, *negative*, *null* or *ambiguous* influences. What type of influence is present for a certain arc is given by the following equations.

There is a *positive* influence between a parent  $A$  and its child  $B$ , where  $C$  is the set of all parents of  $B$  except  $A$ , when the following two equations hold:

$$\forall a_i \forall a_j. a_i \geq a_j \Rightarrow \forall b \forall C (P(B \leq b | a_i, C) - P(B \leq b | a_j, C) \leq 0) , \quad (6.1)$$

and

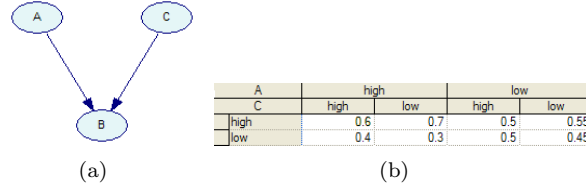


Figure 6.1: An example network and conditional probability table.

$$\exists a_i \exists a_j. a_i \geq a_j \Rightarrow \forall b \forall C (P(B \leq b | a_i, C) - P(B \leq b | a_j, C) < 0) . \quad (6.2)$$

There is a *negative* influence when the following two equations hold:

$$\forall a_i \forall a_j. a_i \geq a_j \Rightarrow \forall b \forall C (P(B \leq b | a_i, C) - P(B \leq b | a_j, C) \geq 0) , \quad (6.3)$$

and

$$\exists a_i \exists a_j. a_i \geq a_j \Rightarrow \forall b \forall C (P(B \leq b | a_i, C) - P(B \leq b | a_j, C) > 0) . \quad (6.4)$$

There is a *null* influence when the following equation holds:

$$\forall a_i \forall a_j. a_i \geq a_j \Rightarrow \forall b \forall C (P(B \leq b | a_i, C) - P(B \leq b | a_j, C) = 0) . \quad (6.5)$$

If there is no *positive*, *negative* or *null* influence, the influence is *ambiguous*.

We will give an example. Figure 6.1(a) shows a simple Bayesian network with the conditional probability table for node  $B$  in Figure 6.1(b).

If we want to determine the sign of influence for the two arcs, we only need the given conditional probability table of node  $B$ .

First, let us consider the arc between  $A$  and  $B$ . For each state of the other parents of  $B$ , which in this case is node  $C$ , we are going to compare the distributions of  $B$ . This comes down to comparing  $[0.5, 0.5]$  with  $[0.6, 0.4]$ , which are the two distributions of  $B$  when  $C$  is in state “high” and  $A$  changes from “low” to “high”, and  $[0.55, 0.45]$  with  $[0.7, 0.3]$ , which are the two distributions of  $B$  when  $C$  is in its other state, “low”, and  $A$  again changes from “low” to “high”. We can see, in this case even without explicitly using the presented formulas, that in both cases the influence is positive. In the first case, the probability of the “high” state of node  $B$  increases from 0.5 to 0.6. In the second case, the probability of the “high” state also increases, from 0.55 to 0.7. This indicates a positive influence, the change of node  $A$  from the state “low” to the higher state “high” also causes the probability distribution of node  $B$  to shift in that same direction. Formally, it can be seen that Equations 6.1 and 6.2 hold.

To determine the sign of influence for the arc between node  $C$  and node  $B$  we need to compare the distribution  $[0.7, 0.3]$  with  $[0.6, 0.4]$  and the distribution  $[0.55, 0.45]$  with  $[0.5, 0.5]$ . The first two indicate the change when node  $C$  changes from state “low” to state “high” where node  $A$  is in state “high”. The latter two indicate the change when node  $C$  changes from state “low” to state “high” where node  $A$  is in its other state, “low”. This influence is negative, because, in both cases, when the state of node  $C$  changes from state “low” to state “high”, the probability of the state “high” of node  $B$  decreases. So a change in one direction in node  $C$  causes a change in the opposite direction in node  $B$ , i.e., the influence is negative. In this case Equations 6.3 and 6.4 hold.



### 6.3 Dynamic coloring

Our dynamic method for determining the sign of influence is, just like the thickness of arcs, context-specific and it takes into account any indirect influences. We are going to determine the sign of influence by looking at the posterior probability distribution of a node, for each possible state of the parent or child node. For a node with  $n$  states, this will result in  $n$  potentially different posterior probability distributions of the connected node(s). These posterior probability distributions will be compared using a slightly simplified version of the equations in Section 6.2. Between a node  $A$  and its child  $B$  there is a *positive* influence when the following two equations hold:

$$\forall a_i \forall a_j. a_i \geq a_j \Rightarrow \forall b (P(B \leq b|a_i) - P(B \leq b|a_j) \leq 0) , \quad (6.6)$$

and

$$\exists a_i \exists a_j. a_i \geq a_j \Rightarrow \forall b (P(B \leq b|a_i) - P(B \leq b|a_j) < 0) . \quad (6.7)$$

There is a *negative* influence when the following two equations hold:

$$\forall a_i \forall a_j. a_i \geq a_j \Rightarrow \forall b (P(B \leq b|a_i) - P(B \leq b|a_j) \geq 0) , \quad (6.8)$$

and

$$\exists a_i \exists a_j. a_i \geq a_j \Rightarrow \forall b (P(B \leq b|a_i) - P(B \leq b|a_j) > 0) . \quad (6.9)$$

There is a *null* influence when the following equation holds:

$$\forall a_i \forall a_j. a_i \geq a_j \Rightarrow \forall b (P(B \leq b|a_i) - P(B \leq b|a_j) = 0) . \quad (6.10)$$

If there is no *positive*, *negative* or *null* influence, the influence is *ambiguous*. The sign of influence can be determined, just like the thickness of an arc, in both directions, i.e., from the parent to the child and from the child to the parent. If these two differ then the sign will also be regarded as being *ambiguous*.

The dynamic method differs from the static method in that it is not local. The sign of influence in our dynamic method signifies the actual behaviour in the current situation, including all indirect influences and observations. Figure 6.2 clarifies this. It shows a Bayesian network consisting of three nodes. All nodes have two states, “low” and “high”. The conditional probability table for node  $A$  is shown in Figure 6.2(b) and for node  $B$  in Figure 6.2(c).

Let us consider the two arcs between nodes  $A$  and  $B$  (arc  $AB$ ) and nodes  $C$  and  $B$  (arc  $CB$ ). With the static method of Section 6.2 the sign of influence of arc  $AB$  is positive and that of arc  $CB$  negative. This is based on the conditional probability table of node  $B$ , which is shown in Figure 6.2(c). From that conditional probability table we can see that node  $C$  has a very small negative influence on node  $B$ . If node  $C$  changes state from “low” to “high”, the probability of the “high” state of node  $B$  drops only 0.001. The positive influence of node  $A$  on node  $B$ , on the other hand, is very strong. If node  $A$  changes state from “low” to “high”, the probability of the “high” state of node  $B$  increases from almost zero to almost one. The signs are visualized in Figure 6.3(a). A green arrow indicates a positive influence and a red arrow indicates a negative influence.

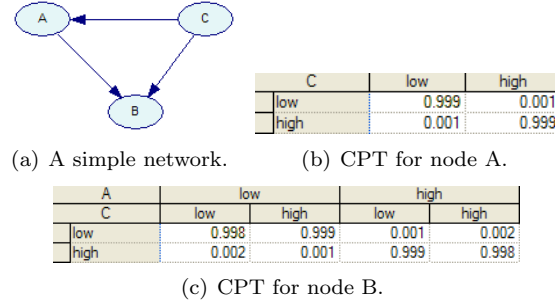


Figure 6.2: An example network and conditional probability tables.

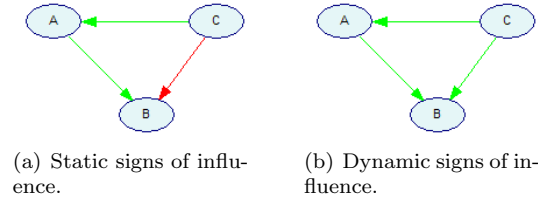


Figure 6.3: Signs of influence, green indicates a positive influence and red a negative influence.

In our dynamic method, the influence that node  $C$  has on node  $B$  is determined to be positive, which is the opposite compared to the static method, that shows a negative influence. The reason for this is that the static method is local. It is true that node  $C$  has a small negative influence on node  $B$ , but what happens if we actually *observe* the two states, “low” and “high” of node  $C$ , and calculate the posterior probabilities of the other two nodes? We will see that changing node  $C$  from state “low” to state “high” actually results in a very large increase in probability of the “high” state of node  $B$ , meaning a positive influence, and not a negative influence like the static method shows. This can be explained by the indirect influence that  $C$  has on  $B$  through  $A$ . When we observe node  $C$  in state “low”, then node  $A$  is almost certainly also in state “low”. When node  $C$  is observed in state “high”, then node “ $A$ ” is almost certainly also in state “high”. This can be seen in the conditional probability table of node  $A$ , shown in Figure 6.2(b). So, if we now look at the conditional probability table of node  $B$  (Figure 6.2(c)), we see that in this case, when node  $C$  is observed in its “low” state, and therefore node  $A$  is also much more likely to be in its “low” state than in its “high” state as explained just now, the posterior probability distribution of node  $B$  will be very close to  $[0.998, 0.002]$ . If node  $C$  is observed in its “high” state, and therefore node  $A$  is also much more likely to be in its “high” state than in its “low” state, the posterior probability distribution of node  $B$  will be very close to  $[0.002, 0.998]$ . This corresponds to a positive influence, because the “high” state of node  $B$  increases in probability when node  $C$  changes state from “low” to “high”. We can conclude that, when node  $C$  is observed, the positive influence that reaches node  $B$  through node  $A$  is much larger than the direct negative influence that node  $C$  has on node  $B$ .

Figure 6.4 shows another example of different behaviour between the static

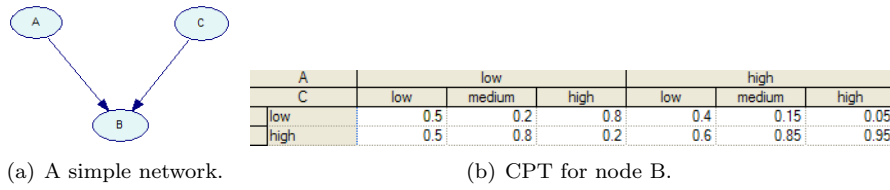


Figure 6.4: An example network and conditional probability table.

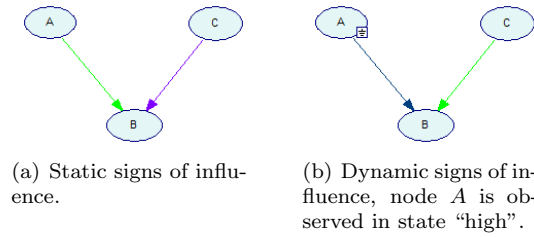


Figure 6.5: Signs of influence, green indicates a positive influence, purple an ambiguous influence and blue that no sign has been determined.

and the dynamic method. It again shows a network with three nodes. Node  $A$  and  $B$  both have two states, “low” and “high”, and node  $C$  has three states, “low”, “medium” and “high”.

In the static method, the influence node  $A$  has on node  $B$  is determined to be positive, and the influence that node  $C$  has on node  $B$  is determined to be ambiguous, as shown in Figure 6.5(a). If we look closely at the conditional probability table of node  $B$ , shown in Figure 6.4(b), we can see that this is correct. Everytime that node  $A$  changes from state “low” to state “high”, the probability of the “high” state of node  $B$ , while keeping the state of node  $C$  unchanged, increases. The ambiguous influence of node  $C$  on node  $B$  can also be explained using the conditional probability table. When node  $A$  is in state “low” and we check what happens when the state of node  $C$  changes from “low” to “medium” to “high”, we see that the probability of the “high” state of node  $B$  first increases, but then decreases again. This indicates an ambiguous influence. When node  $A$  is in state “high” instead of “low”, then the probability of the “high” state of node  $B$  increases from 0.6 to 0.85 to 0.95, which indicates a positive influence. But the ambiguous influence which is also present for this connection causes the final sign to be ambiguous.

