



Building Bayesian Networks from Data: a Constraint-based Approach

Thesis submitted in November 2001
for the degree of Doctor of Philosophy

by

Nicandro Cruz Ramírez

Department of Psychology. The University of Sheffield

Abstract

The main goal of a relatively new scientific discipline, known as Knowledge Discovery in Databases or Data Mining, is to provide methods capable of finding patterns, regularities or knowledge implicitly contained in the data so that we can gain a deeper and better understanding of the phenomenon under study. Because of the very fast growing nature of information, it is necessary to propose novel approaches in order to process this information in a quick, efficient and reliable way. In this dissertation, I use a graphical modelling data mining technique, called a Bayesian network, because of its simplicity, robustness and consistency in representing and handling relevant probabilistic interactions among variables of interest. Firstly, I present an existing algorithmic procedure, which belongs to a class of algorithms known as constraint-based algorithms, that builds Bayesian networks from data based on mutual information and conditional mutual information measures and its performance using simulated and real databases. Secondly, because of the limitations shown by this algorithm, I propose a first extension of such a procedure testing its performance using these same datasets. Thirdly, since this improved algorithm does not show in general a good performance either, I propose a final extension, which provides interesting and relevant results on those same databases, comparable to those of two well-known, accurate and widely tested Bayesian network algorithms. The results show that this final procedure has the potential to be used as a decision support tool that could make the decision-making process much easier. Finally, I evaluate in detail the real-world performance of this algorithm using a database from the medical domain comparing this performance with those of different classification techniques. The results show that this graphical model might be helpful in assisting physicians to reach more consistent, robust and objective decisions.

To Cristina: there are no words to thank you for everything...

Acknowledgements

I am very grateful to CONACyT (National Council for Science and Technology – Mexican Federal Government) who has given me the economic support for studying my PhD, scholarship number 70356.

I am also very grateful to the following people who, in one way or another, have helped me in achieving this goal:

- Dr. Jon May and Prof. Rod Nicolson (supervisors)
- Dr. Simon Cross
- Prof. Mark Lansdale and Dr. John Porrill (examiners)
- Prof. John Mayhew
- Dr. Manuel Martínez Morales
- Nicandro Cruz, Maria Luisa Ramírez, Caridad Cruz and Ana Sofía Juárez
- All my family and old and new colleagues and friends

Contents

1	Antecedents	1
1.1	Introduction	1
1.2	Causal Induction: perspectives from Psychology and Computer Science	6
1.2.1	Psychological approach to Causal Induction	6
1.2.2	Perspective from Computer Science to Causal Induction	18
1.3	Computational aids for handling information accurately and effectively: graphical models	24
1.4	Automatic classification: Data Mining or Knowledge Discovery in Databases	26
1.5	Classification, prediction, diagnosis and decision-making	36
2	Background	40
2.1	Basic concepts on Probability and Graph Theories	40
2.2	Axiomatic characterizations of probabilistic and graph-isomorph dependencies	51
2.3	Bayesian Networks	54
2.4	Representation of uncertainty, knowledge and beliefs in Bayesian Networks	72
3	Learning Bayesian Networks	76
3.1	Typical problems in constructing Bayesian Networks	76
3.2	Traditional approach	78
3.3	Learning approach	79
3.4	Learning Bayesian Networks from data	82

3.4.1	Constraint-based methods	82
3.4.2	Search and scoring based algorithms	90
3.4.3	Advantages and disadvantages of constraint-based algorithms and search and score algorithms	96
3.5	Combining constraint-based methods and search and scoring based methods: a hybrid approach	99
4	Bayes2: a constraint-based algorithm for constructing Bayesian networks from data	105
4.1	Information measures used as independence tests	105
4.2	Bayes2: a first algorithm to build Bayesian Networks from data	112
4.2.1	Description of the Bayes2 algorithm	113
4.3	Experimental results	117
4.3.1	Discussion of the results	125
4.4	Goodness of fit	130
4.4.1	The MDL criterion	133
5	Bayes5: extensions and improvements of Bayes2	141
5.1	Improvements of Bayes2	141
5.2	Description of Bayes5	141
5.3	Experimental results	143
5.3.1	Discussion of the results	144
6	Bayes9: extensions and improvements of Bayes5	147
6.1	Improvements of Bayes5	147
6.2	Description of Bayes9	147
6.3	Experimental results	150
6.3.1	Discussion of the results	151

6.4 Comparison of the performance of Bayes2, Bayes5 and Bayes5 using the MDL criterion	153
6.4.1 Discussion of the MDL results	156
7 A comparison of the performance of three different algorithms that build Bayesian Networks from data	159
7.1 Tetrad II	159
7.2 Power Constructor	160
7.3 Experimental results among Tetrad II, Power Constructor and Bayes9	165
7.3.1 Discussion of the results	165
7.4 Goodness of fit	166
8 Applications	169
8.1 Background of a real-world database from medicine	169
8.2 Tests for measuring accuracy	171
8.3 Experimental results of Bayes9	178
8.4 Discussion of the results	185
8.4.1 Human performance vs. Bayes9	187
8.4.2 Logistic regression vs. Bayes9	189
8.4.3 Decision trees vs. Bayes9	194
8.4.4 MLPs vs. Bayes9	196
8.4.5 ARTMAPs vs. Bayes9	199
8.4.6 ROC curves by logistic regression, MLPs and Bayes9	202
8.4.7 Performance of Tetrad II, Power Constructor and Bayes9 on the breast cancer dataset	203

9 Discussion	213
10 Appendix	A
11 Bibliography	222

Chapter 1

Antecedents

This chapter presents the main ideas from the field of Psychology and Computer Science that support the theoretical and pragmatic aspects of this thesis. It also describes some computational tools to handle information contained in databases accurately and effectively. Finally, it shows how to use these tools to perform important tasks such as prediction, diagnosis and decision-making in order to provide solutions to certain complex problems.

1.1 Introduction.

The central idea for this research began with the motivation of extracting hidden useful knowledge from databases in order to represent and understand some phenomena in the world in an easy, consistent, powerful and beautiful way. That is why the approach of Bayesian networks was chosen. Indeed, this approach is guided by the natural appeal of **Occam's** razor: the best model to explain a phenomenon is the simplest one without losing the adequacy. Of course it is necessary to give precise details about what simplest and adequacy mean. As this thesis progresses, these concepts will be explained.

The main goal of this work is to provide human experts in a certain knowledge area (in this case medicine) with a computational tool to help them discover the underlying principles, mechanisms and causes that govern the phenomenon under study. Once this is done, they can perform important actions such as prediction, diagnosis, decision-making, control and of course a better understanding of the phenomenon being modelled.

First of all, let us describe very briefly what the main task of human experts is: they are to solve very complex problems in a specific domain by using their knowledge obtained

through their academic and research training and through their everyday experience. A computer program capable of producing similar solutions to those obtained by the human experts is called an **expert system**. That is to say, the main goal for an expert system is to reach judgments very similar of those reached by human experts no matter whether or not the system follows the same reasoning process as the human (the final result and not the process is what really matters).

Human experts usually look at a part of the world where a certain complex problem is presented to them. Then, mainly by means of observations and using their expertise, they take actions and propose good solutions, most of the time, to that problem.

An expert system can be used as a support tool to help the human experts process and represent the information coming from a particular part of the world in a more suitable way that permits them to identify the possible solution or solutions to the corresponding problem much more easily. Figure 1.1 presents a slightly modified idea proposed in (Jensen 1996).

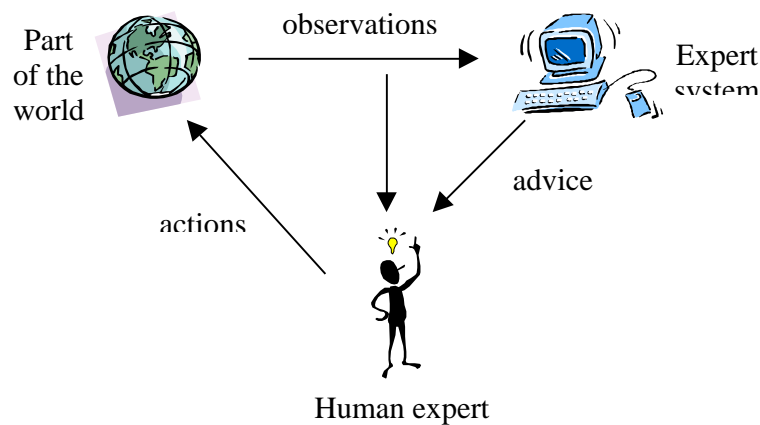


Figure 1.1: An expert system as a support tool for the human expert

It can be argued that even human experts need the support of computers in order for them to perform reliable, fast and accurate calculations that, most of the time, imply the incorporation of uncertainty. In other words, it is not an easy task at all to find out the underlying causes of a determined problem just looking at the data by simple inspection, as figure 1.2 suggests.

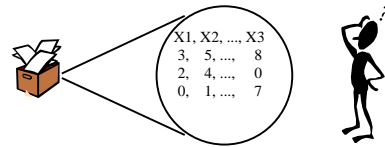


Figure 1.2: Even a human expert finds it very difficult to discover implicit relationships among variables in a database

Because an expert system has to produce similar solutions to those achieved by human experts, it has somehow to incorporate part of their knowledge in the form of a program. Also, this kind of system has to have the capability of dealing with uncertainty since the very nature of complex problems is often of a nondeterministic type for many reasons: uncertain observations, incomplete data, difficulties in measuring the variables, etc. The typical construction of such a system involves a very complex and time consuming task due to many well-known factors that go from the extraction of the experts' knowledge (who do not even themselves know exactly how it is organised) to the problem of understanding, translating and encoding this knowledge in a computer program (Jackson 1990). The reader is referred to Jackson (1990) for an excellent introduction to the topic of expert systems.

An emerging discipline called **Knowledge Discovery in Databases (KDD)** or **Data Mining (DM)** has appeared in order to solve the classical problems presented within the typical approach for constructing expert systems as pointed out above. This discipline argues that knowledge might be implicitly contained (i.e. hidden) in data and combines many ideas and techniques from a variety of areas such as databases, statistics, machine

learning and artificial intelligence, among others, in order to extract that knowledge which can probably be in the form of probabilistic causal relationships or rules. KDD is also useful to cope with the continuing and fast growing nature of information, processing it in an efficient, accurate and reliable way.

This is what this work is all about: a computer program that represents uncertain knowledge from databases in the form of a graph. The approach taken here, which has already been successfully used to build expert systems, is known as **Bayesian Networks (BN)** as well as under some other names: **Probabilistic Networks, Influence Diagrams, Belief Networks** and **Causal Networks** (Pearl 1988; Neapolitan 1990; Heckerman, Mandani et al. 1995). A Bayesian network is a graphical model that encodes probabilistic relationships among variables in a determined problem. An example of a Bayesian network is depicted in figure 1.3 (Pearl 1996). In chapter 2, the main concepts, definitions and the syllabus of what a BN is, will be reviewed.

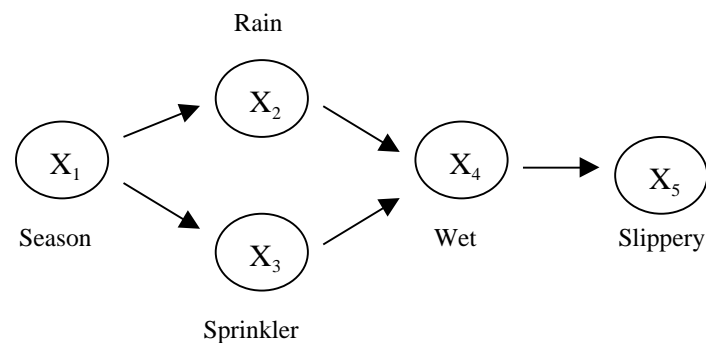


Figure 1.3: A Bayesian network showing the probabilistic relationships among season, rain, sprinkler and wetness and slipperiness of the pavement

Bayesian networks have an intuitive appeal and also are a very powerful tool for representing uncertainty, knowledge and beliefs. As can be seen from figure 1.3, the probabilistic relationships among the variables are captured and represented explicitly

(graphically) in an easily understandable way. It can also be argued that sometimes these relations could be of causal nature so that one is able to perform relevant actions such as decision-making, prognosis, diagnosis and control in order to solve a given problem. In this particular example, the season causes either the rain to be present or the sprinkler to be on with certain probability; if it is either raining or the sprinkler is on, then either of them (or both) can cause the pavement to be wet and finally this wetness can probably cause the pavement to be slippery. Knowing the probabilities of some of the variables allows us to predict the likelihood of, say, X_5 (slipperiness) by a schema called **probability propagation**. This schema uses Bayes' rule to update each node probability given that some of the nodes in the network are instantiated.

Figure 1.4 resembles figure 1.2 and illustrates how a BN could shed some light to the human experts to help them find a possible solution of a given problem.

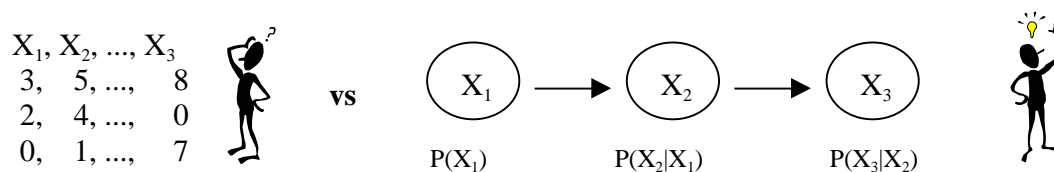


Figure 1.4: An example showing how helpful a Bayesian network can be in representing the relationships among variables

The ideas from the field of **Knowledge Discovery in Databases** can be incorporated in this framework so that the structure of a BN can be then induced from a database. Many algorithms have been proposed for constructing a BN from a database alone, from the experts' knowledge alone or from a combination of the experts' knowledge at hand and a database. There are pros and cons in each approach that will be explained in chapter 3. As mentioned before, Bayesian networks represent probabilistic relationships among the variables taking part in a determined problem. In doing so, it can be argued, some of these probabilistic relations could be of causal nature permitting one to perform prognostic

reasoning, diagnostic reasoning and control. In order to extract causal relationships from data, it is very important to review some of the theories that try to deal with this problem. These are theories coming from the field of Psychology and Computer Science and will be presented in the following subsections.

1.2 Causal Induction: perspectives from Psychology and Computer Science.

In order to explain how causal knowledge is acquired, a fierce debate originated a long time ago. The problem of causal induction was firstly posed by the great philosopher David Hume in 1739 (Cheng 1997) and continued by many other philosophers, psychologists, statisticians and computer scientists to date (Einhorn and Hogarth 1986; Pearl 1988; Cheng 1993; Spirtes, Glymour et al. 1993; Waldmann 1996; Cheng 1997; Pearl 2000). Here, two perspectives will be discussed from two different areas: Psychology and Computer Science.

1.2.1 Psychological approach to Causal Induction.

From the psychological point of view the main interest is to know how humans represent and acquire causal knowledge. Psychologists have adopted two different theories in order to explain such a phenomenon: the **Humean** and the **Kantian** approaches.

a) The Humean approach.

Generally speaking, Hume (Cheng 1993; Cheng 1997; Waldmann and Hagmayer 2001) tried to explain the phenomenon of causal induction in terms of **covariation**. He proposed that the knowledge that one thing (a potential cause) can cause (or prevent) another (the resultant effect) is acquired by means of experience and assuming no prior knowledge. The acquisition of causal knowledge through experience can then be captured by the notion of relative frequencies or probabilities. According to Hume, there are three required conditions in order to identify a potential cause of a certain effect: **temporal** and **spatial contiguity**, **precedence** of the cause in time and the constant occurrence of the

cause and the effect called **constant conjunction**, which can be represented, as said before, by the notion of covariation (Einhorn and Hogarth 1986; Cheng 1993; Waldmann 1996; Cheng 1997; Lien and Cheng 2000). A very intuitive, appealing and beautiful equation for expressing a causal relationship in terms of statistical relevance is the well-known **delta p** rule (ΔP):

$$\Delta P = P(e|c) - P(e|\sim c) \quad (1.2.1)$$

where P represents the (unconditional) contingency between a candidate cause c and an effect e . $P(e|c)$ represents the probability of that effect given that the candidate cause is present and $P(e|\sim c)$ represents the probability of the effect given that the candidate cause is absent. In general if $P \gg 0$ then c can be regarded as a generative cause of e ; if $P \ll 0$ then c can be regarded as a preventive cause of e . Finally if $P = 0$ it can be said that c is not a cause of e . However, one of the main drawbacks of this approach is that correlation does not always imply causation. The various possibilities for this formula will be illustrated with some examples. For the case when $P \gg 0$ imagine that the following classic scenario is given.

In a certain clinic, a number of patients have developed lung cancer (effect). There is a common feature in the majority of patients with this disease: they are strong smokers (potential generative cause). This means that few patients do not smoke but have lung cancer as well. The problem is to find out whether smoking causes lung cancer. Suppose that $P(e|c) = 0.7$ and $P(e|\sim c) = 0.3$. Applying formula 1.2.1, the calculation yields:

$$P = 0.7 - 0.3 = 0.4$$

The rule when $P \gg 0$ applies so the conclusion is that smoking is a potential causal factor for developing lung cancer.

For the second case, when $P \ll 0$, suppose the next given scenario. In a certain hospital, a vaccine (potential preventive cause) against headaches (effect) is being tested. A

sample of patients is taken and the probabilities are $P(e|c) = 0.3$ and $P(e|\sim c) = 0.7$. Again, applying formula 1.2.1, the calculation yields:

$$P = 0.3 - 0.7 = -0.4$$

Now, the rule when $P \ll 0$ applies so the conclusion is that the vaccine is effective most of the time for preventing headaches.

For the last case, when $P \approx 0$, imagine that the next scenario is given. In a certain factory, many workers have developed a strange disease in their eyes. Some studies have been conducted in order to determine the potential cause that could be damaging their eyes and a possible conclusion has been reached: the consumption of garlic is the probable cause. Now, it is necessary to test whether this conclusion is true or false. To do so, a sample of workers in the factory is taken and the probabilities found are $P(e|c) = 0.5$ and $P(e|\sim c) = 0.49$. Applying formula 1.2.1, the calculation yields:

$$P = 0.5 - 0.49 = 0.01 \approx 0$$

The rule that applies for this case is the last one when $P \approx 0$ so it can be concluded that the potential causal factor (the consumption of garlic) is in fact noncausal. Hence, it is necessary to conduct a more profound study to determine the true cause of the eye disease.

In the three examples mentioned above, theoretical ideal situations can be easily identified where it is possible to assert that a potential factor is a generative cause, a preventive cause or simply a noncausal one. However, as stressed earlier, covariation does not, in general, imply causation and therefore some important problems can be found within this approach, as pointed out in the next examples.

The first example is taken from Diez and Nell (1998/99). A certain study carried out in England demonstrated that there was a strong correlation between the number of storks

in each locality and the number of children births. Suppose $P(e|c) = 0.8$ and $P(e|\sim c) = 0.3$ yielding

$$P = 0.8 - 0.3 = 0.5$$

So, given the result, a probable (but not plausible) hypothesis would be that the storks bring the children. Is this hypothesis not very odd to explain the increasing number of births? If the formula 1.2.1 is just applied, then this hypothesis could be taken as true. But somehow humans know that this hypothesis makes no sense at all and therefore we need to look for some other possible and, above all, plausible answers. The end of this example is that there exists a more reasonable alternative: the number of inhabitants of a locality influences the number of churches (the bigger the population, the bigger the number of churches). Hence, on the one hand, there are more belfries where the storks can build their nests and on the other, there is a strong correlation between the number of inhabitants and the number of births.

The second example is taken from Cheng (1997). A person is allergic to certain foods and her skin reacts to them showing hives. Then, she decides to go to the doctor to check to which of these foods she is allergic. The doctor has to make some scratches on her back and put various samples of foods on these scratches to test which foods are causing the allergy. After few minutes the doctor sees that on every scratch hives break out, i.e. $P(e|c) = 1$ (where e corresponds to the hives and c to every sample of food). However, it is discovered that the patient is also allergic to the scratches on her skin, which means that for every scratch alone hives break out as well, i.e., $P(e|\sim c) = 1$. Applying the formula, the calculation yields:

$$P = 1 - 1 = 0$$

From the result, it would be possible to say that neither food is causal. However, according to this situation and the doctor's experience, the doctor does not conclude that the patient is not allergic to either of them. When this kind of situation is presented to humans,

they would then say, most of the time, to be uncertain of the noncausal nature of the factors involved. In other words, under these circumstances, people usually feel undecided to reach a plausible conclusion about the causal nature of the factors.

The third and last example is taken from Cheng too (1997). Suppose that a study to test the efficacy of a drug for relieving headaches is being carried out. There are two different groups, the experimental group and the control one. In the experimental group, the subjects are administered with the drug while in the control group the subjects are administered with a placebo. If the (unconditional) contingency for both groups is the same, i.e., $P = 0 - 0 = 0$, then no difference in the occurrence of headaches can be perceived between the two groups. This would imply, using formula 1.2.1, that the drug is ineffective.

But before confirming that the drug does not work, it is found that the subjects in the control group did not have headaches either before or after taking the placebo. In order to have a sound conclusion, it is necessary that some of the participants in both the experimental and control groups have headaches in order to test the effectiveness of the drug. Thus, from this fact, it is plausible to conclude that the study is uninformative. Once more, under these conditions, humans would be uncertain to assert that the drug does not work and hence prefer to express their uncertainty to produce a reasonable conclusion about the causal nature of the factors.

From these examples, it can be said that somehow, somewhere, a certain kind of knowledge is required to deal under these extreme conditions so that plausible solutions can be proposed or at least, to declare that these solutions cannot be offered because of the contradictory information available.

In order to overcome these different problems to which covariation leads, another different point of view was proposed by the philosopher Kant in 1781 (Cheng 1997). The main features, advantages and disadvantages of such an approach will be reviewed in the next subsection.

b) The Kantian approach.

At the heart of this approach is the very notion of **causal power**. A very good and detailed psychological account of this alternative approach can be found in Bullock et al. (1982). Basically, the notion of causal power refers to the idea or knowledge that some mechanism, source or power has the ability to produce a certain effect; this very knowledge is commonly referred to as **prior knowledge**. According to this view, prior knowledge has the property to overcome the problems that the covariation approach is limited to solve: although a factor covaries with the effect, it is possible to distinguish (in many cases), based on this prior knowledge, **spurious** causes from genuine ones. This term of spurious causes is due to Suppes (Pearl 1996; Lien and Cheng 2000).

So, the idea of the existence of a causal mechanism having the power of producing an effect by means of a physical power (visible or invisible) either directly or indirectly (i.e. through a set of intermediary events) is central to this approach.

This idea will be illustrated with an example taken from Bullock et al. (1982). Imagine that while at home, a family observes a window shattering. It seems very reasonable and plausible for them to find out what caused the window to break. Hence they will look for possible objects that could have broken the window such as a ball, a rock, a bullet, etc. If they, for instance, found any soft object like a sponge, then this object would not be taken into account as a potential cause because their prior knowledge would be telling them that the soft object normally does not cause a window shattering. If incapable of finding the mechanism responsible of producing the breakage, they would prefer to confess their ignorance or uncertainty of what caused the window shattering. Recall the three examples of the last subsection. In the first example, somehow some prior knowledge was already there telling that the storks do not bring the children; in the second example, the doctor's prior knowledge acted as a guide to conclude that the patient was probably allergic to some but not all the foods. Finally, in the third example, the fact that some patients in the experimental and the control group did not show headaches, before and after the administration of the drug and the placebo respectively, was a key point for concluding that the results were uninformative.

However, the power view does not explain how humans come to know that some factors are potential causes whereas some others are simply disregarded because of their lack of power to be probable causes. Hence, a very important question arises: **how** do humans **acquire** that prior knowledge that permits them to recognise, in most of the cases, genuine causes from spurious ones? Recall that in the Kantian approach, it is assumed that causal learning is primarily guided by prior knowledge about causal powers, sources or mechanisms. But now, another question comes out: how do they come to know the causal nature of those mechanisms? As can be noticed, a **circularity** problem appears here. In sum, the power view tries to go one step back but, at the end, gets entangled by its own circularity and fails to provide a plausible explanation.

It is also necessary to recall that the power view was originated by the problems encountered within the covariation approach. But one of the main problems is still there: unless causal prior knowledge is innate, it has to be somehow acquired by means of observable events (Cheng 1993; Cheng 1997; Lien and Cheng 2000; Waldmann and Hagmayer 2001). Moreover, it can be argued that, according to Marr's distinction (Marr 1982), the Kantian view has no definition at the computational level (Cheng 1997) which means that this approach does not provide an explanation of what function has to be computed and why that function is computed. From the advantages and disadvantages of these two classic psychological theories, namely, the Humean and the Kantian theories, some researches have come up with the idea of combining and integrating them into a theory in order to eliminate the problems found in each of them. The next subsection explains one of these theories that appears to be **normative**. A normative theory refers to a theory considered rationally correct (Perales and Shanks 2001).

c) An integration of the Humean and Kantian approaches.

Both approaches per se have appealing characteristics as well as disadvantages. Because it seems that neither of them is complete, it appears reasonable and sound in trying to integrate these two approaches to overcome their intrinsic difficulties mentioned in the two previous subsections. Some different directions have emerged (Einhorn and Hogarth 1986; Cheng 1997; Chase 1999; Lien and Cheng 2000) but only one of them will be briefly

discussed here because of its importance, beauty and powerful nature: the **Power PC Theory** (Cheng 1997). Power PC is the short for causal power theory of the probabilistic contrast model.

As pointed out at the end of the last subsection, if the causal prior knowledge is not innate, then cause-effect relationships have to be, somehow, extracted from direct observations. The key question is exactly **how** to extract those relationships from the available data. Cheng proposed to combine the main advantages of the two approaches (the covariation and the causal power) to overcome the problems presented in both of them. She formalised then her Power PC Theory "by postulating a mathematical concept of causal power and deriving a theory of a model of covariation based on that concept" (Cheng 1997, pp. 369 and 370). The main distinction she made in this paper is that of the relation between laws and models (observable events) in science and the theories (unobservable entities) used to explain such models. This relation can be mapped onto covariation information (observable events) and causal powers (unobservable entities) that discriminate such information. In other words, people can extract, most of the time, useful and correct causal information from data according to their beliefs or knowledge. How can this be reflected by means of a set of algebraic equations?

First of all, it is very important to stress that the Power PC Theory focuses on how a **simple** cause, independently of others, can produce an effect by itself; i.e., it is assumed that the effect is not produced by a joint combination of causes. Another important thing that this theory takes into account is the selection of a **focal set** of possible causes rather than the selection of the universal set. The universal set within a determined experiment is the whole set of events presented in this very same experiment. It is very important to bear in mind that people taking part in such an experiment can easily take into account some other factors (their focal set) that can be not even included within the universal set. These factors are normally those that they believe have potential for being causal. For example, it is often heard that a short circuit can be the cause of a house fire. People do not normally think of the oxygen as being the cause of the fire although it is necessary to start it. In this case, it is possible to establish that the oxygen is merely an enabling condition and the short

circuit is indeed the cause of the fire because in another focal set, say, when oxygen is absent (e.g. in a vacuum chamber), a short circuit will not produce a fire. With this distinction in mind, equation 1.2.1 represents classically, as noticed before, the unconditional contrast whereas equation 1.2.2 represents the **conditional contrast** (Cheng 1993), a generalisation of the former:

$$\Delta P_c = P(e | c, k_1, k_2, \dots k_n) - P(e | \sim c, k_1, k_2, \dots k_n) \quad (1.2.2)$$

P_c represents now the conditional contingency between a candidate cause c and an effect e keeping the alternative causal factors $k_1, k_2, \dots k_n$ **constant**. The same criteria as in equation 1.2.1 apply for p values. It is not necessary at all to know what those alternative causal factors are but only to know that they occur independently of the potential causal factor c . The difference, with respect to equation 1.2.1, is now that the all possible combinations of the presence and absence of the alternative causes can be, in theory, explored and hence computed.

Equation 1.2.2 gives one of the most important clues for constructing the two equations that conform the Power PC Theory: one for explaining the generative causal power (eq. 1.2.3) of a cause and the other for explaining the preventive causal power of such a cause (1.2.4).

$$p_c = \frac{P_c}{1 - P(e | \sim c)} \quad (1.2.3)$$

$$p_c = \frac{- P_c}{P(e | \sim c)} \quad (1.2.4)$$

For equation 1.2.3, for all c , $0 \leq p_c \leq 1$; where p_c represents the power of the cause c to produce the effect e and P_c represents the conditional contrast (eq. 1.2.2). For equation 1.2.4, for all c , $-1 \leq p_c \leq 0$; where p_c represents the power of the cause c to prevent the effect e and P_c represents the conditional contrast (eq. 1.2.2). The minus symbol of P_c in

equation 1.2.4 makes, in general, the overall result negative to capture the preventive nature of the cause c .

Now, let us return to the some of the examples shown in subsection 1.2.1 that proved difficulties. In the example about the allergy to foods, $P(e|c) = 1$ and $P(e|\sim c) = 1$ so $P_c = 0$. Do not forget that the alternatives causes are kept constant. If equation 1.2.3 is applied (because what is wanted to be known is whether some foods have generative causal power) then the result yielded is:

$$p_c = \frac{P_c}{1 - P(e|\sim c)} = \frac{0}{0} = \textit{undefined}$$

Under this boundary condition, the Power PC Theory would say, as in the above result, that the causal power of c cannot be interpreted or is undefined. This can be taken as the indecision of the doctor to conclude that none of the foods is causing the allergy. As can be noticed, this new result is a significant difference with regard to the result of only applying equation 1.2.1 which would say, according to the result it yields, that all foods are noncausal.

In the example about the test of the drug for curing headaches, $P(e|c) = 0$ and $P(e|\sim c) = 0$ so $P_c = 0$. If now, equation 1.2.4 is applied (because what is wanted to be known is whether the drug has preventive causal power) then the result obtained is:

$$p_c = \frac{-P_c}{P(e|\sim c)} = \frac{0}{0} = \textit{undefined}$$

Under this other boundary condition, the Power PC Theory would say that, according to the covariation information at hand, it is not possible to reach a conclusion of the preventive causal power of c . The more plausible conclusion is to say that the study is uninformative instead of, if applying equation 1.2.1 alone, saying that the drug is ineffective.

Out of these boundary conditions, namely, that for a generative cause (when $P(e|c) = 1$ and $P(e|\sim c) = 1$) and that for a preventive cause ($P(e|c) = 0$ and $P(e|\sim c) = 0$), formulas 1.2.3 and 1.2.4 give a conservative estimate of the causal power p_c . It is also very important to observe that, according to the Power PC Theory's equations 1.2.3 and 1.2.4, are exactly opposite to each other.

However, if looked carefully, although the Power PC Theory is to date one of the most complete theories for explaining the phenomenon of causal induction, it still has some important limitations. First of all, Power PC Theory only deals with how a simple cause, independently of each others, can cause a certain effect. This means that this theory does not account for the case when necessarily a joint combination of causes is responsible for producing the effect. If this happens, none of the above equations can be applied.

Another problem is given when the conditional contrasts cannot be computed because the information to do so is unavailable and therefore, according to which is the case, neither the equation 1.2.3 nor 1.2.4 can be calculated. Under these circumstances, humans are still able to produce a reasonable answer about what caused the effect. For instance, in the story about the high correlation between the number of storks and the number of children births, it can be argued that the probability contrast computed was actually the unconditional one (eq. 1.2.1). If the conditional probabilistic contrast is now computed (eq. 1.2.3) and some other possible alternative causal factors such as the number of belfries, number of churches and number of inhabitants in that region are taken into account, then it could be indeed noticed that the number of storks may no longer covary with the number of children births. However, it can be certainly very difficult to find some instances where these alternatives causes occur independent of the potential cause being considered and because of this, the formulas cannot be computed. But because of our prior knowledge, we still know that the conclusion of storks bringing children makes no sense at all.

According to Marr's classification (Marr 1982), Power PC Theory has a definition at the computational level. Hence, it describes what function is required and why that function

is appropriate to be computed. However, it assumes that somehow an asymptotic behaviour has already been reached without saying how that was done. In other words, to date, there exists no algorithm describing how to compute that function; this suggests that such a task can indeed be very complicated. Power PC Theory also assumes that the causes, both potential and alternative ones, have been already chosen in some way without describing the method of how to choose them. Therefore, the selection of causes can involve a computational explosive search so that the use of some heuristics might be needed. Again, it seems that finding an algorithm for the Power PC Theory is not an easy task at all.

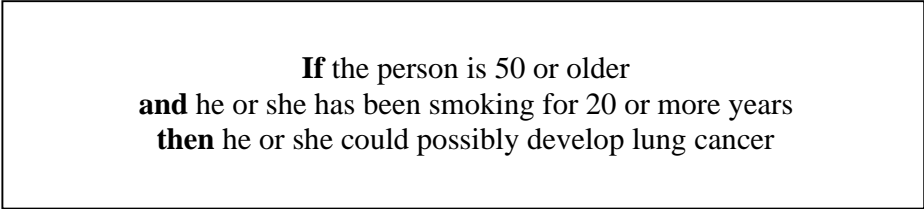
As mentioned at the beginning of this subsection, the Power PC Theory is beautiful and powerful but some things, such as those mentioned earlier, have still to be solved in order to construct a system, based on this theory, for performing causal induction tasks just the way humans do. In spite of these unsolved remaining questions found, the Power PC Theory seems to offer the solution that, under the circumstances mentioned above, has been adopted by humans to solve the problem of causal induction. Needless to say, some other approaches have been proposed for trying to deal with this legendary problem emerging from different disciplines such as philosophy, statistics and computer science (Pearl 1996). The psychological approach of causal induction has in part motivated the search for alternative models in the area of Computer Science and more specifically in the field of Artificial Intelligence. As pointed out in section 1.1, this thesis has to do with the extraction and representation of probabilistic relationships among variables taking part in a problem by means of a graphical model called Bayesian networks as an alternative for discovering useful information and possible causal relations hidden in databases. If soundly and consistently found, these causal relationships can allow us to make certain kind of inference tasks such as prediction, diagnosis, decision-making and control in order to solve a given problem. As in most of scientific areas, there are supporters and detractors of the possible existence of suitable methods for extracting causal relations from observational data (Spirtes, Glymour et al. 1993; Glymour, Spirtes et al. 1999a; Robins and Wasserman 1999a; Glymour, Spirtes et al. 1999b; Robins and Wasserman 1999b). But Power PC Theory sheds light in favour of the existence of such methods that could bridge the gap

between covariation and causation. In the next subsection, the computer science perspective about this will be reviewed.

1.2.2 Perspective from Computer Science to Causal Induction.

Artificial Intelligence (AI) is a branch of Computer Science that has taken two well-differentiated directions: to make intelligent machines or computer programs and to help understand human intelligence by constructing such systems (Winston 1992; Luger and Stubblefield 1993; McCarthy 2000). If one wants to construct, say, an intelligent agent able to interact, learn, act and react on its environment, it must be provided with a very flexible algorithm (Hofstadter 1999) that, through its sensory input, allows it to convert and represent the information contained in that environment in a suitable way for it to perform such actions and even modify the world where it is embedded.

In expert systems, a classic area of AI, the representation of knowledge from human experts was first conceived using the classical logic. The basic idea was to represent causal knowledge in the form of if-then rules such as the figure 1.5 shows below. Because of this, the very first expert systems were called rule-based expert systems.



If the person is 50 or older
and he or she has been smoking for 20 or more years
then he or she could possibly develop lung cancer

figure 1.5: A classic expert system rule

The words in bold represent the logic connectors for the relation of implication and conjunction. So, for the premise to be true, the two conditions need to be true. The two conditions being true make the conclusion true as well. For this rule, if one of the conditions in the premise is false, then the conclusion cannot be drawn. Note in the conclusion the incorporation of uncertainty contained in the word "possibly". Because causal relationships are not of deterministic type (Einhorn and Hogarth 1986; Pearl 1988;

Jackson 1990; Neapolitan 1990; Cheng 1997; Pearl 2000), the system that tries to represent such relations, has to, somehow, incorporate this inherent nature of uncertainty in causality. It is very important to stress the problems that this kind of expert systems have when incorporating such uncertainty in their rules: it leads frequently to contradictory and of course inexact results (Pearl 1988; Diez and Nell 1998/99). The construction of rule-based expert systems is very expensive for many reasons as mentioned in subsection 1.1. These reasons are primarily the very time-consuming task of extracting the knowledge of the human experts (mainly by means of interviews), understanding that knowledge from the point of view of the knowledge engineer and then translating this knowledge into a computer program. Jackson (1990) calculates that for every 8 hours of elicitation process, the knowledge engineer can come up with only 5 useful rules. Taking into account this, for an expert system to reach a solution similar to that offered by the human experts, it would need the order of some hundreds and even thousands of these rules. This would lead to a construction of an expert system that would take months or even some years.

Because of these serious drawbacks, namely, the sound and consistent representation of uncertainty and the matter of time, some other people looked for some other possible new directions. One of them, which is worth mentioning, was that of classifying variables' attribute-value pairs according to the information they provide from a database. This was possible with the aid of Information theory or entropy proposed by Shannon in 1948 (Pearl 1988; Schneider 1995). In this case, the algorithm by Quinlan (Cooper and Herskovits 1992; Winston 1992; Buntine 1996) called **ID3** is of special and remarkable importance. In his algorithm, he tried to **categorise** the variables taking part in the problem being considered in order to find which attribute (or variable) and which value of that attribute divide or partition the set of causes to explain the output (a dependent variable) in the most parsimonious way. Figure 1.6 presents an example of the tree structure produced by such an algorithm. In this example, suppose that people who appear in the leaves of figure 1.6 are in the beach and some of them get sunburned. The variables that can be collected and can probably explain the output (get sunburned or not) are: name, hair colour, height, weight and the use of lotion.

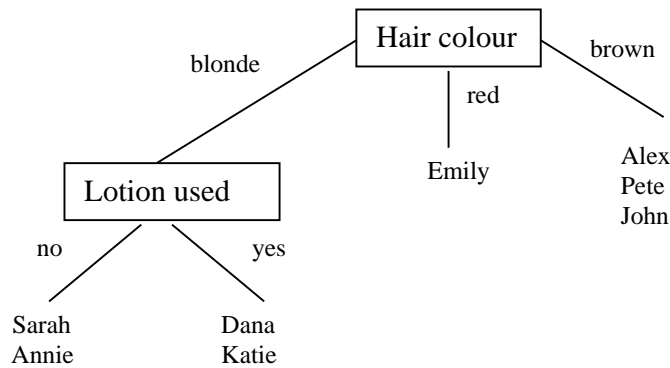


Figure 1.6: A classification tree

As can be seen from figure 1.6, each leaf of the tree has either a single or a set of names in it. The names with the symbol \odot before them are those people who actually get sunburned. From the figure, it can be concluded that the hair colour is the variable that provides information to divide the output in the most parsimonious way, i.e., all the leaves contain people who either get sunburned or not. In the very left branch, (blonde) the single variable hair colour cannot provide enough information to divide the output parsimoniously so another one is needed to preserve this parsimony: the variable lotion-used. One key point of ID3 is that of extracting the knowledge from a database and representing that knowledge in the form of a tree. These kinds of trees are well known as **classification** or **decision trees**. Note that these decision trees are different from those used in decision analysis (Cooper and Herskovits 1992). Once the knowledge has been extracted and represented in the form of a tree, then it is possible to convert it into if-then rules that are better understood by humans. However, one problem with this approach comes when a tree cannot represent the underlying distribution of the data and some more complex structures than trees are needed. But ID3 started giving a good insight of how to construct algorithms with less human supervision in order to save time and, of course, to have support tools more promptly.

Then, about a decade ago, a solid combination with the same basic idea appeared: **graphical models**. The term solid means here that, in contrast to the human counterpart,

such models do not violate the basic axioms of probability theory. These models have taken the better of two worlds, namely, graph theory and probability theory. The very idea of such models is that of modularity: the combination of simpler parts to construct a complex system, as figure 1.4 suggests. To do this, these models use probability theory as the glue to combine the parts providing consistency and ways to interface models to data whereas graph theory provides a natural and intuitive framework to model interactions among variables (Jordan 1998). The terms natural and intuitive suggest that graphical representations are, under certain conditions, easier to understand than other kinds of representations. A number of researchers from a wide range of scientific disciplines (cognitive psychology, developmental psychology, linguistics, anthropology and computer science) have given evidence that supports such a claim: Gattis (2001), Liben (2001), Tversky (2001), Emmorey (2001), Bryant and Squire (2001), McGonigle and Chalmers (2001), Hummel and Holyoak (2001) and Larkin and Simon (1995). It can be argued that these representations aid cognition because they are structured in such a familiar way that people can rely on them to structure memory, communication and reasoning (Gattis 2001). Gattis (2001) also points out that spatial representations are not merely metaphors that help understand cognitive processes but actual internal mechanisms that allow us to perform more abstract cognitive tasks. Larkin and Simon argue that a diagram can be superior to a verbal description because, when well used, the former “automatically supports a large number of perceptual inferences, which are extremely easy for humans” (Larkin and Simon 1995, p. 107). Graphical representations are useful in reasoning tasks because, through their structure (which can represent order, directionality and relations) and the partial knowledge about their elements and the relations among them, it is possible to infer the values of the elements and their relationships that are unknown (Gattis 2001). In a similar vein, Larkin and Simon (1995) also claim that these representations have the power to group together all the information that is used together, which avoids the problem of searching large amounts of data to find the elements needed for performing inference. As Tversky (2001) points out, graphical representation can be used to reveal hidden knowledge, providing models that facilitate inference and discovery. It is very important to remark that, in words of Tversky, “long before there was written language, there were depictions, of myriad varieties” (Tversky 2001, p. 80). In sum, graphs can represent abstract concepts and information in

such a way that this information can be accessed and integrated quickly and easily. Graphs also facilitate group communication (Tversky 2001).

Regarding the relationships between graphical models and probabilistic models, Pearl (1988) was one of the firsts to find the way probabilistic relationships could be represented in a graph without violating the very basic axioms of probability. This great discovery was a breakthrough in the construction of expert systems because since then, it is much easier and sound to represent uncertain knowledge in a very easily understandable, economic and convenient way. The advantages and disadvantages of such an approach will be reviewed in chapter 2. Moreover, the power of these models is that, because of their inherent features, they can go beyond the representation of only probabilistic relations and represent cause-effect relationships within their structure.

It is important to note, however, that the way these models are built can be perfectly the same as for the case of the rule-based expert system implying that the elicitation process can take a long time as well. In this work, this is not the case. The way the algorithm proposed here builds a Bayesian network is to take a database as an input (with the potential relevant factors to explain a certain problem), process the information contained in it and then to output the structure of such a network making the knowledge extraction process easier and quicker, as figure 1.7 shows. The emerging area for "**mining**" the data and discover patterns, rules and relationships hidden in collections of data is called, as said in section 1.1, knowledge discovery in databases or data mining. This kind of algorithm is called **unsupervised** because the output it produces, whichever form it has, is a result of processing data and not of external human supervision (Frey 1998). The details of this sort of algorithms will be discussed in detail in chapters 4, 5 and 6.

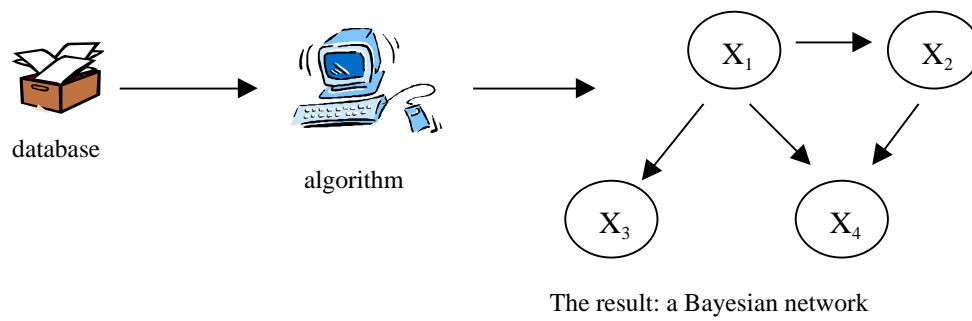


Figure 1.7: an algorithm for learning Bayesian networks from data

1.3 Computational aids for handling information accurately and effectively: graphical models.

Although the psychological approach is mainly concerned with the problem of knowing how humans represent and acquire causal knowledge, this point of view and some of the theories supporting it give some insight about the importance of covariation information for extracting that causal knowledge. It is this insight that, at least in part, has motivated the use of computational methods for extracting causal knowledge from data automatically.

Before trying to extract causal knowledge, patterns, rules, etc. in the data, it is very important to remember the dynamic growing nature of information. The amount of information grows so fast that new methods are needed for processing it in an efficient and reliable way as figure 1.8 suggests.

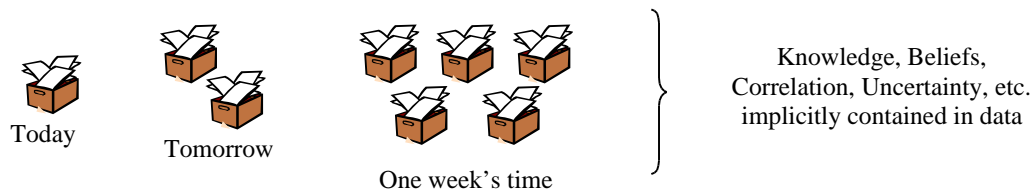


Figure 1.8: The dynamic growing nature of information

Problems encountered in many knowledge domains usually contain many random variables to be analysed. This implies two big problems to deal with: **uncertainty** and **complexity**. To overcome the problem of uncertainty, as said before, it is necessary to find a suitable model capable of representing and managing uncertainty in a sound and consistent way. The problem of complexity has to do with the impossibility of performing an extensive search and processing over the variables taking part in the problem because it is actually computationally intractable, which means that not even computers are able to solve this problem in a short period of time (Russell and Norvig 1994; Chickering 1996; Chickering, Heckerman et al. 1997; Friedman and Goldszmidt 1998a). Thus, powerful and convenient heuristics are needed for solving this complexity problem. As can be inferred from these two problems, the proper and accurate analysis of the information might include much more computing than people and even classic statistical methods can indeed do. Because of this, people or even human experts can find it very difficult to extract useful information such as causal patterns from data.

An excellent solution for dealing with these two problems of complexity and uncertainty has been offered by the so-called graphical models. They, as said above, have the interesting characteristic of combining the methods from graph and probabilistic theories to represent in an elegant and easy way the interactions among variables of interest (Heckerman 1998), as figure 1.4 suggests. Generally speaking, graphical models represent the variables in the form of a circle called a **node** and the interactions between any pair of variables with a line connecting these two variables (which can have either an arrow at one of its extremes or not) called an **edge** or an **arc**. These models have both common and different features that make them suitable for one specific task or another. Here are some of

them: **Markov networks** (Buntine 1996), **Bayesian networks** (which are also known under different names as stated in section 1.1) (Pearl 1988; Neapolitan 1990), **structural equation models** (Spirtes, Glymour et al. 1993), **factor graphs** and **Markov random fields** (Frey 1998). In this work, a particular kind of graphical model has been chosen because of its natural way to perform prediction and diagnosis: Bayesian networks.

Graphical models are, in conclusion, effective tools for analysing and processing information. However, now an important question arises: can causality be reliably extracted from data by algorithmic means and represented in the form of a graph? As the nature of this question suggests, this has been of course cause of a great debate (Spirtes, Glymour et al. 1993; Friedman and Goldszmidt 1998a; Glymour, Spirtes et al. 1999a; Robins and Wasserman 1999a; Glymour, Spirtes et al. 1999b; Robins and Wasserman 1999b; Pearl 2000). If the answer of the question is yes, then the problem is now to find out how to do it by implementing and testing algorithms in a number of different situations.

1.4 Automatic classification: Data Mining or Knowledge Discovery in Databases.

The traditional method to extract human expert knowledge has been by interview with experts. This process has proved, as mentioned before, very time-consuming and hence expensive. The first problem for this approach to be carried out is to find an expert or experts in the knowledge area for which a computer system needs to be built. After finding them, it has now to be checked whether they are available and want to cooperate in building this system. Another very big problem is when even human experts themselves realise the great difficulties they find to express verbally how their knowledge is organised and if uncertainty has to be incorporated, they usually make mistakes violating the basic laws of probability (Pearl 1988; Diez and Nell 1998/99). Finally, the person who is responsible for eliciting the knowledge via interviews (called the knowledge engineer) has very often big difficulties in understanding and translating the experts' knowledge into a computer

program. As said before, the average number of useful rules extracted after 8 hours of interview is 5 (Jackson 1990).

These serious problems motivated a new direction of research in order to make the elicitation and representation processes much easier. The area of KDD emerged in the late 1980's from a variety of areas such as statistics, databases, machine learning, artificial intelligence and others to deal with such problems (Han and Kamber 2001). The main idea is basically to automate the elicitation, analysis, classification, discovery and coding processes or at least to perform such tasks with the minimum amount of external supervision in order to save time and money. With the aid of new methods for collecting data, such as hand-held terminals, remote sensors, scanners, etc., the amount of data is so vast that, without the availability of suitable methods for analysing the information at hand, these data are often just archived or filed and not used for carrying out important tasks such as control and decision-making (Keim 2001). Also, large databases may be used to confirm prior hypotheses but rarely to test alternative hypotheses, which may explain data better. So, the key point is to find out the way to **extract** knowledge from data and present it in such a manner that permits an easily understandable intuitive visualisation of interesting patterns implicitly contained in the data, as figure 1.9 shows.

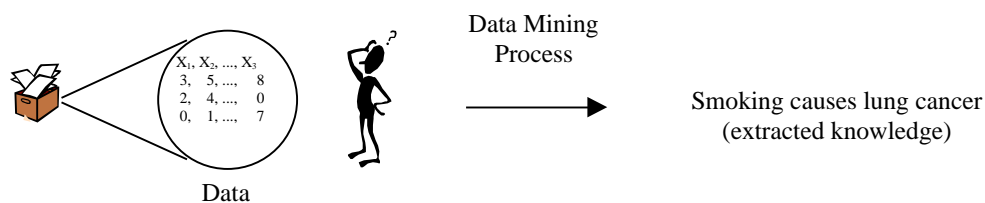


Figure 1.9: A data mining process helps to discover possible causal relationships hidden in the data

Because it is required to mine patterns from data, i.e., to infer possible causal relations from covariation, this is where the importance of the feasibility of obtaining knowledge from data given by the psychological approach can come to help. There are different ways to represent the output patterns: association rules, decision trees and Bayesian networks, among others. The if-then rule of figure 1.5 shows an example of an

association rule. These rules have the form $A \rightarrow B$ where the premise A can contain a single or a joint set of premises and the conclusion B can be a single or a conjugated one as well. The symbol \rightarrow represents the logical operator called **implication**. A and B are attribute-value pairs; then the rule would read: **If** the conditions in A hold **then** B is highly likely to be true. If the rule in figure 1.5 is taken and if the premises are known to be true then it is highly probable that the person will develop lung cancer. These association rules are well understood by experts so that they can perform some important actions to solve a given problem when looking at those rules.

In order to extract knowledge from data and code it in the form of an association rule a good method to do so is to construct a decision tree from the data as shown in figure 1.6. Two very well-known and classic algorithms that extract knowledge from data in the form of a tree are **Chow and Liu's** algorithm and **Quinlan's** algorithm (Pearl 1988; Cooper and Herskovits 1992; Winston 1992; Buntine 1996).

In its simplest form, a classification tree is a binary tree meaning that it has only two different branches representing two disjoint values of a certain variable. These disjoint values can perfectly be value ranges (when the variable is continuous). The whole idea of a tree is representing, whenever possible, a probability distribution responsible of generating the data. If for instance, the mechanism underlying the data does not have the form of a tree, then the algorithms such as the mentioned above build a tree trying to approximate the probability distribution with this tree-like form as close as possible. To do so, the criterion of **cross-entropy** is often used to measure the closest approximation. The main measures of **information theory** or **entropy** are reviewed in chapter 4. A classification tree, as its name suggests, looks for maximizing the classification accuracy on new cases. The following example taken from Han and Kamber (Han and Kamber 2001) illustrates much better the basic idea of a classification tree.

Suppose that from the table in figure 1.10, a certain enterprise, called AllElectronics, wants to know which attribute-value pair or attribute-value pairs determine if a customer is likely to buy a computer or not (the class attribute).

Age	Income	Student	Credit_rating	Class: buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no

figure 1.10: training data set from the AllElectronics customer database

As can be clearly seen, it is not an easy task to mine the data by simple inspection; i.e., to find some useful patterns that can explain the behaviour of the output which is, in this case, whether a person is buying or not a computer. This is true even when the number of variables is small. In this example, the number of independent variables taken to predict the outcome of one single dependent variable is 4. The dependent variable is frequently known as the **class variable** or **class attribute**. However, to get a pattern able to explain the output in terms of these 4 variables is not a straightforward task without the help of tools such as automatic classification tools for instance. In order to extract knowledge from the table above and using the ideas of, say, algorithm ID3, first of all, a measure such as information gain is needed in order to construct such a tree-like model. This measure has to be able to select the attribute (variable) which provides the highest amount of information in order to divide the sample in the most parsimonious possible way. Doing this permits to construct a tree, which allows visualising in a simple manner the knowledge contained in the data. The final result (the numerical calculations are not presented here) is drawn in

figure 1.11, which gives a good insight of how powerful these automatic classification methods can be.

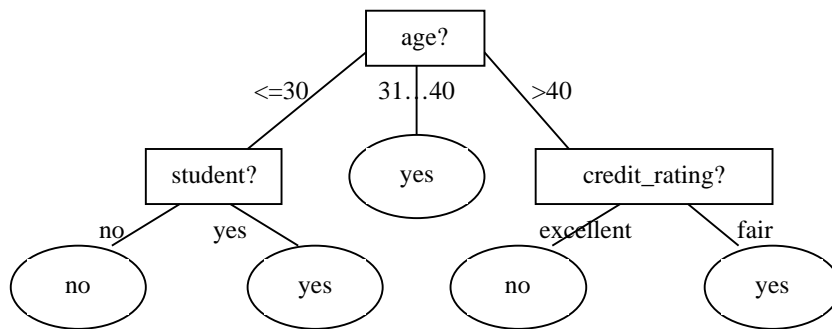


Figure 1.11: a classification tree for the AllElectronics database

The variable which provides the highest amount of information to explain the class attribute (buying a computer) is age; that is why it is the root node of the tree. Age has three different possible values that are represented by the three different arcs. These three arcs are the different possible ways or branches to follow in order to know the value of the outcome. When age ≤ 30 , it is possible to observe, from the table in figure 1.10, that there are cases with both possible results of the class attribute: yes and no. So it is necessary to find another partition variable for when age ≤ 30 that allows to divide the cases in the sample and put them in a same class. This attribute or variable is student. Note that now, two more branches are added. The left one is for the case when age ≤ 30 and the person is not a student. If looked at the table carefully, then it is possible to see that all the cases that have these two values for age and student have the same result: the person does not buy a computer. The right branch tells that if age ≤ 30 and the person is indeed a student, then for all the cases having this combination of values the result is the same: the person does buy a computer.

For the middle branch of the variable age, i.e., when age = 31...40, all the cases having a value in this range, have the same output: the person buys a computer.

Finally, when $\text{age} > 40$, the cases having this value do not belong to an only one category. So it is necessary to find a variable that divides the cases that share the same output. `Credit_rating` is such a variable. Note that now two more branches are added. In the left branch, all the cases having $\text{age} > 40$ and `credit_rating = excellent` share the same result: the person does not buy a computer. For the right branch, all the cases having the value of the variable $\text{age} > 40$ and `credit_rating = fair` have the same result: the person does buy a computer. Notice the shape of the variables in the tree of the previous figure. The rectangles represent the independent variables and the leaves (circles) represent the class attribute (buys a computer).

Now, this induction tree method makes perfectly possible generate classification rules from this classification tree. The rules that can be extracted are those shown in figure 1.12.

<p>R1: If <code>age</code> ≤ 30 and <code>student</code> = no then <code>buys_computer</code> = no R2: If <code>age</code> ≤ 30 and <code>student</code> = yes then <code>buys_computer</code> = yes R3: If <code>age</code> = 31...40 then <code>buys_computer</code> = yes R4: If <code>age</code> > 40 and <code>credit_rating</code> = excellent then <code>buys_computer</code> = no R5: If <code>age</code> > 40 and <code>credit_rating</code> = fair then <code>buys_computer</code> = yes</p>
--

Figure 1.12: classification rules from the classification tree of figure 1.11

As can be seen, this procedure to extract knowledge from data seems very powerful and indeed it is. The complexity of the acquisition of expert knowledge by classic means appear to be reduced with good results because an important feature of such systems is that they do not need or use domain knowledge but only the data in the form of a database. Moreover, once the structure of the tree is built, the generation of classification rules from this structure seems straightforward. Also, it can be noticed that the variable `income` did not take part in the resultant tree because it was not relevant to make a partition of the output variable or, in other words, did not provide enough information to do so. This method of automatic classification has proved very useful in solving a wide range of problems and because of that, it has been used in a variety of domains (Pearl 1988; Cooper and Herskovits 1992; Winston 1992; Han and Kamber 2001).

However, this approach has some disadvantages. One of them is, for instance, when two different attributes give exactly the same amount of information. Thus the procedure is unable to decide which variable is to be used as the main one causing the procedure to fall into a deadlock. This could seem an odd situation and difficult to happen but it actually does. So, it is necessary to add a criterion or heuristics, in the event of a draw, to decide which variable is to be chosen. Another problem comes when the underlying probability distribution of the data cannot be represented by a tree but by other more complex structure and therefore this will lead to inexact results; i.e., another graphical structure more complex than a tree can represent products of higher order distributions (Pearl 1988). This can be because, in order to construct a classification tree, it is necessary to designate a classification variable. This produces a restriction, namely, that the probability distribution has to be represented over one variable of interest (which is this very same classification variable).

To finish this section, an example to illustrate another model of automatic classification will be presented. This useful tool is known as Bayesian networks. Bayesian networks are a powerful tool to represent uncertainty in a natural, consistent and suitable way. A BN captures the probabilistic relationships among variables of interest by means of a graph consisting of nodes (circles) representing the variables and arcs (arrows) connecting nodes representing interactions among them. These interactions can of course be of causal nature. This graph also has to be acyclic which means that no cycles are presented within the structure of such a network. To see the power of this approach and how it generally works, consider the data in figure 1.13. The example is taken from Cooper (Spirtes, Scheines et al. 1994). In this example, there are 5 different variables: metastatic cancer (mt), brain tumour (bt), serum calcium (sc), coma (c) and severe headache (h). Now, the intention is to know, say, how these variables are related each other. Once these relationships are established, it can be argued, the classification, prediction, diagnosis or decision-making processes can be carried out much more easily. For instance, suppose that you want to know which variables cause a certain patient to fall into a coma. Note that all the variables are binary: 0 represents the absence of a variable and 1 represents the presence of that variable.

c	mc	sc	bt	h
0	0	0	0	1
0	0	0	0	0
0	0	0	0	0
0	0	0	0	1
0	0	0	0	0
1	1	1	1	1
1	0	1	0	0
0	0	0	0	1
1	0	1	0	0
0	0	0	0	1
...

Figure 1.13: database about potential causes for a patient to fall into a coma: metastatic cancer, serum calcium, brain tumour and severe headache

As in the example of the construction of the tree, it can easily be noticed that the probabilistic relationships among the variables just by looking at the data cannot be determined or identified straightforwardly. If an algorithm to induce the structure of a Bayesian network from data is applied, then the result obtained is that shown in figure 1.14.

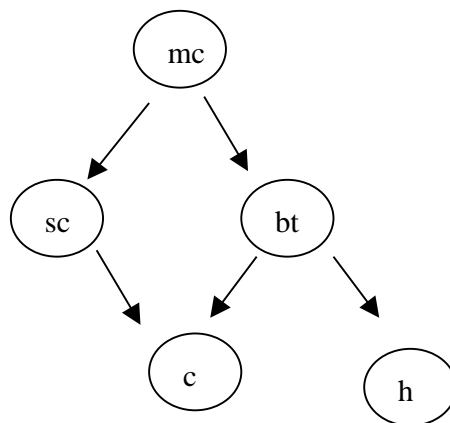


Figure 1.14: the resultant Bayesian network for the database of figure 1.13

As can be easily visualised from figure 1.14, the relationships among the variables in this particular example are explicitly represented by a directed acyclic graph that permits

one to recognise them in a simple and intuitive way. From the result, it is possible to say (of course under the supervision of the medical experts who have the last word) that if the patient has a brain tumour and his or her total level of serum calcium has increased, then it is very likely that this person falls into a coma. Also note that brain tumours cause severe headaches. Finally, it is possible to assert that both the increasing serum calcium level and brain tumours are caused by metastatic cancer. If looked carefully, some other implicit relationships among the variables can be identified. For instance, once known that the serum calcium and brain tumours are instantiated (i.e. their values are known) it is not necessary to know the value of metastatic cancer because it does not provide additional information to explain the behaviour of coma. In other words, metastatic cancer and coma are **conditionally independent** given that the values of serum calcium and brain tumour are known. This characteristic property of Bayesian networks is known as **d-separation** (Pearl 1988; Neapolitan 1990; Spirtes, Glymour et al. 1993; Jensen 1996) and it is one of the most powerful features of such networks as will be explained in detail in chapter 2.

There also exists, for each node in the graph, a marginal or conditional probability distribution according to which is the case; these probability tables are computed from the sample or from the human experts. Node mc has no parents so its probability distribution is marginal. Node c for instance has two parents: sc and bt. Its probability is conditional on the values that its parents take, i.e., $P(c|sc, bt)$. There are 8 possible combinations whose probabilities are calculated from the database and do not violate the basic axioms of probability. In doing so, the result yielded is consistent and sound. Tables in figure 1.15 show this idea.

<p>Marginal probability of mc</p> <p>$P(mc = 0) = 0.8$</p> <p>$P(mc = 1) = 0.2$</p>	<p>Conditional probability of c given sc and bt</p> <p>$P(c = 0 \mid sc = 0, bt = 0) = 0.950$</p> <p>$P(c = 1 \mid sc = 0, bt = 0) = 0.050$</p> <p>$P(c = 0 \mid sc = 0, bt = 1) = 0.200$</p> <p>$P(c = 1 \mid sc = 0, bt = 1) = 0.800$</p> <p>$P(c = 0 \mid sc = 1, bt = 0) = 0.200$</p> <p>$P(c = 1 \mid sc = 1, bt = 0) = 0.800$</p> <p>$P(c = 0 \mid sc = 1, bt = 1) = 0.200$</p> <p>$P(c = 1 \mid sc = 1, bt = 1) = 0.800$</p>
---	---

Figure 1.15: tables showing the marginal and conditional probabilities for the network of figure 1.14

Of course, every node has either a marginal or conditional probability table attached to itself. In this example only two of them are shown (those of variable mc and variable c). These probabilities allow one to represent and deal with uncertainty as well as providing the capability to perform classification, decision-making, prediction and diagnosis. The potential applications of this graphical modelling approach are, among others, classification, automated scientific discovery, automated construction of probabilistic expert systems, diagnosis, forecasting, automatic vision, sensor fusion, manufacturing control, information retrieval, planning and speech recognition (Pearl 1988; Cooper and Herskovits 1992; Heckerman, Mandani et al. 1995). This thesis will focus on the construction of Bayesian networks from data for performing tasks such as classification and diagnosis.

1.5 Classification, prediction, diagnosis and decision-making.

It is perhaps much simpler to explain the concepts of classification, prediction, diagnosis and decision-making with the help of some examples using the frameworks of classification trees and Bayesian networks.

For the case of classification, take the tree of figure 1.11. Now imagine that you want to know whether a client will buy or not a computer; in other words, you want to determine to which class he or she belongs to. In order to classify this new subject, it is necessary first to check if the values of the variables that describe him or her are present in table of figure 1.10. If so, then what is finally necessary is to follow the path (branches) that better suits his or her case. For instance, suppose that the person is 23 years old, is a student, his or her income is medium and the credit rating is excellent. Because this case is present in the data, therefore, the branches that better characterises these facts are reflected by rule R2 of figure 1.12, which says that the person will indeed buy a computer.

Now suppose that, taking the same tree structure, you want to know if it is likely that a client with the following values for the variables in the table of figure 1.10 will buy a computer. This example has been taken from Han and Kamber (2001). The values are: age ≤ 30 , income = medium, student = yes and credit_rating = fair. As can be identified, this case is not present in the database so it is not possible to apply any of the rules induced from the classification tree for this example. Now, in order to solve this problem, it is necessary to apply an idea that permits one to do so. Since it is possible to calculate marginal and conditional probabilities from the data, then Bayes' theorem can be used. This formula, applied to this specific case, would look like the following one:

Let $X = \text{age} \leq 30, \text{income} = \text{medium}, \text{student} = \text{yes}, \text{credit_rating} = \text{fair}$, then in probabilistic terms, the question would be: $P(\text{buys_computer} = \text{yes} \mid X)$?

By applying Bayes' theorem it is possible to compute this required probability and thus to predict the likelihood for the client to buy a computer given X . It is not the intention

here to write down all the numerical details but only the general idea underlying this principle. Bayes' theorem will be explained in chapter 2.

For the problem of diagnosis, imagine the next scenario, taken and adapted from Lauritzen (1996), is given. A patient who has visited Asia lately visits a clinic because he has dyspnoea (shortness of breath). It is known that the patient does not smoke. Now, the question is: what is the diagnosis for this patient? Imagine that the medical knowledge about the interaction among these variables and some others is captured in the structure of the Bayesian network depicted in figure 1.16. The variables taking part in the problem are: visit to Asia (a), smoking (s), tuberculosis (t), lung cancer (l), bronchitis (b), tuberculosis or cancer (tol), x-ray result (x) and dyspnoea (d). All the variables are binary, which means that either each of one is present or absent.

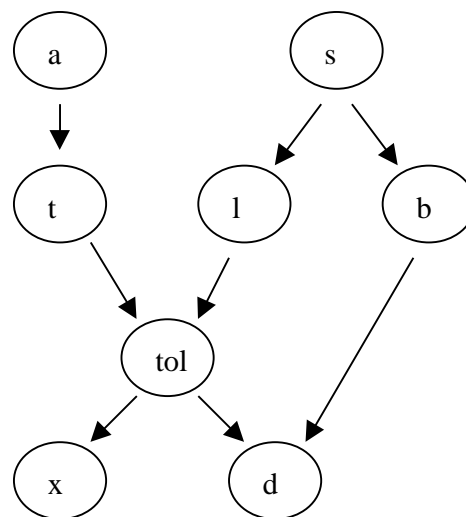


Figure 1.16: a Bayesian network for diagnosing the possible causes of a patient having dyspnoea

From figure 1.16 it can be seen that dyspnoea (d) can be caused by bronchitis (b) or by a combination of tuberculosis and lung cancer (tol). The x-ray study (x) is not able to distinguish between tuberculosis and lung cancer (tol). Also, a visit to Asia (a) could have produced tuberculosis (t) and smoking (s) increases the risk of having or developing lung

cancer (l) and bronchitis (b). For the Bayesian network of the above figure to be complete, it is necessary to specify marginal and conditional distributions over all these variables. These distributions can be extracted from a database containing cases of patients with these common characteristics. Now what is left to do is to instantiate or substitute the values of the variables that are known and then propagate their probabilities to the other nodes for which values are not explicitly known using Bayes' theorem. The values of the variables that are known are: a = yes, s = no and d = yes. Of course the most plausible diagnostic, given the data at hand, depends precisely on these data. Suppose that, for this example, the most plausible explanation or diagnostic is that the patient has bronchitis and possibly tuberculosis. Once again, the numerical calculations are not expressed here but only the general idea of how a diagnostic reasoning task could be performed using this framework of Bayesian networks.

Finally, for the case of decision-making and control, taking into account the same previous example and the same previous figure, policy makers for instance would be very interested in the relation between smoking and lung cancer. Once they learn this causal relationship, they could create a special policy for making the smokers reduce or stop their habits in order for them not to develop this terrible and mortal disease. The health policy makers could also make smokers save money if they succeed in their enterprise and reduce health costs about treatment, medicines and specialists who have to do with the development of this disease. As can be noticed, this kind of deliberative actions can be carried out once the causal relationships among variables are discovered.

In the next chapter, the necessary background that will be useful to understand more deeply all the technical concepts and the general idea of this work will be reviewed.

Chapter 2

Background

This chapter presents relevant results and concepts from Probability and Graph Theories as well as axiomatic characterizations of probabilistic and graph-isomorph dependencies, which are necessary to arrive to a formal definition of a Bayesian, network. Finally, it presents the possible and plausible representation of uncertainty, knowledge and beliefs using this graphical modelling approach.