If we now observe node  $A$  to be in its “high” state, the dynamic mode redetermines the sign of influence and considers the sign between node  $C$  and  $B$  to be positive, as shown in Figure 6.5(b). This is because now that we know for a fact that node  $A$  is in state “high” the situation has changed such that the influence no longer is ambiguous but positive. There is no sign of influence determined for the arc between node  $A$  and node  $B$ . This is because node  $A$  has been observed, and therefore there is no potential influence anymore.



## Chapter 7

# Implementation

The theories presented in Chapter 5 and 6 have been implemented, and integrated into GeNIe, the software package developed at the Decision Systems Laboratory of the University of Pittsburgh. GeNIe makes use of the SMILE library as its reasoning engine, which is also developed at the Decision Systems Laboratory of the University of Pittsburgh.

### 7.1 Implementation in SMILE

SMILE, which is introduced in Section 2.7.1, is built up of a collection of classes, all written in C++. We have done our implementation by creating two new classes, also written in the C++ programming language. One class represents an arc and is called `DSL_arc`, the other holds and manages all arcs that exist in a Bayesian network and is called `DSL_arcs`. The class diagram is shown in Figure 7.1. For a particular network there exists exactly one `DSL_arcs` class, which holds multiple `DSL_arc` classes, one for each arc in the network.

The `DSL_arcs` class provides top-level functions. Arcs can be retrieved and the distance measure to use when determining the thickness of an arc can be set for all arcs at once. The `DSL_arcs` class holds a reference to the network it belongs to and stores all arcs in a vector for easy manipulation.

The `DSL_arc` class represents a single arc in a network. It holds a reference to the network it belongs to and references to its child and parent node. Furthermore it stores probability distributions of the parent and child nodes. For the dynamic mode posterior probability distributions need to be stored for the parent/child node for each possible state of the child/parent node. The `DSL_arc` class has various structures to manage this. It also is able to store the direction in which the dynamic influence should be calculated, and provides functions to manage the changing of direction, because not all changes are permitted. For example, when the parent node is a decision node the direction is always to the child, as explained in Section 5.3.1. There are also functions in the `DSL_arc` class to provide access to both the static and the dynamic influence values. These functions return a value between 0 and 1 of type `double`, based on the currently set distance function, where 0 indicates no influence and 1 indicates a maximum influence. Both for the static and the dynamic influence there are three separate functions. One returns the maximum influence value, another returns the

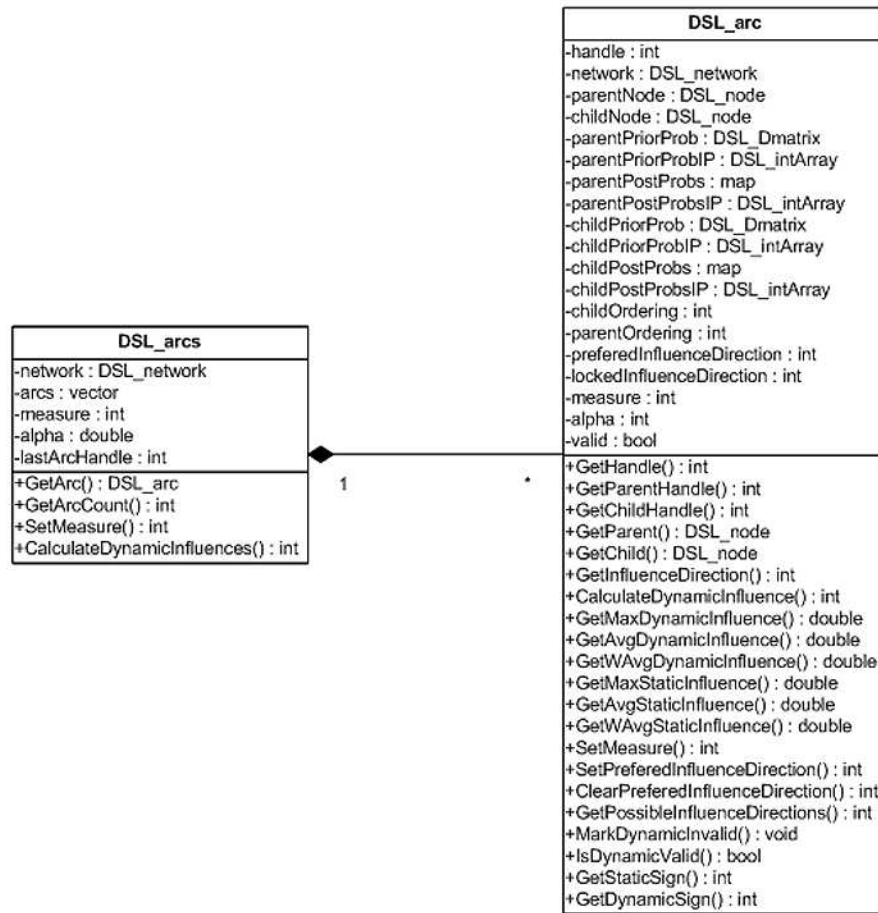


Figure 7.1: Class diagram of the arcs classes.

average influence value and the third one returns a weighted average. This has all been discussed in Chapter 5. Finally, there are two functions in the **DSL\_arc** class to retrieve the sign of influence. One returns the static sign of influence and the other returns the dynamic sign of influence. This is done as a simple integer, internally constants are defined to make things easy to understand.

All that is needed to get the signs of influence and the strengths of influence for the arcs in a network is to create a **DSL\_arcs** object, and then to use the functions provided by the **DSL\_arc** class to get the needed information. If the state of the network changes, e.g., a new observation has been done, the dynamic data needs to be recalculated. This can be triggered by marking the current values invalid, for which the **DSL\_arc** class provides a function. When this has been done, a subsequent call to any of the functions regarding dynamic values will return an updated value.

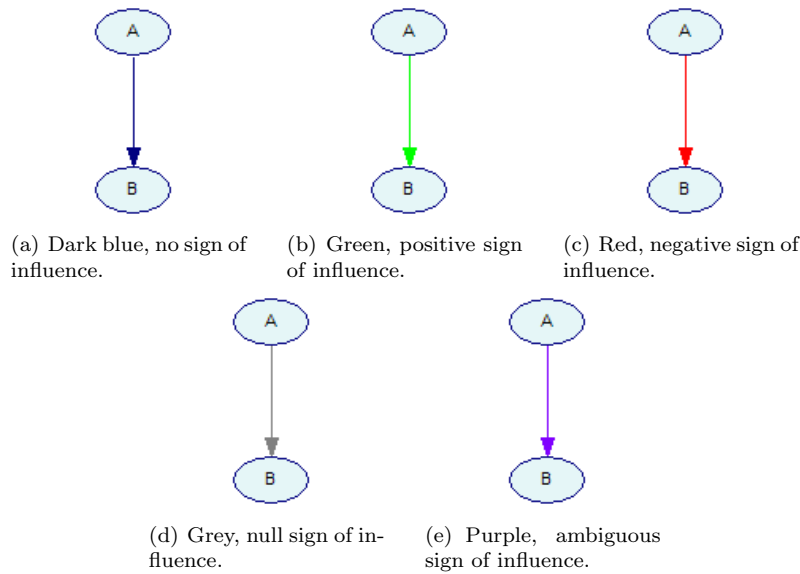


Figure 7.2: The different possible colors.

## 7.2 Integration into GeNIe

The implementation of Section 7.1 has been integrated into GeNIe. We created an extra module within GeNIe that provides all the functionality for the thickness and the color of the arcs. GeNIe is created and maintained solely by Tomasz Sowinski, a member of the Decision Systems Laboratory of the University of Pittsburgh. The integration into GeNIe has been done in close cooperation with him. The actual programming of the graphical user interface has been done by Mr. Sowinski, according to our specifications and ideas.

The signs of influence are visualized by adjusting the color of an arc. In that way a user can easily study the relations and pick out any arc that does not match with his or her belief. We will color positive arcs green, negative arcs red, ambiguous arcs purple and null arcs grey. Green is mostly associated with something positive or good, and red is mostly associated with something negative or bad. Grey for the null influence has been chosen because something that is not present or irrelevant is often “greyed out” in computer programs, therefore we think grey will be easy to associate with a null influence. The purple color for an ambiguous influence has been chosen because it fits nicely with the other colors. Figure 7.2 shows all the possible colors.

The sign of influence between two nodes, visualized by the color of the arc, only has meaning if both nodes have some kind of ordering, i.e., are ordinal. The ordinality of a node can be set by accessing its property dialog box. This dialog box is shown in Figure 7.3. If the ordering of the outcomes is said to be *none*, then the color of the arcs connected to that node will be left at the default color of GeNIe, dark blue.

In the dynamic mode, the user can select in which direction the influences should be calculated by right-clicking on an arc. These directions can be visualized by icons. There are four different icons, shown in Figure 7.4.

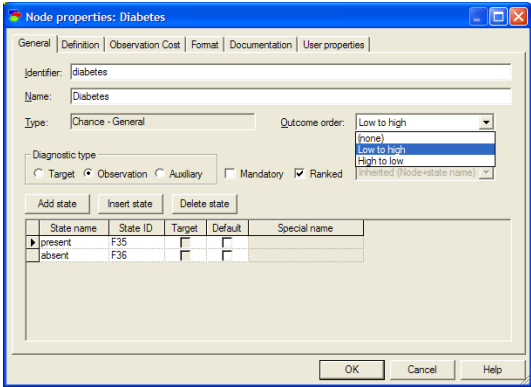


Figure 7.3: Properties of a node, the ordinality can be set using the option *Outcome order*.

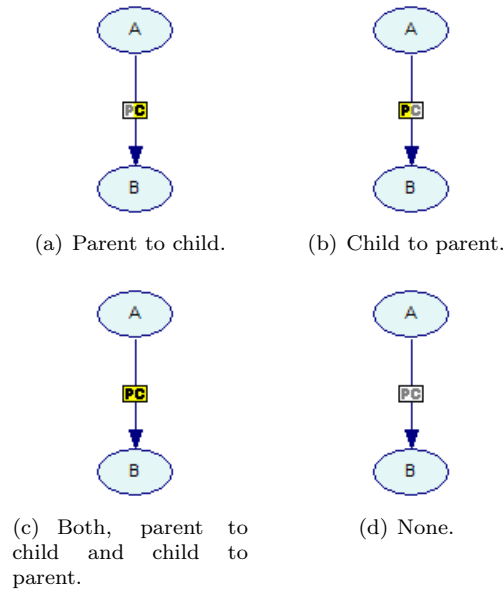


Figure 7.4: The different possible icons indicating the direction.



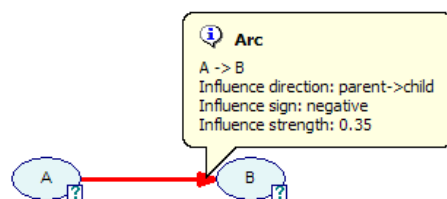


Figure 7.5: Properties of an arc, shown in a popup balloon.