2.1 Basic concepts on Probability and Graph Theories.

As mentioned in chapter 1, Bayesian networks, as a member of the so-called graphical models, make use of some sound ideas from probability and graph theories in order to represent the probabilistic interactions among **random** variables in a suitable, intuitive and easily understandable way. In this section, all the necessary concepts supporting this graphical modelling approach will be reviewed. Some useful results from probability theory will also be presented. The connection between these results and the definition of a Bayesian network will be clearly seen in section 2.3.

A good question to start with is the following one: why is it important to capture probabilistic dependencies / independencies in the form of a graph?

It can be argued that many problems in everyday life and science are in fact probabilistic, i.e., a deterministic behaviour cannot be defined with the data available at hand at a particular time period. That is why, tools for representing and handling uncertainty provided by probability theory are needed in order to solve those kinds of problems. In probability theory, the most important definition to represent probabilistic relationships is the joint distribution function $P(x_1, x_2, \dots, x_n)$. Once this function is defined, any inference on any variable taking part of the problem being modelled can be performed.

However, defining such a function will involve, most of the time, a very complex problem. For instance, for the case of n variables, a table with 2^n entries will be required for storing that function. This means a huge number of different instances which, in the real world, would be almost impossible to find and collect. Moreover, it can be argued (Pearl 1988) that human beings do not need such an astronomic amount of data to perform inferences tasks such as prediction, diagnosis or decision-making. On the contrary, they seem to make good judgements based only on a small number of those instances in the form of **conditional probabilities** rather than in the form resembling joint probabilities. It can also be argued (Pearl 1988; Neapolitan, Morris et al. 1997; Plach 1997; Waldmann and Martignon 1998; Plach 1999; Gattis 2001; McGonigle and Chalmers 2001) that graphs could powerfully and plausibly provide a good hypothesis of how causal relationships are organised in the human mind (see section 1.2.2 of chapter 1). Furthermore, it seems that people do not carry out numerical manipulations while trying to find out dependence / independence relations among variables but **qualitative** ones. Graphs give the same power: the ability of inferring dependencies / independencies relations using only **logical** manipulations. Let us first review some important concepts from probability and graph theories in order to present the connection and integration between these two theories, which will permit us to represent effectively and easily the dependencies / independencies embedded in a probability distribution by means of a graph.

Definition 2.1. Let Ω be a random experiment. Let Ω be the set of possible outcomes called sample space. If an experiment Ω has a sample space Ω and an event A is defined in Ω , then $P(A)$ is a real number denominated as the probability of A . The function $P(\cdot)$ has the following properties:

$$0 \leq P(A) \leq 1 \text{ for each event } A \text{ in } \Omega \quad (2.1)$$

$$P(\Omega) = 1 \quad (2.2)$$

$$P(A \text{ or } B) = P(A) + P(B) \text{ if } A \text{ and } B \text{ are mutually exclusive} \quad (2.3)$$

Equation 2.3 can be generalized as follows. For each finite number k of events mutually exclusive defined in Ω :

$$P\left(\bigcup_{i=1}^k A_i\right) = \sum_{i=1}^k P(A_i) \quad (2.4)$$

Definition 2.2. If B_k , $k = 1, 2, \dots, n$, is a set of mutually exclusive and exhaustive events of S and $B_1 \cup B_2 \cup \dots \cup B_k = S$, then it is said that these events form a **partition** of S .

In general, if k events, B_i ($i = 1, 2, \dots, k$), form a partition of A , then $P(A)$ can be computed from $P(A, B_i)$ written as:

$$P(A) = \sum_i P(A, B_i) \quad (2.5)$$

where $P(A, B_i)$ is the short for $P(A \text{ and } B_i)$ or $P(A \cap B_i)$. Figure 2.1 (Hines and Montgomery 1997) represents graphically this definition.

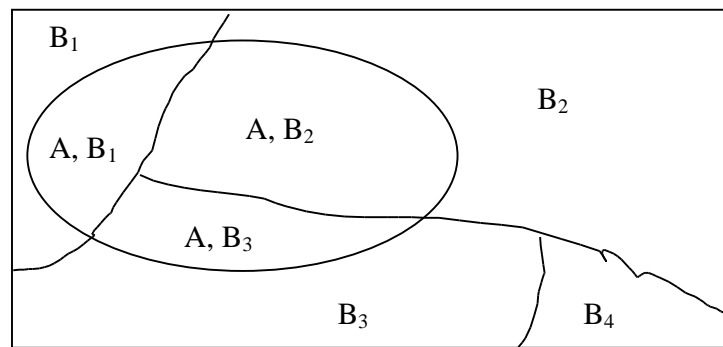


Figure 2.1: Partition of

From this figure, $k = 4$:

$$P(A) = P(A \cap B_1) + P(A \cap B_2) + P(A \cap B_3) + P(A \cap B_4).$$

It does not matter if $P(A \cap B_i) = 0$ for one or all i since $P(\emptyset) = 0$.

Definition 2.3. The conditional probability of the event A given the event B can be defined as follows:

$$P(A | B) = \frac{P(A, B)}{P(B)} \quad \text{if } P(B) > 0 \quad (2.6)$$

The conditional probability definition satisfies the basic axioms of the probability, i.e., equations 2.1 to 2.4. Equation 2.6 can also be rewritten as:

$$P(A, B) = P(A | B)P(B) \quad (2.7)$$

Taking equation 2.7, equation 2.5 can then be rewritten as:

$$P(A) = \sum_i P(A | B_i)P(B_i) \quad (2.8)$$

A useful generalisation of equation 2.7 is known as the **chain rule** or multiplication rule. The chain rule indicates that, when there are sets of n events E_1, E_2, \dots, E_n , the probability of the joint event (E_1, E_2, \dots, E_n) can be written as follows:

$$P(E_1, E_2, \dots, E_n) = P(E_n | E_{n-1}, \dots, E_2, E_1) \dots P(E_2 | E_1)P(E_1) \quad (2.9)$$

Another useful result that can be obtained from equation 2.8 is the famous formula known as **Bayes'** theorem (see equations 2.11 and 2.12). Before arriving to such a theorem, we start with the following definition. If B_1, B_2, \dots, B_k are a partition of Ω and A is an event in Ω , then for $r = 1, 2, \dots, k$:

$$P(B_r | A) = \frac{P(B_r, A)}{P(A)} \quad (2.10)$$

If equation 2.7 is used to substitute the numerator in equation 2.10, then this equation becomes now:

$$P(B_r | A) = \frac{P(A | B_r)P(B_r)}{P(A)} \quad (2.11)$$

Furthermore, if denominator in equation 2.11 is now substituted by equation 2.8, equation 2.11 becomes:

$$P(B_r | A) = \frac{P(A | B_r)P(B_r)}{\prod_{i=1}^k P(A | B_r)P(B_r)} \quad (2.12)$$

Some important theorems can be deduced from the previous definitions. These theorems are shown below:

$$\text{If } \emptyset \text{ represents the empty set, then } P(\emptyset) = 0 \quad (2.13)$$

$$P(\sim A) = 1 - P(A) \quad (2.14)$$

$$P(A \cup B) = P(A) + P(B) - P(A \cap B) \quad (2.15)$$

Before presenting definition 2.4, some notations taken from Pearl (1988) are given. Let \mathbf{U} be a finite set of discrete random variables where each variables $X \in \mathbf{U}$ can take values from a finite domain D_X . Capital letters, X, Y, Z , will denote variables while lowercase letters, x, y, z , will denote specific values of the correspondent variables. Sets of variables will be represented by boldfaced capital letters $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$. Boldfaced lowercase letters, $\mathbf{x}, \mathbf{y}, \mathbf{z}$, will represent values taken by these sets. Boldfaced lowercase letters represent what is called a **configuration**. For instance, if $\mathbf{Z} = \{X, Y\}$, then $\mathbf{z} = \{x, y : x \in D_X, y \in D_Y\}$. Greek letters can also represent individual variables and can be used to avoid confusion between single variables and sets of variables.

Definition 2.4 (Pearl 1988). Let $\mathbf{U} = \{X_1, \dots, X_n\}$ be a finite set of variables with discrete values. Let $P(\bullet)$ be a joint probability function over the variables in \mathbf{U} and let $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ stand for any three subsets of variables in \mathbf{U} . \mathbf{X} and \mathbf{Y} are said to be **conditionally independent** given \mathbf{Z} if

$$P(\mathbf{x} \mid \mathbf{y}, \mathbf{z}) = P(\mathbf{x} \mid \mathbf{z}) \quad \text{whenever } P(\mathbf{y}, \mathbf{z}) > 0 \quad (2.16)$$

If equation 2.16 holds, \mathbf{X} and \mathbf{Y} are conditionally independent given \mathbf{Z} and this relation can be expressed as follows:

$$I(\mathbf{X}, \mathbf{Z}, \mathbf{Y})_p \text{ or simply } I(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$$

The previous relation means:

$$I(\mathbf{X}, \mathbf{Z}, \mathbf{Y}) \quad P(\mathbf{x} \mid \mathbf{y}, \mathbf{z}) = P(\mathbf{x} \mid \mathbf{z}) \quad (2.17)$$

for all values \mathbf{x} , \mathbf{y} and \mathbf{z} such that $P(\mathbf{y}, \mathbf{z}) > 0$.

Marginal or **unconditional independence** is denoted by:

$I(\mathbf{X}, \emptyset, \mathbf{Y})$ which means:

$$I(\mathbf{X}, \emptyset, \mathbf{Y}) \quad P(\mathbf{x} \mid \mathbf{y}) = P(\mathbf{x}) \quad \text{whenever } P(\mathbf{y}) > 0 \quad (2.18)$$

Equations 2.17 and 2.18 are the backbone of Bayesian networks because through their structure they represent, besides the probabilistic relationships among variables (dependencies), the independencies among them as well. But before connecting these equations to Bayesian networks, some important results from graph theory will be reviewed in the remainder of this section.

Definition 2.5 (Neapolitan 1990). A **binary relation** R on a set X is a set of ordered pairs of elements (x, x') where $x \in X$ and $x' \in X$. If $(x, x') \in R$, then x' is said to be a relative of x . Notice that this does not imply that x is a relative of x' . For each $x \in X$, the set of all relatives of x is denoted by $\text{Rel}(x)$. We have

$$(x, x') \in R \text{ if and only if } x' \in \text{Rel}(x).$$

For example, let $X = \{a, b, c, d, e, f\}$ and $R = \{(a, b), (a, c), (a, d), (c, a), (c, d), (d, d), (e, f)\}$. The set of relatives of a , $\text{Rel}(a) = \{b, c, d\}$, while the set of relatives of b , $\text{Rel}(b) = \emptyset$.

Definition 2.6 (Neapolitan 1990). A binary relation R is **irreflexive** if $x \notin \text{Rel}(x)$ for all $x \in X$. For example, let $X = \{a,b,c,d\}$ and $R = \{(a,b),(a,c),(b,c),(c,a),(d,c)\}$. Therefore, it can be said that R is irreflexive.

Definition 2.7 (Neapolitan 1990). A directed graph (or digraph) G consists of a finite set V of vertices or nodes and an irreflexive binary relation E on V . The graph G is denoted as (V,E) .

If a node $w \in \text{Rel}(v)$ and $(v,w) \in E$, then w is adjacent to v and it can be said that there exists an arc or edge from v to w . The set of all nodes that are adjacent to v is denoted as $\text{Adj}(v)$. So,

$$(v,w) \in E \text{ if and only if } w \in \text{Adj}(v)$$

An example of a directed graph is depicted in figure 2.2.

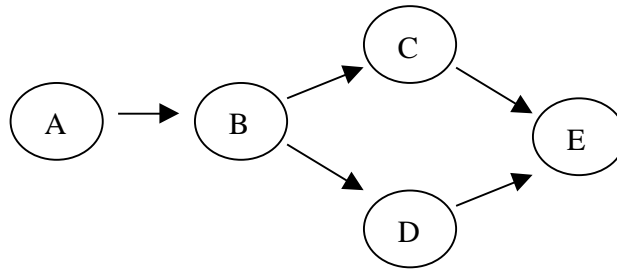


Figure 2.2: A directed graph

As can be seen from this example, the nodes are represented by circles and the arcs by arrows. It can also be noticed that:

$$V = \{A,B,C,D,E\} \text{ and } E = \{(A,B),(B,C),(B,D),(C,E),(D,E)\}$$

Definition 2.8 (Neapolitan 1990). An undirected graph $G = (V,E)$ is a graph in which the adjacency relation is symmetric. That is,

$$(v,w) \in E \text{ implies } (w,v) \in E$$

An example of an undirected graph is depicted in figure 2.3.

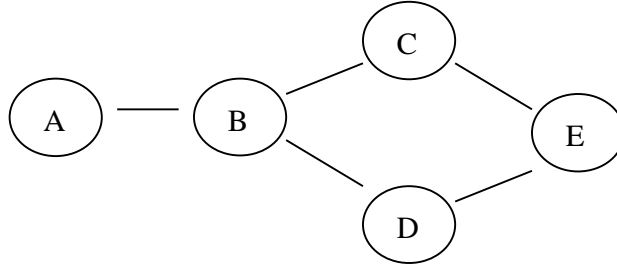


Figure 2.3: An undirected graph

As can be seen from figure 2.3, because there is an arc from v to w if and only if there is an arc from w to v , then the arcs are represented simply by lines connecting two nodes. In this kind of graphs, if $(v,w) \in E$, it is said that v and w are adjacent.

Definition 2.9 (Neapolitan 1990). Let $G = (V,E)$ be a graph (directed or undirected). A sequence of vertices $[v_0, v_1, \dots, v_m]$ is a chain (or an adjacency or undirected path) of length m in G between v_0 and v_m if

$$(v_{i-1}, v_i) \in E \quad \text{or} \quad (v_i, v_{i-1}) \in E \quad \text{for } i = 1, 2, \dots, m$$

Definition 2.10 (Neapolitan 1990). Let $G = (V,E)$ be a graph (directed or undirected). A sequence of vertices $[v_0, v_1, \dots, v_m]$ is a path of length m in G from v_0 to v_m if

$$(v_{i-1}, v_i) \in E \quad \text{for } i = 1, 2, \dots, m$$

In an undirected graph, the concepts of a chain and a path are identical. The concept of a chain is often needed in learning in Bayesian networks to perform inferences applying Bayes' rule.

In figure 2.2, there is a chain from node D to node A but there is no a path from node D to node A. On the other hand, there is a path from node A to node D and also there is a chain from node A to node D.

Definition 2.11 (Neapolitan 1990). A cycle of length m is a path $[v_0, v_1, \dots, v_{m-1}, v_0]$ from a vertex v_0 to v_0 .

Figure 2.4 (Neapolitan 1990) explains much better the previous definition of what a cycle is.

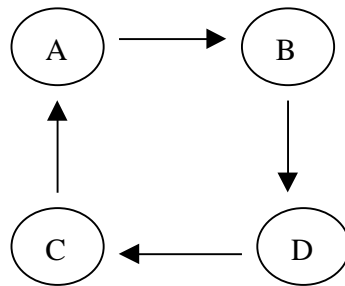


Figure 2.4: A cyclic graph

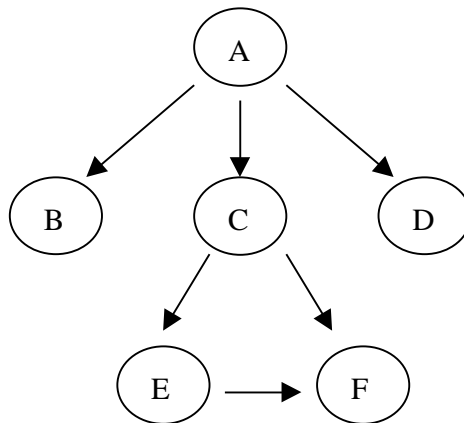


Figure 2.5: An acyclic graph (there is no path from F to C)

In figure 2.4 , there is a cycle of length 4 from every node to itself. On the contrary, in graph 2.5 $[C, E, F, C]$ is not a cycle because there is no a path from F to C.

Definition 2.12 (Neapolitan 1990). Let $G = (V, E)$ be a graph (directed or undirected). Then G is acyclic if G contains no cycles.

Definition 2.13 (Neapolitan 1990). A graph $G = (V, E)$ is a directed acyclic graph (DAG) if G is both directed and acyclic. Figures 2.2 and 2.5 show examples of a DAG.

Definition 2.14 (Neapolitan 1990). Let $G = (V, E)$ be a digraph and u and v vertices in V . Then u is a parent (predecessor) of v and v is a child (successor) of u if $(u, v) \in E$; u is an ancestor of v and v is a descendent of u if there is a path from u to v . A node with no parents is called a root. A node with no children is called a leaf. In figure 2.5, node A is the root of the graph and nodes B , D and F are the leaves of the graph.

Note that a digraph is a directed graph which is not necessarily acyclic.

Definition 2.15 (Neapolitan 1990). Let $G = (V, E)$ be a digraph with n vertices. Then $\pi = [v_1, v_2, \dots, v_n]$ is an ancestral ordering of the vertices in V if for every $v \in V$ all the ancestors of v are ordered before v .

Two important theorems, which will be useful for the algorithms presented in chapters 4 and 5, are stated below.

Theorem 2.1 (Neapolitan 1990). Let $G = (V, E)$ be a digraph. Then an ancestral ordering of the nodes in V exists if and only if G is a DAG.

Theorem 2.2 (Neapolitan 1990). Let $G = (V, E)$ be a DAG and $v \in V$. Then it is always possible to obtain an ancestral ordering of the nodes in V such that only the descendants are labelled after v .

After reviewing the main results from probability and graph theories, it is now possible, as said at the beginning of this chapter, to use those results and combine them in order to represent in a sound and consistent way probabilistic knowledge by means of a

graphical representation. In other words, a graph can be used to represent effectively, in many cases, the dependencies / independencies embedded in a probability distribution.

Pearl (1988) has proposed a set of axioms for the probabilistic relation represented by equation 2.17: \mathbf{X} is independent of \mathbf{Y} given \mathbf{Z} . These axioms allow one to infer new independencies using nonnumeric but only logical manipulations. With the aid of graphical models, it is possible to identify **structural** properties of probabilistic models. In the next section, some axioms, theorems and definitions showing probabilistic and graph-isomorph dependencies will be presented. These axioms provide the basis for inferring new independencies without having to make reference to numerical manipulations. They will also serve as a basis for arriving to a formal definition of a Bayesian network.

2.2 Axiomatic characterizations of probabilistic and graph-isomorph dependencies.

All the axioms, theorems and definitions presented in this section are taken from Pearl (1988).

Theorem 2.3. Let \mathbf{U} be a finite set of discrete random variables and let \mathbf{X} , \mathbf{Y} and \mathbf{Z} be three disjoint subsets of variables from \mathbf{U} . If $I(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$ stands for the relation " \mathbf{X} is independent of \mathbf{Y} given \mathbf{Z} " in some probabilistic model P , then the relation I must satisfy the following four independent conditions:

- Symmetry:

$$I(\mathbf{X}, \mathbf{Z}, \mathbf{Y}) \quad I(\mathbf{Y}, \mathbf{Z}, \mathbf{X}) \quad (2.19)$$

- Decomposition:

$$I(\mathbf{X}, \mathbf{Z}, \mathbf{Y} \cup \mathbf{W}) \quad I(\mathbf{X}, \mathbf{Z}, \mathbf{Y}) \text{ \& } I(\mathbf{X}, \mathbf{Z}, \mathbf{W}) \quad (2.20)$$

- Weak Union:

$$I(\mathbf{X}, \mathbf{Z}, \mathbf{Y} \cup \mathbf{W}) \quad I(\mathbf{X}, \mathbf{Z} \cup \mathbf{W}, \mathbf{Y}) \quad (2.21)$$

- Contraction:

$$I(\mathbf{X}, \mathbf{Z}, \mathbf{Y}) \& I(\mathbf{X}, \mathbf{Z} \cup \mathbf{Y}, \mathbf{W}) \implies I(\mathbf{X}, \mathbf{Z}, \mathbf{Y} \cup \mathbf{W}) \quad (2.22)$$

If P is strictly positive, then a fifth condition holds:

- Intersection:

$$I(\mathbf{X}, \mathbf{Z} \cup \mathbf{W}, \mathbf{Y}) \& I(\mathbf{X}, \mathbf{Z} \cup \mathbf{Y}, \mathbf{W}) \implies I(\mathbf{X}, \mathbf{Z}, \mathbf{Y} \cup \mathbf{W}) \quad (2.23)$$

Before giving a formal definition of what a Bayesian network is, it is necessary to provide some other definitions that have to do with how a graph can represent a probabilistic model. In other words, there exists an isomorphism between a graph (either directed or undirected) and the independence relationships characterized by equations in theorem 2.3. The following definitions that involve undirected graphs will be very useful for obtaining similar definitions regarding directed graphs.

Definition 2.16. Let $\mathbf{U} = \{ \dots \}$ be a finite set of variables and let \mathbf{X} , \mathbf{Y} and \mathbf{Z} stand for three disjoint subsets of variables in \mathbf{U} . Let M be a **dependency model**, i.e., a rule that assigns truth values (true or false) to the three-place predicate $I(\mathbf{X}, \mathbf{Z}, \mathbf{Y})_M$. Any probability distribution P is a dependency model because, for any triplet $(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$, it is possible to test the validity of $I(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$ by using equation 2.17.

It is important to recall that an undirected graph $G = (\mathbf{V}, \mathbf{E})$ has a set of nodes (\mathbf{V}) and a set of symmetric adjacency relations (\mathbf{E}). A graphical representation of a dependency model M represents a direct correspondence between the elements in \mathbf{U} and the set of nodes in \mathbf{V} such that the structure of G reflects some properties of M . Whenever this correspondence holds, it is possible to write $G = (\mathbf{U}, \mathbf{E})$.

If a subset of nodes \mathbf{Z} in a graph G blocks all the paths between the nodes of \mathbf{X} and the nodes of \mathbf{Y} , then this situation is written $\langle \mathbf{X} \mid \mathbf{Z} \mid \mathbf{Y} \rangle_G$. This interception can represent conditional independence between \mathbf{X} and \mathbf{Y} given \mathbf{Z} , i.e.:

$$\langle \mathbf{X} \mid \mathbf{Z} \mid \mathbf{Y} \rangle_G \implies I(\mathbf{X}, \mathbf{Z}, \mathbf{Y})_M$$

and conversely,

$$I(\mathbf{X}, \mathbf{Z}, \mathbf{Y})_M \quad < \mathbf{X} \mid \mathbf{Z} \mid \mathbf{Y} >_G$$

Definition 2.17. An undirected graph G is a **dependency map** (or D-map) of M if there is a one-to-one correspondence between the elements of \mathbf{U} and the nodes \mathbf{V} of G , such that for all disjoint subsets \mathbf{X} , \mathbf{Y} , \mathbf{Z} of elements we have

$$I(\mathbf{X}, \mathbf{Z}, \mathbf{Y})_M \quad < \mathbf{X} \mid \mathbf{Z} \mid \mathbf{Y} >_G \quad (2.24)$$

Similarly, G is an **independency map** (or I-map) of M if

$$I(\mathbf{X}, \mathbf{Z}, \mathbf{Y})_M \quad < \mathbf{X} \mid \mathbf{Z} \mid \mathbf{Y} >_G \quad (2.25)$$

G is said to be a **perfect map** of M if it is both a D-map and an I-map.

It is important to make two distinctions here. First, according to the definition of D-mapness, a D-map assures that the nodes that are connected are dependent in M (if the contrapositive form of equation 2.24 is taken). However, a D-map can probably display a pair of dependent variables in M as separated nodes in G . On the other hand, according to the definition of I-mapness, an I-map assures that separated nodes are independent variables in M but it does not assure that all connected nodes are dependent. An empty graph (a graph with no edges) is a trivial D-map. A complete graph (every node has an edge with every other node) is a trivial I-map. But remember that these definitions have to do with undirected arcs and not with directed ones. In the following section, the advantages of directed over undirected graphs will be mentioned.

Now, a definition and a theorem will be presented so as to show the isomorphism between an undirected graph and a probabilistic model.

Definition 2.18. A dependency model M is said to be a **graph-isomorph** if there exists an undirected graph $G = (\mathbf{U}, \mathbf{E})$ that is a perfect map of M , i.e., for every three disjoint subsets \mathbf{X} , \mathbf{Y} and \mathbf{Z} of \mathbf{U} , we have

$$I(\mathbf{X}, \mathbf{Z}, \mathbf{Y})_M \iff \langle \mathbf{X} \mid \mathbf{Z} \mid \mathbf{Y} \rangle_G \quad (2.26)$$

Theorem 2.4. Pearl and Paz (Pearl 1988). A necessary and sufficient condition for a dependency model M to be a graph-isomorph is that $I(\mathbf{X}, \mathbf{Z}, \mathbf{Y})_M$ satisfies the following five independent axioms (the subscript M is dropped for clarity):

- Symmetry:

$$I(\mathbf{X}, \mathbf{Z}, \mathbf{Y}) \iff I(\mathbf{Y}, \mathbf{Z}, \mathbf{X}) \quad (2.27)$$

- Decomposition:

$$I(\mathbf{X}, \mathbf{Z}, \mathbf{Y} \cup \mathbf{W}) \iff I(\mathbf{X}, \mathbf{Z}, \mathbf{Y}) \& I(\mathbf{X}, \mathbf{Z}, \mathbf{W}) \quad (2.28)$$

- Intersection:

$$I(\mathbf{X}, \mathbf{Z} \cup \mathbf{W}, \mathbf{Y}) \& I(\mathbf{X}, \mathbf{Z} \cup \mathbf{Y}, \mathbf{W}) \iff I(\mathbf{X}, \mathbf{Z}, \mathbf{Y} \cup \mathbf{W}) \quad (2.29)$$

- Strong Union:

$$I(\mathbf{X}, \mathbf{Z}, \mathbf{Y}) \iff I(\mathbf{X}, \mathbf{Z} \cup \mathbf{W}, \mathbf{Y}) \quad (2.30)$$

- Transitivity:

$$I(\mathbf{X}, \mathbf{Z}, \mathbf{Y}) \iff I(\mathbf{X}, \mathbf{Z}, \mathbf{U}) \text{ or } I(\mathbf{U}, \mathbf{Z}, \mathbf{Y}) \quad (2.31)$$

is a singleton element of \mathbf{U} and all the three arguments of function $I(\bullet)$ must represent disjoint subsets.

In the following section, the formal definition of a Bayesian network will be presented as well as the advantages of directed graphs over undirected ones.

2.3 Bayesian Networks.

Ideally, having in mind undirected graphs, it would be very convenient that all dependency models have perfect maps in the sense of definition 2.17. Unfortunately this is

not the case; there are situations where dependency models do not have perfect maps. An example taken from Pearl (1988) will be useful to illustrate this case.

A special case called **induced dependencies** is presented when two different and unrelated variables become relevant to each other when some other facts or variables become known or learned. This situation can be represented as follows:

$$I(\mathbf{X}, \mathbf{Z}_1, \mathbf{Y})_M \text{ and } \sim I(\mathbf{X}, \mathbf{Z}_1 \cup \mathbf{Z}_2, \mathbf{Y})_M$$

which cannot be represented by an undirected graph because of the axiom in equation 2.30.

Figure 2.6 shows such a situation. Imagine that there is an experiment with two coins being tossed and a bell ringing if the outcomes of the two coins are the same. If the sound / no sound of the bell is ignored, then the outcomes of the coins, namely, X and Y , are mutually independent. This situation can be represented as $I(X, \emptyset, Y)$. On the other hand, if the outcome of the bell is taken into account, then knowing the outcome of one coin will change one's opinion about the outcome of the other coin, i.e., $\sim I(X, Z, Y)$.

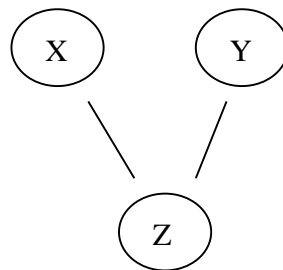


Figure 2.6: A graph showing an induced dependency

The graph of figure 2.6 is not an I-map because it is showing that X and Y are independent given Z . If an edge is added between X and Y , this will result in a complete graph (a trivial I-map) which does not reflect anymore the situation of the coins being really independent and that the outcome of the bell, which is a passive device, does not in fact affect their interaction.

This problem of induced dependencies can be solved using directed graphs. As can be noticed from the previous example, because of this problem, undirected graphs cannot represent many independencies through their structure. On the contrary, directed graphs can indeed represent such induced dependencies using a special criterion called **d-separation** (direction-dependent separation) as the next definition states.

Definition 2.19 (Pearl 2000). A path p is said to be **d-separated** (or blocked) by a set of nodes \mathbf{Z} if and only if

- a) p contains a serial connection $i \rightarrow m \rightarrow j$ or a fork (called also diverging connection) $i \leftarrow m \rightarrow j$ such that the middle node m is in \mathbf{Z} , or
- b) p contains an inverted fork (converging connection or collider) $i \rightarrow m \leftarrow j$ such that the middle node m is not in \mathbf{Z} and such that no descendent of m is in \mathbf{Z} .

A set \mathbf{Z} is said to **d-separate** \mathbf{X} from \mathbf{Y} if and only if \mathbf{Z} blocks **every** path from a node in \mathbf{X} to a node in \mathbf{Y} .

It is very important to mention that, as proved by Geiger and Pearl (Cheng 1998; Cheng, Bell et al. 1998), the d-separation criterion can show all the conditional independencies encoded in a DAG, which means that no other criterion has a better performance than this one. In other words, the d-separation criterion cannot be improved. It is also important to stress that for every path if either one of the two conditions in definition 2.19 is true implies that \mathbf{X} and \mathbf{Y} are d-separated by \mathbf{Z} .

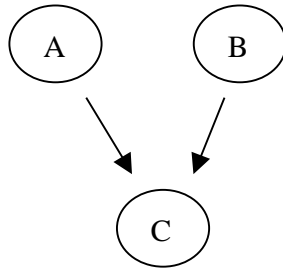


Figure 2.7: A directed graph can solve the problem of induced dependencies

Figure 2.7 resembles figure 2.6 with the difference that this time the edges connecting two different nodes have directionality. If this time the criterion of d-separation is taken into account, then the problem of induced dependencies disappears, as shown below.

Let $\mathbf{X} = \{A\}$ and $\mathbf{Y} = \{B\}$ be the outcomes of the two coins as in the previous example and $\mathbf{Z} = \emptyset$ (which means that the outcome of the bell is not taken into account). As can be seen from figure 2.7, condition 1 no longer holds because the path $A \rightarrow C \leftarrow B$ does not contain either a serial connection or a fork to make this condition true. However, the path between \mathbf{X} and \mathbf{Y} contains a collider on node C which is not in \mathbf{Z} . Hence, it holds that $I(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$. From this result, it is possible to assert that the path $A \rightarrow C \leftarrow B$ is **blocked**, i.e., \mathbf{X} and \mathbf{Y} are d-separated given \mathbf{Z} .

Now, let $\mathbf{X} = \{A\}$ and $\mathbf{Y} = \{B\}$ be the outcomes of the two coins and $\mathbf{Z} = \{C\}$ (which means that now the outcome of the bell is indeed taken into account). As before, condition 1 no longer holds because the path $A \rightarrow C \leftarrow B$ does not contain either a serial connection or a fork to make this condition true. The only node with a collider, namely C , is now actually in \mathbf{Z} . So, condition 2 no longer holds either. Therefore, it can be said that $\sim I(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$. From this result, it is possible to assert that the path $A \rightarrow C \leftarrow B$ is now **active**, i.e., \mathbf{X} and \mathbf{Y} are **d-connected** given \mathbf{Z} . From this example, it is shown that, with the use of directed graphs, the problem of induced dependencies can be solved.

A more complex example to show the criterion of d-separation is depicted in figure 2.8, which has been taken from Spirtes et al. (Spirtes, Glymour et al. 1993).

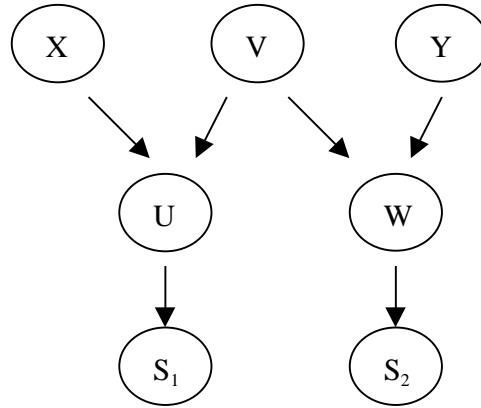


Figure 2.8: An example of d-separation

From the DAG in figure 2.8 the following relations hold:

- a) X and Y are d-separated given \emptyset
- b) X and Y are d-connected given $\{S_1, S_2\}$
- c) X and Y are d-separated given $\{S_1, S_2, V\}$

In case a) let $\mathbf{X} = \{X\}$, $\mathbf{Y} = \{Y\}$ and $\mathbf{Z} = \emptyset$. As can be seen, the path between X and Y is $X \rightarrow U \rightarrow V \rightarrow W \rightarrow Y$. \mathbf{Z} does contain a fork, namely the path $U \rightarrow V \rightarrow W$ but the middle node V is not in \mathbf{Z} making condition 1 of definition 2.19 false. However, the path $X \rightarrow U \rightarrow V \rightarrow W \rightarrow Y$ contains two colliders, i.e. U and W, which neither themselves nor their descendents, S_1 and S_2 , are in \mathbf{Z} making the condition 2 of definition 2.19 be true. Therefore, **X and Y are d-separated** given $\mathbf{Z} = \emptyset$.

In case b) let $\mathbf{X} = \{X\}$, $\mathbf{Y} = \{Y\}$ and $\mathbf{Z} = \{S_1, S_2\}$. The path between **X** and **Y** is, as in the previous case, $X \rightarrow U \rightarrow V \rightarrow W \rightarrow Y$. \mathbf{Z} does contain a fork, namely the path $U \rightarrow V \rightarrow W$

$V \rightarrow W$ but the middle node V is not in \mathbf{Z} making condition 1 of definition 2.19 false. The nodes containing colliders are U and W that are not in \mathbf{Z} but their descendants are now in fact in \mathbf{Z} making condition 2 false as well. Therefore, \mathbf{X} and \mathbf{Y} are **d-connected** given \mathbf{Z} .

In case c) let $\mathbf{X} = \{X\}$, $\mathbf{Y} = \{Y\}$ and $\mathbf{Z} = \{S_1, S_2, V\}$. The path between \mathbf{X} and \mathbf{Y} is, as in the two previous cases, $X \rightarrow U \rightarrow V \rightarrow W \rightarrow Y$. \mathbf{Z} does contain a fork, namely the path $U \rightarrow V \rightarrow W$ and now the middle node V is indeed in \mathbf{Z} making condition 1 of definition 2.19 true. The nodes containing colliders are U and W that are not in \mathbf{Z} but their descendants are in fact in \mathbf{Z} making condition 2 false. Because one of the conditions is true and there is no other path between \mathbf{X} and \mathbf{Y} , path $X \rightarrow U \rightarrow V \rightarrow W \rightarrow Y$ is said to be **d-separated** by \mathbf{Z} .

A final example is presented below to illustrate that it is very important to take into account every possible path between a pair of nodes so that the criterion of d-separation can be applied correctly. This example is taken from Pearl (1988) and is shown in figure 2.9.

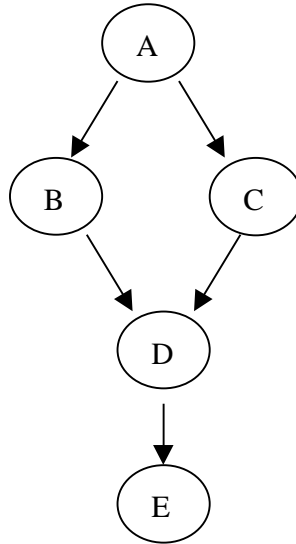


Figure 2.9: Another example for d-separation

Let $\mathbf{X} = \{B\}$, $\mathbf{Y} = \{C\}$ and $\mathbf{Z} = \{A\}$. The paths between \mathbf{X} and \mathbf{Y} are: $B \rightarrow A \rightarrow C$ and $B \rightarrow D \rightarrow C$. Remember that **every** path between \mathbf{X} and \mathbf{Y} has to be blocked by the set of nodes in \mathbf{Z} to correctly affirm that all the paths between \mathbf{X} and \mathbf{Y} are d-separated by \mathbf{Z} . In the first path, $B \rightarrow A \rightarrow C$, condition 1 of definition 2.19 is true, namely that there is a fork in the path in which the middle node (A) is in \mathbf{Z} . Condition 2 holds as well because in this path, $B \rightarrow A \rightarrow C$, there is not a collider in which A or its descendants can make the condition false. In path $B \rightarrow D \rightarrow C$, condition 1 does not hold because there is not a serial connection or a fork that could make the condition true. However, the path $B \rightarrow D \rightarrow C$ contains a collider on D and is not in \mathbf{Z} or has descendants in \mathbf{Z} either. Therefore, \mathbf{X} and \mathbf{Y} are **d-separated** by \mathbf{Z} .

Now, let $\mathbf{X} = \{B\}$, $\mathbf{Y} = \{C\}$ and $\mathbf{Z} = \{A, E\}$. The paths between \mathbf{X} and \mathbf{Y} are: $B \rightarrow A \rightarrow C$ and $B \rightarrow D \rightarrow C$. In the first path, $B \rightarrow A \rightarrow C$, condition 1 of definition 2.19 is true, namely that there is a fork in the path in which the middle node (A) is in \mathbf{Z} . Condition 2 holds as well because in this path, $B \rightarrow A \rightarrow C$, there is no collider in which A or its descendants can make the condition false. In path $B \rightarrow D \rightarrow C$, condition 1 does not hold because there is not a serial connection or a fork that could make the condition true. As in the previous case, the path $B \rightarrow D \rightarrow C$ contains a collider on D but now a descendent of

D, namely E, which is actually in **Z** making condition 2 false. Therefore, because Z blocks not every path from X to Y, then it is said that **X** and **Y** are **d-connected** given **Z**.

Before arriving to the formal definition of what a Bayesian network is, another one is needed, which has to do with the concept of I-mapness, but this time taking in account a directed acyclic graph (DAG).

Definition 2.20 (Pearl 1988). A DAG D is said to be an **I-map** of a dependency model M if every d-separation condition displayed in D corresponds to a valid conditional independence relationship in M, i.e., if for every three disjoint sets of vertices **X**, **Y** and **Z** we have

$$\langle \mathbf{X} \mid \mathbf{Z} \mid \mathbf{Y} \rangle_D \implies I(\mathbf{X}, \mathbf{Z}, \mathbf{Y})_M$$

A DAG is a **minimal I-map** of M if none of its arrows can be deleted without destroying its I-mapness.

The concept of I-mapness of a DAG can also be expressed as in Spirtes' et al. (1993) terminology, which is called the Markov Condition as the following definition states.

Definition 2.21 (Spirtes, Glymour et al. 1993). **Markov Condition.** A directed acyclic graph G over a set of nodes **V** and a probability distribution P(**V**) satisfy the Markov Condition if and only if for every W in **V**, W is independent of $\mathbf{V} \setminus (\mathbf{Descendants}(W) \cup \mathbf{Parents}(W))$ given **Parents**(W).

In simple terms, the Markov condition can be stated as follows: “Any node is conditionally independent of its nondescendants, given its parents” (Cooper 1999 p. 4).

Recall that W is its own descendent. Comparing definition 2.21 to definition 2.20, it is possible to say that D is an **I-map** of P. An example taken from Spirtes et al. (1993) (figure 2.10) will be useful to clarify these equivalent definitions.

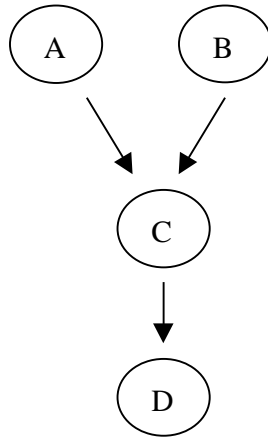


Figure 2.10: An example of the Markov condition

Let $\mathbf{X} = \{A\}$, $\mathbf{Y} = \{B\}$ and $\mathbf{Z} = \emptyset$. The Markov Condition or the I-mapness definition entails $I(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$. Let now $\mathbf{X} = \{D\}$, $\mathbf{Y} = \{A, B\}$ and $\mathbf{Z} = C$. The Markov Condition or the I-mapness definition entails $I(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$.

There is another definition in the terminology of Spirtes et al. (1993) called the Minimality Condition.

Definition 2.22 (Spirtes, Glymour et al. 1993) **Minimality Condition.** If G is a DAG over the set of nodes \mathbf{V} and P a probability distribution over \mathbf{V} , (G, P) satisfies the Minimality Condition if and only if for every proper subgraph H of G with node set \mathbf{V} , (H, P) does not satisfy the Markov Condition.

According to Spirtes et al.'s (1993) criteria, if a distribution $P(\mathbf{V})$ satisfies both the Markov and the Minimality conditions for a DAG G , then it is said that, as in Pearl's (Pearl 1988) terms, G is a **minimal I-map** of P .

Definition 2.23 (Pearl 1988). Given a probability distribution P on a set of variables \mathbf{U} , a DAG $D = (U, E)$ is called a **Bayesian Network** of P if and only if D is a minimal I-map of P .

Before presenting some examples of how a Bayesian network can represent a joint probability distribution in an economical and compact way, some other definitions are necessary. The following definitions also serve as a support theory for the algorithm proposed in this work.

Definition 2.24 (Pearl 1988). Let M be a dependency model defined on a set $U = \{X_1, X_2, \dots, X_n\}$ of elements and let d be an ordering $(X_1, X_2, \dots, X_i, \dots)$ of the elements of U . The **boundary strata** of M relative to d is an ordered set of subsets of U , $(B_1, B_2, \dots, B_i, \dots)$, such that each B_i is a Markov boundary of X_i with respect to the set $U_{(i)} = \{X_1, X_2, \dots, X_{i-1}\}$, i.e., B_i is a minimal set satisfying $B_i \subseteq U_{(i)}$ and $I(X_i, B_i, U_{(i)} - B_i)$. The DAG created by designating each B_i as parents of vertex X_i is called a boundary DAG of M relative to d .

Theorem 2.5 Verma 1986 (Pearl 1988). Let M be any semigraphoid (i.e., any dependency model satisfying the axioms of equations 2.19 through 2.22). If D is a boundary DAG of M relative to any ordering d , then D is a minimal I-map of M .

As Pearl (1988) points out, theorem 2.5 is the key for building and testing Bayesian networks as it is presented via the following three corollaries.

Corollary 2.1 (Pearl 1988). Given a probability distribution $P(x_1, x_2, \dots, x_n)$ and any ordering d of the variables, the DAG created by designating as parents of X_i any minimal set B_i of predecessors satisfying

$$P(x_i | B_i) = P(x_i | x_1, \dots, x_{i-1}), \quad B_i \subseteq \{X_1, X_2, \dots, X_{i-1}\} \quad (2.32)$$

is a Bayesian network of P .

Equation 2.32 means that for every X_i , there is some subset

$$B_i \subseteq \{X_1, X_2, \dots, X_{i-1}\}$$

such that $\{X_1, X_2, \dots, X_{i-1}\} \setminus B_i$ are conditionally independent given B_i .

Equation 2.32 resembles the chain rule of probability of equation 2.9. An example shown in figure 2.11 will be very useful to show the implications of corollary 2.1.

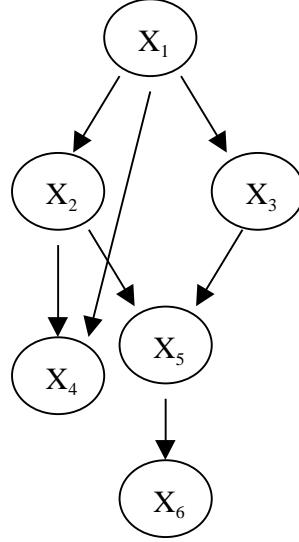


Figure 2.11: A Bayesian network representing the probability distribution $P(x_1, x_2, x_3, x_4, x_5, x_6) = P(x_6|x_5) \cdot P(x_5|x_2, x_3) \cdot P(x_4|x_1, x_2) \cdot P(x_3|x_1) \cdot P(x_2|x_1) \cdot P(x_1)$

As corollary 2.1 points out, the conditional probabilities $P(x_i | x_{1..i-1})$ shown in figure 2.11 are sufficient to reconstruct the original joint probability distribution. In this case, the distribution of the variables X_1, X_2, X_3, X_4, X_5 and X_6 . Taking the chain rule formula and a certain ordering of the variables d , then equation 2.32 will yield:

$$\begin{aligned}
 P(x_1, x_2, \dots, x_n) &= P(x_n | x_{n-1}, \dots, x_1) P(x_{n-1} | x_{n-2}, \dots, x_1) \dots P(x_3 | x_2, x_1) P(x_2 | x_1) P(x_1) \\
 &= \prod_i P(x_i | x_{1..i-1}) \quad (2.33)
 \end{aligned}$$

For this example, the joint distribution can be denoted as:

$$P(x_1, x_2, x_3, x_4, x_5, x_6) = P(x_6|x_5) \cdot P(x_5|x_2, x_3) \cdot P(x_4|x_1, x_2) \cdot P(x_3|x_1) \cdot P(x_2|x_1) \cdot P(x_1)$$

If the figure 2.10 is taken now, then the joint probability distribution it represents can be factorised as follows:

$$P(A,B,C,D) = P(D|C) \cdot P(C|A,B) \cdot P(B) \cdot P(A)$$

As can be seen, taking into account corollary 2.1, the structure of a Bayesian network depends heavily on the node ordering d . In other words, if the order d of the variables is chosen carelessly, then it is possible that a graph becomes a complete one (a fully connected network). For instance, in the example of figure 2.10, using the following node ordering, A, B, C, D, the conditional independencies are:

$$P(A \mid \emptyset) = P(A)$$

$$P(B \mid A) = P(B)$$

$$P(C \mid A,B) = P(C \mid A,B)$$

$$P(D \mid A,B,C) = P(D \mid C)$$

However, if the ordering is reversed, i.e., D, C, B, A, then the conditional independencies are:

$$P(D \mid \emptyset) = P(D)$$

$$P(C \mid D) = P(C \mid D)$$

$$P(B \mid C,D) = P(B \mid C,D)$$

$$P(A \mid B,C,D) = P(A \mid B,C,D)$$

which would represent a fully connected network structure as shown in figure 2.12. So, in the worst case, $n!$ different variable ordering would have to be explored in order to find the best one. This could be a very time-consuming and complex task.

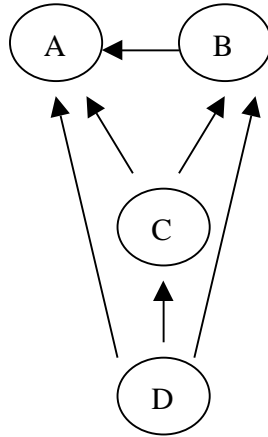


Figure 2.12: A fully connected Bayesian network

Recall that, in section 2.2, it was said that a complete graph represents a trivial I-map. Since this is true, then the graph in figure 2.12 is still an I-map, which means that all conditional independencies implied by the network structure (according to the d-separation criterion) are valid in the probability distribution P . Because of this, any chosen variable ordering will be an I-map; this observation determines a corollary where an order-independent definition of what a Bayesian network is can be established.

Corollary 2.2 (Pearl 1988). Given a DAG D and a probability distribution P , a necessary and sufficient condition for D to be a Bayesian network of P is that each variable X be conditionally independent of all its non-descendants, given its parents pa_X , and that no proper subset of pa_X satisfy this condition.

The last corollary of the series of the three mentioned above is the next one that follows from corollary 2.2.

Corollary 2.3 (Pearl 1988). If a Bayesian network D is constructed by the boundary-strata method in some ordering d , then any ordering d' consistent with the direction of arrows in D will give rise to the same network topology.

Corollary 2.3 ensures that the set X_i satisfies equation 2.32 if and only if the new set of X_i 's predecessors does not contain any of X_i 's old descendants. From this corollary, it is possible to assert that, once the network structure is built, the original order is no longer important but only the partial ordering shown in the network.

The following corollary provides an important definition for the concept of independence in Bayesian networks.

Corollary 2.4 (Pearl 1988). In any Bayesian network, the union of the following three types of neighbours is sufficient for forming a **Markov blanket** of a node X : the direct parents of X , the direct successors of X and all direct parents of X 's direct successors.

The definition of a Markov blanket is very important since the node X is conditionally independent (d-separated) of all other nodes in the Bayesian network given its Markov blanket.

In the example of figure 2.9, the Markov blanket of node C is $\{A,D,B\}$. Node A corresponds to the direct parent of C , node D corresponds to the direct successor of C and node B corresponds to the direct parent of C 's direct successor. The only node that is outside the Markov blanket of C is node E . Therefore, according to corollary 2.4, C and E are conditionally independent given $\{A,D,B\}$.

As said previously, the fact that the d-separation criterion cannot be improved by any means is given by the next theorem.

Theorem 2.6 Geiger and Pearl, 1988 (Pearl 1988). For any DAG D there exists a probability distribution P such that D is a **perfect map** of P relative to d-separation, i.e., P embodies all the independencies portrayed in D and no others.

Definition 2.25 (Pearl 1988). A dependency model M is said to be **causal** (or **DAG isomorph**) if there is a DAG D that is a perfect map of M relative to d-separation, i.e.,

$$I(\mathbf{X}, \mathbf{Z}, \mathbf{Y})_M \quad < \mathbf{X} \mid \mathbf{Z} \mid \mathbf{Y} >_D$$

This DAG-isomorph characteristic of a model M is also known as **stability** (Pearl 2000) and **faithfulness** (Spirtes, Glymour et al. 1993).

In sum, a Bayesian network is a directed acyclic graph (DAG) consisting of (Cooper 1999):

- a) nodes representing random (normally) discrete variables, arcs representing probabilistic relationships among these variables and
- b) for each node there is a probability distribution associated to that node given the state of its parents.

Recall that if a node V does not have any parents, i.e., $\text{pa}_V = \emptyset$ then the probability distribution of node V is just the marginal one: $P(V)$. In words of Cooper (Cooper 1999, p. 3) "A Bayesian network specifies graphically how the node probabilities factor to specify a joint probability distribution over all the nodes (variables)". This can be written as equation 2.33. As can be noticed from this equation, this factorisation of the joint probability permits to represent this joint distribution in a compact and economical way. An example taken from Cooper (1999) (figure 2.13) will be presented to illustrate this idea.

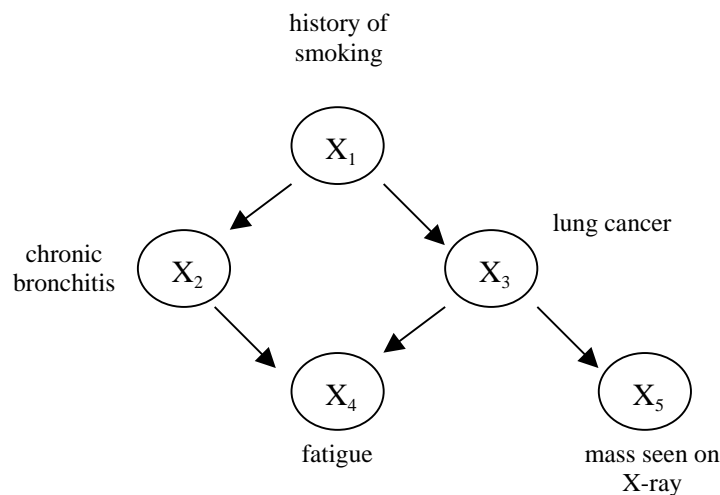


Figure 2.13: A Bayesian network showing the probabilistic relationships among 5 different variables

Table of figure 2.14 illustrates the probabilities associated to each node taking part in the example.

$P(X_1=0)=0.80$	$P(X_1=1)=0.20$
$P(X_2=0 \mid X_1=0)=0.95$	$P(X_2=1 \mid X_1=0)=0.05$
$P(X_2=0 \mid X_1=1)=0.75$	$P(X_2=1 \mid X_1=1)=0.25$
$P(X_3=0 \mid X_1=0)=0.99995$	$P(X_3=1 \mid X_1=0)=0.00005$
$P(X_3=0 \mid X_1=1)=0.997$	$P(X_3=1 \mid X_1=1)=0.003$
$P(X_4=0 \mid X_2=0, X_3=0)=0.95$	$P(X_4=1 \mid X_2=0, X_3=0)=0.05$
$P(X_4=0 \mid X_2=0, X_3=1)=0.50$	$P(X_4=1 \mid X_2=0, X_3=1)=0.50$
$P(X_4=0 \mid X_2=1, X_3=0)=0.90$	$P(X_4=1 \mid X_2=1, X_3=0)=0.10$
$P(X_4=0 \mid X_2=1, X_3=1)=0.25$	$P(X_4=1 \mid X_2=1, X_3=1)=0.75$
$P(X_5=0 \mid X_3=0)=0.98$	$P(X_5=1 \mid X_3=0)=0.02$
$P(X_5=0 \mid X_3=1)=0.40$	$P(X_5=1 \mid X_3=1)=0.60$

Figure 2.14: The marginal and conditional probabilities for the Bayesian network of figure 2.13

For the variable X_1 , 0 represents no and 1 represents yes. For the rest of the variables, 0 represents that the symptom is absent and 1 represents that the symptom is present.

For this example, the factorisation of the joint probability distribution would be calculated as follows:

$$P(X_1, X_2, X_3, X_4, X_5) = P(X_5 \mid X_3) \cdot P(X_4 \mid X_2, X_3) \cdot P(X_3 \mid X_1) \cdot P(X_2 \mid X_1) \cdot P(X_1)$$

If for instance, the probability $P(X_1=1, X_2=1, X_3=1, X_4=1, X_5=1)$ is required to be calculated, then that probability would be computed taking in account the conditional probabilities from table of figure 2.14; as follows:

$$P(X_5=1 \mid X_3=1) = 0.60$$

$$P(X_4=1 \mid X_2=1, X_3=1) = 0.75$$

$$P(X_3=1 \mid X_1=1) = 0.003$$

$$P(X_2=1 \mid X_1=1) = 0.25$$

$$P(X_1=1) = 0.20$$

$$P(X_1=1, X_2=1, X_3=1, X_4=1, X_5=1) = 0.60 \times 0.75 \times 0.003 \times 0.25 \times 0.20 = 0.0000675$$

To stress the importance of the factorisation of the joint probability distribution in order to represent it in a more compact and economical way, the previous example can be taken. Recall that there are 5 binary variables, i.e., 2^5-1 parameters are needed if the joint probability were required to be calculated. In other words, the number of probabilities in the table of figure 2.14 would be 31. The 32nd probability can be inferred using equation 2.14. Imagine now that the number of variables is not 5 but 37. Then the number of parameters needed for defining the joint distribution would be (at least) $2^{37}-1$.

In contrast, from figure 2.14, only 11 probabilities are required to represent the joint distribution from the local conditional probability distributions portrayed by the structure of the Bayesian network of figure 2.13. The complementary 11 probabilities in table of figure 2.14 are also shown but they can be calculated using equation 2.14 as well; that is why only 11 probabilities are in fact needed to represent the joint probability distribution.

As can be seen from the previous example, it is possible to provide answers of global nature (joint probability) from these local distributions, as Pearl (2000) points out. Furthermore, because of the power of conditional independencies as well as the local conditional distributions being represented by a Bayesian network, it is possible, most of the time, to save substantial amounts of storage space and processing time. However, when a Bayesian network is a highly-dense connected graph, this is not the case.

There is a formula to calculate the number of parameters (probabilities) needed to specify a joint distribution from conditional ones in a Bayesian network. This formula defines what is called the **dimension** of the Bayesian network and is written below.

$$d = \prod_{i=1}^n q_i(r_i - 1) \quad (2.34)$$

where r_i represents the number of possible states of the variable X_i and q_i represents the number of configurations of the parents of X_i , i.e., X_i . If a node X_i has no parents then $q_i=1$. The number of configurations of the parents of node X_i is determined by the principle of multiplication (Hines and Montgomery 1997).

In the Bayesian network of figure 2.13, X_1, X_2, X_3, X_4 and X_5 are binary variables so $r_i = 2$ for $i = 1, 2, 3, 4, 5$. The number of configurations of X_i are as follows:

For X_1 , $q_1 = 1$

For X_2 , $q_2 = 2$

For X_3 , $q_3 = 2$

For X_4 , $q_4 = (2 \times 2) = 4$

For X_5 , $q_5 = 2$

Once the r_i and the q_i are determined, then equation 2.34 can be applied. The result is shown below.

$$d = 1(1) + 2(1) + 2(1) + 4(1) + 2(1) = 1 + 2 + 2 + 4 + 2 = 11$$

which corresponds to the half number of probabilities of table of figure 2.14. Recall that the complementary 11 probabilities can be calculated using equation 2.14.

This result shows that, unless the graph is highly-dense connected as mentioned above, the saving of storage space and processing time is substantial. Thus, Bayesian networks provide a very good and compact way to represent a joint probability distribution by means of local conditional probabilities which, most of the time, implies that each variable in the network depends only on a small subset of the others.

In the next section the suitability that this framework gives for representing uncertainty, knowledge and beliefs is presented.

2.4 Representation of uncertainty, knowledge and beliefs in Bayesian Networks.

As mentioned in the previous section, a Bayesian network is a graphical formalism that permits an efficient and compact representation of probability distributions. In other words, Bayesian networks are graphical representations of probabilistic relationships among variables of interest. In such networks, two different parts can clearly be detected: a qualitative part and a quantitative part. The qualitative part is represented through the structure of the network, which contains important information about the dependencies / independencies portrayed by such networks. The quantitative part corresponds to the local probability models represented by each node as conditional probabilities. It is important to recall the example shown in figure 2.13 and table of figure 2.14. Using equation 2.33, it was possible to recover the whole joint probability distribution from the local conditional ones. So, summing the qualitative and the quantitative parts, namely, the structure of the network and the conditional probabilities attached to each node of this network, will be enough to represent a unique joint distribution over a certain domain. In the qualitative part, the well-known criterion of d-separation enters into consideration. In the quantitative part, as said above, a set of conditional probability distributions now enters into consideration. These distributions are usually multinomial if the variables are discrete; linear Gaussian if they are continuous and a combination of those if the variables in the network are both discrete and continuous.

Pearl (1988) points out clearly how the representation of uncertainty, knowledge and beliefs can be carried out using Bayesian networks. First of all, one of the most important concepts for representing such entities is the notion of relevance. In the context of Bayesian networks, if a variable is relevant to another one, then this relevance relationship is represented, in the simplest case, by connecting the two variables with a directed arc pointing from the first variable to the second one. As can be noticed, this

relevance is intrinsically represented by the dependency relationship between two variables given another one, which can be the empty set, i.e., $\sim I(X,Z,Y)$. On the other hand, if two variables are not relevant to each other, now this irrelevance is denoted by the notion of independence, i.e., $I(X,Z,Y)$. In sum, Bayesian networks are very useful engines to represent and manipulate relevance relations among variables. The concepts of d-separation and d-connection previously presented can be used to know, at a determined time, what information seems to be relevant to another.

Although probability theory manipulates numbers, it has the capability to express qualitative and primitive relationships of the probability language such as likelihood, conditioning, relevance and causation providing a coherent framework that dictates how beliefs should change if partial or uncertain information is at hand. For a good discussion about this see Pearl (1988).

The classic relation of **likelihood** can be stated as "A is more likely than B"; probabilities are used to assess how probable is A or B to happen at a determined moment and with the available information at hand. Knowing the likelihood of A and B is important for performing actions such as decision-making, prediction and diagnosis. The quantitative part of a Bayesian network, say the marginal or conditional probabilities attached to each node in the network, is the key to represent and manipulate this likelihood relationship.

Conditioning can be compared to the English phrase "if C then A"; say for example, if the pavement is wet then this conditioning relationship can be captured by the probability theory statement known as Bayes' conditionalization " $P(A|C) = p$ ". In its simple form, Bayes' conditionalization reflects the representation of uncertainty, knowledge and beliefs: given that C is known, A takes a possible value with an uncertainty p which is a real number between 0 and 1 as stated by the equation 2.1. In other words, the previous statement combines knowledge and beliefs attributing to A a degree of belief p given the knowledge C. C is called the **context** (knowledge) of the belief in A. This Bayes' conditionalization is the heart of well-known definition of the conditional probability shown in equation 2.6. Bayes' conditionalization is a very powerful device that permits retraction of previous conclusions if new knowledge (C) is acquired at a posterior time.

This new knowledge will probably produce a revision of beliefs, which will be changed if it is the case. Thus, the quantitative part of a Bayesian network allows to represent and manipulate then uncertainty, knowledge and beliefs as claimed at the beginning of this section.

The qualitative part of a Bayesian network is the one that now permits to represent the notion of **relevance**. Relevance has the power to indicate potential changes of beliefs due to a change in knowledge. As mentioned above, this relationship can be captured using the concept of d-connection that does not need any numerical manipulations but only logical ones; hence the qualitative nature of relevance is captured only by the structure of the network.

Finally, Bayesian networks can also capture the relationship of **causation**. Bayesian networks are also known as causal networks, as mentioned in section 1.1 of chapter 1. This name provides an insight of how to construct this kind of structures. Before commenting on how this is done in more detail in the following chapter, it can be said that arcs connecting variables can be taken as knowledge about causes and effects in a certain domain. Causation has the ability to separate the relevant information from the superfluous one as the whole structure of a Bayesian network, altogether with the notions of d-separation and d-connection, does it. Because of causation's asymmetry, it is possible to represent in a Bayesian network more complex relevance relationships such as nontransitive and induced dependencies as explained in figure 2.7. In the next chapter, some important aspects on the different ways to construct a Bayesian network, as well as the main and typical problems for doing this, will be reviewed.

Chapter 3

Learning Bayesian Networks

In the previous chapter, the formal definition of a Bayesian network and the power and suitability of such networks for representing and manipulating uncertainty, knowledge and beliefs were presented. However, in both chapter 1 and chapter 2 very little was said about the problems usually found when constructing such networks. Thus, this chapter firstly presents the typical problems encountered when a Bayesian network is being built. Secondly, it describes different approaches for building such a network. Finally, it explains a hybrid methodology that could lead to the construction of Bayesian networks in a more robust and sound way.

3.1 Typical problems in constructing Bayesian Networks.

The qualitative and quantitative nature of Bayesian networks determine basically what Friedman and Goldszmidt (1998a) call the **learning problem**, which comprises a number of combinations of the following subproblems:

- Structure learning
- Parameter learning
- Probability propagation
- Determination of missing values (also known as missing data)
- Discovery or determination of hidden or latent variables

Structure learning is the part of the learning problem that has to do with finding the topology of the Bayesian network; i.e., to construct a graph that shows qualitatively the

dependence / independence relationships among the variables involved in a certain problem in order to know and identify easily which variables are relevant / irrelevant for the hypothesis being considered and tested (Sucar 1994; Buntine 1996; Friedman and Goldszmidt 1998a).

Parameter learning refers to the quantitative part of a Bayesian network which has to do with the determination of probability distributions, either marginal or conditional, for each node in the network given a certain structure or topology of this network (Sucar 1994).

The **probability propagation** problem has also to do with the quantitative part of a Bayesian network referring to the updating of the probability values for all the nodes once one or more specific values in certain nodes become known. In other words, the process of instantiating some of the variables in the network and propagating their effects through this network is called probability propagation (Sucar 1994).

In order to support empirically both the topology and the parameters of a certain Bayesian network, a common practice is to collect and analyse data about the domain being modelled. It is also common that, because of many conditions and circumstances, the data collected have some of the variables with missing values (empty cells in a database). The problem of determining what the most likely value or values missing in the data are, is what is called the determination of **missing data** or **missing values**. It also happens very often that, as Spirtes et al. point out (1993), some relevant variables for explaining the phenomenon under study simply fail to be measured. That is to say, if it is noticed that the production of the data are being influenced by a probable not easily perceived cause, then it is possible to postulate the existence of one or some hidden variables to explain what can be responsible for this abnormal data production. The problem of discovering such unmeasured common causes is known as the determination or discovery of **hidden** or **latent variables** also known as **confounders** (Spirtes, Glymour et al. 1993). It is this last term which gives another name for this problem: **confounding** (Spirtes, Glymour et al. 1993).

This thesis focuses on the determination of the structure of a Bayesian network from data; this is why it is only this problem that will have further elaboration from now on. The reader is referred to (Buntine 1996) for an extensive literature review on all the above subproblems.

Basically there are three different ways for determining the topology of a Bayesian network: the **manual** or **traditional** approach (Diez 1996), the **automatic** or **learning** approach (Pearl 1988; Spirtes, Glymour et al. 1993), in which the work presented in this dissertation is inspired, and the **Bayesian** approach, which can be seen as a combination of the previous two (Heckerman 1998).

3.2 Traditional approach.

In the traditional approach, the structure of a Bayesian network is usually given by the human expert helped by the knowledge engineer who carries out the elicitation process, as mentioned in section 1.1 of chapter 1. However, as also pointed out in that section, the construction of either a rule-based expert system or a Bayesian network from human experts, is often a very complex and time-consuming task. This knowledge elicitation process is very difficult mainly because these very same experts have big problems in making their knowledge explicit. Moreover, such a process is very time-consuming because the information has to be collected manually (Jackson 1990; Neapolitan 1990; Sucar 1994; Buntine 1996; Cruz-Ramirez 1997; Monti and Cooper 1998).

Although this is a very complex and time-consuming task, the typical, easiest and most consistent way to build such a structure within the traditional approach is to think about the relationships among the variables as if they were **causal** (Heckerman 1998; Cooper 1999; Pearl 2000). This means that if two variables are connected and the first one is thought as the cause of the second one (the effect) then this causal relationship is represented by directing an arc (arrow) from the first variable to the second one. This process can be done recursively, namely, for every pair of variables the expert needs to determine if one variable is cause of the other and keep applying this criterion until there

are no more variables. As Pearl and Heckerman point out (Heckerman 1997; Heckerman 1998; Pearl 2000), constructing a Bayesian network this way, i.e., using causal knowledge, has one main advantage: it is easier to incorporate prior knowledge, which is more meaningful, accessible and reliable. This way of constructing Bayesian belief networks is also useful, though not completely easy or straightforward for the reasons mentioned above. But it does provide a method that, if combined with statistical data, becomes a very powerful and more consistent tool for building such networks (Heckerman, Geiger et al. 1994; Chickering 1996; Pearl 1996; Chickering, Heckerman et al. 1997; Pearl 2000).

3.3 Learning approach.

Friedman and Goldszmidt (1998a), Chickering (1996), Heckerman (1997, 1998) and Buntine (1996) give a very good and detailed account of the learning problem within this approach in Bayesian networks. The motivation for this approach is basically to solve the problem of the manual extraction of human experts' knowledge encountered in the traditional approach. This can be done using the data at hand about the problem under study in order to "mine" or determine the structure and the parameters of the Bayesian network, including the determination of hidden variables and missing values. In many situations and areas, the data (in the form of databases) can be more easily obtained than the experts' knowledge (Geiger, Heckerman et al. 1998; Monti and Cooper 1998). By making extensive use of these databases, the complexity of the learning problem mentioned above can be dramatically reduced. Thus, when the learning problem is solved by using either the data available alone or a combination of these data and the human experts' knowledge, this problem is known as **learning Bayesian networks from data** (Pearl 1988; Chickering 1996; Friedman and Goldszmidt 1998a; Heckerman 1998).

It is often believed, when collecting the data of the phenomenon under investigation, that some underlying process is the responsible for the production of these data. The principal idea in this learning approach is to build (learn) one out of various models (Bayesian networks) to represent that process which generated the data. Hence, it is

possible to discover the laws and principles governing the phenomenon under study. Many researchers have pursued this task with very good and promising results (Pearl 1988; Cooper and Herskovits 1992; Heckerman, Geiger et al. 1994; Buntine 1996; Chickering 1996; Cheng, Bell et al. 1998). A very important question arises when trying to look for such models: how to choose a model, if there are more than one, that best captures the features of the underlying process? The answer to this question has been guided by the criterion known as **Occam's razor**: the model that best fits the data in the simplest way is the best one (Pearl 1988; Bozdogan 2000; Grunwald 2000; Wasserman 2000; Zucchini 2000). This issue is very well known under the name of **model selection** (Geiger, Heckerman et al. 1998; Heckerman 1998; Cooper 1999; Bozdogan 2000; Grunwald 2000; Wasserman 2000; Zucchini 2000). Furthermore, according to Heckerman (1998), this problem has been also addressed using another approach called **selective model averaging** (Cheng, Bell et al. 1998; Heckerman 1998). Either approach has been subject of many controversies and discussions (Bozdogan 2000; Browne 2000; Busemeyer and Wang 2000; Cutting 2000; Grunwald 2000; Wasserman 2000; Zucchini 2000).