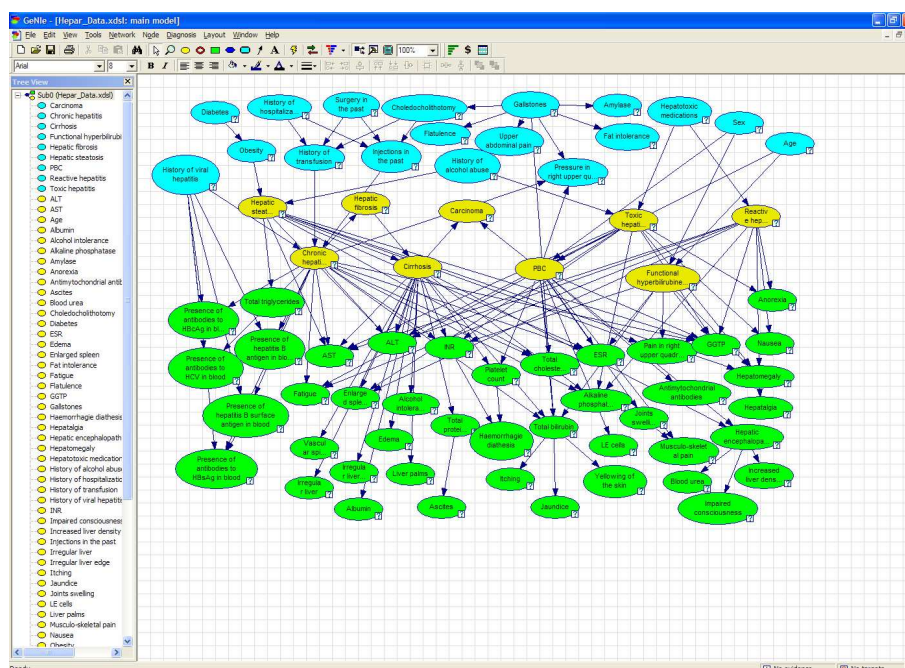


Figure 7.6: GeNIe showing the HEPAR II network.

When hovering over the tip of an arc, a popup balloon appears that holds detailed information concerning that arc. An example is shown in Figure 7.5.

## 7.3 Examples

Figure 7.6 shows GeNIe with the HEPAR II network loaded. Using the toolbar at the top the new module can be invoked, by pressing the button with the horizontal red and green arrows. Figure 7.7 shows GeNIe with the module activated and displaying the thickness of the arcs. A floating toolbar can be seen in both screenshots. This toolbar holds all the controls concerning the module. All its features are discussed in detail in Appendix B.

Figure 7.8 shows the coloring of the arcs, visualizing the signs of influence. It can be seen that a lot of ambiguous influences (purple arcs) in the static mode displayed in Figure 7.8(a) are replaced by non-ambiguous influences in the dynamic mode, displayed in Figure 7.8(b). This is due to the fact that the

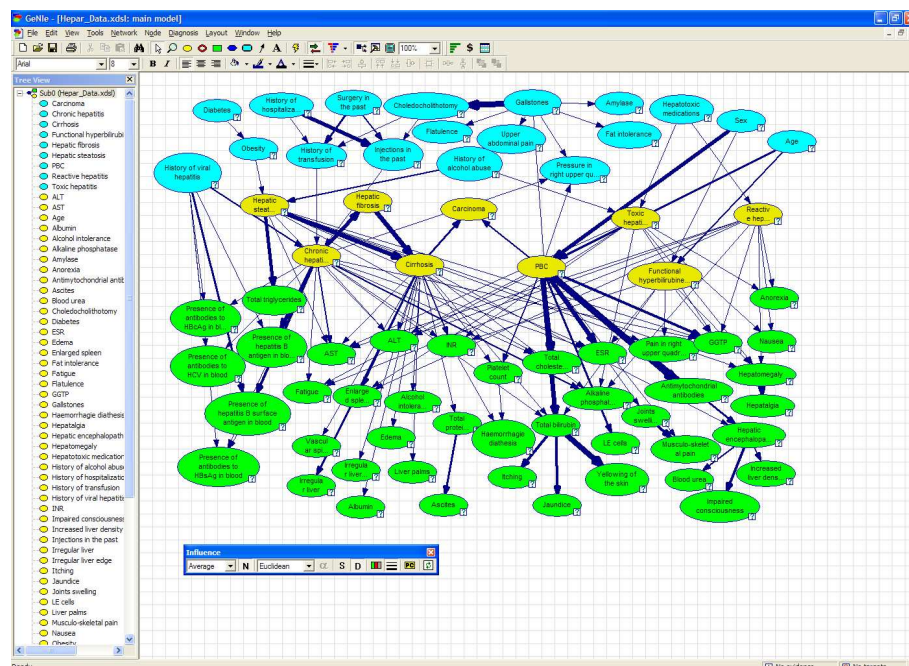
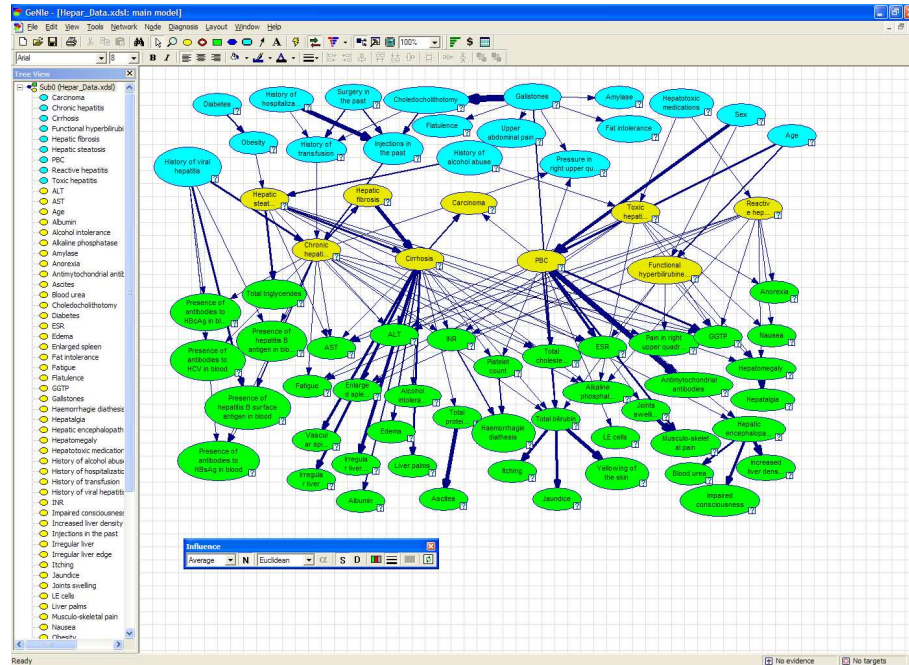
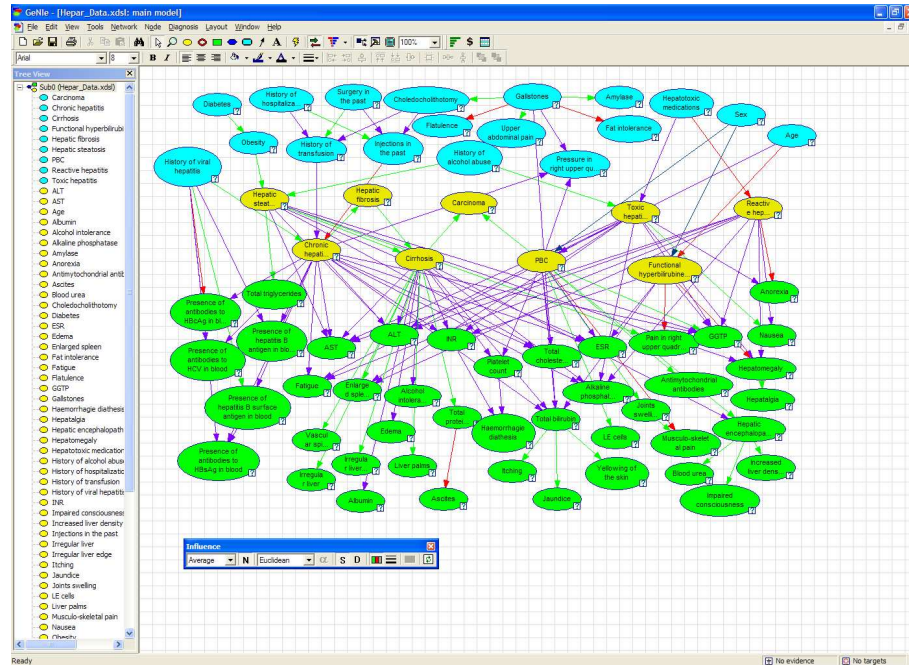
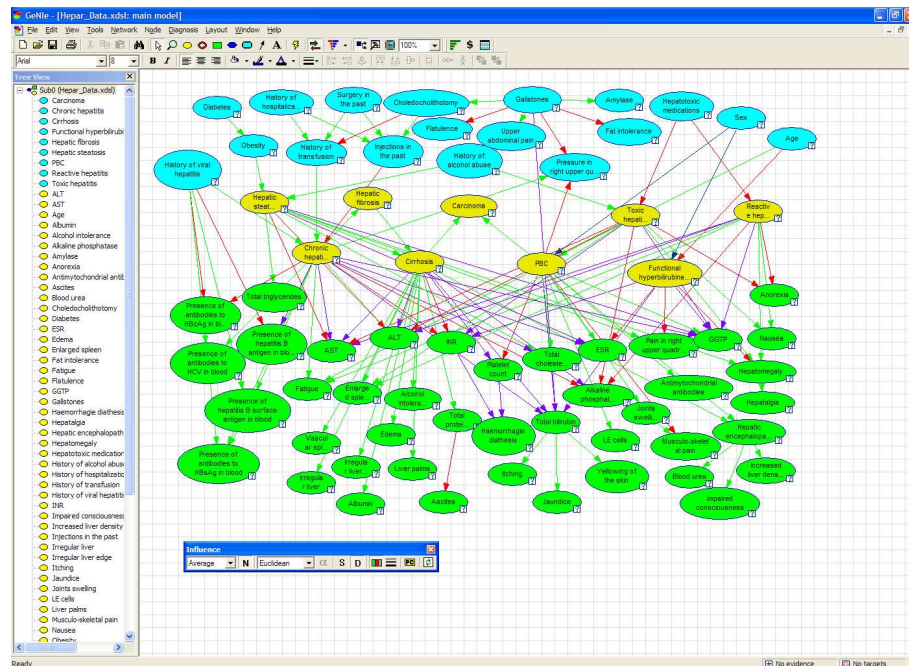


Figure 7.7: Thickness of arcs in the HEPAR II network.

dynamic mode is context-specific and not purely local, as discussed in Chapters 5 and 6. Finally, Figure 7.9 shows GeNIe when both the thickness and color of the arcs are displayed simultaneously.



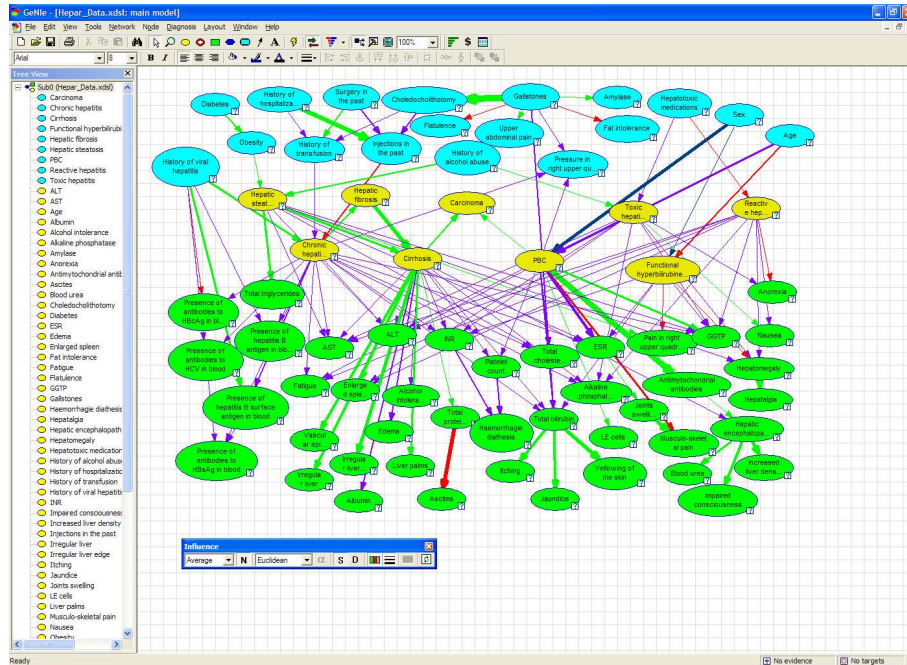
(a) Color of arcs in static mode.



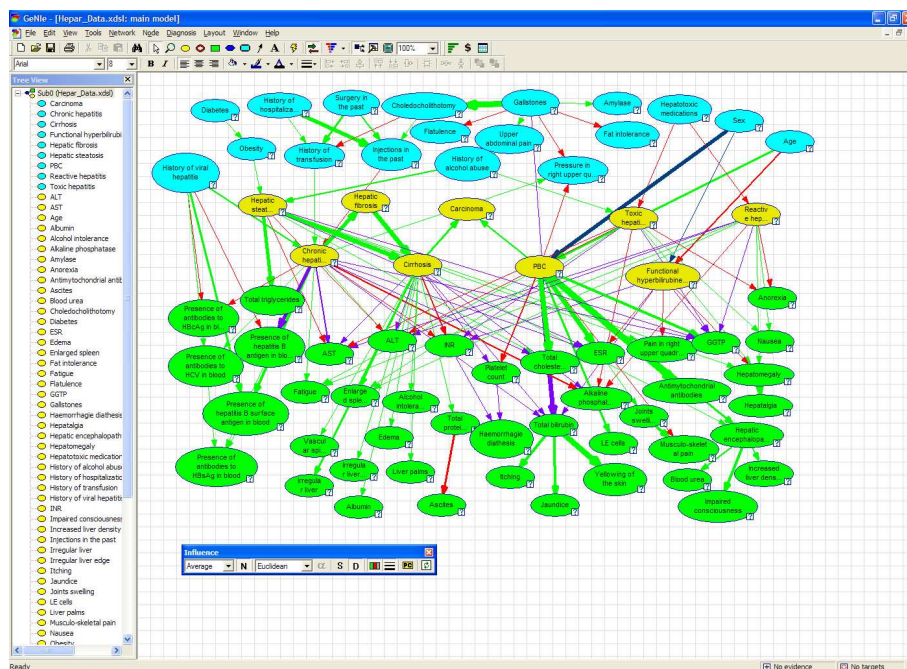
(b) Color of arcs in dynamic mode.

Figure 7.8: Color of arcs in the HEPAR II network.





(a) Color and thickness of arcs in static mode.



(b) Color and thickness of arcs in dynamic mode.

Figure 7.9: Color and thickness of arcs in the HEPAR II network.



## Chapter 8

# Empirical evaluation

We have evaluated our implementation, both quantitatively and qualitatively. Section 8.1 treats the quantitative evaluation and Section 8.2 the qualitative evaluation.

### 8.1 Quantitative evaluation

To get an idea of the time needed to calculate the data needed for our dynamic mode of both the thickness and color of the arcs, we measured the performance on various benchmark networks. The properties of the networks are listed in Figure 8.1. The results of the test are shown in Figure 8.2. The first thing to notice is that, while the calculations for many of the networks are completed within seconds, the Diabetes network is particularly slow. Keeping in mind that the most intensive part of the process is performing inference, this long running time is caused by the structure of the network. The structure is such that the inference algorithm takes more time to perform one inference step. Besides this the Diabetes network has many states per node, an average of more than eleven per node, and has quite a few arcs. For each node and for each of the states inference must be done.

Another thing that can be seen is that the running time often increases when there are observed variables. This is caused by the fact that observed variables can create extra dependencies among variable, resulting in more complex inference.

Overall, the time needed to calculate all the needed data for the dynamic mode is often negligible, but this depends on the structure of the network. If the network is very large one can ask the question whether showing thickness and color of all arcs at once is really useful. In such situations it may be more insightful to consider a subset of arcs, in which case the running time will decrease dramatically.

### 8.2 Qualitative evaluation

We have performed a qualitative evaluation of our newly developed module for thickness and color of arcs. We wanted to find out if we had succeeded in meeting our goals, i.e., creating an explanation that is both easy to use and

Table 8.1: Properties of the various networks.

Network	Nodes	Max Parents	Avg Parents	Max States	Avg States	Arcs
Alarm	37	4	1.24324	4	2.83784	46
CPCS179	179	8	1.3352	4	2.2905	239
Diabetes	413	2	1.45763	21	11.3366	602
Hailfinder	56	4	1.17857	11	3.98214	66
Hepar	70	6	1.75714	4	2.31429	123
Link	724	3	1.55387	4	2.53177	1125
Munin	1041	3	1.34198	21	5.42843	1397
Pathfinder	109	5	1.78899	63	4.11009	195

Table 8.2: Calculation time for determining thickness and color of all arcs, in seconds.

Network	No observations	10 observations
Alarm	0.016	0.001
CPCS179	0.297	0.437
Diabetes	3114.17	6343.77
Hailfinder	0.063	0.031
Hepar	0.063	0.078
Link	4.016	78.625
Munin	15.734	23.188
Pathfinder	2.297	2.657

easy to understand for a user that is familiar with Bayesian network. In order to evaluate this we designed an evaluation in which the subject is provided with two particular Bayesian network models and is asked, after a thorough introduction, to gain insight in these models using our module.

The qualitative evaluation has been performed by members of the Decision Systems Laboratory and other people. All were familiar with Bayesian networks and GeNIe and were therefore suitable candidates for our evaluation. They were given a short introduction and tutorial to the newly developed module, which is printed in Appendix A. They were also provided with a manual explaining all the options of the module. This manual can be seen in Appendix B. They were asked to explore two provided Bayesian networks, *Car* and *Hepar*, using the features of our module. The *Car* network is shown in Figure 8.1. It is a diagnostic network that can be used to diagnose certain problems with a car. The *Hepar* network [25], modeling various liver disorders, is shown in Figure 8.2.

Directly afterwards, the subjects were asked to answer the following questions:

1. How long, approximately, did you experiment with the module?
2. On a scale from one to ten, how would you rate the easiness of interpretation of the thicknesses of the arcs?
3. On a scale from one to ten, how would you rate the easiness of interpretation of the colors of the arcs?



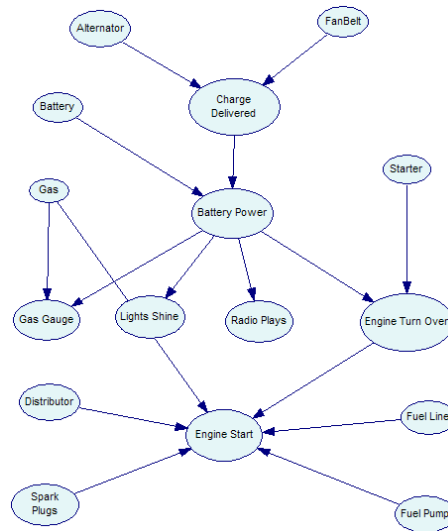


Figure 8.1: The Car network.

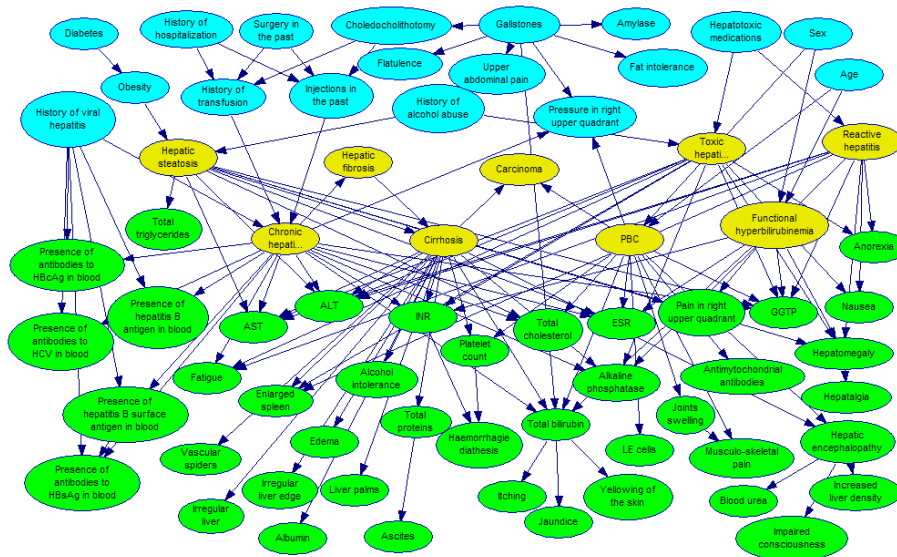


Figure 8.2: The Hepar II network.

4. On a scale from one to ten, how would you rate the intuitiveness of the thicknesses of the arcs?
5. On a scale from one to ten, how would you rate the intuitiveness of the colors of the arcs?
6. Are there any particularly positive or negative things you encountered during the use of the module?
7. What is your overall impression of the module?
8. Do you think you would use the module again to explore a model?

In the next section we will discuss the outcome of the evaluation.

### 8.2.1 Results of the evaluation

This section discusses the outcome of the qualitative evaluation. First we will treat questions 1 to 5, according to the list given in Section 8.2. These questions are answered numerically and are therefore suitable for summarization in a table. This is done in Table 8.3. We can see that subjects spent, on average, about forty minutes using the module. The easiness of interpretation of both the thickness and color of the arcs is, on average, given a slightly lower rating than the intuitiveness. But both get a rating that is more than satisfactory.

Table 8.3: Results for questions 1 to 5.

Subject #	Questions				
	1	2	3	4	5
Subject 1	45 min.	8	5	9	6
Subject 2	15 min.	8	6	9	7
Subject 3	60 min.	10	10	9	10
Subject 4	45 min.	7	7	9	9
Subject 5	60 min.	7	8	6	9
Subject 6	15 min.	7	7	8	8
Subject 7	30 min.	8	7	8	6
Subject 8	20 min.	7	6	9	8
Average	36 min.	7.8	7	8.4	7.9

To the question if there were any particularly positive or negative things (question 6), the subjects gave a variety of answers. Three subjects explicitly praise the module for its intuitive visualizations. The chosen colors are said to be intuitive, especially green for positive and red for negative. The other colors needed some time to adjust to, according to two subjects. Three subjects mentioned that it would be helpful if the recalculation in the dynamic mode is done automatically. In the current implementation a user has to press a recalculate button manually after a new observation. One user also indicated that, in the dynamic mode, after setting a new observation and recalculating it is sometimes hard to see the difference with the previous situation. It would be insightful, according to that subject, to somehow see the difference between the current situation and the previous one. Three subjects mention that the

different distance measures can be confusing. They say that it is not clear when to use which distance measure. One subject notes that the normalization option for the thickness of arcs can be dangerous in the dynamic mode, because if a new observation is done and the maximum influence changes, a direct comparison with the previous situation is impossible. Finally, one user says that he found the dynamic mode to be more useful.

To the next question, question number 7, on the overall impression of the module, every subject gave a positive answer, with, in two cases, one or two negative side notes. One subject mentions that he found the colors not to be intuitive enough, apart from red and green. One other subject mentions that the difference between the different distance measures is unclear. On the other hand, the module is called a useful extra tool for model analysis and a useful tool to get a feel for the strength of influences between nodes. It is said to be helpful in understanding the relations among the variables in a network and that it can help in model building and decision making. Also, multiple subjects say the graphical user interface looks good and that the module is easy to use.

To the final question, if the subject would use the module again in the future, all but one subject gave a positive answer. The subject who did not give a positive answer was currently not using Bayesian networks, but may do so again in the future. So the reason for not using it again is not because of the performance of the module but because the subject has no need to. All other subjects say they would, sometimes even definitely, use the module again.



## Chapter 9

# Conclusions and future work

### 9.1 Conclusions

This thesis has presented a technique to make inference in Bayesian networks more insightful. This has been done by automatically adjusting the thickness of the arcs in a Bayesian network to indicate the strength of influence between two directly connected nodes. Also, the color of an arc is automatically adjusted to indicate the sign of influence between the two directly connected nodes. These two visualizations can be done in a static way, using only the definition of a Bayesian network, but also in a dynamic way. The dynamic variant is, as opposed to the static method, context-specific, takes into account any indirect influences and is non-local.