The philosophy behind model selection is to choose only one model among all possible models; this single model is treated as the “good” one and used as if it were the correct model. On the other hand, the philosophy behind selective model averaging is to choose a (computationally) manageable number of models among all possible models; these models are treated as the “good” ones and as if they were exhaustive.

The obvious question that comes immediately to one's mind for both approaches is how to measure this **goodness of fit** of the suggested models in order to decide whether these models are good or not (this notion of goodness of fit will be explained in chapter 4). For the sake of simplicity, for now just imagine that we have a certain goodness of fit measure; then the most intuitive and secure way to know which structure (and hence the parameters) is the best one for representing the underlying process responsible of producing the data is to start with a structure with the fewest number of arcs and then to find the parameters, if possible, that can fit the data best. This process is repeated until all the

possible structures are tested (assuming that there exist no hidden variables). Then, the best structure is chosen. As can be seen, it is always possible to find the best structure when such a process is followed. But the problem is far more complicated than it appears.

Robinson (Cooper and Herskovits 1992) has mathematically shown that finding the most probable Bayesian network structure has an exponential complexity on the number of the variables taking part in a determined problem. This means that, as the number of variables grows, the more difficult it becomes, if not practically (computationally) impossible, to test exhaustively. Robinson's formula (Cooper and Herskovits 1992) is presented in equation 3.1.

$$f(n) = \sum_{i=1}^n (-1)^{i+1} \sum_i^n (2^{i(n-i)}) f(n-i) \quad (3.1)$$

If for instance, two variables take part in a problem, i.e., $n = 2$, then the number of possible structures is 3. If $n = 3$, the number of possible structures is 25; for $n = 5$, the number of possible structures is now 29,000; for $n = 10$, the number of possible structures is 4.2×10^{18} . From these results it is possible to conclude that an exhaustive enumeration of all possible structures is, in general, not feasible even though the number of variables is not too large.

This complex problem resulted in much work on **heuristic** methods, namely, methods that use a certain kind of reliable criteria to avoid exhaustive enumeration (Pearl 1988; Neapolitan 1990; Whittaker 1990; Cooper and Herskovits 1992; Spirtes, Glymour et al. 1993; Heckerman, Geiger et al. 1994; Sucar 1994; Martinez-Morales 1995; Spirtes and Meek 1995; Buntine 1996; Chickering 1996; Chickering, Heckerman et al. 1997; Cruz-Ramirez 1997; Cheng, Bell et al. 1998; Friedman and Goldszmidt 1998a; Heckerman 1998; Jordan 1998; Glymour and Cooper 1999; Pearl 2000).

In other words, heuristic methods are used in order to reduce the search space so that the complexity of this space is computationally tractable and a good Bayesian network candidate, which has high probability of representing closely and reliably the real underlying probabilistic model P , can be proposed.

The work carried out in this research has to do only with the determination of the structure of a Bayesian network from data. The original algorithms proposed in this thesis will be presented in chapters 5 and 6. In the next subsection, some important heuristic algorithms and the philosophy behind them will be presented.

3.4 Learning Bayesian Networks from data.

Generally speaking, there exist two different kinds of heuristic methodologies for constructing the structure of a Bayesian network from data: **constraint-based algorithms** (also known as dependency analysis methods) and **search and scoring based algorithms** (also known as Bayesian methods) (Pearl 1988; Cooper and Herskovits 1992; Spirtes, Glymour et al. 1993; Chickering 1996; Cheng, Bell et al. 1998; Friedman and Goldszmidt 1998a; Heckerman 1998; Cooper 1999; Pearl 2000).

3.4.1 Constraint-based algorithms.

The motivation for the birth of constraint-based algorithms was that of the possibility of representing probabilistic dependence / independence relationships in probabilistic models using graphical models, as seen in chapter 2. Such connection is due to Pearl (1988). It is this connection that provides the key point for constructing Bayesian networks from data: if all the probabilistic relationships portrayed by a probabilistic model P can be represented by a Bayesian network D , a condition called **faithfulness** in Spirtes et al. terms (1993) (see also definition 2.25 of chapter 2), then it is possible to test the dependence / independence relationships, contained implicit in the data, with a certain conditional independence measure in order to construct such a network. Spirtes et al. (1993) have shown that most probabilistic models found in the real world are faithful to Bayesian networks. As mentioned in definition 2.25 in chapter 2, in Pearl's terms, this

faithfulness condition is equivalent to the **DAG-isomorphism** condition (Pearl 1988) also called **stability** (Pearl 1996; Pearl 2000). The faithfulness condition can be illustrated graphically to make this concept clearer. Figure 3.1 represents this condition (Friedman and Goldszmidt 1998a). If the faithfulness condition holds, then it is not possible to have **accidental** independencies; i.e., it is not possible to have simultaneously an arrow pointing from smoking to lung cancer (in **D**) and the conditional independence relation between smoking and lung cancer given \emptyset (in **P**).

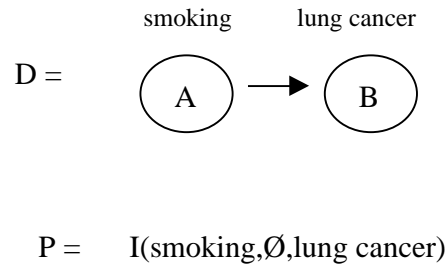


Figure 3.1: If the faithfulness condition holds, then **D** and **P** cannot be true simultaneously

The conditional independence measures usually used are χ^2 , G^2 and the **mutual** and **conditional mutual information** measures (Shannon and Weaver 1949; Pearl 1988; Spirtes, Glymour et al. 1993; Cheng 1998; Cheng, Bell et al. 1998). A slight variant of these two last measures (information measures), proposed originally by Kullback (1959), is used in the algorithms proposed in this thesis. The most representative algorithms that fall into this category (constraint-based algorithms) are: the **Chow-Liu** method for constructing trees (Pearl 1988; Cooper and Herskovits 1992), the **boundary DAG** algorithm (Pearl 1988; Cheng, Bell et al. 1998), the **IC** (inductive causation) algorithm (Pearl 2000), the **SGS** algorithm (Spirtes, Glymour et al. 1993; Cheng, Bell et al. 1998), the **Wermuth-Lauritzen** algorithm (Spirtes, Glymour et al. 1993; Cheng, Bell et al. 1998) and the **PC** algorithm (Spirtes, Glymour et al. 1993; Cheng, Bell et al. 1998), among others. All these algorithms assume that the **faithfulness** (stability) condition and the **Markov**

condition hold. Recall that the Markov condition can be stated easily as follows: “Any node is conditionally independent of its nondescedendants, given its parents” (Cooper 1999, p. 4). An example taken from Cooper and Herskovits (1992), which is presented below, will help to clarify the way this kind of algorithms generally works.

Imagine that, from table of figure 3.2, the task is to discover the structure underlying the probabilistic model responsible of producing the data in such a table.

case	a	b	c
1	present	absent	absent
2	present	present	present
3	absent	absent	present
4	present	present	present
5	absent	absent	absent
6	absent	present	present
7	present	present	present
8	absent	absent	absent
9	present	present	present
10	absent	absent	absent

Figure 3.2: 10-case database containing three different variables (a, b and c)

The first step is to determine which variables are dependent / independent to each other. For the sake of simplicity, the numerical calculations will not be presented but only the general form that the conditional independence (CI) measures can take. Following the terminology of Pearl (1988) presented in chapter 2, it is possible to ask the next questions:

1.- $I(a, \emptyset, b)$?

2.- $I(b, \emptyset, c)$?

3.- $I(a, b, c)$?

Each question can be answered using the frequencies (cell counts) in table of figure 3.2. If the value of the calculated conditional or marginal independence measure is smaller than a determined threshold ϵ , then the variables are (marginally or conditionally) independent; otherwise they are (marginally or conditionally) dependent. This threshold is known as the **significance level** represented in statistics by the Greek letter α (alpha). One very important thing to remark is the number of variables in the conditioning set. In the above three questions, the first and second one are independence tests of order zero because the conditioning set is the empty set. In the case of the third question, the independence test is of first order because the conditioning set contains only one member, namely, variable b . Notice that the bigger the conditioning set is, the more the data are needed to perform the required independence tests because the majority of such tests need the specification of the joint probability of the variables taking part in the conditioning set. Let us simply assume that the answers of the above questions are as follows:

1.- No

2.- No

3.- Yes

According to these answers and assuming that the faithfulness and the Markovian (definition 2.21 of chapter 2) conditions hold, it is possible now to construct a Bayesian network capable of representing such dependencies / independencies implicitly contained in this probabilistic model. Figure 3.3 depicts such a network.

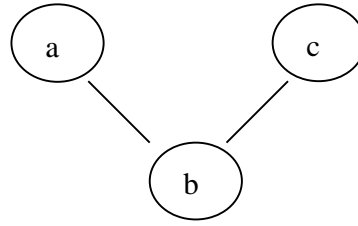


Figure 3.3: the undirected graph resultant from running a constraint-based algorithm with the data of figure 3.2

As it can be easily noticed, the arcs have no direction; this is because it is not possible to learn about arc direction from any conditional independence measure (Friedman and Goldszmidt 1998a) or probabilities alone (Spirtes, Glymour et al. 1993; Pearl 2000). The problem is that it is possible to choose different orderings of the arc directions that can represent the same independence constraints. That is to say, if the results of the above questions are taken, then the only one conditional independence relationship is $I(a,b,c)$ and hence there are three different possible Bayesian networks capable of representing this relation. Such networks are depicted in figures 3.4, 3.5 and 3.6.

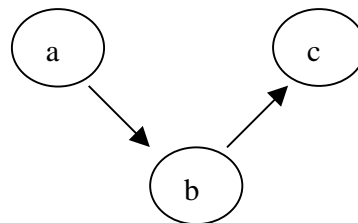


Figure 3.4: A first possible Bayesian network resulting from directing the arcs in figure 3.3 without violating the independence relationships represented in that network

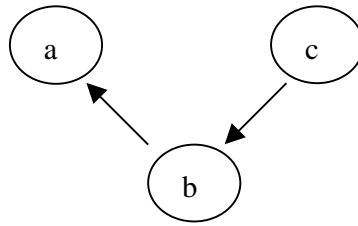


Figure 3.5: A second possible Bayesian network resulting from directing the arcs in figure 3.3 without violating the independence relationships represented in that network

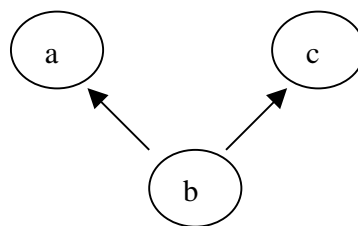


Figure 3.6: A third possible Bayesian network resulting from directing the arcs in figure 3.3 without violating the independence relationships represented in that network

Following Chickering and Heckerman’s notation (Chickering 1996; Heckerman 1997) then it can be said then that two structures are **equivalent** if they encode the same independence constraints. Moreover, Verma and Pearl have shown (Chickering 1996; Heckerman 1997) that two Bayesian network structures are equivalent if and only if they have the same **skeleton** and the same **v-structures**. Chickering describes the skeleton of any directed acyclic graph (DAG) as “the undirected graph resulting from ignoring the directionality of every edge” (Chickering 1996, p. 37). Pearl describes v-structures as “two converging arrows whose tails are not connected by an arrow” (Pearl 2000, p. 19). Figure 3.7 represents a v-structure.

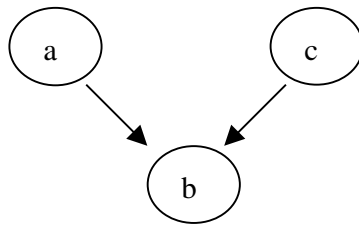


Figure 3.7: A Bayesian networks representing a v-structure

Note that figures 3.4, 3.5, 3.6 and 3.7 have the same skeleton, i.e., the same structure without taking in account the direction of the arcs. The skeleton of all these figures is then that shown by figure 3.3. However, figure 3.7 does not share the same v-structure with the other ones, namely, the fact that b has two parents: a and c. No other figure, apart from figure 3.7, has a v-structure. So figure 3.7 is not equivalent to figures 3.4, 3.5 or 3.6. A v-structure is also called a **converging connection** (Jensen 1996) or an **unshielded collider** (Spirtes, Glymour et al. 1993; Spirtes, Scheines et al. 1994; Scheines, Glymour et al. 1999a). This problem of the directionality of the arcs can be solved basically in three different ways. One solution is to ask the experts about the most probable direction of the arcs. In order to perform such a task, they can use their, say, causal knowledge about the problem being modelled. This problem of directing the arcs in a Bayesian network is for much, less complex than proposing a Bayesian network from **tabula rasa** (Spirtes and Meek 1995). Another criteria that the experts usually take into account to direct the arcs is temporal information. If a Bayesian network is constructed as a causal network, then temporal information gives a good clue of how to direct the mentioned arcs.

Another solution is a heuristic qualitative (graphical) criterion used once an undirected graph is obtained from running any constraint-based algorithm. For example, the PC algorithm (Spirtes, Glymour et al. 1993) has two different steps for suggesting possible direction of the arcs. Moreover, Verma and Pearl (Pearl 2000) have shown that four different rules are required to obtain a maximally oriented pattern.

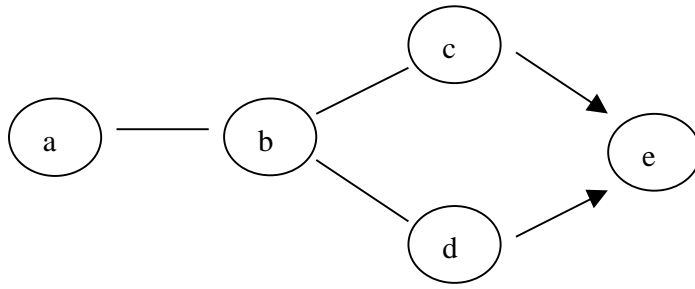


Figure 3.8: A graph containing both directed and undirected arcs (a mixed graph)

As figure 3.8 shows, a **pattern** is defined as a mixed graph containing both a (possibly empty) set of directed arcs and a set of undirected arcs (Spirtes, Glymour et al. 1993). These rules are (Pearl 2000):

R_1 : Orient $b - c$ into $b \rightarrow c$ whenever there is an arrow $a \rightarrow b$ such that a and c are nonadjacent.

R_2 : Orient $a - b$ into $a \rightarrow b$ whenever there is a chain $a - c - b$.

R_3 : Orient $a - b$ into $a \rightarrow b$ whenever there are two chains $a - c - b$ and $a - d - b$ such that c and d are nonadjacent.

R_4 : Orient $a - b$ into $a \rightarrow b$ whenever there are two chains $a - c - d$ and $c - d - b$ such that c and b are nonadjacent.

A partially oriented acyclic graph or pattern DAG, commonly known as a **pDAG**, forms what is called an **I-equivalence** class of Bayesian networks (Heckerman 1997; Friedman and Goldszmidt 1998a). The graph in figure 3.3 represents as a pDAG containing the probabilistic information $I(a,b,c)$. Thus, figure 3.3 represents an I-equivalence class of Bayesian networks depicted in figures 3.4, 3.5 and 3.6. The Bayesian network of figure 3.7, as said before, is not equivalent to figures 3.4, 3.5 or 3.6 so the pDAG of figure 3.3 does not include the Bayesian network of figure 3.7 as part of its I-equivalent network class.

Note also that the solution of this problem is partial since sometimes not even a single arc can be oriented.

The last solution to the problem of directing arcs is a combination, whenever possible, of the two first solutions; this combination can provide a more complete solution than that offered by the second solution: after a constraint-based algorithm proposes a pDAG, the experts could propose directions for some of the arcs, then an iterative application of the above rules can lead to a graph more oriented, if not totally, than that obtained by only applying the above rules alone.

In sum, the constraint-based approach performs tests of conditional independence, given a database, to search a network structure consistent with the dependencies / independencies portrayed by the probabilistic model P responsible of generating such a database.

3.4.2 Search and scoring based algorithms.

The motivation for the appearance of this kind of algorithms was not to use conditional independence tests as an oracle that decides the inclusion or exclusion of an arc between two variables but to find another method to do this same thing (the advantages and disadvantages of each of these approaches are explained in section 3.4.3). Having this in mind, the philosophy of the search and scoring methodology has the two following typical characteristics:

- 1.- define a measure (score) to evaluate how well the data fit with the proposed Bayesian network structure (goodness of fit) and

2.- define a searching engine that seeks for a structure that maximizes or minimizes, according to which is the case, the score mentioned above.

For the first step, there are a number of different scoring metrics such as the Bayesian Dirichlet scoring function (**BD**), the cross-validation criterion (**CV**), the Bayesian information criterion (**BIC**), the minimum description length (**MDL**), the minimum message length (**MML**) and the Akaike's information criterion (**AIC**), among others (Cooper and Herskovits 1992; Buntine 1996; Chickering 1996; Chickering, Heckerman et al. 1997; Heckerman 1998). For the second step, some well-known and classic search algorithms such as **greedy-hill climbing**, **best-first search** and **simulated annealing** (also known as **Metropolis**) can be applied (Chickering 1996; Chickering, Heckerman et al. 1997; Heckerman 1997; Friedman and Goldszmidt 1998a). The details of some of these criteria for steps 1 and 2 will be explained in chapter 4. Steps 1 and 2 are applied iteratively until there is not any modification that can improve the score, i.e., no new structure is better than the previous one.

The search and scoring based algorithms are also known as **greedy** algorithms (Cooper and Herskovits 1992; Chickering 1996) because in each step, when applying some or all defined operators, they choose the operator that, according to the scoring function, increases the probability of the resulting structure. In the framework of Bayesian networks, these operators can be:

- add a directed arc in either direction
- add an undirected arc
- reverse an existing arc
- delete either a directed arc or a undirected arc

As said before, in each step, every operator is applied and scored; the chosen one is that which has more potential to succeed, i.e., the one having the highest (or lowest) score which, according to this heuristic, is the most likely to reflect in the best way the underlying probability distribution. If the search and score procedure is applied alone, then there are basically three different search space initialisations: an empty graph, a complete graph or a random graph. The search initialisation chosen determines which operators can be used and applied. The general way the search and scoring algorithms works can be explained clearly using some graphical examples presented below.

The data in table of figure 3.2 are again taken as an example. As before, the problem is to represent the probabilistic model P responsible of producing such a table by means of a graphical model, say, a Bayesian network. However, the way a search and scoring algorithm works is different from that of the constraint-based methods. Suppose that the empty graph is taken as the search space initialisation. The next step is to take a pair of variables, say a and b and apply every possible operator. In this specific case, the operators can be to add a directed arc from a to b , from b to a or an undirected arc between these two variables; this part is the **search** part. Then, in the **scoring** part, the scoring function is applied and the best operator (the one which got the maximum or minimum score) is chosen, say, the one that added an undirected arc. The search and scoring algorithm continues taking pairs of variables (nodes), scoring once the operators are applied and choosing the best operator. The process stops when no application of any operator improves the score. Figures 3.9, 3.10, 3.11, 3.12, 3.13, 3.14 and 3.15 show the steps taken by this kind of algorithms.

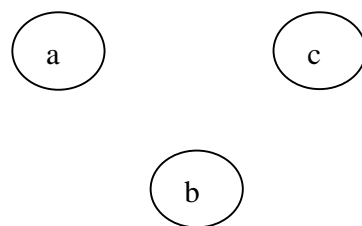


Figure 3.9: the empty network as the search space initialisation

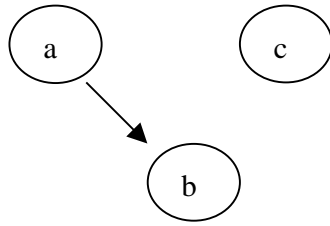


Figure 3.10: adding a directed arc from a to b (score 5.0)

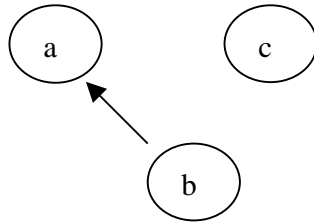


Figure 3.11: adding a directed arc from b to a (score 5.0)

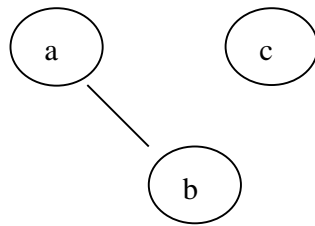


Figure 3.12: adding an undirected arc between a and b (score 7.0)

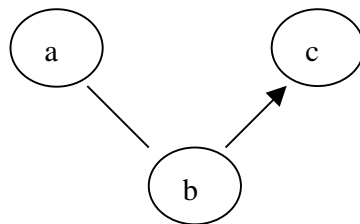


Figure 3.13: adding a directed arc from b to c (score 5.0)

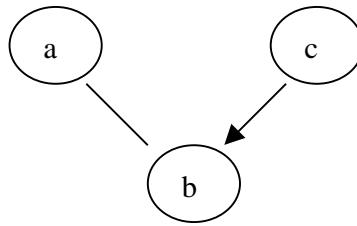


Figure 3.14: adding a directed arc from c to b (score 4.0)

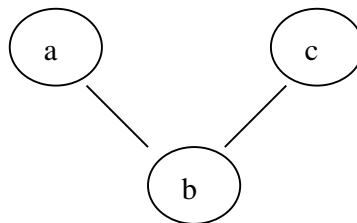


Figure 3.15: adding an undirected arc between b and c (score 8.0)

Figure 3.15 shows the last structure that the search and scoring algorithm chooses. Adding an arc between a and b (either directed or undirected) does not improve the score so, when arriving to the structure of this last figure, the algorithm stops. Moreover, these algorithms have the property of not applying an operator if the application of such an operator produces a cycle since, by definition, a Bayesian network is an acyclic directed graph. The search initialisation space could be a random graph containing a certain number of arcs, as mentioned before. This would result in the application of other operators that possibly could lead to a different result. This is likely to happen since the heuristics chosen can get stuck in a **local maximum** which is explained below (Heckerman 1997; Friedman and Goldszmidt 1998a; Heckerman 1998).

If the problem of search and score is seen as to follow certain branches of a tree, then getting in a local maximum means that one final branch is chosen (with the best local score) but there can be another ones having better score than the chosen branch. This problem can easily be understood using figure 3.16. Taking equation 3.1, the possible number of different Bayesian network structures is 25. For the lack of space, figure 3.16 does not depict every possible structure but only a portion of the all search space. Now, imagine that the algorithm reaches somehow the very left branches and returns, as the best local structure, $a \rightarrow b \rightarrow c$. Since an exhaustive enumeration of all the possible structures implies, in many cases, a computationally intractable problem (**NP-hard**) (Chickering 1996; Heckerman 1998) then the use of heuristic methods is justifiable.

The heuristic method in this kind of algorithms chooses only one possible way in the tree. Now, imagine that the structure resulting of following the very right branches, $a \rightarrow b \rightarrow c$, is better than the previous one. However, there is no way of reaching this better structure since the application of any operator cannot lead us to this structure. A possible solution is to perform random restarts, i.e., to initiate the algorithm with different structures; this process is repeated iteratively a certain manageable number of times preserving, at each stage, the resultant networks. When the iterations are done, then the best structure is chosen.

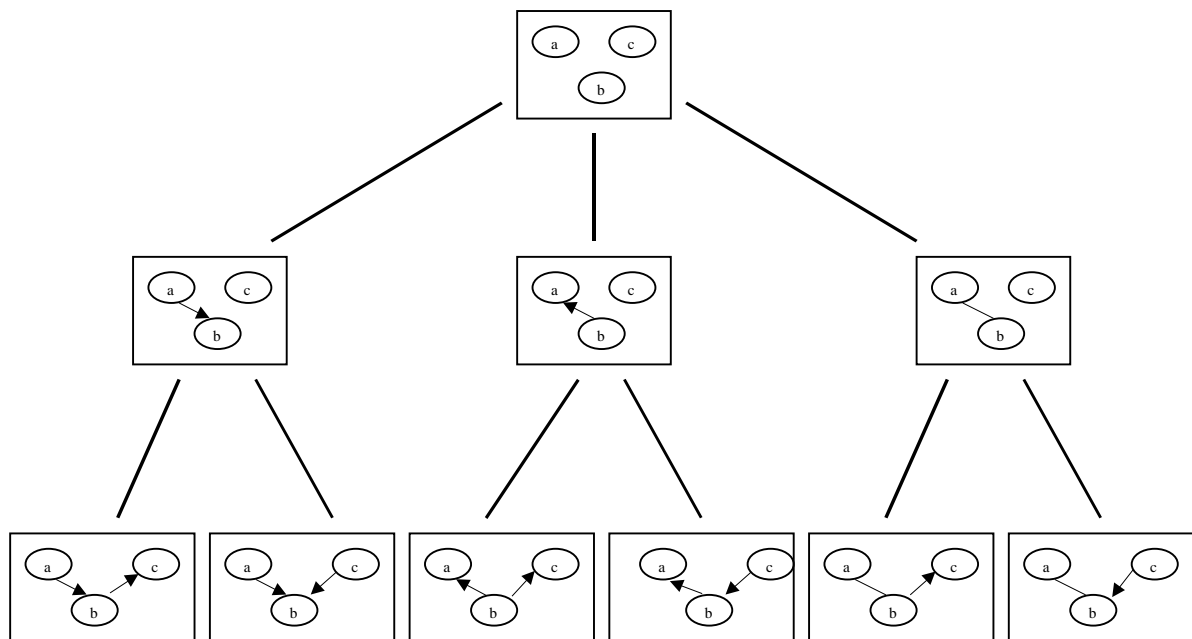


Figure 3.16: different possible structures resultant from applying different operators

Another problem that random restarts can solve is that called **plateaus**. This problem appears when some operators are applied leaving the score intact (Friedman and Goldszmidt 1998a). Heckerman et al. have shown (Heckerman, Geiger et al. 1994; Heckerman 1997) that equivalent network structures produce the same score. For instance, figures 3.4, 3.5 and 3.6 have the same score (using for instance MDL as a criterion). Therefore, at some step, the score could not be improved by the application of any operator. Hence, random restarts can help solve this problem in the same way that the local maximum problem is solved. However, it is necessary to notice that applying random restarts can lead to a solution that might take substantially more time than when this

strategy is not used. The compromise between time complexity and accuracy has to be decided according to the situation being considered.

In sum, search and scoring algorithms are also a very good option for learning the structure of a Bayesian network from data. The selection of any choice depends heavily on many factors such as the available data and the knowledge of the problem being considered.

It is very important to mention the respective advantages and disadvantages of both kinds of methodologies; this will be done in the next subsection. By the way, some of these advantages and disadvantages have motivated the proposal of a synergy of these two different methodologies: a **hybrid** approach. This approach must not be confounded with hybrid Bayesian networks which are networks containing both discrete and continuous variables (Cooper and Herskovits 1992).

3.4.3. Advantages and disadvantages of constraint-based algorithms and search and scoring based algorithms.

The **constraint**-based algorithms, as mentioned before, carry out conditional independence tests to induce, from data, a Bayesian network structure that is consistent with the observed dependencies / independencies contained implicitly in these data. One very important **advantage** of this methodology over the search and scoring methodology is that it performs a more efficient (faster) search when reconstructing **sparse** networks, i.e., networks that are not densely connected. Another **advantage** is that, when the size of the sample grows arbitrarily large, then the results from running the independence tests will be more accurate (Chickering 1996).

Since these algorithms make their decisions of including or not an arc between two variables based on the statistical tests they perform, one important **disadvantage** has to do with the volume of the data. So, the reliability of these tests is a function of the size of the database and the number of variables taking part in the conditioning set as mentioned before. In other words, these algorithms need infinite data in order to learn independence with certainty; independence tests of high order can be unreliable unless the size of the sample is really huge (Friedman and Goldszmidt 1998a). Furthermore, some of the algorithms pertaining to this class need an exponential number of independence tests; the result of this will lead to a computationally intractable problem when the number of variables is large.

Another **disadvantage** comes when one needs to choose the value of the threshold ϵ (the significance level) in order to decide whether the independence test is passed or not. There is not any standard recipe that tells us which threshold has to be chosen; such a threshold is normally determined by experience and knowledge about the problem being considered (Cheng 1998; Friedman and Goldszmidt 1998a). Hence, these algorithms are sensitive to the significance level chosen, therefore, the learned Bayesian network structures are susceptible to errors produced by this chosen significance level, namely, errors in the independence tests (Friedman and Goldszmidt 1998a). Moreover, if in a determined problem there are many independence tests to be performed, then the more statistical tests of this nature are carried out the more likely the independence decision classifies incorrectly. For instance, if a significance level of 0.05 is chosen, then 1 out of 20 significant independence tests will be classified erroneously. If a large number of these independence are to be performed, then many such tests will have the same error. Some adjustment measures can be used in order to correct this error, such as Bonferroni's correction factor (Howell 1997). In other words, when many statistical tests for determining independence are required, the global significance level is greater than the nominal one. Then, for each particular case, and especially when the size of the sample is small, it is suggested to review the significance level to be used (Cruz-Ramirez 1997). Finally, as Chickering points out (Chickering 1996), the constraint-based algorithms identifies only one model which has to do with the approach of model selection mentioned before.

Therefore, these algorithms are unable to produce more than one model, which has to do with the approach of selective model averaging. This means that when the data are insufficient then it is possible that there be more than one good model.

The **search and scoring** based algorithms define a metric (score) that evaluates how well the dependencies / independencies portrayed by a certain Bayesian network structure fit the data while a search engine looks for structures that maximizes (minimizes) this score. One main **advantage** of these algorithms is that they do not need the specification of the statistical threshold ϵ mentioned previously (Cruz-Ramirez 1997). Another **advantage** is that of the implicit introduction of Occam's razor philosophy; i.e., the scoring criteria introduce naturally, within the equations they use, the preference of selecting simpler models over the more complicated ones. Introducing this criterion of **parsimony** (Occam's razor) avoids the **overfitting** (which will be described in chapter 4) of data (Heckerman 1997; Friedman and Goldszmidt 1998a; Heckerman 1998).

One important **disadvantage** of these methods is that of computational intractability (Cooper 1999). In order to reduce this intractability (the search space), some algorithms such as K2 (Cooper and Herskovits 1992) assume that an **ancestral** node ordering exists. Such an ordering means that the variables form part of a list where a variable on the left can possibly be parent of a variable on its right but by no means vice versa. This ancestral node ordering implies, many times, that considerable work on knowledge elicitation has been carried out. Another important **disadvantage** of these algorithms is that, because of their heuristic nature, the best network structure cannot be found but only a good local one. Because of this, it is necessary to run the algorithm many times in order to avoid getting stuck in a **local maximum**, which implies of course a considerable amount of time. Moreover, if the initial model given to this kind of algorithms is the empty graph, much more time can be indeed needed in order for these algorithms to converge to a (possible local) optimal network.

Finally, it is important to remark that when the size of a database is large enough the constraint-based approach and the search and scoring approach are equivalent (Chickering 1996) and there is basically no preference in choosing one over the other.

3.5 Combining constraint-based methods and search and scoring based methods: a hybrid approach.

As can be clearly seen in the previous subsection, some problems arise within the two different approaches as well as some potential advantages. The idea of combining the advantages offered by both methods in order to ameliorate the problems encountered in them has proved a very good one (Spirtes and Meek 1995; Friedman and Goldszmidt 1998a; Cooper 1999). The main features of such a combination are explained below.

First of all, it is necessary to review a kind of search and scoring approach called the **Bayesian approach** (Cooper and Herskovits 1992; Heckerman, Geiger et al. 1994; Chickering 1996; Chickering, Heckerman et al. 1997; Heckerman 1997; Friedman and Goldszmidt 1998a; Heckerman 1998; Cooper 1999). The term Bayesian refers to the idea of combining human (commonly) expert knowledge with statistical data collected from the problem being considered and incorporating them into a single formula such as Bayes' rule (see equations 2.11 2.12 of chapter 2) where the term, which can be determined by the person's belief or knowledge, is usually called the **prior** probability or simply the prior (Pearl 1988). This prior probability is the unconditional probability appearing as part of the numerator of Bayes' formula. This probability can be determined either by the data available or by the human expert. Under the Bayesian interpretation, the probability of any event happening is seen as a degree of belief (Chickering 1996; Heckerman 1998). That is why the Bayesian probability is also known as the **subjective** probability.

The philosophy for constructing Bayesian network structures behind the Bayesian approach is very similar to that of the search and scoring approach, namely, there exists a measure to evaluate the goodness of fit of the structure and a search engine to look for structures that maximizes or minimizes such a measure as well. As mentioned in the

previous subsection, if the search and scoring methodology is applied alone, then the search space initialisation can be either an empty graph or a random graph with a particular number of edges (arcs). The Bayesian approach differs from this methodology in that the search space initialisation, i.e., the initial network structure that the search and scoring algorithm takes as input, is now proposed by the expert instead of being the empty graph or any random graph. This new proposed network structure is known as the **prior** network (Heckerman, Geiger et al. 1994; Chickering 1996; Heckerman 1998). In doing this, the results obtained by this algorithm can be much more accurate and converge in an optimal value much faster (Cooper and Herskovits 1992; Heckerman, Geiger et al. 1994; Chickering 1996; Chickering, Heckerman et al. 1997; Cooper 1999). Of course, the availability of the prior network implies that, again, considerable amount of time has been spent in the acquisition of such expert knowledge which is, as mentioned previously in chapter 1, not always a straightforward task.

The Bayesian approach differs also from the constraint-based approach. For instance, when presented with different network structures, the experts can give their preference for one specific structure over the others. Or, as mentioned above, they can propose a certain network structure, which of course means that they have special preference for that proposed structure as the possible correct one. A very good **advantage** of the Bayesian methodology is that if the experts do not find it easy to propose such a structure or to specify the information for doing that then priors called **noninformative** priors can be used instead. The reader is referred to (Heckerman 1997; Heckerman 1998) for a more detailed description on this. Moreover, as with the search and scoring algorithms, Bayesian methods do not need the specification of statistical testing thresholds but the specification of prior network structures or probabilities for ranking such networks. Some of the most representatives algorithms that fall into this category are: the **K2** algorithm (Cooper and Herskovits 1992), the **Kutato** algorithm, the **HGC** algorithm (Heckerman, Geiger et al. 1994; Cheng, Bell et al. 1998), the **Lam-Bacchus** algorithm (Buntine 1996; Cheng, Bell et al. 1998), the **Suzuki's** algorithm (Buntine 1996; Cheng, Bell et al. 1998) and some others (Jordan 1998; Cooper 1999; Pearl 2000).

Once again, as with the search and scoring algorithms, Bayesian algorithms have the problem of computational intractability. The possibility of representing prior knowledge about the likelihood of a given structure is both a strength and a weakness. The strength is found when this prior knowledge can be incorporated even when the knowledge comes from different sources. The weakness appears when the number of variables in the problem taken into consideration is large causing the determination of prior probabilities to be difficult and even infeasible. An important feature of a Bayesian system is that, as with the constraint-based approach (see section 3.3.1), the condition of **stability** (or faithfulness) holds as well (Heckerman 1998; Pearl 2000). This assumption does not need to be made explicit since, in general, all practical implementations of these approaches prefer simpler models (those with fewer parameters) hence head to **minimality** (Occam's razor) and establish no accidental independencies, hence head to **stability** (see section 3.4.1).

In sum, the prior information or background knowledge can be of the form of the structure of a Bayesian network (possibly incomplete), prior probabilities of the variables in a determined problem and a certain time ordering of such variables so that it is possible to use this temporal information to represent possible causal relationships among these variables. It is in this Bayesian approach that the name of Bayesian networks makes more sense: the input information has a subjective nature and Bayes' formula is used as the basis for updating the information (Pearl 2000).

After reviewing the Bayesian approach, it is now possible to stress the motivation for the proposal of a hybrid method, namely, a method capable of combining a constraint-based algorithm and a search and scoring algorithm, as figure 3.17 suggests.

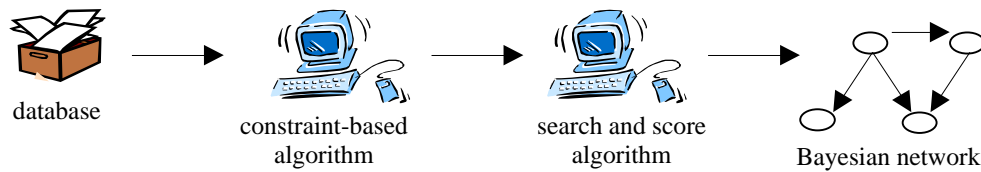


Figure 3.17: a combination of a constraint-based algorithm and a search and score algorithm to build a Bayesian network from data

One of the main disadvantages with the Bayesian approach is that of the elicitation process which represents a “bottleneck” problem in the construction of a Bayesian network. It is very important to remark that, as said before, the capability of integrating personal beliefs in the Bayesian framework can represent both an advantage and a disadvantage. In chapter 1, the time-consuming problem of the manual acquisition for constructing expert systems was pointed out. So, instead of giving the prior network as an input to the search and scoring algorithm, it might be a good idea to give this algorithm a graph determined or proposed by a constraint-based algorithm as a good candidate for the initial graph. This process can save much time and offer a good solution to the problem at hand. Intuitively, it does make sense to combine constraint based algorithms and search and scoring algorithms to produce much more robust results. One of the main motivations for such a combination is actually the difficulty in accessing and extracting expert knowledge in a certain domain. Moreover, this idea has been explored by some other researchers (Spirtes and Meek 1995; Cooper 1999; Scheines 1999b). The key point of a hybrid algorithm is to use both the power of a constraint-based algorithm, which provides a fast first good approximation of a Bayesian network structure, and the power of a search and scoring algorithm, which refines the given structure using its scoring metric and the application of its operators to improve such structure. Chickering et al. (Spirtes and Meek 1995) have shown that the greedy nature of a search and scoring algorithm performs well and successfully when the initial proposed Bayesian network (which can be given by the constraint based algorithm) is close to the gold-standard Bayesian network (this notion will be presented in chapter 4). They have also suggested that this performance will not be correct when the initial Bayesian network is the

empty graph. This is confirmed by simulation studies carried out by Spirtes and Meek (1995).

A hybrid method can also reduce the search space of a Bayesian algorithm alone that is generally intractable, as mentioned above. Cooper's insight (Cooper 1999) is that the use of these hybrid methods seems to have potential for future research. Friedman and Goldszmidt (1998a) also stress that the combination of constraint based methods and score based methods is consistent and can learn, with sufficient amounts of data, the **correct** structure (Chickering 1996) of a Bayesian network, in the sense of faithfully representing the joint probability distribution from the local conditional probability distributions, as eq. 2.32 of chapter 2 shows (Chickering 1996). Hybrid methods can also avoid the overfitting of data.

Before closing this chapter, it is necessary to comment on a point from chapter 1: the Power PC Theory. It can be argued that this powerful psychological theory could be used as an insight to favour the possibility of building computational models capable of combining background knowledge and statistical data in the form of Bayesian networks. In order to further refine the results, a hybrid approach can be used altogether with a Bayesian approach, namely, the output of the hybrid algorithm can be presented to the human experts and then they can improve more on the proposed model. Although it can take much longer, this level of refinement might be an appropriate, sensible and sound combination.

As figure 3.17 suggests, the first step for the creation of a hybrid approach is the proper, reliable and robust construction of a constraint-based algorithm capable of proposing a Bayesian network structure close to the real underlying probability distribution. Thus, this dissertation focuses on that first and important step for building Bayesian networks from data.

Chapter 4

Bayes2: a constraint-based algorithm for constructing Bayesian networks from data

This chapter introduces the measures used in the algorithm presented here for carrying out independence tests among the variables of a given problem. Then, it presents the first version of this algorithm that builds Bayesian networks from data and the experimental results obtained by running such an algorithm on five different databases. Finally, it provides an explanation of the quantitative criterion that is used to measure the goodness of fit of the networks proposed by the algorithm presented in this chapter as well as those presented in chapters 5 and 6.

4.1 Information measures used as independence tests.

As mentioned in section 3.4.1 of chapter 3, in order to represent probabilistic dependencies / independencies among the variables of a certain problem in the form of a Bayesian network, a measure to test the independence between any two variables given a (possibly empty) set of other variables is needed. The algorithms developed in this thesis use a slight variant of the marginal and conditional independence measures defined by information theory, known as the **mutual information** and the **conditional mutual information** (Shannon and Weaver 1949; Pearl 1988; MacKay 1995; Martinez-Morales 1995; Schneider 1995; Cruz-Ramirez 1997; Cheng 1998; Cheng, Bell et al. 1998), proposed originally by Kullback (1959). Before presenting the formulas describing such measures, it is useful to give a very brief explanation about the concept of **information** or **entropy** and the motivation for using this notion to construct Bayesian networks from data. The reader is referred to (Shannon and Weaver 1949) for a detailed account on information theory.

It is important to recall that the variables in Bayesian networks are, by definition, **random** variables. Thus, at any given time, each variable will have a specific value that can be chosen from a set of different possible values. Information theory provides a very nice way to measure the information content of a random variable. But, why would we want to measure the information content of such a variable?

Let us take an example from Schneider (1995). Imagine two different devices; one that produces four different symbols A, B, C and D and the other that produces two different symbols, 1 and 2. Suppose that the probability for each device to produce any symbol is the same; so, for the first device the probability for each symbol to appear is $1/4$ and for the second device the probability for each symbol to appear is $1/2$. When a certain symbol is produced by any of the two above devices, then it is possible to assert that some information has been received and therefore the uncertainty associated to that device decreases. If the logarithm (of any base) is used to measure the uncertainty associated to each device, then for the first device its uncertainty will be $\log(4)$ while for the second its uncertainty associated will be $\log(2)$. In fact, the base of the logarithm determines the units: for logarithms of base 2 the units are bits, for logarithms of base 10 the units are digits, for the natural logarithms (base e) the units are nits. If the base 2 is taken, then for the first device the uncertainty is $\log_2(4) = 2$ bits and for the second device the uncertainty is $\log_2(2) = 1$ bit.

In other words, for the first device, 2 bits (at most) are sufficient to represent its four possible outcomes and for the second device, 1 bit (at most) is enough to represent its two possible outcomes. There exists the possibility that, for instance, two of the symbols in the first device never appears; this means that the uncertainty now will be reduced to $\log_2(2) = 1$ bit. Of course, it is possible also to have situations where some of the symbols appear more frequently than others. In this case, it is necessary to find a formula capable of representing that some symbols are more likely to appear than others. Shannon (Shannon and Weaver 1949; MacKay 1995; Schneider 1995) arrived to that formula which is shown in equation 4.1.

$$H(X) = - \sum_x P(x) \log P(x) \quad (4.1)$$

This formula represents the **entropy** or uncertainty associated with variable X. If the base of the logarithm is 2, then this formula tells how many bits are needed to represent, in average, any possible value that X can take. If, as mentioned before, all the symbols are equally likely to appear, then the formula is reduced to simply $\log M$ (where M is the number of symbols). See (Schneider 1995) for an easy derivation of this result.

According to equation 4.1, the maximum entropy is reached when the probability distribution is the uniform distribution: $p(x_1) = p(x_2) = \dots = p(x_n) = 1/n$. So from this result it is possible to deduce that the output of an unbiased coin is harder to guess than that of a biased coin because the entropy associated with the former is bigger than that of the latter. To see much clearer this picture, imagine the uncertainty when there are 10 symbols and only one of them appears; if equation 4.1 is applied (using l'Hopital's rule for this special case; i.e., if $\lim_{p \rightarrow 0} p \log p = 0$, $p = 1/M$, then $p \log p = 0$) then the result is 0. This means that there is no uncertainty about what symbol will come out next. When a reduction in uncertainty occurs, then it is possible to consider that a **gain in information** also occurs. These two entities are inversely related to each other: while uncertainty decreases information increases and vice versa.

Information theory has to do with how to communicate reliably from a point A to a point B through a noisy channel. Let us illustrate this idea by means of an example taken from (Schneider 1995). Imagine a teletype receiving a message (a string of characters) through a phone line. If this channel (the phone line) does not have any noise at all, then it is logical to deduce that the text will be printed out perfectly. However, in practical matters, such noiseless channels do not exist; therefore the text will have an implicitly contained amount of uncertainty, i.e., there will be an uncertainty regarding whether the string of characters printed is the right one or whether there are some symbols in the printed string that do not correspond to those sent. In other words, this uncertainty measures the amount

of noise reflecting the probability of error that a certain symbol of the string that has been received is not equal to the symbol that was actually sent. In the present example, in contrast to the example of the devices responsible of producing three and two different symbols, this problem involves the communication of the outcome of such devices through a noisy channel. Another formula that reflects this situation is needed because now two variables, not only one, have to be taken in consideration: variable X, which is the actual output of any of the two devices mentioned above and variable Y, which is the output of the teletype (which of course can be different from that of X due to the noisy channel).

The new formula (**conditional entropy**) has to incorporate these two different variables and reflect the remaining amount of uncertainty of variable X given that the output of variable Y is known. Shannon (Shannon and Weaver 1949) also arrived to such a formula which is presented in equation 4.2.

$$H(X | Y) = - \sum_{x,y} P(x,y) \log P(x | y) \quad (4.2)$$

Using equations 4.1 and 4.2 it is possible now to define the gain of information or **mutual information** of X given Y; i.e., a measure to determine the average reduction in uncertainty about X that results from learning the value of Y or vice versa. This formula is written in equation 4.3.

$$H(X;Y) = H(X) - H(X | Y) \quad (4.3)$$

Equation 4.3 can also be rewritten in another different form, which is represented by equation 4.4.

$$H(X;Y) = I(X,Y) = \sum_{x,y} P(x,y) \log \frac{P(x,y)}{P(x)P(y)} \quad (4.4)$$

Finally, imagine that not only are there two variables X and Y but also a set of variables represented by \mathbf{Z} . Of course this set can be empty; if this is the case, then equation 4.4 applies. If this is not the case, then it is necessary to find a formula that is able to represent the conditional information gain or **conditional mutual information** provided by variable Y to explain variable X conditional on set \mathbf{Z} . Such a formula is shown in equation 4.5 (Cheng 1998; Cheng, Bell et al. 1998).

$$H(X;Y | Z) = I(X,Y | \mathbf{Z}) = \sum_{x,y,\mathbf{z}} P(x,y,\mathbf{z}) \log \frac{P(x,y | \mathbf{z})}{P(x | \mathbf{z})P(y | \mathbf{z})} \quad (4.5)$$

Taking the results shown in equations 4.1 to 4.5 and returning to the Bayesian network framework, if two nodes in a Bayesian network (which represents random variables) are dependent, then knowing the value of one of those nodes will provide some information regarding the value of the other node. This gain of information provided by one of the nodes (or variables) can be measured using **mutual information** applying either equation 4.3 or 4.4. If these two nodes are dependent but this time conditional on a set \mathbf{Z} , then the respective information gain can be measured using **conditional mutual information** applying equation 4.5.

The previous relationships (equations 4.1 to 4.5) suppose that all the probability distributions involved are known. However, in real-life problems this is not usually the case, thus these distributions have to be estimated from a dataset (sample). Hence, if the probability distributions are calculated from a sample, then the previous formulas will be expressed in terms of the estimators $\hat{H}(\bullet)$ and $\hat{I}(\bullet)$.

In sum, it is possible to use these information measures to establish, from a data sample, whether two nodes in a Bayesian network are dependent or independent. The information measures, used in this thesis to perform these tests of independence, were proposed by Kullback (1959). Equation 4.6 presents such a case.

$$T = 2N \hat{I} \quad (4.6)$$

where N is the size of the sample (the number of cases in the database) and \hat{I} is either the mutual information (eq. 4.4) or the conditional mutual information (eq. 4.5), according to which is the case.

Kullback has shown (Kullback 1959) that, under the independence assumption and under the hypothesis that the data come from a multinomial distribution, this statistic T is asymptotically distributed as a χ^2 (chi-square) variable with $(X-1)(Y-1)$ degrees of freedom for the case of the mutual information and $(X-1)(Y-1)Z$ degrees of freedom for the case of conditional mutual information (where X is the number of possible values taken by X , Y is the number of possible values taken by Y and Z is the number of possible values taken by the variables included in Z determined by the principle of multiplication (Hines and Montgomery 1997)). From this result, it is possible then to perform an independence statistical test to check whether two variables in a Bayesian network are marginally or conditionally dependent or independent. This assumption of independence means that when equation 4.4 is used to calculate the value of \hat{I} in equation 4.6 and the result T is smaller than a certain threshold ϵ , then it can be said that X and Y are marginally independent. If it is the case that equation 4.5 needs to be applied to compute the value of \hat{I} in equation 4.6 and the result T is smaller than a certain threshold ϵ (recall section 3.4.1 of chapter 3), then it can be said that X and Y are conditionally independent given Z . Otherwise, X and Y are dependent (either marginally or conditionally). To calculate the degrees of freedom, the following calculation taken from (Spirtes, Glymour et al. 1993) has to be carried out.

Let $Cat(X)$ be the number of the possible values taken by X . Let $Cat(Y)$ be the number of the possible values taken by Y . And let n be the number of variables in \mathbf{Z} . The number of degrees of freedom (df) in the test is calculates as follows:

$$df = (Cat(X) - 1) \times (Cat(Y) - 1) \times \prod_{i=1}^n Cat(Z_i) \quad (4.7)$$

So, if H_0 is considered as the null hypothesis that two variables are independent and H_1 as the alternative hypothesis that two variables are **not** independent, then the decision rules of the statistical test can be written as follows:

For the case of mutual information:

$$(i) \quad \text{Reject } H_0 \text{ if } T \geq \chi^2_{(X-1)(Y-1)}(\alpha) \quad (4.8)$$

$$(ii) \quad \text{Do not reject } H_0 \text{ if } T < \chi^2_{(X-1)(Y-1)}(\alpha) \quad (4.9)$$

For the case of conditional mutual information:

$$(i) \quad \text{Reject } H_0 \text{ if } T \geq \chi^2_{(X-1)(Y-1)Z}(\alpha) \quad (4.10)$$

$$(ii) \quad \text{Do not reject } H_0 \text{ if } T < \chi^2_{(X-1)(Y-1)Z}(\alpha) \quad (4.11)$$

where α is the significance level or threshold of the statistical test against which T is compared.

Taking in account the information measures, the T statistic, the two different decision rules and the fact that a database is provided, it is possible now to design an algorithm for constructing Bayesian networks from data. In the next section, the first algorithm that builds Bayesian networks from data is presented. But before doing so, some

important assumptions are introduced to describe under which situations this algorithm works.

4.2 Bayes2: a first algorithm to build Bayesian Networks from data.

The following assumptions are taken mainly from (Cooper and Herskovits 1992); they also are described in (Cheng 1998) and (Chickering 1996). These assumptions detail the situations that the algorithm presented in this chapter needs to work properly. Since such an algorithm builds a Bayesian network from data, these assumptions have to do with the database that is taken as input.

Assumption 1. The variables in the database (sample) are **discrete**.

Assumption 2. The cases occur **independently** given the underlying probabilistic model of the data.

Assumption 3. There are **no** cases with **missing** values.

Assumption 4. The **volume** of the data is large enough for the reliable independence tests used in the algorithm proposed here.

It is clear that, for the case of the first assumption, all the variables that take part in a certain problem need to be discrete. If continuous variables are part of this problem being considered, it is possible to discretize them in order for the proposed algorithm to work properly. However, such a procedure is not part of a pre-processing module of this algorithm and, if the variables are to be discretized, then a pre-processing module is needed.

In the case of the second assumption, a well-known example of this can be presented so that this assumption is better understood. Suppose there is an unbiased coin which is tossed two times. The fact that it is a fair coin results in the probability of the coin landing heads equal to 0.5 and the probability of landing tails equal to 0.5. Assumption 2 tells us that, because cases occur independently, the fact that the first time the coin was tossed landed heads does not influence that the result of second toss will be heads. In other words, knowing the values in one case gives us no information about values in any other case.

Assumption 3 limits databases to have all the records on them observed. It is important to mention that there exist algorithms capable of handling missing values. However, all the algorithms proposed in this thesis can only cope with complete databases.

And finally, assumption 4 guarantees that, because of the length of the sample size, the statistical independence tests carried out by the proposed algorithm are reliable and hence correct.

4.2.1 Description of the Bayes2 algorithm.

The algorithm presented in this chapter, called **Bayes2**, was originally proposed by Martinez-Morales (1995) and later extended by Cruz-Ramirez (Cruz-Ramirez 1997; Cruz-Ramirez and Martinez-Morales 1997). The basic idea behind Bayes2 is that the information measure in equation 4.4 can form a descending succession of variables; i.e., the variables can be ordered according to the information gain they provide to a dependent variable from maximum to minimum (ancestral ordering). Theorem 2.1 of chapter 2 establishes that, if such an ordering exists, then it is possible to induce a directed acyclic graph (GAD). By definition, a Bayesian network is a GAD so, once this ancestral ordering is induced, it is possible to build a Bayesian network from that ordering. The way Bayes2 works permits the use of theorem 2.2 of chapter 2 to induce such an ancestral ordering.

One of the strongest assumptions of Bayes2 is to suppose that if only the variable which provides the biggest amount of information (according to eq. 4.4) is considered, then this variable will be the only one forming part of the conditioning set \mathbf{Z} and, because the rest of the variables give little information, they can be discarded as forming part of that conditioning set. In the section of the experimental results of this chapter and in chapter 6, it will become evident why this strong assumption is, in general, not enough to produce accurate and sound results. However, it can be claimed that Bayes2 is efficient in the sense that it does not test all the possible associations (independence tests) among variables but only those which could be significant based on the information gain they provide. In doing this, Bayes2 avoids an exponential complexity on the number of such independence tests (Cheng 1998).

In order for Bayes2 to construct a Bayesian network from data, it first performs the necessary independence tests (marginal or conditional); then, based on those results, it checks whether the null hypothesis (eq. 4.8 or 4.10) holds or not. If the independence hypothesis (the null hypothesis) does not hold, then Bayes2 draws an arc from the independent variables (Y_1, Y_2, \dots, Y_n) to the dependent one (X). In other words, Bayes2 first assumes that all the variables are disconnected and then starts drawing arcs among them when this is the case. This class of algorithms is known as **stepwise forward** algorithms. Other examples of such algorithms are K2 (Cooper and Herskovits 1992) and Power Constructor (Cheng 1998; Cheng, Bell et al. 1998). The algorithms that first assume a complete graph, i.e., that all the variables are connected to each other, and then starts removing arcs as the correspondent independence tests hold are known as **stepwise backward** algorithms. Examples of this kind of algorithms are PC (Spirtes, Glymour et al. 1993) and MIM (Edwards 1995).

Bayes2 can be applied to a class of problems where the variables taking part in those problems can be divided naturally into dependent variables and independent variables. For the sake of simplicity, Bayes2 assumes that there exists only one dependent variable. In the next section, the pseudocode of Bayes2 procedure is described.

Procedure Bayes2

Let X be the dependent random variable and let Y_1, Y_2, \dots, Y_n be the **unordered** independent random variables.

1. Compute the value of the mutual information (equation 4.4) that each variable Y_i ($1 \leq i \leq n$) provides to X and order them according the information they provide from the largest to the smallest value. The **ordered** variables will be labelled now $Y_{(1)}, Y_{(2)}, \dots, Y_{(n)}$. Draw an arc from $Y_{(1)}$ to X . Then use formula 4.8 to check whether the null hypothesis H_0 (two variables are independent from each other) holds or not. If H_0 does not hold then draw a **directed** arc from $Y_{(i)}$ ($2 \leq i \leq n$) to X .
2. Compute the value of the conditional mutual information (eq. 4.5) that each variable $Y_{(i)}$ ($2 \leq i \leq n$) provides to X given $Y_{(1)}$. Then use formula 4.10 to check whether the null hypothesis H_0 (two variables are independent from each other given a set of variables \mathbf{Z}) holds or not. If H_0 holds then remove the arc from $Y_{(i)}$ to X . If H_0 does not hold, leave the arc intact.
3. Do $X = Y_{(1)}$ and delete $Y_{(1)}$ from the set of independent random variables. Let the number of independent variables $n = n-1$. If $n \geq 2$ then go to step 1; otherwise STOP.

End of Procedure Bayes2

As seen in chapter 1, the process to extract human experts' knowledge involves a very complex and time-consuming task; Bayes2 can help reduce such complexity by building a Bayesian network from data. If this Bayesian network is built accurately, then the main probabilistic relationships among the variables that take part in a specific problem are represented in a suitable and easily recognizable way, as shown in chapters 1, 2 and 3. It is also very important to remark that some search and scoring based algorithms reviewed in chapter 3 (such as K2 for instance) usually need an ancestral node ordering as a part of their inputs; this implies that such an ordering has to be given by someone (generally the human expert) assuming much work (manual elicitation mainly) previously done. Also, if this ordering is not chosen properly, the results produced by this kind of algorithms are inaccurate and imprecise (Cooper and Herskovits 1992; Chickering 1996; Cheng 1998).

In contrast, Bayes2 does not need a complete ordering of the variables; instead, all it needs is the specification of one dependent variable which must have no children at all, i.e., a terminal node or a leaf (see definition 2.14 of chapter 2). This task is far less complex than the total ordering specification, hence it needs less time for knowledge elicitation. Once this variable is determined, Bayes2 procedure finds an ancestral ordering for the independent variables, as explained above.

One of the main contributions of Bayes2 is the way it generates this ancestral ordering among the variables using the mutual information measure (eq. 4.4). However, the quality of the learned structure of the Bayesian network is highly dependent on this ordering and an improper ordering chosen by Bayes2 will lead to poor results. The basic idea to generate the ancestral ordering is very intuitive and, at first sight, it appeared to be sound and consistent: the induction of a decreasing succession of the variables according to the information they provide in such a way that if only the variable that gives the greatest information gain is considered, the rest of the variables give little information once this initial variable has been taken in account. Furthermore, the use of mutual information lets one induce a complete ordering on the independent variables which assures that the resultant network is a directed acyclic graph, as required by theorem 2.1 of chapter 2.

In the next subsections the experimental results running Bayes2 using real and simulated databases will be presented. From these results the advantages, as well as the disadvantages or limitations of this algorithm, will also be presented using as an evaluation methodology a special goodness of fit measure called **minimum description length** (MDL). Finally, some extensions to improve the performance of Bayes2 are proposed.

4.3 Experimental results.

The methodology to test experimentally the performance of the procedures presented here has been mainly taken from (Chickering 1996). It consists of the following steps:

1. Select a specific Bayesian network structure, which will be called the **gold-standard** network, to represent a joint probability distribution over the set of variables taking part in the problem.
2. Using a Monte-Carlo technique, simulate a set of cases by generating a database from this joint probability distribution and the gold-standard network.
3. Give this generated database as input to the algorithm (in this case Bayes2). The resultant network, product of running the algorithm, will be called the **learned** network.
4. Obtain the accuracy of the result by comparing the difference between the gold-standard network and the learned network using a goodness of fit measure.

Figure 4.1 also describes graphically this four-step process.

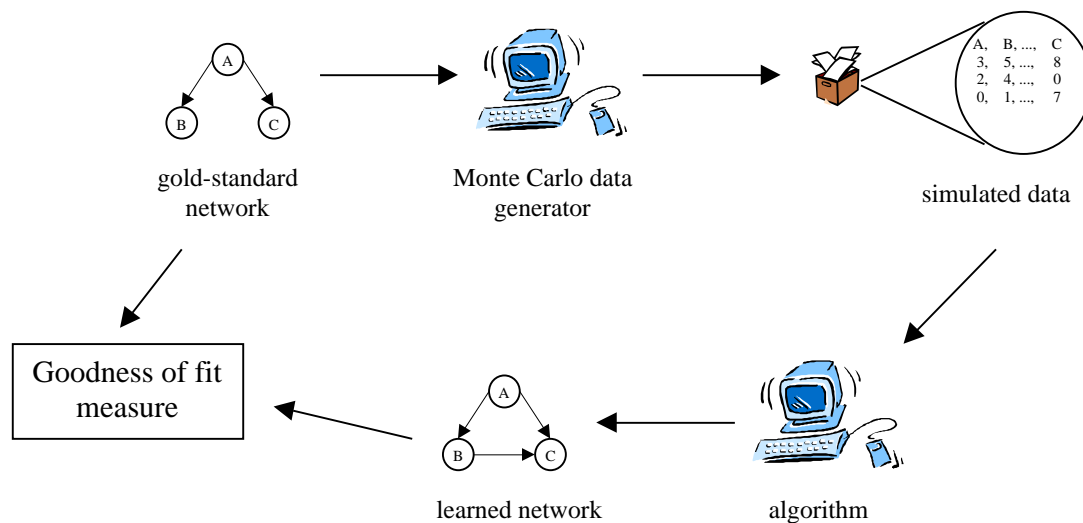


Figure 4.1: The four-step methodology used to test the performance of an algorithm that builds Bayesian networks from data

As Chickering (1996), Cooper (Cooper and Herskovits 1992; Cooper 1999) and Cheng (Cheng 1998; Cheng, Bell et al. 1998) point out, an advantageous point of using this methodology is that there exists a clear correct answer against which the learned network can be compared: the gold-standard network. The use of simulated datasets is a common practice to test and evaluate the performance and accuracy of the algorithms of this and some other types (Buntine 1996; Chickering 1996; Cheng 1998; Cheng, Bell et al. 1998). The Monte Carlo simulation technique used to generate the data to test Bayes2 is a technique developed by Henrion for Bayesian networks and is known as the **probabilistic logic sampling** method (Cooper and Herskovits 1992; Cooper 1999). This simulation technique is an unbiased generator of cases; i.e., the probability that a specific case is generated is equal to the probability of the case according to the Bayesian network and it is also embedded in a well-known software called **Tetrad II** (Spirtes, Glymour et al. 1993; Spirtes, Scheines et al. 1994), which has been developed to construct, among other things, Bayesian networks from data. Thus, Tetrad II was used to generate databases from the description of a Bayesian network structure and a probability distribution in order to test the performance of Bayes2.

Chickering (1996) mentions that there could be an argument against using simulated data generated this way: the results could not be robust because some of the four assumptions presented above may not hold when the data come from a real-world generating process. It is possible to use more sophisticated simulation techniques that can account for this argument and then to test the robustness of the algorithm. In this dissertation, these more complicated techniques are not explored. However, there is a better approach than that of only generating synthetic databases from a joint probability distribution and a Bayesian network structure: a Bayesian network (structure and parameters) being defined by an expert or group of experts based on their personal knowledge and knowledge found in the literature about a specific problem (Cooper 1999). This is the case of the **ALARM** network (Cooper and Herskovits 1992; Cooper 1999) and the **CHILD** network (Spiegelhalter, Dawid et al. 1993), both of which were used to test Bayes2 and are presented below.

Networks constructed using this approach have the favourable feature that their structures are known precisely and their probabilities have a high likelihood to resemble those probabilities generated by the real-world processes being modelled. On the other hand, an unfavourable point when manually constructing such networks is that it involves, as mentioned previously in chapters 1, 2, 3 and in this chapter, a very time-consuming and considerably complex task. Another disadvantage comes out because, in order to construct Bayesian networks this way, it is possible that even human experts have an incomplete and / or incorrect knowledge about the problem under consideration.

Five different databases were used to test the performance of Bayes2. Four of them, ALARM, CHILD, DIAGCAR and ASIA were generated using the Tetrad software. The remaining one, SEWELL & SHAH, is a real-world database which was collected by these researchers (Heckerman 1998) and hence was not generated using the Tetrad software. The description, references of each database and the results running Bayes2 with these datasets are presented below. In section 4.3.1, after presenting all the results given by Bayes2, a discussion on these obtained results is also presented.

The first Bayesian network given to the Tetrad program to generate a database and test the performance of Bayes2 was the ALARM network. ALARM stands for “A Logical Alarm Reduction Mechanism”. This network was constructed by Beinlich (Cooper and Herskovits 1992; Cooper 1999) as an initial research prototype to model potential anaesthesia problem in the operating room. ALARM has 37 variables (nodes) and 46 arcs. From these 37 variables, 8 variables represent diagnostic problems, 16 variables represent findings and 13 variables represent intermediate variables that connect diagnostic problems to findings. Each node (variable) has from 2 to 4 different possible values. It took Beinlich about 10 hours to build the ALARM Bayesian network structure and about 20 hours to fill in all the marginal and conditional probability distributions. Beinlich constructed ALARM based on both his own experience as an anaesthetist and knowledge from the literature.

It is very important to mention that the ALARM network is the most well-known and widely used benchmark to assess the performance of algorithms that recover the

structure of a Bayesian network from data. According to Cheng (Cheng 1998; Cheng, Bell et al. 1998), when generating cases from the ALARM network, three different probability distributions are used. The one used here is described at the Norsys Software Corporation web site (Norsys 2001). The size of the sample of the ALARM database is 10,000 cases. The ALARM network (gold-standard) and the result obtained from running Bayes2 on ALARM are presented in figure 4.2 and 4.7 respectively.

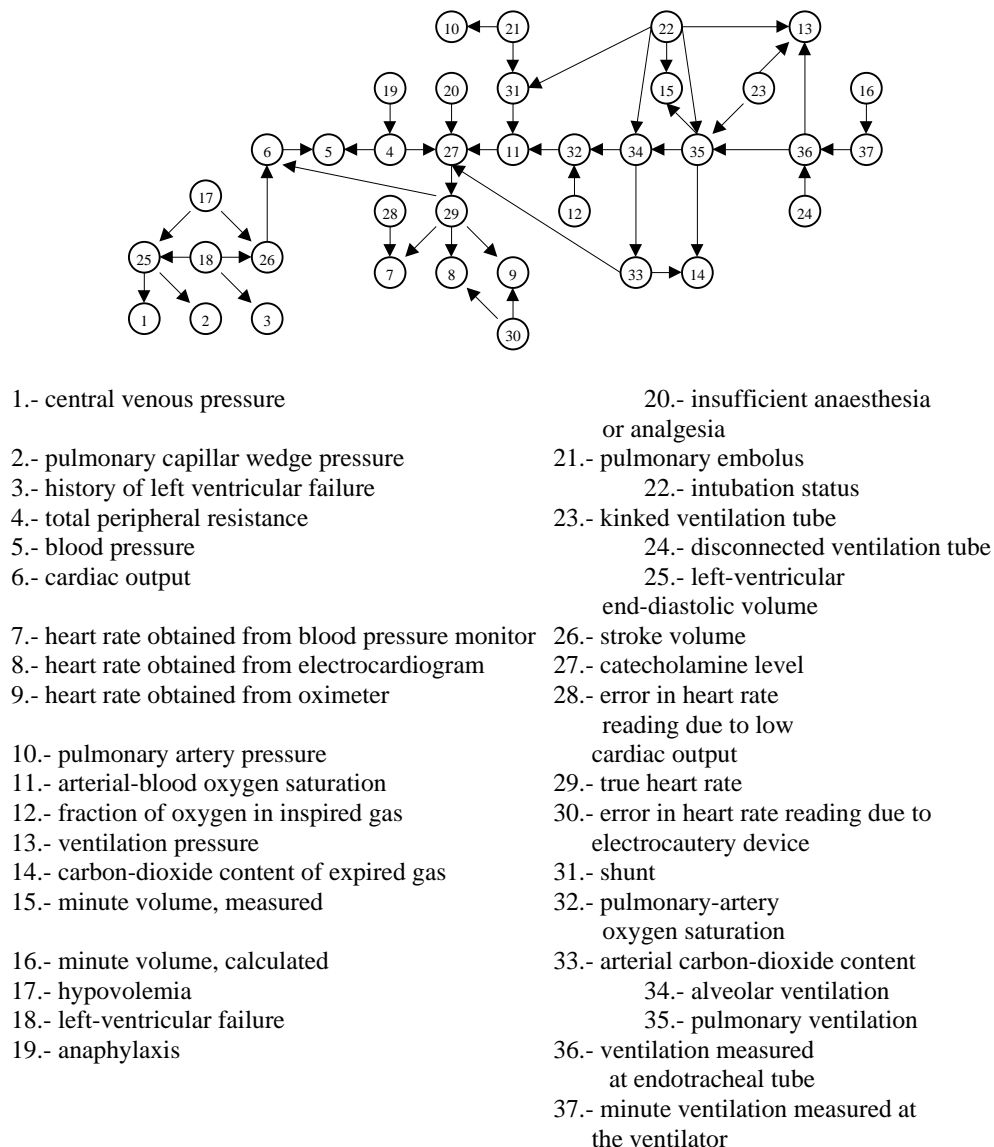


Figure 4.2: The ALARM network

The second Bayesian network given to the Tetrad program was the CHILD network (Spiegelhalter, Dawid et al. 1993). CHILD has 20 variables and 25 arcs. This network represents a real, moderately complex medical example, which has to do with the diagnosis of congenital heart diseases in children. Each node (variable) has from 2 to 6 different possible values. As Spiegelhalter et al. point out (1993), the aim of this network is to provide a mechanism capable of properly combining clinical expertise and data so that a reasonably transparent diagnostic tool can be produced. In the original paper by Spiegelhalter et al. (1993), the conditional probability distributions are not complete; i.e., they are not defined for all the nodes in the network. Hence, the Tetrad software was allowed to randomly populate these missing probabilities. The nonmissing probability distributions were also taken, as in the case for the ALARM network, from the Norsys web site (Norsys 2001). The size of the sample of the CHILD database is 5,000 cases. The CHILD network (gold-standard) and the result obtained from running Bayes2 on CHILD are presented in figure 4.3 and 4.7 respectively.