To come to this result we first did a thorough review of previous research in this area in Chapter 3. After reviewing and discussing what we found, we formulated our initial ideas in Chapter 4. Two of these ideas, thickness of arcs and color of arcs, were eventually chosen as the two most promising ideas. These two ideas were designed in detail in Chapters 5 and 6. According to the designs we did our implementation in Chapter 7. Our implementation was done in two classes in the C++ programming language. The implementation has been integrated into GeNIe, creating a fully working explanation facility using the normally unused arcs to visualize the strength and sign of influence between two directly connected nodes. Finally, in Chapter 8, we presented our quantitative and qualitative evaluation. The quantitative evaluation consisted of evaluating the performance, in terms of calculation time, on various benchmark networks. It showed that our dynamic method can, in some situations, be quite slow, but in most Bayesian networks speed is no issue. A qualitative evaluation was performed by eight people, all familiar with Bayesian networks. They were asked a variety of questions, after having used the newly developed module of GeNIe. The reactions were very positive, both in terms of our technique giving more insight into a Bayesian network and in terms of easiness of use. The most important criticism, mentioned by quite a few subjects, was the unclarity about the available distance measures.

## 9.2 Future work

During our research, we did not pursue every idea we had. Also, during the qualitative evaluation, some opportunities arose to improve our implementation. We propose the following items for future work:

- Find a way to make the difference between the different distance measures more clear. During the qualitative evaluation multiple subjects mentioned a need for clarification of the distance measures. Giving a user insight in this is not a trivial task. A user does not want to read a long document explaining the, sometimes subtle, differences between the measures. Also, there is no golden rule as to when to use which measure. It is sometimes more a matter of feel, which takes time to get. Somehow this needs to be solved. Maybe by introducing two modes in which the module can operate. A “simple” and “advanced” mode. In the simple mode many of the settings are set by the program, and in the advanced mode the user has more options, like changing the distance function.
- In the dynamic mode, the normalization option can deceive a user. If the normalization option is selected and a new piece of evidence is observed, which causes the maximum influence present in the network to go up or down, all other arrows are scaled up or down as well. This makes a direct comparison with the previous situation impossible. This problem needs to be taken care of.
- In the dynamic mode, the user has to explicitly press a “recalculate” button after setting or removing observations. During the qualitative evaluation more than one user found this to be a burden. The solution is to make the module recalculate automatically in these situations.
- In Sections 4.1 and 4.3 we proposed the idea of visualizing the ancestors, descendants and Markov blanket of a node. We think this can be a very useful tool for a user to help him or her explore a model more efficiently.
- In Section 4.2 we proposed to visualize the subgraph containing all paths of influence between a pair of nodes. How this can be done efficiently and in what way this can be presented to a user is open for research.
- In Section 4.4 we identified showing the relevance of findings as an interesting subject. It can show which findings are most significant for a certain target variable, and could be able to indicate if those findings conflict or agree with the overall inference result. Maybe there are opportunities in the way in which these findings can be identified, but most certainly there are opportunities in finding a good way to present the gathered statistics to a user.
- In Section 4.5 we mentioned the possible improvement of insight in a model by being able to display multiple cases at once in GeNIe. In that way a user can compare various cases without much effort. An intuitive way to display and work with more than one case has to be developed. Interesting statistics could also be generated for the user, such as the amount of change between the active cases for a certain variable. Ways to visualize this efficiently have to be explored.

# Appendix A

## Qualitative Evaluation

For my MSc project I have developed an extension for GeNIe. I would like to ask you to experiment with this module and afterwards evaluate it by answering a few questions. I will start with a short theoretical introduction, followed by a tutorial of the module.

### A.1 Introduction

The developed module helps a user explore and understand inference in a Bayesian network. This is done by changing the thicknesses of the arcs and the colors of the arcs. The thickness of an arc is proportional to the amount of influence between two directly connected nodes. The color of an arc shows, simply said, if that influence is positive or negative.

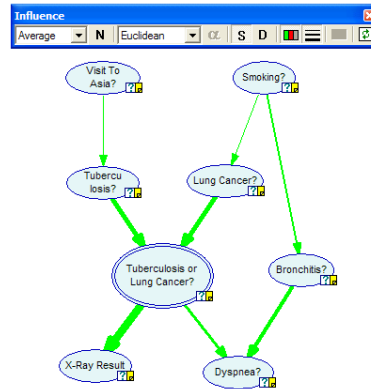
There are two main modes of operation: *static* and *dynamic*. In the *static* mode only the conditional probability tables of the nodes are used to determine the thickness and color. This means that the influence in the direction of the parent to the child is visualized. Also, the thicknesses and colors are local and not context-specific, meaning that they visualize the interaction between two directly connected nodes, but they do not take into account any observed nodes or indirect influences.

In the *dynamic* mode the actual potential influence is visualized. The thickness of an arc between two nodes indicates the actual amount of change that would occur in the probability distribution of one of the nodes if the other would be observed. The color of an arc shows the actual direction of change of the probability distribution of one of the nodes if the other would be observed. In the *dynamic* mode, the direction in which the thickness and color of an arc is calculated is selectable, i.e., the influence the parent has on the child can be visualized, but the influence the child has on the parent can also be shown. These two influences can be very different from each other. This mode is context-specific, meaning that when another piece of evidence has been observed and the influences between nodes might have changed, the *dynamic* mode can visualize the new situation by recalculating the thicknesses and colors.

### A.1.1 Tutorial

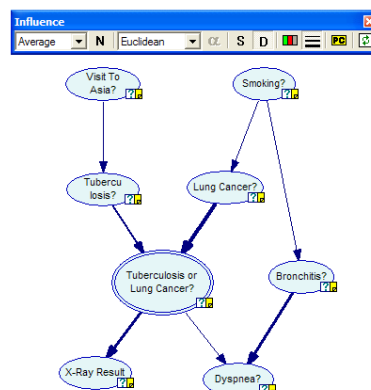
To introduce you to the module I will give a short tutorial, making use of the *Asia* network. For specific questions about the usage of the module you can refer to the provided manual, which explains every option the module has.

After loading the *Asia* network and invoking the module, the situation is as pictured below.



We can see that both coloring and thickness of arcs are activated by default, in *static* mode (the “S” icon is pressed). All arcs are colored green, which means that all influences are positive, which is what we expect for this model because *Asia* is a causal network. If there would be an arc with a different color we would be alerted immediately that something might be wrong with the definition of the model. From the thicknesses we can, for example, see that the deterministic node “Tuberculosis or Lung Cancer?” has a bigger influence on “X-Ray Result” than on “Dyspnea?”.

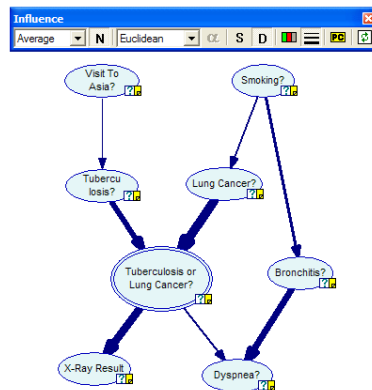
Next, we switch to the *dynamic* mode and turn off coloring.



Now that we are in *dynamic* mode, the thicknesses of the arcs indicate how much an observation of one of the two nodes of an arc would impact the other.

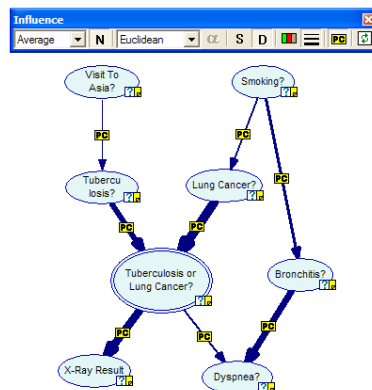
To bring out the differences in thicknesses some more, we can turn on *normalization* by pressing the “N” button.





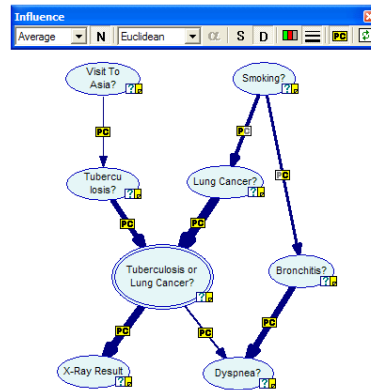
The arc with the largest influence has now been given the thickest arc, in this case the arc between “Lung Cancer?” and “Tuberculosis or Lung Cancer?”, the thicknesses of all other arcs are proportional to that arc.

In the *dynamic* mode, the direction in which the thickness of an arc is determined is selectable. To see how the directions are set at this moment, press the yellow “PC” button.



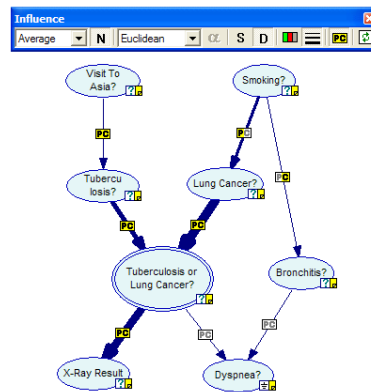
In the middle of each arc an icon has now appeared, which indicates the direction in which the influence for that arc is calculated. In the situation pictured here, all thicknesses are determined in both direction, and averaged.

We can change the direction of any arc. Let us do so for the arc between “Smoking?” and “Lung Cancer?”, by right-clicking on that arc and changing the direction to “child → parent”. Also, we are going to change the direction for the arc between “Smoking?” and “Bronchitis?” to “parent → child”.



Now the two icons have changed. The thickness of the arc between “Smoking?” and “Lung Cancer?” is now determined from the child to the parent, i.e., it shows the influence “Lung Cancer?” will have on “Smoking?” if it would be observed next. We see that that arc is thicker than in the previous situation. So if we know that a person has lung cancer, the probability of him/her being a smoker would increase more than that being a smoker would influence the probability of having lung cancer.

Finally, in the *dynamic* mode, we can observe a variable, and then recalculate the thicknesses of the arcs. Let us set “Dyspnea?” to “present” and recalculate by pressing the rightmost button, the “recalculate” button.



There are a few things to notice now. First, arcs connected to the observed variable have no thickness anymore (the icons are greyed out). This is because there is no potential influence anymore for these arcs, because there is an observed node connected to these arcs. Second, we can see that after observing “Dyspnea?”, the potential influence of “Smoking?” on “Bronchitis?” has clearly diminished, compared to the previous situation. So “Dyspnea?” has influenced “Bronchitis?” in such a way that additionally observing “Smoking?” will not have much impact on “Bronchitis?”.

## A.2 Evaluation

What I would like you to do is to read the short manual on how to use the new module, if you have not already done so, and then to run the provided version of GeNIe and play around with it, using the two provided models *Car* and *Hepar*. Please keep the following few things in mind:

- There are two modes, *static* and *dynamic*.
- The *dynamic* mode is context-specific.
- In *dynamic* mode, press the “recalculate” button after setting or removing an observation.
- It is possible to make a selection of nodes before invoking the module, only arcs between those nodes will then be used.

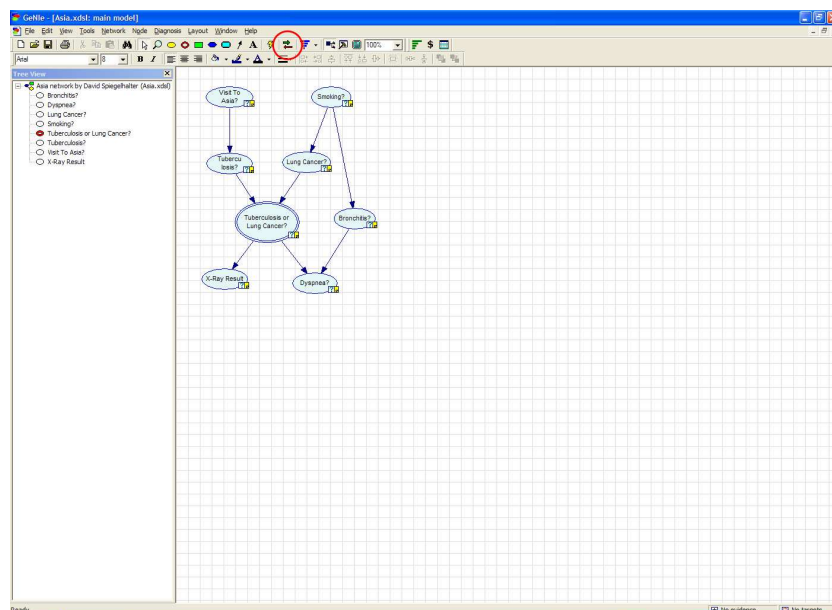
When you feel you have experimented enough to form your opinion, please go to <http://www.bingopaleis.com/joost/questions/> to answer a few questions.