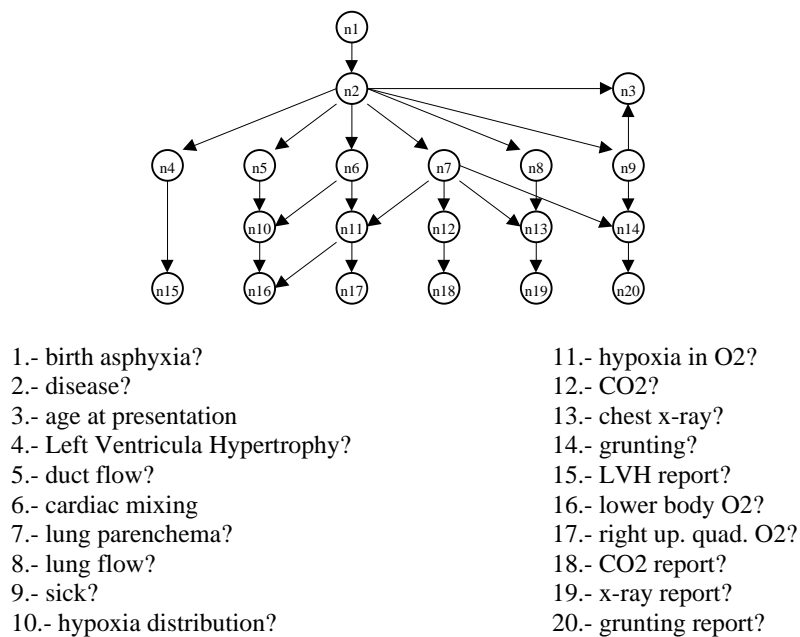


Figure 4.3: The CHILD network

The third Bayesian network given to the Tetrad program was the DIAGCAR network (Norsys 2001). DIAGCAR has 18 variables and 20 arcs. This network is an example of a Bayesian network for diagnosing why a car will not start, based on spark plugs, headlights, main fuse, etc. Each node (variable) has from 2 to 3 different possible values. The probability distributions for each node are described at the Norsys Software Corporation web site (Norsys 2001). The size of the sample of the DIAGCAR database is 5,000 cases. The DIAGCAR network and the result obtained from running Bayes2 on DIAGCAR are presented in figure 4.4 and 4.7 respectively.

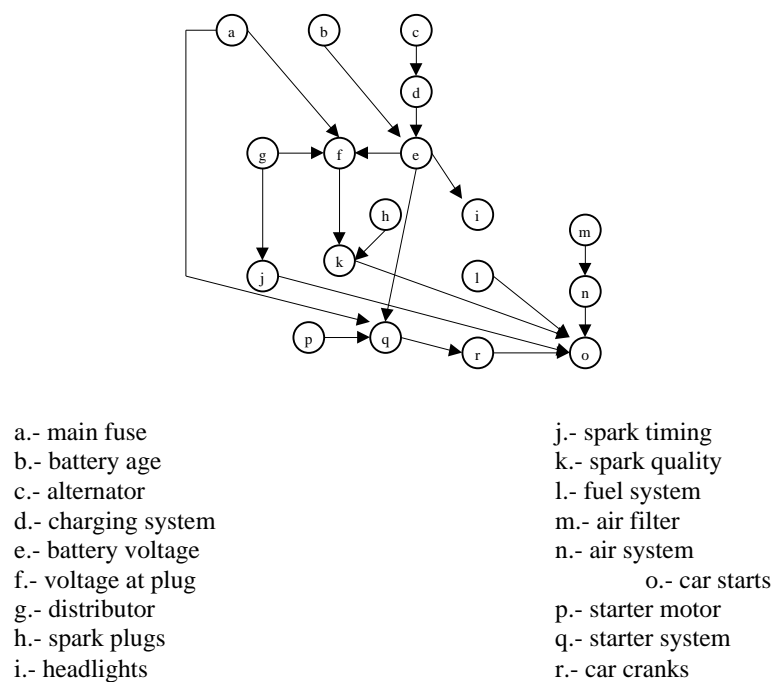


Figure 4.4: The DIAGCAR network

The fourth Bayesian network given to the Tetrad program was the ASIA network (Norsys 2001). ASIA has 8 variables and 8 arcs. This network is a very small Bayesian network for a fictitious medical example about whether a patient has tuberculosis, lung cancer or bronchitis, related to their X-ray, dyspnoea, visit-to-Asia and smoking status; it is also called "Chest Clinic". Each node (variable) has 2 different possible values. The

probability distributions for each node are described at the Norsys Software Corporation web site (Norsys 2001). The size of the sample of the ASIA database is 1,000 cases. The ASIA network and the result obtained from running Bayes2 on ASIA are presented in figure 4.5 and 4.7 respectively.

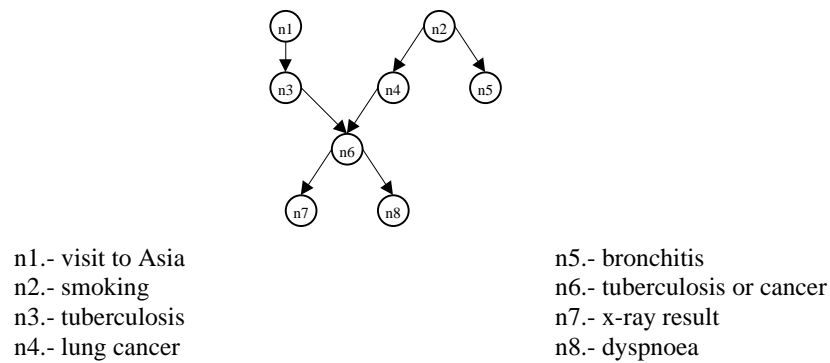


Figure 4.5: The ASIA network

The last data set is the SEWELL & SHAH database (Heckerman 1998). As said above, this is a real-world database collected by Sewell and Shah so there was no need to use simulation to generate it. This database has 5 variables and 7 arcs. Sewell and Shah investigated potential factors responsible of possibly influencing the intention of high school students to attend college. They measured variables such as sex, socio-economic status, intelligent quotient, parental encouragement and college plans for 10,318 Wisconsin high school seniors. Each node (variable) has from 2 to 4 different possible values. The network depicted in figure 4.6 is the most likely structure according to the Bayesian algorithm proposed by Heckerman (1998). The result obtained from running Bayes2 on these same data is presented in the table of figure 4.7.

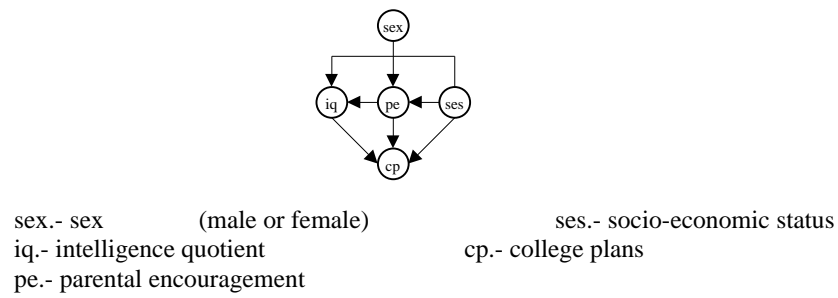


Figure 4.6: The most likely structure proposed by the algorithm of Heckerman

Databases	dep. variable	cases no. var	Bayes2		Total of arcs
			m.a.	e.a.	
Alarm	node 5	10000;37	0	192	46
Child	node 19	5000;20	0	68	25
Diagcar	node o	5000;18	3	22	20
Asia	node 7	1000;8	1	1	8
Sewell and Shah	node cp	10318;5	0	0	7

Figure 4.7: Results of Bayes2 from running the Alarm, Child, Diagcar, Asia and Sewell and Shah databases. Dep. variable refers to the dependent variable chosen for the analysis. Cases refer to the sample size while no. var refers to the number of variables involved in each database; m.a. and e.a. stand for missing arcs and extra arcs respectively. Finally, total of arcs corresponds to the total number of arcs in the gold-standard network

4.3.1 Discussion of the results.

Some important points can be extracted from all the results shown above. These points can be obtained by simple visual inspection comparing the gold-standard networks against the learned networks. In section 4.4, these results obtained by Bayes2 will be compared using a goodness of fit measure; that is to say, a quantitative criterion rather than a qualitative one will be used to test the performance of Bayes2.

As can be noticed from the above results, the only errors counted are those when there are missing and extra arcs in the learned network compared to the gold-standard network. So, there can be errors involving the wrong orientation of the arcs but these are

not given here. The main reason is because the algorithm presented in chapter 6 has the special feature that it may produce a pattern; i.e., a mixed graph containing both directed arcs and undirected arcs, as mentioned in section 3.4.1 of chapter 3, and Bayes2 always outputs a directed graph. Hence, it is more convenient to only take the errors involving missing and extra edges in order to compare more easily the results obtained by all these algorithms. Moreover, two other constraint-based algorithms by Cheng et al. (1998) and Spirtes et al. (Spirtes, Glymour et al. 1993; Spirtes, Scheines et al. 1994) presented in chapter 7 also produce a pattern; so again the comparison among all these algorithms is more easily performed if only the missing and added arcs are counted.

As reviewed in section 3.4.1 of chapter 3, it is not possible to learn about arc direction from any conditional independence measure or probabilities alone. However, the way Bayes2 works is that it always adds **directed** arcs; feature that can be clearly seen in step 1 of the procedure Bayes2. Due to this characteristic, Bayes2 is very likely to make mistakes regarding the direction of arcs. Since Bayes2 is intended to be a support tool for experts in some knowledge areas, this problem can be fixed if the experts are asked to change the arc directions based on their knowledge and experience; a practice which seems to be far less difficult for them than constructing the whole network structure from scratch. This is another reason why the only errors taken into account are those when there are missing and added arcs.

In the first three results of table of figure 4.7, it is very clear that Bayes2 produces a huge number of extra arcs. These results show that, in general, Bayes2 does not give a sound and useful support tool for making the knowledge elicitation process much easier, especially when the number of variables in the database is large. Needless to say, these results are very inaccurate and lead to the **overfitting** of data (see section 4.4 for an explanation of this concept). This overfitting of data is not a desired characteristic because the model obtained is too complex (many added arcs) so it will fail to capture and exploit independencies in the domain being studied; i.e., it contradicts Occam's razor. Moreover, such a model will be very far from reflecting the real pattern underlying the data, which will lead to unreliable estimates of the dependent variable. From these results, it appears

that one of the main problems with Bayes2 is that it is unable to carry out, when necessary, conditional independence tests of second or higher order as can be seen in step 2 of procedure Bayes2 where only one variable ($Y_{(i)}$) forms part of the conditional set. Therefore, the cardinality of the conditional set is 1 and that is why Bayes2 can carry out only conditional independence tests of first order. From this procedure and according to definition 2.19 of chapter 2, it is possible to infer that only one path coming from any node to the dependent variable X can be blocked by the most informative variable $Y_{(i)}$. So, it seems not enough to have only one node in the conditional set, even if this node is the most informative. Another problem, not as easily perceived but as important as the previous one, appears when the use of the mutual information of just a couple of variables does not provide the correct ancestral ordering. Figure 4.8 helps to visually clarify these inherent problems of Bayes2.

Suppose that 5 random variables A, B, C, D and E take part in the problem and a database has the underlying Bayesian network structure of the figure 4.8(a); variable E is the designated dependent variable. That is to say, figure 4.8(a) is the gold-standard network. Figure 4.8(b) shows that initially there are no arcs connecting any variable (stepwise forward algorithm). Figure 4.8(c) shows the result after calculating the mutual information (eq. 4.4) between the independent variables A, B, C and D and the dependent variable E. Figure 4.8(d) shows the final result Bayes2 produces (3 extra arcs).

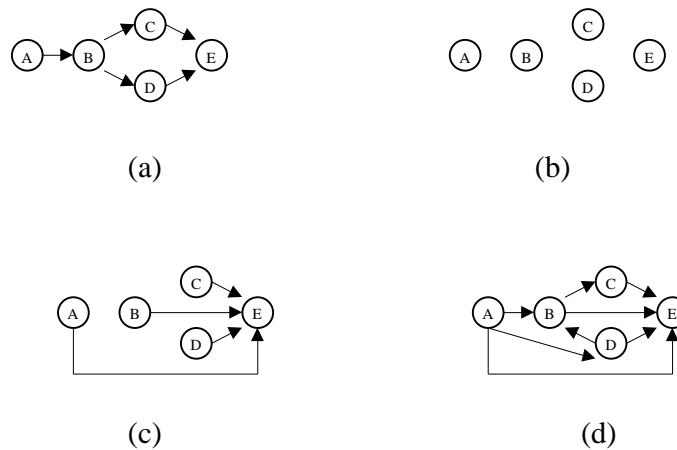


Figure 4.8: An example showing the incapability of Bayes2 to perform higher order conditional independence tests

From this example, as mentioned above, two important problems can be observed. In the first problem (about the cardinality of the conditional set), take as reference the arc $B \rightarrow E$ of figure 4.8(d). This arc cannot be removed because Bayes2's lack of capability to perform higher conditional independence tests. To explain this better, take the two paths between B and E : the path $B \rightarrow C \rightarrow E$ and the path $B \rightarrow D \rightarrow E$. The actual ancestral ordering (see theorem 2.2 of chapter 2) of the variables is A, D, B, C, E (E is the dependent variable). As can be seen, C is the most informative variable but it only blocks (because it is the only variable taking part in the conditional set) one of the two paths between B and E , namely, the path $B \rightarrow C \rightarrow E$. Because of the impossibility of Bayes2 to perform higher conditional independence tests, it cannot take variable D as being part of the conditional set. In fact, the set of variables that d-separates B and E is $\{C, D\}$. So, if this set were taken into account as the separation set, the arc $B \rightarrow E$ could then be removed. But because of the intrinsic nature of Bayes2, it cannot ever pick this set as the d-separation set because this set has cardinality 2 and Bayes2's conditional set has at most cardinality 1. Therefore, the arc $B \rightarrow E$ cannot be removed.

In the second problem (about the incorrect induction of the ancestral ordering), recall what was said at the end of section 4.2: that one of the main contributions of Bayes2 is the way in which the ancestral ordering between variables is obtained. But unfortunately, at the same time, it is the main drawback of the algorithm since the quality of the learned structure is highly dependant on this ordering. If watched carefully, this becomes more or less evident when figures 4.8(a) (the gold-standard network) and 4.8(d) (the learned network) are compared. Take as reference the arc $A \rightarrow D$ of figure 4.8(d). Take now the path $A \rightarrow B \rightarrow D$. In order to remove this arc, the descendent ordering of informativeness would have to be D, B and A so that we can have the path $A \rightarrow B \rightarrow D$ instead of $A \rightarrow D$.

D. If this ordering were the correct, then it would be possible to d-separate A from D conditional on B . As can be inferred, this is not the case since, if it were, the arc $A \rightarrow D$ would not exist. In fact, the ancestral ordering chosen by Bayes2, according to the information the variables provide is A, D, B, C, E , as mentioned above. As can be seen, to d-separate A from D conditional on B , it would be necessary to have the following ordering: A, B, D, C, E .

From the results of the table of figure 4.7 and from this example, it can be concluded firstly that the limitation of Bayes2 to perform conditional independence tests of second and higher order will lead, in general, to the impossibility of removing extra arcs and this in course will lead to the derivation of large networks. Secondly, the use of mutual information involving only a pair of variables to determine the next variable to be considered as the dependent variable (step 3 of procedure Bayes2) will also lead to get improper orderings and hence to obtain poor results reflected in the derivation of large networks as well.

The realisation of these problems led to the design of two other algorithms. The first one, presented in chapter 5, is capable of performing conditional independence tests of higher order; the second one, presented in chapter 6, is capable of performing this feature as well but also capable of getting rid of the ancestral ordering avoiding the problems caused by step 3 of procedure Bayes2.

Before closing this section, it is also very important to discuss the last result: that presented when running Bayes2 on the SEWELL & SHAH database. It seems that Bayes2 performs considerably well in this special case. Recall that this is the only database that was not simulated so the “gold-standard” network was provided by Heckerman’s algorithm. Under such a situation, another criterion, apart from the qualitative one, is needed to grade the result given by Bayes2 having another reference to measure the performance of this procedure. In section 4.4.2, a quantitative criterion is used to grade this result.

In general, it can be argued that, as the number of arcs increases with respect to the number of variables (as it is the case in the networks of figures 4.2, 4.3 and 4.4), it is more likely to have more than one arc pointing to a specific node, which makes the error regarding the cardinality of conditional independence tests more likely to occur. Furthermore, it can also be argued that, even when the number of variables and arcs is not too many, it is still likely for Bayes2, because of the selection of the next dependent variable according to the information it provides (ancestral ordering), to choose an incorrect ordering.

In the next section, we present a quantitative criterion called **MDL** (minimum description length) that will help us identify plausible, likely and parsimonious models (Bayesian networks) that can account well for the data at hand.

4.4 Goodness of fit.

The principle that has been proposed as guidance for solving the model selection problem is the criterion referred to as **Occam's razor**; also known as **parsimony**, as mentioned in chapter 3. Basically, the concepts of adequacy and simplicity are embedded in it: a good model is that which explains or fits the data coming from the phenomenon under investigation in the best possible way (accuracy) and in the least complex way (Myung, Forster et al. 2000). One of the first and obvious questions to ask is how to decide whether a model is good or not; i.e., how to measure if a proposed model fits the data well and in the simplest way. Some quantitative criteria, called **goodness of fit** metrics, have been proposed in order to measure both adequacy and simplicity of a model (Chickering 1996; Heckerman 1998; Grunwald 2000; Zucchini 2000). As the name suggests, these metrics measure how well a specific model fits the data, as said above. But before presenting the criterion used in this work, it is necessary to explain some important concepts that are usually found when applying a goodness of fit measure.

Generally speaking, there are two situations when the goodness of fit of a model is being measured: whether the model **overfits** the data or whether the model **underfits** the data. In the ideal case none of these situations happens, it can be said that the proposed model perfectly fits the data. However, in many real-world cases, this last situation cannot be proved because the underlying process responsible of generating the data is not known, thus, a model selection technique has to be applied.

Having this in mind, it is natural then to consider that every possible model that can be taken into account will have some errors when representing a certain phenomenon. Thus, the goal is searching for an **approximate** model capable of helping us understand the problem under study in a plausible way. If the real process is not known (as usually happens), then it is very hard to assess this model with no reference points (Browne 2000).

The reference point against which the learned networks (given by the procedures presented in chapters 4, 5, 6 and 7) are compared are the gold-standard networks. The performance of such procedures is measured using the goodness of fit metric explained in the next section. According to this metric, the best model is the one which is the closest to the gold-standard network. Browne (2000) suggests that, even when a procedure containing a goodness of fit measure suggests a certain model, the ultimate decision of which is the model to select has to be left to the human judgment. This is exactly the aim of the algorithms proposed here: to work as a support tool only suggesting to the human experts possible good routes where to find the solution. So it is the experts who have the final decision about the validity of the proposed model.

Every model has a set of parameters that can take a different set of values in a determined situation. Of course, some models will have a greater number of parameters than others or perhaps the same number. If the parsimony principle is taken as the guidance principle, then a trade-off between model complexity and accuracy has to be found (Friedman and Goldszmidt 1998a; Sucar and Martinez-Arroyo 1998; Grunwald 2000). The problem of **overfitting** appears when, compared to the real process, the selected model uses more parameters than necessary in order to describe properly the data at hand produced by this process. This situation can be seen making use of simulation studies such as those presented in section 4.3. In that section, the gold-standard networks of figures 4.2, 4.3, 4.4, 4.5 and 4.6 are the “real” processes responsible of generating the data. The first three results in table of figure 4.7 show the huge number of extra arcs obtained by running Bayes2 compared to those arcs corresponding to the gold-standard networks. In all these cases, it can be concluded that the proposed models overfit the data. In the context of Bayesian networks, this overfitting of data is reflected in the number of extra arcs which, in turn, defines more complex conditional probability distributions that are much more difficult to determine, provided more cases in the database are needed to represent such distributions. In other words, since the resultant networks are not sparse, i.e., some nodes will have a big number of parents, the number of cases in the database needs to be enormous if we want to estimate all the probabilities of interest (Chickering 1996; Cruz-Ramirez 1997). This implies that the factorisation of the joint probability from the

conditional ones (equation 2.33 of chapter 2) cannot be properly exploited. In sum, the overfitting problem increases the number of parameters to be fitted (Friedman and Goldszmidt 1998a).

On the other hand, the problem of **underfitting** appears when the opposite situation happens: given the “real” process of figure 4.5, the selected model (given by Bayes2) uses fewer parameters than necessary when describing the data. This case can be seen from table of figure 4.7 (ASIA) where there is one missing arc. Hence, it can be concluded that, in the context of Bayesian networks, this underfitting of data is reflected in the number of missing arcs (Friedman and Goldszmidt 1998a). Because of its intrinsic nature, accurate fitting of parameters cannot solve this problem of underfitting.

However, it can be claimed that underfitting is better than overfitting; i.e., simpler (parsimonious) models are then preferred over more complex ones following Occam’s razor philosophy. To support this claim, many methods have been proposed for avoiding the overfitting of data instead of avoiding the underfitting of data (Linhart and Zucchini 1986; Li and Vitányi 1993; Heckerman, Geiger et al. 1994; Buntine 1996; Chickering 1996; Chickering, Heckerman et al. 1997; Friedman and Goldszmidt 1998a; Heckerman 1998; Sucar and Martinez-Arroyo 1998; Bozdogan 2000; Browne 2000; Grunwald 2000; Zucchini 2000). Grunwald expresses this situation in a simply way paraphrasing Occam’s razor: “If you overfit, you think you know more than you really know. If you underfit, you do not know much but you know that you do not know much. In this sense, underfitting is relatively harmless while overfitting is dangerous” (Grunwald 2000, p. 148). In a similar vein, Grunwald also sustains that “a model that is too complex is typically worthless while a model that is much too simple can still be useful” (Grunwald 2000, p. 150).

In the next subsection, the quantitative criterion used for measuring how good a model is, according to the data at hand, is presented.

4.4.1 The MDL criterion.

The **MDL** criterion (minimum description length) also known as **stochastic complexity**, proposed originally by (Rissanen 1989), was chosen because it introduces very naturally and intuitively the parsimony principle reviewed previously (Friedman and Goldszmidt 1998b; Forster 2000; Grunwald 2000). The general idea behind MDL is that the regularities underlying a specific dataset can be captured in a compressed manner. In other words, if some regularities, uniformities or patterns are intrinsically present in the data, then it is possible to use such a feature to represent the data in a much more compact and economic way. Thus, MDL looks for models that facilitate the shortest encoding of data. A slightly modified example, originally proposed by (Chaitin 1975), will be very useful to explain this idea.

Imagine that the two following sequences have to be somehow described. Suppose that those sequences are 10,000 bits long:

010101010101010101... 01 (a)

01101100110111100010... 11 (b)

If observed carefully, it is possible to find a regularity or a pattern in sequence (a), which is the repetition of 01 five thousand times whereas sequence (b) does not seem to have any capturable regularity. Now, if a computer program were written to represent the first sequence, it would look like the following computer program:

Print 01 five thousand times (1)

Whereas, if a computer program were written to represent the second sequence, it would look like the following computer program:

Print 01101100110111100010... 11 (2)

It can be clearly seen that program (1) is much shorter (in bits) than the sequence it represents (a). On the other hand, program (2) is about the same size the sequence it specifies (b). Therefore, sequence (a) can be compressed in a sequence of instructions much shorter than (a) itself while sequence (b) cannot.

Now, the whole idea of MDL is not using a computer program to compress the regularities found in the data but a description method (model) in a similar way. This description method can perfectly be, for instance, the set of polynomials of degree n , the set of all neural networks, the set of Bayesian networks and so on. Thus, in order to compress the data (if the data are compressible), it is necessary to find a suitable model that can represent a compact version of these data. It is then the **total description length** (the sum of the length of any of the possible models used to compress the data and the length of the compressed version of the data) that is taken into account by MDL to select the best model. That is to say, MDL chooses the model that minimizes such a total description length (Rissanen 1989; Li and Vitányi 1993; Friedman and Goldszmidt 1998b; Grunwald 2000). In the context of this dissertation, the model is a Bayesian network. The MDL principle then selects the Bayesian network with the minimum length (among different network structures) combined with the length of the encoded data. As said before, this represents a trade-off between complexity and accuracy. A formula to compute this trade-off needs to incorporate basically two terms; one for measuring how well the model fits or predicts the data and the other for punishing the complexity of that model (Heckerman 1998). Equation 4.12 represents such a formula:

$$MDL = -\log P(D | \theta) + \frac{k}{2} \log n \quad (4.12)$$

where D represents the data, θ represents the parameters of the Bayesian network, k represents the dimension of the network and n represents the sample size (number of cases).

Let us analyse carefully the terms of equation 4.12 and their impact on the final behaviour of the MDL score. Figure 4.9 (Bouckaert 1994) will help us to clarify graphically such an interaction of these two terms.

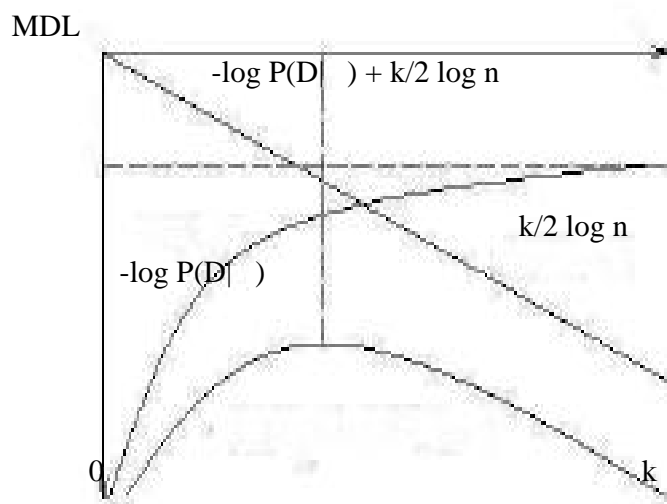


Figure 4.9: The interaction of the terms in the MDL score

As can be seen from figure 4.9, the x-axis represents the dimension of the model, which is more or less directly proportional to the number of arcs in the Bayesian network while the y-axis represents the MDL score.

The first term of equation 4.12 is the conditional entropy given by the Bayesian network structure under consideration. Thus, this term describes the amount of uncertainty of the data D given the parameters (see section 4.1). In other words, it describes how well the parameters define the probability distribution implicitly contained in the data. Recall that entropy and information are inversely related to each other; hence the first term of equation 4.12 is zero when there exists complete knowledge (or no uncertainty) and is maximal (non-negative) when uncertainty is maximal. Figure 4.9 shows that, for the first term, when the number of arcs increases the entropy decreases since the probability distribution will be more accurate when the network becomes less sparsely connected (the more variables with an arc to a certain variable, the more information they provide to explain that variable). That is why, the MDL criterion introduces a term (the second one in

equation 4.12) for punishing the complexity for that network structure. Thus, for the second term, its value increases when more arcs are added (the more complex the network, the bigger the value for the second term). This second term allows us to introduce the parsimony principle: a Bayesian network with fewer number of arcs (simpler structure) is preferred over another with more arcs (more complex structure) unless the conditional entropy (first term) of the more complex model is much lower than that of the simpler model. This fact can be seen from figure 4.9 as well. The MDL score will first decrease when more arcs are added to the network structure and eventually increase if the number of arcs keeps growing. So, in theory, there is a global minimum for this score, which represents the optimal trade-off between accuracy and complexity (shown by the vertical dotted line). The problem is that, as shown by equation 3.1 of chapter 3, it is not possible, in general, to carry out an exhaustive enumeration of the possible number of structures. Hence, we cannot be sure, most of the time, if the model chosen by any search procedure, such as Bayes2, is the best one. The best we can do is to search among a manageable list of different models (perhaps proposed by different procedures) and use the MDL criterion to select the best model among those within this list. In chapter 6, we compare the performance of the algorithms proposed in this dissertation using this criterion. In chapter 7, we compare the models given by two well-known algorithms and those suggested by the best procedure presented in this thesis using this criterion.

By the moment, we first present a slightly modified example, used in (Buntine 1996), to illustrate how the right-hand side terms in equation 4.12 are coded in a Bayesian network. Figures 4.10 and 4.11 show respectively a sample database and a Bayesian network structure possibly representing the probability distribution underlying such database.

case	A	B	C
1	T	F	T
2	T	T	T
3	F	T	T
4	F	T	T
5	F	F	F

Figure 4.10: A sample database

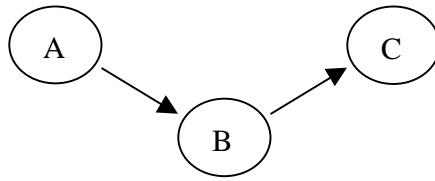


Figure 4.11: A three-variable Bayesian network

The probability of the data given the parameters $P(D | \theta)$ can be decomposed in the probability of each case of table of figure 4.10 given the parameters. In this context, the real values that are used to represent or specify the conditional probability distributions in a database or in some formula (such as the covariance matrix) are referred to as the **parameters** of the Bayesian network. Of course these parameters have a specific value for each different structure of a Bayesian network so the first term on the right-hand side of equation 4.12 can be rewritten taken now into account the structure of the network, which will result in equation 4.13:

$$MDL = -\log \prod_{i=1}^n P(case_i | \theta, S) + \frac{k}{2} \log n \quad (4.13)$$

where $case_i$ is the i th case of the database and S is the specific Bayesian network structure to be measured. All the remaining variables are the same as in equation 4.12. Using the logarithmic properties, equation 4.13 becomes equation 4.14:

$$MDL = - \sum_{i=1}^n \log P(case_i | S) + \frac{k}{2} \log n \quad (4.14)$$

In this particular example, using table of figure 4.10 and the Bayesian network of figure 4.11, the numerical calculations will yield:

For case₁:

$$\log P(case_1 | S) = \log P(A = T)P(B = F | A = T)P(C = T | B = F) = -1$$

For case₂:

$$\log P(case_2 | S) = \log P(A = T)P(B = T | A = T)P(C = T | B = T) = -0.6989$$

For case₃:

$$\log P(case_3 | S) = \log P(A = F)P(B = T | A = F)P(C = T | B = T) = -0.3979$$

For case₄:

$$\log P(case_4 | S) = \log P(A = F)P(B = T | A = F)P(C = T | B = T) = -0.3979$$

For case₅:

$$\log P(case_5 | S) = \log P(A = F)P(B = F | A = F)P(C = F | B = F) = -1$$

The result of the first term of the right-hand side of equation 4.14 will then be:

$$- \sum_{i=1}^5 \log P(case_i | S) = 3.4947$$

Now, in the second term of the right-hand side of equation 4.14, the dimension k of the model (Bayesian network) is calculated as equation 2.34 of chapter 2 shows (Heckerman 1998):

$$k = \sum_{i=1}^n q_i(r_i - 1) \quad (4.15)$$

where q_i is the number of possible configuration of the parents of variable X_i and r_i is the number of states of that variable. If a variable has no parents, then $q_i = 1$.

Let us see what the result of k is in this example. Let $A = 1$, $B = 2$ and $C = 3$. Recall that each variable is a binary variable; thus, we have:

For A (no parents):

$$q_1 = 1 \quad r_1 - 1 = 2 - 1 \quad (q_1)(r_1 - 1) = (1)(1) = 1$$

For B (with A as a parent which has two possible values):

$$q_2 = 2 \quad r_2 - 1 = 2 - 1 \quad (q_2)(r_2 - 1) = (2)(1) = 2$$

For C (with B as a parent which has two possible values):

$$q_3 = 2 \quad r_3 - 1 = 2 - 1 \quad (q_3)(r_3 - 1) = (2)(1) = 2$$

So, the dimension of the model is:

$$k = 1 + 2 + 2 = 5$$

Then the second term will become:

$$\frac{k}{2} \log n = \frac{5}{2} \log 5 = 1.7474$$

So, the final result will be:

$$\mathbf{MDL} = 3.4947 + 1.7474 = 5.2421$$

This final result per se only represents an index which indicates that for the dataset of figure 4.10 and for the network of figure 4.11, the MDL score is 5.2421. As said above, such a score can start making sense when compared through various network structures so

that one of them can be selected, under this criterion, as the best one (among all these structures). In other words, as Browne points out (2000), it is hard to assess a model on its own with no reference points. Hence, it is necessary to consider different competing models simultaneously and keep the one which minimizes the MDL score. Browne (2000) also points out that no numerical index calculated from the data can say anything regarding the interpretability of a model. Thus, as said also previously, the final word about which model is to be selected is that of human experts. However, metrics such as MDL helps us select parsimonious models which are, it can be argued, more easily interpretable and more likely to make sense than more complex models. In section 6.4 of chapter 6, the MDL results given by Bayes2 will be compared against those given by the procedures presented in chapters 5 and 6.

In the next chapter, an extension of the Bayes2 algorithm, called Bayes5, is presented, which, in contrast with its predecessor, will be able to perform conditional independence tests of second and higher order.

Chapter 5

Bayes5: extensions and improvements of Bayes2

This chapter presents an extension of the Bayes2 algorithm comparing the qualitative results of this extended procedure against those by Bayes2. Based on such results, it concludes that this new algorithm does not recover properly the structure of a Bayesian network from data when the number of variables of the problem under consideration is large, leading to the overfitting of the data. It finally proposes another extension of these two algorithms to build correctly and accurately Bayesian networks from data.