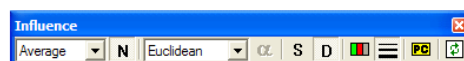
# Appendix B

## Manual

This is a short manual explaining the various possibilities of the developed module for displaying color and thickness of arcs. To start the module, open a model and then press the button with the red and green arrow, which is surrounded by a red circle in the screenshot below.



The toolbar that is pictured below will then appear. This toolbar holds all the controls concerning the module.



In the remainder of this section each option of the toolbar will be explained. First a screenshot will be shown with a certain option circled, after which that option will be explained.

### Thickness type



Using this option the way the thickness of the arcs is determined can be changed. There are three possibilities: *average*, *maximum* and *weighted*. The first one averages all possible influences two nodes can have on each other. If *maximum* is selected, the thickness of an arc is proportional to the highest possible influence between the two nodes. The third option, *weighted*, also averages the various influences, but also takes into account the prior probabilities of the various states of the nodes. If a certain state has a low prior probability, the influence belonging to that state will not have a large part in determining the thickness of the arc.

### Normalize



This button toggles between the *normalized* and *non-normalized* mode. If the button is pressed, the thickest possible arc is given to that arc that has the highest strength of influence. The thicknesses of all other arcs are calculated proportionally to the thickest arc. If the button is depressed the *non-normalized* mode is activated. This way the thickest possible arc will only be given to an influence value of 1 (influence values always range from 0 to 1).

### Distance measure



The module allows for four different distance measures: *Euclidean distance*, *Hellinger distance* [14], *J-Divergence* [17, 18] and *CDF distance* [19]. The selected distance measure is used to determine the amount of difference between two probability distributions.

### Alpha parameter



When the *J-Divergence* is selected as the distance measure, its *alpha* parameter can be changed using this button. The *alpha* parameter is used to control the normalization of the *J-Divergence*.

## Static mode



When the *S* button is pressed, the *static* mode is activated. This mode only makes use of the conditional probability tables present in the model and is therefore not context-specific. The thicknesses of the arcs indicate the strength of influence that a parent has on a child, while the colors of the arcs show the sign of that local influence.

## Dynamic mode



When the *D* button is pressed, the *dynamic* mode is activated. This mode is context-specific and essentially shows the potential influence two directly connected nodes can have on each other, i.e., the amount of change that would occur in one node if the other would be observed next is visualized by the thickness of the arc, while the sign of that change is displayed by the color of the arc.

## Coloring of arcs



When this button is pressed coloring of the arcs is activated. The colors indicate the sign of influence. This sign can be *positive* (green), *negative* (red), *null* (grey), or *ambiguous* (purple). The color of an arc indicates what kind of influence there is between the two nodes connected by that arc. The sign of influence in the *static* mode can be different from the sign in the *dynamic* mode. In *static* mode the color indicates the sign of influence the parent has on the child, but there could be other paths of influence to the childnode that have a larger influence in a different direction, so that when the parent is observed, the change in probability is in a different direction than the color of the arrow indicates. In the *dynamic* mode the color indicates the actual net direction of change when the parent would be observed next.

## Thickness of arcs



When this button is pressed the thickness of the arcs is activated. The thickness of an arc indicates the strength of influence between the two nodes connected by that arc. The thickness in the *static* mode can be different from the thickness in the *dynamic* mode.

## Direction icons



In the *dynamic* mode, the strength of influence can be calculated in three different directions: from the parent to the child, from the child to the parent or in both directions. To visualize which option is used for a particular arc, icons can be shown on the arcs to indicate the used direction. This button can be used to toggle these icons on and off. This option is only valid in the *dynamic* mode. In the *static* mode the direction is always from parent to child, because the *static* mode relies on conditional probability tables.

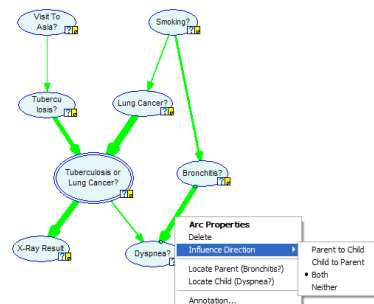
## Recalculate



In the *dynamic* mode, the final button can be used to recalculate the thickness and coloring of the arcs. This is needed when you, for example, have observed a new piece of evidence. This option is only valid in the *dynamic* mode.

All the features of the toolbar have now been mentioned. There are two more ways in which interaction with the module is possible. These will be explained in the next two sections.

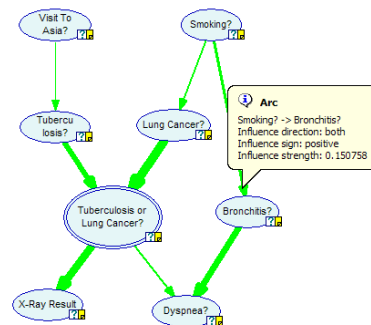
## Change direction of influence



In the *dynamic* mode it is possible to change the direction in which the influence is determined. This can be done by right-clicking on an arrow. A popup menu will appear like the one shown above. This popup menu can be used to set the desired direction.



## Arc information



If the mouse pointer is moved onto the tip of an arrow, a balloon tooltip will appear like the one shown above. This tooltip holds all the information concerning the strength of influence and the sign of influence.



# Appendix C

## Paper version

### Visualizing Inference in Bayesian Networks

Joost R. Koiter

Delft University Of Technology

Faculty of Electrical Engineering, Mathematics,  
and Computer Science, Department of Man-Machine  
Interaction

#### Abstract

*We propose a technique to visualize important aspects of a Bayesian network, in order to make the process of inference more insightful. We have used the arcs in a Bayesian network to show additional information: (1) the thickness of an arc is automatically adjusted to represent the strength of influence between two directly connected nodes and (2) the color of an arc is automatically adjusted to indicate the sign of influence between two directly connected nodes. Our technique does this in a novel, dynamic way, which is context-specific and takes into account any indirect influences. We have implemented our technique and performed a qualitative empirical evaluation. This evaluation showed that our technique and implementation are easy to use and understand and give a user more insight into a Bayesian network.*

#### C.1 Introduction

A Bayesian network [27] consists of two parts: a qualitative part and a quantitative part. The qualitative part is a directed, acyclic graph in which the nodes are random variables and the arcs represent probabilistic dependen-

cies among the nodes. The arcs can model causal relationships, but this is not necessary. The quantitative part consists of conditional probability tables and prior probabilities. A node has a prior probability if that node has no parents. When a node has one or more parents, it has a conditional probability table, representing the probabilities of each state given the states of the parent nodes. A Bayesian network encodes the full joint probability distribution. With the full joint probability distribution, any query in the domain can be answered. The most common task performed in a Bayesian network is the computation of the posterior probability distribution for a set of query variables, given an observation of a set of evidence variables. This process is called *inference*, but is also called Bayesian updating, belief updating or reasoning. The resulting posterior probability distributions of the query variables are used to draw conclusions and are the basis for decisions.

But the results of inference, the posterior probability distributions, are not always easy to explain. Why and how the probability distribution of a certain query variable has been affected by a set of evidence variables is often difficult to understand for even the most experienced Bayesian network user. The field of *explanations* in Bayesian networks tries to give the user more insight into the workings of particular network [23].

An explanation should be presented in a way that is effective, convenient, as well as easily accessible. A distinction that can be made in this respect is that between *verbal* and *graphical* explanations.

A *verbal* explanation could be, for example: “Variable A is dependent on variable B, but given variable C they are independent”, or “State zero is somewhat more likely than state one”.

A *graphical* explanation uses graphical means to communicate an explanation. The most obvious and basic explanation of this type is the visualization of the

network structure. If the user has enough knowledge about Bayesian networks, he can deduce the dependencies and independencies between the variables in the modeled domain from this view. Another example is to display the probabilities of the various states of a variable using graphical bars that range from zero to one hundred percent.

There has been some interesting work in this field of research. The INSITE method [31], by H.J. Suermondt, tries to explain which observations have influenced a certain target variable and to what extent. It also determines the paths through which the relevant findings influence the target variable, the so called “chains of reasoning”. The result is a very clear and insightful explanation of why a certain target variable has been influenced in a certain way. A user can see which observations have the largest influence on the target variable, and the paths through which they reach the target variable. Only the most relevant observations are included in the explanations. Observations that have little or no impact on the posterior probability distribution of the target variable are discarded.

The software package Elvira [3, 21, 22, 23] incorporates many forms of explanations, both verbal and graphical. The verbal explanations come down to descriptions about the various nodes and their type. For this to work all nodes must be classified. The classification, in conjunction with the network structure, is used to build up a verbal description of the network, in a causal way. Other verbal explanations use likelihood ratios, saying that one state is, for example, “3.77” times more likely than some other state. The graphical capabilities are probably the best part of Elvira. Colors are used throughout the program to indicate the direction of change of probability distributions of nodes. Also, Elvira is able to show the “chains of reasoning” by using the INSITE method, along with coloring of nodes and links to indicate the changes in probability.

BayesiaLab, a commercial software package for modeling Bayesian networks, also features quite a few options that help a user understand what is going on in a model. The most interesting parts make use of a target node, which the user has to set. Various statistics can then be generated. BayesiaLab makes use of the graphical representation of the network by augmenting it with various symbols to signify changes and characteristics and by adjusting the thickness of the arcs to indicate the contribution of that arc to the current situation of the network. Besides this BayesiaLab can generate various textual statistical reports.

In this paper, we propose a way to use the arcs in a Bayesian network to indicate the sign of influence and strength of influence between two directly connected nodes, by varying the color and thickness of the arcs.

The approach that we are proposing is a dynamic one. It considers the network in its current state, including any observations. It essentially indicates how much potential influence a node has on a direct successor or predecessor, so the influence that a node could have if it was observed next. Our method is targeted towards people who have a fairly good understanding of what Bayesian networks are, for example researchers that build Bayesian networks to aid in their research.

The remainder of this paper is structured as follows. Section C.2 gives an introduction to Bayesian networks. Section C.3 discusses the thickness of arcs and Section C.4 discusses the color of arcs. Our implementation of the presented techniques is discussed in Section C.5. Finally, the empirical evaluation we did is presented in Section C.6.

## C.2 Bayesian networks

A Bayesian network is a probabilistic graphical network. It represents variables in a certain domain and visualizes the probabilistic relationships between them. These relationships can also be thought of as causal relationships. The formal definition of a Bayesian network is as follows [28]:

1. A set of random variables makes up the nodes of the network. Variables may be discrete or continuous.
2. A set of directed links or arrows connects pairs of nodes. If there is an arrow from node  $X$  to node  $Y$ ,  $X$  is said to be a parent of  $Y$ .
3. Each node  $X_i$  has a conditional probability distribution  $P(X_i|Parents(X_i))$  that quantifies the effect of the parents on the node.
4. The graph has no directed cycles (and hence is a directed, acyclic graph, or DAG).

A Bayesian network defines a complete joint probability distribution over  $X$  given by:

$$P(X_1, \dots, X_i) = \prod_{i=1}^n P(X_i|Parents(X_i)). \quad (C.1)$$

To further illustrate these concepts we will introduce an example network in Figure C.1 [28].

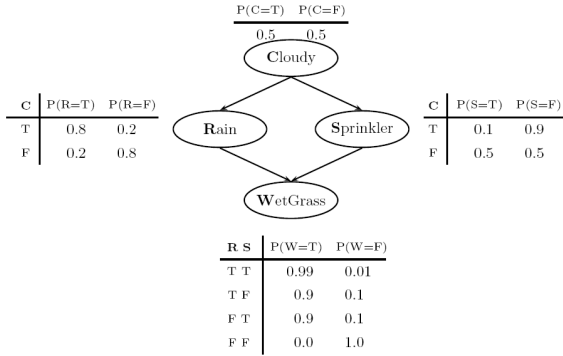


Figure C.1: An example Bayesian network.

It shows a Bayesian network with four nodes and a conditional probability table for each node. It models the following situation: Whether it is cloudy or not influences the chance that it rains and the chance that the sprinkler will be on. If it is cloudy the sprinkler will most likely not be on. The wetness of the grass is influenced by both the rain and the sprinkler. If there is rain and the sprinkler is on, the probability of the grass being wet is the highest, i.e., 0.99. If there is no rain and the sprinkler is off, it is certain that the grass is not wet, the probability is 1.0.

An arrow between two nodes indicates that the two nodes are dependent, meaning that they influence each other. If there is no arc present between two nodes, then they have no influence on each other, at least not directly. Also, if we see that, for example, the grass is wet, then we have *observed* the variable (or node) *WetGrass*, in which case it has become *evidence*, an *observation* or a *finding*. These three terms can be used interchangeably.

This network can be used to infer probabilities like that of the sky being cloudy when we know that the grass is wet but sprinkler is off, or the probability of the sprinkler being on when we know the grass is wet and there is no rain. We will explain why this is the case.

Every query about the domain, including the ones just posed, are specified by the full joint probability distribution  $P(\text{Cloudy}, \text{Rain}, \text{Sprinkler}, \text{WetGrass})$ . It consists, in this case, of  $2^4 = 16$  entries, the probability of every possible combination of variables is specified. The Bayesian network of Figure C.1 represents the exact same distribution, but only has nine probabilities specified in its conditional probability tables. There are eighteen numbers present, but all variables are binary and therefore the probability of one state is one minus the probability of the other state. So only nine numbers

are needed. This reduction is an important advantage of Bayesian networks and it is caused by the (conditional) independence assumptions made by the network. The larger the network or domain, the bigger the savings. Using Equation C.1 the joint probability distribution  $P(\text{Cloudy}, \text{Rain}, \text{Sprinkler}, \text{WetGrass})$  can be decomposed into:

$$P(\text{Cloudy}) \cdot P(\text{Rain}|\text{Cloudy}) \cdot P(\text{Sprinkler}|\text{Cloudy}) \cdot P(\text{WetGrass}|\text{Rain}, \text{Sprinkler}),$$

all of which are given in the model as conditional probability tables.

### C.3 Thickness of arcs

The information that we want to provide for a user by varying the thickness of arcs is the amount of influence one node has on the other. The approach by *BayesiaLab* uses the joint probability distribution, while *Elvira* determines the influence by looking at the conditional probability tables and determining the influence a parent node has on a child node. This approach to determine the influence of a parent node on a child node is static. This means that the calculations do not take into account any current observations. But it could be that, with certain observations, the influence of a certain parent node on a child node is significantly different from the observation-free situation, in which case the static information would be incorrect. The static information can be used to get a global impression of the interactions between the nodes, but it is not tailored to a certain situation.

Besides that, while it is true that a parent influences its child(ren) if it is observed, a child, when observed, can also influence the probability distribution of its parent(s). These two influences can be quite different from each other. To give an example, if a laptop is dropped from a high building, we almost know for certain that it will end up getting smashed into many pieces. But if we find a laptop that is smashed into many pieces, we cannot be just as sure about what caused this. It could have been dropped from a high building, but it could just as well have been run over by a car, or maybe someone got angry and stamped on it. So while the probability of “dropped from a high building” will increase when finding a smashed laptop, the probability will not increase as much the other way around, i.e., that of “laptop will get smashed” when we drop it from a high building.

Therefore, we are going to do this differently. Many networks contain one or more *target* or *hypothesis* nodes. See for example the Hepar II network [25] shown in Figure C.2, modeling various liver disorders.

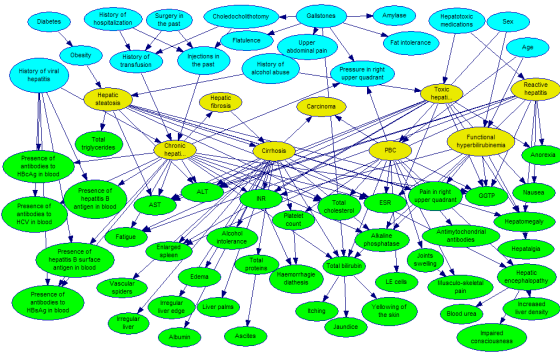


Figure C.2: The Hepar II network.

The yellow colored nodes represent the diseases. What we would be interested in most is the influence that the other nodes have on these disease, or *target*, nodes. When an arrow connects a target node with a non-target node, we will determine the influence the non-target node has on the target node, regardless of the direction of the arrow. When two non-target or two target nodes are connected by an arrow, we will, by default, use the average of the influences in both directions. In total there are four situations possible, shown in Figure C.3. In the first situation, shown in Figure C.3(a), we will visualize the influence  $A$  has on  $B$ . In the second situation, that of Figure C.3(b), we will consider the influence that  $B$  has on  $A$ . In the last two situations, depicted in Figures C.3(c) and C.3(d), we will consider the influence in both directions, and average them. The user will have the ability to override any of these default actions by specifying in which direction the influence for a particular arrow should be calculated.

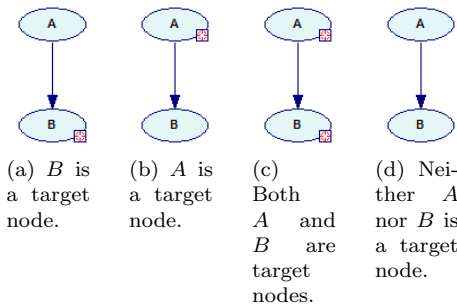


Figure C.3: Four different situations.

Furthermore, the approach that we are proposing here is a dynamic one. It considers the network in its current state, including any observations. It essentially

indicates how much potential influence a node has on a direct successor or predecessor, so the influence that a node could have if it was observed next.

Another advantage of our approach as opposed to the static one is the fact that in some situations it does not need to account for the *synergy* between the different parents of a node with more than one parent, simply because it is not there anymore in those situations. The definition of synergy can be given as: “the interaction of two or more agents or forces so that their combined effect is greater than the sum of their individual effects”. In case of a Bayesian network, this applies to the combined effect that the observation of multiple parents of one node can have on that node. The combined effect can be greater than the individual effects. This phenomenon cannot be accurately captured by varying the thickness of the arcs, which is one dimensional. In our dynamic approach, though, when all but one of the parents or children of a certain node are observed, i.e., there is no synergy anymore, we are able to accurately display the actual situation, because we are considering potential influences in the current state of the network. As soon as there is a change in the network, for example another observation is done, the thickness of an arc is recalculated if necessary.

We are going to determine the strength of the influence by looking at the posterior probability distribution of a node, for each possible state of the parent or child node, depending on the type of connection as discussed earlier in Figure C.3. For a node with  $n$  states, this will result in  $n$  potentially different posterior probability distributions of the connected node(s). We will compute the amount of difference between these distributions and base our final determination of the thickness of the arc on either the average of all the differences, the maximum of all the differences, or the weighted average. The weighted average is defined as

$$\sum_{i=0}^n a_i \cdot D(P(A), P(B|A = a_i)), \quad (C.2)$$

where  $A$  and  $B$  are two directly connected nodes,  $A$  has  $n$  states and  $D$  is a function measuring the distance between two distributions. As distance functions we are going to use the *Euclidean distance*, *Hellinger distance* [14], a normalized version of the *J-Divergence* [17, 18] and the *CDF distance* [19].

## C.4 Color of arcs

If a network, or a part of a network, consists of ordinal nodes, it is meaningful to determine the *sign of influence*. A node is ordinal if the states of that node are ordered in some way, for example from good to bad,

from large to small, from high to low or from desirable to not desirable. The sign of influence between a node  $A$  and a direct successor or predecessor  $B$  can be *positive*, meaning that higher values of  $A$  always lead to higher values of  $B$ , *negative*, meaning that higher values of  $A$  always lead to lower values of  $B$ , *null*, meaning that higher values of  $A$  always lead to values of  $B$  that are neither higher nor lower, or *ambiguous*, meaning that the influence is neither *positive*, *negative* nor *null*. In this context, 'always' refers to the fact that  $A$  can have more parents than just  $B$ , in which case the stated relations must hold for every possible configuration of the other parents of  $A$ , i.e., always. These definitions stem from the field of *qualitative probabilistic networks* [33], in which the relations between nodes in a network are not defined by conditional probability tables, but by signs of influence among nodes.

Up to now the approach to determine the sign of influence has been based on calculations using the conditional probability tables, which results in a *static* and local method. We will implement this feature but we will also extend it so that it is *context-specific* and non-local, i.e., taking into account any observed variables and indirect influences. We will discuss both in the following two sections. Eventually, the signs of influence will be visualized by adjusting the color of an arc.

### C.4.1 Static coloring

The static determination of the sign of influence uses the conditional probability tables of a Bayesian network. As mentioned earlier, there can be *positive*, *negative*, *null* or *ambiguous* influences. What type of influence is present for a certain arc is given by the following equations. There is a *positive* influence between a parent  $A$  and its child  $B$ , where  $C$  is the set of all parents of  $B$  except  $A$ , when the following two equations hold:

$$\forall a_i \forall a_j. a_i \geq a_j \Rightarrow \forall b \forall C (P(B \leq b|a_i, C) - P(B \leq b|a_j, C) \leq 0) ,$$

and

$$\exists a_i \exists a_j. a_i \geq a_j \Rightarrow \forall b \forall C (P(B \leq b|a_i, C) - P(B \leq b|a_j, C) < 0) .$$

There is a *negative* when the following two equations hold:

$$\forall a_i \forall a_j. a_i \geq a_j \Rightarrow \forall b \forall C (P(B \leq b|a_i, C) - P(B \leq b|a_j, C) \geq 0) ,$$

and

$$\exists a_i \exists a_j. a_i \geq a_j \Rightarrow \forall b \forall C (P(B \leq b|a_i, C) - P(B \leq b|a_j, C) > 0) .$$

There is a *null* influence when the following equation holds:

$$\forall a_i \forall a_j. a_i \geq a_j \Rightarrow \forall b \forall C (P(B \leq b|a_i, C) - P(B \leq b|a_j, C) = 0) .$$

If there is no *positive*, *negative* or *null* influence, the influence is *ambiguous*.

### C.4.2 Dynamic coloring

Our dynamic method for determining the sign of influence is, just like the thickness of arcs, context-specific and it takes into account any indirect influences. We are going to determine the sign of influence by looking at the posterior probability distribution of a node, for each possible state of the parent or child node. For a node with  $n$  states, this will result in  $n$  potentially different posterior probability distributions of the connected node(s). These posterior probability distributions will be compared using a slightly simplified version of the equations in the previous section. Between a node  $A$  and its child  $B$  there is a *positive* influence when the following two equations hold:

$$\forall a_i \forall a_j. a_i \geq a_j \Rightarrow \forall b (P(B \leq b|a_i) - P(B \leq b|a_j) \leq 0) ,$$

and

$$\exists a_i \exists a_j. a_i \geq a_j \Rightarrow \forall b (P(B \leq b|a_i) - P(B \leq b|a_j) < 0) .$$

There is a *negative* influence when the following two equations hold:

$$\forall a_i \forall a_j. a_i \geq a_j \Rightarrow \forall b (P(B \leq b|a_i) - P(B \leq b|a_j) \geq 0) ,$$

and

$$\exists a_i \exists a_j. a_i \geq a_j \Rightarrow \forall b (P(B \leq b|a_i) - P(B \leq b|a_j) > 0) .$$

There is a *null* influence when the following equation holds:

$$\forall a_i \forall a_j. a_i \geq a_j \Rightarrow \forall b (P(B \leq b|a_i) - P(B \leq b|a_j) = 0) .$$

If there is no *positive*, *negative* or *null* influence, the influence is *ambiguous*. The sign of influence can be determined, just like the thickness of an arc, in both directions, i.e., from the parent to the child and from

the child to the parent. If these two differ then the sign will also be regarded as being *ambiguous*.

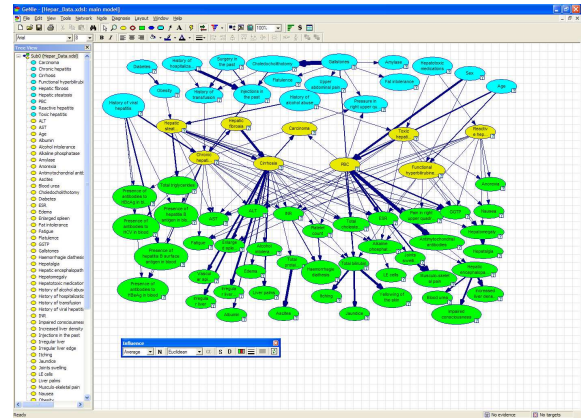
The dynamic method differs from the static method in that it is not local. The sign of influence in our dynamic method signifies the actual behaviour in the current situation, including all indirect influences and observations.

## C.5 Implementation

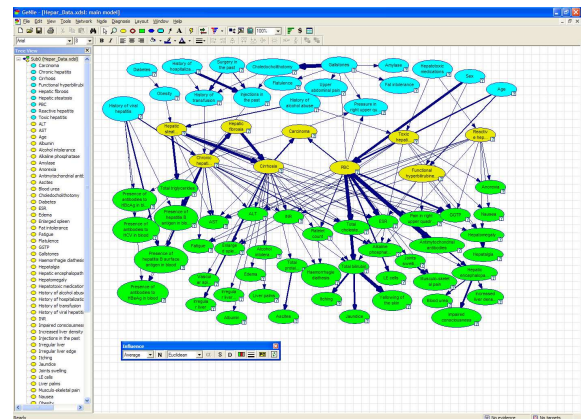
We have done an implementation of thickness and color of arcs in the C++ programming language. This implementation has been integrated into GeNIe, a development environment for building graphical decision-theoretic models developed at the Decision Systems Laboratory of the University of Pittsburgh. We created an extra module within GeNIe that provides all the functionality for the thickness and the color of the arcs. The signs of influence are visualized by adjusting the color of an arc. In that way a user can easily study the relations and pick out any arc that does not match with his or her belief. We will color positive arcs green, negative arcs red, ambiguous arcs purple and null arcs grey. Green is mostly associated with something positive or good, and red is mostly associated with something negative or bad. Grey for the null influence has been chosen because something that is not present or irrelevant is often “greyed out” in computer programs, therefore we think grey will be easy to associate with a null influence. The purple color for an ambiguous influence has been chosen because it fits nicely with the other colors.

The sign of influence between two nodes, visualized by the color of the arc, only has meaning if both nodes have some kind of ordering, i.e., are ordinal. The ordinality of a node can be set by accessing its property dialog box. There are three options: “low to high”, “high to low” and “none”. If the ordering of the outcomes is said to be “none”, then the color of the arcs connected to that node will be left at the default color of GeNIe, dark blue.

Figure C.4 shows GeNIe with the module activated and displaying the thickness of the arcs. Figure C.5 shows the coloring of the arcs, visualizing the signs of influences. It can be seen that a lot of ambiguous influences (purple arcs) in the static mode displayed in Figure C.5(a) are replaced by non-ambiguous influences in the dynamic mode, displayed in Figure C.5(b). This is due to the fact that the dynamic mode is context-specific and not purely local.

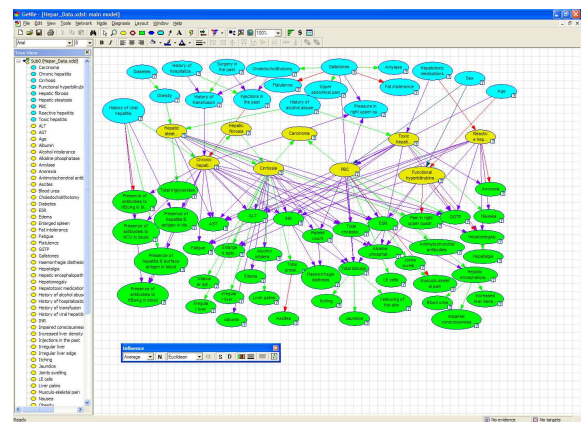


(a) Thickness of arcs in static mode.



(b) Thickness of arcs in dynamic mode.

Figure C.4: Thickness of arcs in the HEPAR II network.



(a) Color of arcs in static mode.





to two subjects. Three subjects mentioned that the different distance measures can be confusing. They say that it is not clear when to use which distance measure. The module is called a useful extra tool for model analysis and a useful tool to get a feel for the strength of influences between nodes. It is said to be helpful in understanding the relations among the variables in a network and that it can help in model building and decision making. Also, multiple subjects say the graphical user interface looks good and that the module is easy to use.

## C.7 Concluding remarks

This paper presented a technique to make inference in Bayesian networks more insightful. This has been done by automatically adjusting the thickness of the arcs in a Bayesian network to indicate the strength of influence between two directly connected nodes. Also, the color of an arc is automatically adjusted to indicate the sign of influence between the two directly connected nodes. These two visualizations can be done in a static way, using only the definition of a Bayesian network, but also in a dynamic way. The dynamic variant is, as opposed to the static method, context-specific, takes into account any indirect influences and is non-local.

A qualitative evaluation was performed by eight people, all familiar with Bayesian networks. They were asked a variety of questions, after having used the newly developed module of GeNIe. The reactions were very positive, both in terms of our technique giving more insight into a Bayesian network and in terms of easiness of use. The most important criticism, mentioned by quite a few subjects, was the unclarity about the available distance measures.

# Bibliography

- [1] Rev. T. Bayes. An Essay Toward Solving a Problem in the Doctrine of Chances. *Philosophical Transactions of the Royal Society of London*, 53:370–418, 1763.
- [2] J. Cheng and M.J. Druzdzel. AIS-BN: An adaptive importance sampling algorithm for evidential reasoning in large Bayesian networks. *Journal of Artificial Intelligence Research*, 13:155–188, 2000.
- [3] Elvira Consortium. Elvira: An environment for creating and using probabilistic graphical models. *First European Workshop on Probabilistic Graphical Models*, 2002.
- [4] G. F. Cooper. A method for using belief-network algorithms to solve decision-network problems. *Fourth Workshop on Uncertainty in Artificial Intelligence, Minneapolis, MN*, pages 55–63, 1988.
- [5] G.F. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42:393–405, 1990.
- [6] Luis M. de Campos, José A. Gámez, and Serafín Moral. Simplifying Explanations in Bayesian Belief Networks. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, Vol. 9, No. 4:461–489, 2001.
- [7] Marek J. Druzdzel. *Probabilistic Reasoning in Decision Support Systems: From Computation to Common Sense*. PhD thesis, Department of Engineering and Public Policy, Carnegie Mellon University, Pittsburgh, PA, 1993.
- [8] Marek J. Druzdzel and H.J. Suermondt. Relevance in Probabilistic Models: "Backyards" in a "Small World". *Working Notes of the AAAI-94 Workshop on Fall Symposium Series: Relevance*, pages 60–63, 1994.
- [9] R. Fung and K.C. Chang. Weighting and integrating evidence for stochastic simulation in Bayesian networks. *Proceedings of the Fifth Conference of Uncertainty in Artificial Intelligence*, 1989.
- [10] R. Fung and B. Del Favero. Backward simulation in Bayesian networks. *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pages 227–234, 1994.
- [11] I.J. Good. Explicativity: A mathematical theory of explanation with statistical applications. *Proceedings of the Royal Society of London*, 354:303–330, 1977.

- [12] H. Guo, P. Boddhireddy, and W. Hsu. An Ant Colony Optimization Algorithm for the Most Probable Explanation Problem. *The 17th Australian Joint Conference on Artificial Intelligence*, 2004.
- [13] Peter Haddawy, Joel Jacobson, and Charles E. Kahn Jr. BANTER: A Bayesian network tutoring shell. *Artificial Intelligence in Medicine*, Vol. 10:177–200, 1997.
- [14] E. Hellinger. *Die Orthogonalinvarianten quadratischer Formen von unendlich vielen Variablen*. PhD thesis, University of Göttingen, 1907.
- [15] M. Henrion. Propagation of uncertainty by Bayesian networks by probabilistic logic sampling. *Uncertainty in Artificial Intelligence*, 2:149–163, 1988.
- [16] R.A. Howard and J.E. Matheson. Influence diagrams. *Readings on the Principles and Applications of Decision Analysis*, 2:721–762, 1984.
- [17] H. Jeffreys. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London*, 186:453–454, 1946.
- [18] D. Johnson and S. Sinanovic. Symmetrizing the Kullback-Leibler distance. *IEEE Transactions on Information Theory*, 2001.
- [19] P.C. Kraaijeveld. GeNIeRate: An Interactive Generator of Diagnostic Bayesian Network Models. Master’s thesis, Man-machine interaction group, Delft University of Technology, 2005.
- [20] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.
- [21] Carmen Lacave, Roberto Atienza, and Francisco J. Díez. Graphical Explanation in Bayesian Networks. *Lecture Notes In Computer Science; Proceedings of the First International Symposium on Medical Data Analysis*, Vol. 1933:122–129, 2000.
- [22] Carmen Lacave and Francisco J. Díez. Explanation for causal Bayesian networks in Elvira. *Proceedings of the Workshop on Intelligent Data Analysis in Medicine and Pharmacology (IDAMAP-2002)*, pages 47–52, 2002.
- [23] Carmen Lacave and Francisco J. Díez. A review of explanation methods for Bayesian networks. *The Knowledge Engineering Review*, 17:107–127, 2002.
- [24] David Madigan, Krzysztof Mosurski, and Russell G. Almond. Graphical Explanation in Belief Networks. *Journal of Computational and Graphical Statistics*, Vol. 6, No. 2:160–181, 1997.
- [25] Agnieszka Oniśko, Marek J. Druzdzel, and Hanna Wasyluk. Extension of the Hepar II model to multiple-disorder diagnosis. *Intelligent Information Systems, Advances in Soft Computing Series*, pages 303–313, 2000.
- [26] J. Pearl. Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, 29(3):241–288, 1986.

- [27] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1988.
- [28] S.J. Russell and P. Norvig. *Artificial Intelligence A Modern Approach*. Pearson Education International, 2003.
- [29] P. Sember and I. Zukerman. Strategies for generating micro explanations for bayesian belief networks. *Uncertainty in Artificial Intelligence*, 5:295–302, 1990.
- [30] R.D. Shachter and M.A. Peot. Simulation approaches to general probabilistic inference on belief networks. *Proceedings of the Fifth Conference of Uncertainty in Artificial Intelligence*, 1989.
- [31] H. J. Suermondt. *Explanation in Bayesian Belief Networks*. PhD thesis, Departments of Computer Science and Medicine, Stanford University, Stanford, CA, 1992.
- [32] Kardi Teknomo. Similarity measurement. <http://people.revoledu.com/kardi/tutorial/Similarity/>.
- [33] M.P. Wellman. Fundamental concepts of qualitative probabilistic networks. *Artificial Intelligence*, 4:257–303, 1990.
- [34] C. Yuan and M.J. Druzdzel. An Importance Sampling Algorithm Based on Evidence Pre-propagation. *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence*, pages 624–631, 2003.
- [35] Changhe Yuan, Tsai-Ching Lu, and Marek J. Druzdzel. Annealed MAP. *Proceedings of the 20th Annual Conference on Uncertainty in Artificial Intelligence*, pages 628–635, 2004.