5.1 Improvements of Bayes2.

As clearly seen from the experimental results shown in the previous chapter (see section 4.3), one of the main problems with Bayes2 was its incapability of performing conditional independence tests of second and higher order. As suggested by (Martinez-Morales 1995), since such a problem was the first to be recognised because of its evidence, it was the first to be solved. The first impression was actually that only extending Bayes2's features regarding its ability to carry out higher conditional independence tests could improve the overall results. In section 5.3, the results of such an extension will be presented and discussed. But first, in the next section, the Bayes5 algorithm will be described.

5.2 Description of Bayes5.

As an extension of Bayes2, Bayes5's general idea is the same as its predecessor. The only difference has to do with the increasing of the number of variables in the conditional set, as will be seen in the procedure itself.

Procedure Bayes5

Let X be the dependent random variable and let Y_1, Y_2, \dots, Y_n be the **unordered** independent random variables.

1. Compute the value of the mutual information (equation 4.4) that each variable Y_i ($1 \leq i \leq n$) provides to X and order them according the information they provide from the largest to the smallest value. The **ordered** variables will be labelled now $Y_{(1)}, Y_{(2)}, \dots, Y_{(n)}$. Draw an arc from $Y_{(1)}$ to X . Then use formula 4.8 to check whether the null hypothesis H_0 (two variables are independent from each other) for $Y_{(i)}$ ($2 \leq i \leq n$) holds or not. If H_0 does not hold then draw a **directed** arc from $Y_{(i)}$ ($2 \leq i \leq n$) to X .
2. Compute the value of the conditional mutual information (eq. 4.5) that each variable $Y_{(i)}$ ($2 \leq i \leq n$) provides to X given $Y_{(1)}$. Then use formula 4.10 to check whether the null hypothesis H_0 (two variables are independent from each other given a set of variables Z) holds or not. If H_0 holds then remove the arc from $Y_{(i)}$ to X . If H_0 does not hold, leave the arc intact.
3. Let $Y_{(1)}, Y_{(2)}, \dots, Y_{(k)}$ be the ordered variables that have a directed arc to X ; $1 \leq k \leq n$.
For $a = 2$ to $a = 5$
 Let $b = a + 1$
 If $Y_{(b)} \in \emptyset$
 Compute the value of the conditional mutual information (eq. 4.5) that each variable $Y_{(b)}$ provides to X given the set $Y_{(1)}, \dots, Y_{(a)}$. Then use formula 4.10 to check whether the null hypothesis H_0 (two variables are independent each other given a set of variables Z) holds or not. If H_0 holds then remove the arc from $Y_{(b)}$ to X . If H_0 does not hold, leave the arc intact.
4. Do $X = Y_{(1)}$ and delete $Y_{(1)}$ from the set of independent random variables. Let the number of independent variables $n = n - 1$. If $n \geq 2$ then go to step 1; otherwise STOP.

End of Procedure Bayes5

As can be easily compared, procedure Bayes5 has one additional step with respect to procedure Bayes2, namely, step 3 of procedure Bayes5. Such a step permits to, whenever possible, carry out conditional independence tests from second to a maximum of fifth order

(variable **a** in this step is what controls the order of these tests). It also must be noted that the variables $Y_{(1)}, Y_{(2)}, \dots, Y_{(k)}$ in step 3 will change over time.

As in the case of Bayes2, Bayes5 does not need a complete ordering of the variables but only the specification of one dependent variable. Once this is done, as happens with its ancestor, procedure Bayes5 finds an ancestral ordering for the independent variables. In other words, because Bayes5 is a successor of Bayes2, it inherits all the properties of its predecessor and extends Bayes2's capabilities in the sense of performing higher order independence tests. This also means that Bayes2's limitations already mentioned in the previous chapter, which have to do with the improper ordering chosen, are inherited by Bayes5 as well. However, in the first place, this problem was not recognised but only the problem regarding higher order independence tests. That is why Bayes5 was implemented; i.e., we thought that it would be enough if only the latter problem were solved. The results presented in sections 5.3 and 5.4 show that this was not the case.

5.3 Experimental results.

As in the previous chapter, the same methodology to test experimentally the performance of Bayes5 applies. The same Bayesian network structures with their respective five databases presented in chapter 4 are used in this chapter to test such a performance.

Databases	dep. variable	cases no. var	Bayes2		Bayes5		Total of arcs
			m.a.	e.a.	m.a.	e.a.	
Alarm	node 5	10000;37	0	192	6	47	46
Child	node 19	5000;20	0	68	2	33	25
Diagcar	node o	5000;18	3	22	3	13	20
Asia	node 7	1000;8	1	1	1	1	8
Sew. and Shah	node cp	10318;5	0	0	0	0	7

Figure 5.1: Comparison of the results given by Bayes2 and Bayes5 from running the Alarm, Child, Diagcar, Asia and Sewell and Shah databases. Dep. variable refers to the dependent variable chosen for the analysis. Cases refer to the sample size while no. var refers to the number of variables involved in each database; m.a. and e.a. stand for missing arcs and extra arcs respectively. Finally, total of arcs corresponds to the total number of arcs in the gold-standard network

5.3.1 Discussion of the results.

As can be seen from table of figure 5.1, although Bayes5 produces much fewer extra arcs than Bayes2 does, it is evident that Bayes5 still generates a large number of additional arcs. Therefore, it can be concluded that the idea of only augmenting step 3 in procedure Bayes5 was not enough to produce accurate results.

In the last 2 databases (ASIA and SEWELL & SHAH), Bayes5 reproduces the results of Bayes2. The added feature of Bayes5 with respect of Bayes2 is more easily perceivable when the number of variables is big (37, 20 and 18 variables for the first three databases).

These results show that, in spite of Bayes5's added feature, it does not represent a sound and useful support tool for making the knowledge elicitation process much easier, as in the case of Bayes2. These results are still very inaccurate and lead also to the **overfitting** of data. The model obtained using this procedure (Bayes5) is still too complex and therefore it will fail too in capturing and exploiting independencies in the domain under study. From these results, it seems that now the main problem with Bayes5 appears when the use of the mutual information of just a couple of variables does not provide the correct ancestral ordering. The example seen in section 4.3.1 of chapter 4 will again be used to help visually clarify this inherent problem of Bayes5.

Figure 5.2(a) shows the gold-standard network. Figure 5.2(b) shows that initially there are no arcs connecting any variable (stepwise forward algorithm). Figure 5.2(c) shows the result after calculating the mutual information (eq. 4.4) between the independent variables A, B, C and D and the dependent variable E. Figure 5.2(d) shows the final result Bayes5 produces (3 extra arcs).

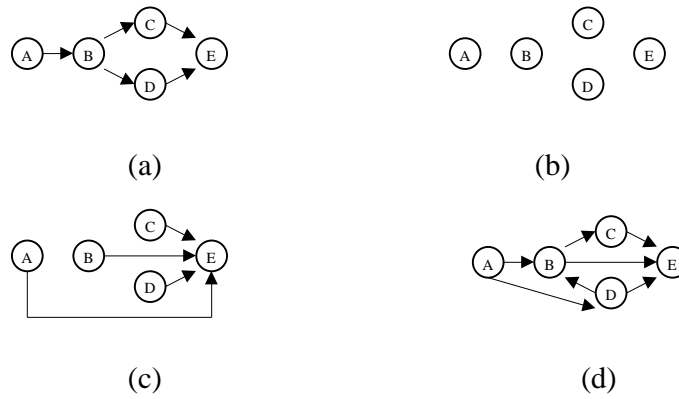


Figure 5.2: An example showing the problem of Bayes5 when it selects an incorrect ancestral ordering

From this example, it is possible to see that the added feature of Bayes5 (that of the ability to perform higher order independence tests) is not enough to recover the original structure of figure 5.2(a). If compared to the structure generated by Bayes2 (see figure 4.8(d) of chapter 4), the difference is only one arc removed, namely, the arc $A \rightarrow E$ (see figure 5.2(d)). The remaining arcs, which Bayes5 cannot get rid of, are $A \rightarrow D$ and $B \rightarrow E$. To see the picture much clearer, remember that the ancestral ordering (see theorem 2.2 of chapter 2) produced by both procedure Bayes2 and procedure Bayes5 is A, D, B, C, E . Thus, the arc $A \rightarrow E$ can be removed because B blocks the path between them. On the other hand, the arc $A \rightarrow D$ cannot be removed because there is no variable (in the ancestral ordering) that could block the path between them. Moreover, the arc $B \rightarrow E$ cannot be removed because there are two paths from B and E , namely $B \rightarrow C \rightarrow E$ and $B \rightarrow D \rightarrow E$, and only one variable (in the ancestral ordering) that blocks the path between them, namely, variable C .

These errors show a key result: they indicate that the augmentation in the number of variables of the conditional set (or d-separation set) is not enough to fix and improve the overall performance of Bayes2. This is because it is necessary to throw out the requirement of the ancestral ordering, given by the mutual information involving only a pair of variables, which determines the next variable to be considered as the dependent variable (step 4 of procedure Bayes5). This leads to improper orderings; hence poor results will be

reflected in the production of large networks, as in the case of the first three databases of table of figure 5.1.

Once this problem was detected, another algorithm had to be proposed in order to cope with the restrictions imposed by Bayes5. This new algorithm, called Bayes9 and which is presented in chapter 6, is capable of performing not only higher order independence tests but also capable of getting rid of the ancestral ordering, avoiding the problems caused by step 4 of procedure Bayes5, as seen above. Moreover, a quantitative comparison given by Bayes2, Bayes5 and Bayes9, according to the MDL criterion, will be also presented in the next chapter.

Chapter 6

Bayes9: extensions and improvements of Bayes5

This chapter presents an extended algorithm, with respect to those presented in chapters 4 and 5, which provides very good, accurate and encouraging results about the power and suitability of using the mutual information and conditional mutual information measures to correctly build Bayesian networks from data. Such a claim is supported by both the qualitative and quantitative results presented in this chapter.

6.1 Improvements of Bayes5.

As can be seen from the experimental results shown in chapter 5 (see section 5.3), one of the main problems with Bayes5 was the way it uses the mutual information and the conditional mutual information measures to determine the next variable to be considered as the new dependent variable. Independence tests of second and higher order had not been sufficient to produce accurate results, so the less easily perceivable problem, namely, the need for an ancestral ordering of the variables, was addressed. The new extension was then incorporated in an algorithm called Bayes9, which will be described in the next section.

6.2 Description of Bayes9.

The assumptions presented in section 4.2 of chapter 4 also hold for procedure Bayes9.

Although Bayes9 is an extension of Bayes5, it only resembles the latter in the sense of the use of information measures to carry out independence tests. Regarding the capability of performing independence tests of second and higher order, both of them can

handle this situation. However, Bayes5 is restricted to perform these higher order tests according to the ancestral ordering that it generates, which still prevents it from producing networks close to the gold-standard networks, as presented in table of figure 5.1 of chapter 5. In contrast, because procedure Bayes9 avoids the need for an ancestral ordering (and an initial dependent variable has no longer to be selected), such a procedure is capable of carrying out these higher order tests using all the possible combinations (of different cardinality) of the variables that are adjacent to a particular node.

Thus, procedure Bayes9 looks quite different compared to procedures Bayes2 and Bayes5, as can be seen in the description of the algorithm itself below. Bayes9 needs neither the specification of one dependent variable (as part of its input) nor the induction of an ancestral ordering of the independent variables based on the information they provide. If the independence test between a pair of variables fails (step 1), then an **undirected** arc is drawn between such variables rather than a directed arc, as it is the case of procedures Bayes2 and Bayes5. This is because Bayes9 algorithm takes into account what was said in chapter 3 about the impossibility of learning about arc direction from any independence measure of probabilities alone (see section 3.4.1). Secondly, step 2 permits conditional independence tests between a pair of variables given a conditional set whose content and cardinality change over time (this is basically the power set of order n). Finally, step 3 of Bayes9, which is based on step 3 of procedure PC (see section 7.1 of chapter 7) by Spirtes, Glymour et al. (1993), and step 4, which is based on a subset of the rules given by Verma and Pearl (Pearl 2000), make it able to leave some arcs undirected. Thus, procedure Bayes9 can represent a class of faithfully indistinguishable graphs by means of a pattern, as explained in chapter 3 (see also section 3.4.1). Accordingly, human experts, based on their knowledge and experience of the domain being modelled, can then orient these undirected arcs. Procedure Bayes9 is presented below.

Procedure Bayes9

Let Y_1, Y_2, \dots, Y_n be the random variables.

1. For $a = 1$ to $a = n$

For $b = a$ to $b = n$

If $a \neq b$

Compute the value of the mutual information (equation 4.4) for each pair of variables $Y_{(a)}, Y_{(b)}$. Then use formula 4.8 to check whether the null hypothesis H_0 (two variables are independent from each other) holds or not. If H_0 does not hold then draw an **undirected** arc from $Y_{(a)}$ to $Y_{(b)}$ and form a queue \mathbf{W} such that $\mathbf{W} = \bigcup (Y_{(a)}, Y_{(b)})$

end for b

end for a

2. Let $(\mathbf{W}, Y_{(a)})$ be the set of vertices adjacent to $Y_{(a)}$ in the undirected acyclic graph \mathbf{W}

while $\mathbf{W} \neq \emptyset$

Select the pair of variables $Y_{(a)}, Y_{(b)}$ from the beginning of \mathbf{W} ; then form the adjacency **power** set of $Y_{(a)}$ called $\mathbf{Z} = \{(\mathbf{W}, Y_{(a)}) \setminus Y_{(b)}\}$. Compute the value of the conditional mutual information (eq. 4.5) between each pair of variables $Y_{(a)}$ and $Y_{(b)}$ given \mathbf{Z} . Then use formula 4.10 to check whether the null hypothesis H_0 (two variables are independent from each other given a set of variables in \mathbf{Z}) holds or not. If H_0 holds then remove this pair of variables from \mathbf{W} . If H_0 does not hold, form a new set $\mathbf{X} = \bigcup (Y_{(a)}, Y_{(b)})$ (\mathbf{X} is the set of the final accepted adjacency relations) and remove this same pair of variables from \mathbf{W}

end while \mathbf{W}

3. Let $Y_{(a)} \in \mathbf{X}$, $Y_{(b)} \in \mathbf{X}$ and $Y_{(c)} \notin \mathbf{X}$ be three different variables. If $Y_{(a)} - Y_{(b)} \in \mathbf{X}$, $Y_{(b)} - Y_{(c)} \in \mathbf{X}$, $Y_{(a)} - Y_{(c)} \notin \mathbf{X}$ and $Y_{(b)}$ is not in the d-separation set of (a, c) , then orient the triplet $Y_{(a)} - Y_{(b)} - Y_{(c)}$ as the directed triplet $Y_{(a)} \rightarrow Y_{(b)} \rightarrow Y_{(c)}$

4. Do

If $Y_{(b)} - Y_{(c)} \in \mathbf{X}$ and $Y_{(a)} \rightarrow Y_{(b)}$ exist and $Y_{(a)} - Y_{(c)} \notin \mathbf{X}$, then orient $Y_{(b)} - Y_{(c)}$ as $Y_{(b)} \rightarrow Y_{(c)}$

If $Y_{(a)} - Y_{(b)} \in \mathbf{X}$ and $Y_{(a)} \rightarrow Y_{(c)}$ $Y_{(b)} \rightarrow Y_{(c)}$ exists, then orient $Y_{(a)} - Y_{(b)}$ as $Y_{(a)} \rightarrow Y_{(b)}$

while no more arcs can be oriented

End of Procedure Bayes9

In the next section, the experimental results are presented.

6.3 Experimental results.

As in the previous 2 chapters, the same methodology to test qualitatively the performance of Bayes9 applies. Also, the same five Bayesian network structures with their respective databases presented in chapters 4 and 5 are used in this chapter to test such a performance.

Databases	cases no. var	Bayes2		Bayes5		Bayes9		Total of arcs
		m.a.	e.a.	m.a.	e.a.	m.a.	e.a.	
Alarm	10000;37	0	192	6	47	3	1	46
Child	5000;20	0	68	2	33	0	2	25
Diagcar	5000;18	3	22	3	13	4	2	20
Asia	1000;8	1	1	1	1	4	0	8
Sew and Shah	10318;5	0	0	0	0	0	0	7

Figure 6.1: Comparison of the results given by Bayes2, Bayes5 and Bayes9 from running the Alarm, Child, Diagcar, Asia and Sewell and Shah databases. Cases refer to the sample size while no. var refers to the number of variables involved in each database; m.a. and e.a. stand for missing arcs and extra arcs respectively. Finally, total of arcs corresponds to the total number of arcs in the gold-standard network

6.3.1 Discussion of the results.

As can be seen from table of figure 6.1, Bayes9's performance is for much, the best of all of the three algorithms presented, namely, Bayes2, Bayes5 and Bayes9. Its very nature is to be, most of the time, very conservative; i.e., it tends to avoid overfitting of data, as can be seen in its trend of having more missing arcs than extra arcs. Bayes9 is much more likely to lead to the correct structures than Bayes2 and Bayes5; i.e., the structures it generates are very similar to those of the gold-standard networks. Moreover, as can be also

compared to the results presented in table of figure 4.7 of chapter 4 and table of figure 5.1 of chapter 5, when Bayes2 and Bayes5 run on medium and large networks, they are very likely to lead to improper structures due mainly to the lack of second and higher order conditional independence tests in the former and the lack of obtaining a proper ancestral ordering in both the former and the latter. Both algorithms, Bayes2 and Bayes5, tend to overfit the data; feature easily observed in the large number of extra arcs produced by such procedures. This overfitting of data is an undesired characteristic because the model obtained is too complex, which will cause unreliable estimates and not reflect the real pattern underlying the data (Grunwald 2000).

In contrast, the heuristics inside Bayes9 seem very appropriate, leading to quite accurate results. Bayes9 tends to underfit the data. As mentioned in section 4.4 of chapter 4, the underfitting of data is preferred over the overfitting because it allows getting more reliable estimates and discovering with certain precision the real pattern underlying the data. That is to say, the models obtained using procedure Bayes9 are likely to capture and exploit the independencies present in the domain under study.

The example seen in section 4.3.1 of chapter 4 will again be used to visually appreciate the extended features of Bayes9.

Figure 6.2(a) represents the gold-standard network. In contrast with the same example of chapters 4 and 5, this time there is no need to designate E as the dependent variable. Figure 6.2(b) shows that initially there are no arcs connecting any variable; i.e., Bayes9 is still a stepwise forward algorithm. Figure 6.2(c) shows the result after calculating the mutual information (eq. 4.4) between all pairs of variables. Figure 6.2(d) shows that Bayes9 reproduces the gold-standard network with the only difference that, because of its intrinsic capacity of representing patterns or mixed graphs, some arcs are not directed.

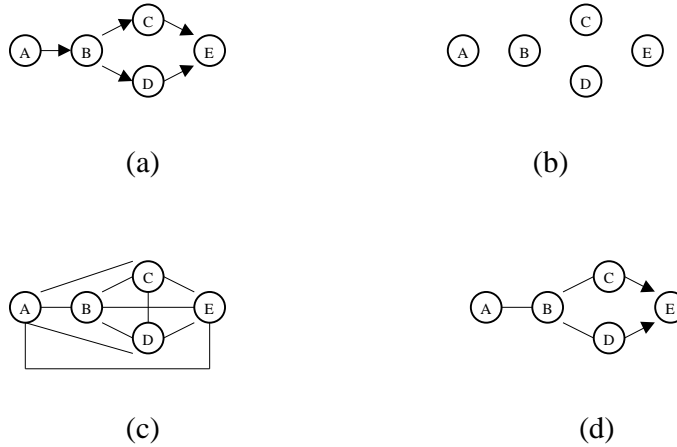


Figure 6.2: An example showing the performance of Bayes9

From this example, it is possible to see that the added features of Bayes9, with respect to those of Bayes2 and Bayes5, make it considerably better than its predecessors. It seems that the solutions of the key limitations found in Bayes2 and Bayes5 about the augmentation in the number of variables of the conditional set and the requirement of the ancestral ordering respectively, were enough to build a correct procedure that builds Bayesian networks from data accurately. See the **appendix** for a detailed description on how Bayes9 produces the final graph of figure 6.2(d).

To support such a claim, it is now imperative to compare quantitatively the results given by Bayes2, Bayes5 and Bayes9 using the **MDL** criterion. This will be done in the next section.

6.4 Comparison of the performance of Bayes2, Bayes5 and Bayes9 using the MDL criterion.

Table of figure 6.3 shows a comparison among the five gold-standard networks and the results obtained by Bayes2, Bayes5 and Bayes9 given the databases of these corresponding gold-standard networks.

Databases	MDL for golden nets	MDL for Bayes2	MDL for Bayes5	MDL for Bayes9
Alarm	40827.19	not computed	42496.67	167231.77
Child	36809.88	not computed	37981.43	75544.70
Diagcar	24874.50	25723.38	25817.81	59160.18
Asia	979.28	983.66	983.66	1270.97
Sewell & Shah	19779.70	19881.07	19881.07	38305.07

Figure 6.3: Comparison of the MDL results among the gold-standard networks and the networks proposed by Bayes2, Bayes5 and Bayes9

First of all, table of figure 6.3 shows that the MDL score for the results produced by Bayes2 on the databases ALARM and CHILD cannot be efficiently and consistently computed for two main reasons: the huge number of extra arcs combined with the sample size prevent the proper calculation of the MDL score. This is because there are many combinations of different variables' values that are not present in the sample. Therefore, the calculated probabilities will be zero; such a situation will result in unstable estimators for the probabilities and statistical tests. For the case of the DIAGCAR database, although Bayes2 also produces a large number of extra arcs (see table of figure 6.1), it is possible to reliably compute the value of its MDL.

Furthermore, from these quantitative results shown in table of figure 6.3, a very strange situation can be detected: the qualitative results presented in section 6.3 are not at all consistent with their quantitative counterpart. The results given by Bayes9 are far bigger than those given by Bayes2 and Bayes5. On the one hand, the qualitative results shown in section 6.3 seem to tell that Bayes9 is the best of the three procedures presented in chapters 4, 5 and 6. But, on the other hand, the quantitative results presented in this section seem to tell the opposite thing: that the results given by Bayes9 are the worst of the three procedures.

The answer to such a contradiction lies in the way the MDL criterion is calculated. Equation 4.14 in chapter 4 indicates that the probabilities usually computed are conditional probabilities. Moreover, the dimension of the model given by equation 4.15 of the same

chapter takes into account that the whole Bayesian network is directed. In contrast, because Bayes9 produces a pattern (a mixed graph), there will be some arcs undirected. Therefore, instead of computing a conditional probability, a joint probability will be computed resulting in a bigger number. Also, because of the lack of direction in some of the arcs, formula 4.15 cannot be used so that the calculation of the total number of parameters will result in a much bigger number as well. In other words, the total number of parameters for a connected pair of nodes, say $A - B$, will be $(q \times r) - 1$, where q is the number of states of variable A and r is the number of states of variable B . Thus, since the resultant network produced by Bayes9 is a pattern, the joint probability cannot be factored according to formula 2.33 of chapter 2 hence the power of Bayesian networks regarding the compact representation of a joint probability cannot be exploited in its totality.

This problem can be corrected directing the undirected arcs. If this is done according to the expert's knowledge at hand, then much better results are obtained compared to those in figure 6.3, as shown in figure 6.4.

Databases	MDL for golden nets	MDL for Bayes2	MDL for Bayes5	MDL for Bayes9 with all the arcs directed
Alarm	40827.19	not computed	42496.67	42403.26
Child	36809.88	not computed	37981.43	36818.71
Diagcar	24874.50	25723.38	25817.81	25305.17
Asia	979.28	983.66	983.66	1045.04
Sewell & Shah	19779.70	19881.07	19881.07	19779.70

Figure 6.4: Comparison of the MDL results among the gold-standard networks and the networks proposed by Bayes2, Bayes5 and Bayes9 (with all the arcs directed)

Table of figure 6.4 shows an overall much better performance of Bayes9 over Bayes2 and Bayes5. Both the quantitative and the qualitative results (figure 6.4 and figure 6.1 respectively) show that these two different criteria to measure the goodness of fit of the resultant Bayesian network structures are consistent with each other; i.e., simpler networks

(given by Bayes9) are preferred over more complex ones (given by Bayes2 and Bayes5), according to the MDL score. For the case of Bayes2 and Bayes5, the MDL results support analytically that the observations (made in sections 4.3 and 5.3 respectively) about the big number of extra arcs are, as said before, correct and consistent with the numerical results of the MDL goodness of fit measure. Thus, it can be argued that MDL provides a sensible solution to the model selection problem mentioned in section 3.3 of chapter 3.

Regarding the performance of Bayes2 and Bayes5 on the real-world SEWELL & SHAH database, a peculiar situation can be recognised: under the qualitative criterion, the structures proposed by both procedures do not have either missing arcs or extra arcs; in fact, the only differences are two reversed arcs, one between SES and IQ and the other between PE and IQ (although these differences are not presented in table of figure 6.1) whereas, under the quantitative criterion, due to such differences, the result proposed by Heckerman's algorithm (Heckerman 1998) is slightly better than that suggested by Bayes2 and Bayes5.

Although Bayes5 is an improvement of Bayes2, it is still evident that such an algorithm still produces a large number of extra arcs, mainly in the ALARM, CHILD and DIAGCAR networks. In sum, the main feature added to Bayes5 with respect to Bayes2 (that of the augmentation of the number of variables in the conditional set) does not improve the results considerably. Even though such a feature produces a better performance regarding the total number of extra arcs, Bayes5 still inherits the poor heuristics of choosing an incorrect ancestral ordering based on the amount of information the variables provide, which still produces a large number of extra arcs. The overall performance of Bayes5 is still not good enough to produce accurate results. Hence, we concluded that this procedure needed another refinement phase, namely, the elimination of the requisite of choosing an ancestral ordering, which also involves the removal of the requisite of selecting a dependent variable. That is why procedure Bayes9 was designed and built.

6.4.1 Discussion of the MDL results.

According to the quantitative results shown in figure 6.4, it is possible to make some interesting comments. As the Occam's razor criterion suggests, it is in general easier to understand models that have few parameters; i.e., the tendency guided by such a criterion is to prefer simpler (more parsimonious) models over more complex ones. MDL provides a very good metric that combines (trade-off) goodness of fit (accuracy) and complexity in a single measure so that the models chosen are those which get, under this criterion, the minimum score. However, it is important to have in mind that MDL is based on heuristic principles (Sucar and Martinez-Arroyo 1998; Grunwald 2000); therefore it will not always produce the optimal result. But once more, being guided by Occam's razor, it can then be argued that it is more likely to find a sensible solution searching among the parsimonious models.

In the above results, mostly simulated databases (4 out of 5) were used to check the performance of Bayes2, Bayes5 and Bayes9. This was done because it is much easier to check the performance of these algorithms by comparing their results against each other having the gold-standard networks or the "real" processes underlying the data as a point of reference. These quantitative results, as well as the qualitative results already presented (figure 6.1), show that the inaccuracies that Bayes2 and Bayes5 are likely to produce are due mainly to the natural constraint of only being able to perform first order conditional independence tests in the first procedure and the impossibility for avoiding the need of an ancestral ordering in both of them. Such characteristics make them overfit the data (large networks with many extra arcs).

In order to justify the use of MDL as a reliable goodness of fit measure, it is very important to mention that a strong theoretical background supports the claim that such a measure represents a good criterion as a proposed solution to the model selection problem (Rissanen 1989; Friedman and Yakhini 1996; Grunwald 2000). The philosophy behind MDL is that, instead of looking for a true model, the goal is to search for a simple model which fits the data reasonably well. In other words, MDL looks for a good trade-off between accuracy and complexity. Furthermore, it is important also to keep in mind that not

only is the accuracy of a given model important but also its complexity since, sometimes, a complex model may give a good fit of the data without providing a good understanding or resemblance with the phenomenon under investigation. In other words, if only the accuracy of a model is taken into account, then it is likely to have models that overfit the data (Friedman and Goldszmidt 1998a). In integrating both accuracy and complexity, MDL, as stressed previously, avoids, in general, the overfitting of data.

Before closing this chapter, it is necessary to recall some important remarks. The added feature of Bayes9 with respect to Bayes5 regarding both the elimination of the selection of a dependent variable and the elimination of the induction of an ancestral variable ordering, makes it perform significantly much better. It can also be seen from the previous results that the very conservative nature of Bayes9 is to underfit the data, which, according to the claims made in section 4.4 of chapter 4, is much preferable than to overfit the data because the former characteristic leads to more appropriate, accurate and useful results. But it is very important to bear in mind that, for the results to be accurate, we need enough data to carry out reliable independence tests, as pointed out in assumption 4 of section 4.2. The size of these data will then be a function on the number of variables of the problem being considered, the number of variables that each variable can take and finally whether the graph is sparse or not (the average degree of the graph).

In the next chapter, we will compare the performance of Bayes9 against two well-known constraint-based systems that build Bayesian networks from data: Power Constructor and Tetrad II (Spirtes, Glymour et al. 1993; Spirtes, Scheines et al. 1994; Cheng 1998; Cheng, Bell et al. 1998).

Chapter 7

A comparison of the performance of three different algorithms that build Bayesian Networks from data

In this chapter, the performance of Bayes9 is compared against those of two important and well-known systems that, among other things, also construct Bayesian networks from data: Tetrad II (Spirtes, Glymour et al. 1993; Spirtes, Scheines et al. 1994) and Power Constructor (Cheng 1998; Cheng, Bell et al. 1998). The algorithms embedded in such systems use similar hypothesis testing principles to the procedures already presented in chapters 4, 5 and 6: the G^2 statistics for Tetrad II and the **Mutual Information** and **Conditional Mutual Information** for Power Constructor. They all are asymptotically distributed as χ^2 with appropriate degrees of freedom. Firstly, the procedure within the system Tetrad II is described. Secondly, the procedure within the system Power Constructor is described as well. Thirdly, the experimental results among these 3 algorithms are presented and compared. Finally, the goodness of fit for all these procedures are also presented and compared.

7.1 Tetrad II.

The algorithm within the Tetrad II system is called the **PC** algorithm and is presented below.

Procedure PC

1. Form the complete undirected graph C on the vertex set V
2. $n = 0$

```

repeat
  repeat
    select an ordered pair of variables X and Y that are adjacent in C such that
    Adjacencies(C,X)\{Y} has cardinality greater than or equal to n, and a
    subset S of Adjacencies(C,X)\{Y} of cardinality n, and if X and Y are
    d-separated given S delete edge X – Y from C and record S in Sepset(X,Y)
    and Sepset(Y,X)
  until all ordered pairs of adjacent variables X and Y such that
  Adjacencies(C,X)\{Y} of cardinality n have been tested for d-separation
  n = n + 1
until for each ordered pair of adjacent vertices X, Y, Adjacencies(C,X)\{Y} is of
cardinality less than n
3. For each triple of vertices X, Y, Z such that the pair X, Y and the pair Y, Z are each
adjacent in C but the pair X, Z are not adjacent in C, orient X – Y – Z as X → Y → Z if
and only if Y is not in Sepset(X,Z)
4. repeat
  If A → B, B and C are adjacent, A and C are not adjacent, and there is no
  arrowhead at B, then orient B – C as B → C
  If there is a directed path from A to B, and an edge between A and B, the orient the
  pair A – B as A → B
until no more edges can be oriented.

```

End of Procedure PC

As can be seen from above, procedure PC is a stepwise backward algorithm because it starts assuming that every pair of nodes is connected and then it tries to remove the arcs between these pairs whenever the independence test holds. Moreover, step 4 corresponds to rules R_1 and R_2 given in (Pearl 2000, p. 51).

7.2 Power Constructor.

The Power system contains two algorithms in it, which build Bayesian networks from data: algorithm **A** and algorithm **B**. In this section only algorithm B will be presented

because, in contrast with algorithm A, it does not need the specification of an ancestral ordering of the variables to work properly.

Algorithm B

Phase I: (Drafting)

1. Initiate a graph $G(V,E)$ where $V = \{\text{all the nodes of a data set}\}$, $E = \{\}$. Initiate an empty list L .
2. For each pair of nodes (v_i, v_j) where $v_i, v_j \in V$ and $i \neq j$, compute mutual information $I(v_i, v_j)$ using eq. 4.4. For all the pair of nodes that have mutual information greater than a certain small value ϵ , sort them by their mutual information and put these pairs of nodes into list L from large to small. Create a pointer p that points to the first pair of nodes in L .
3. Get the first two pairs of nodes of list L and remove them from it. Add the corresponding edges to E . Move the pointer p to the next pair of nodes.
4. Get the pair of nodes from L at the position of pointer p . If there is no adjacency path between the two nodes, add the corresponding edge to E and remove this pair of nodes from L .
5. Move the pointer p to the next pair of nodes and go back to step 4 unless p is pointing to the end of L , or G contains $N - 1$ edges. (N is the number of nodes in G . When G contains $N - 1$ edges, no more edges can be added without forming a loop).

Phase II: (Thickening)

6. Move the pointer p to the first pair of nodes in L .
7. Get the pair of nodes from L at the position of the pointer p . Call Procedure `try_to_separate_A(current graph, node1, node2)` to see if this pair of nodes can be separated in current graph. If so, go to next step; otherwise, connect the pair of nodes by adding a corresponding edge to E . (Procedure `try_to_separate_A` will be presented below).
8. Move the pointer p to the next pair of nodes and go back to step 7 unless p is pointing to the end of L .

Phase III: (Thinning)

9. For each edge in E , if there are other paths besides this edge between the two nodes, remove this edge from E temporarily and call Procedure `try_to_separate_A` (current graph, node1, node2). If the two nodes cannot be separated, add this edge back to E ; otherwise remove the edge permanently.
10. For each edge in E , if there are other paths besides this edge between the two nodes, remove this edge from E temporarily and call Procedure `try_to_separate_B` (current graph, node1, node2). If the two nodes cannot be separated, add this edge back to E ; otherwise remove the edge permanently. (Procedure `try_to_separate_B` will be also presented below).
11. Call Procedure `orient_edges` (current graph). (This procedure will be presented below too).

End of Algorithm B

Procedure `try_to_separate_A` (current graph, node1, node2)

1. Find the neighbours of node1 and node2 that are on the adjacency paths between node1 and node2. Put them into two sets $N1$ and $N2$ respectively.
2. Remove the currently known child-nodes of node1 from $N1$ and child-nodes of node2 from $N2$.
3. If the cardinality of $N1$ is greater than that of $N2$, swap $N1$ and $N2$.
4. Use $N1$ as the condition-set C .
5. Conduct a conditional independence test using eq. 4.5. Let $v = I(\text{node1}, \text{node2} \mid C)$. If $v < \alpha$, return ('separated').
6. If C contains only one node, go to step 8; otherwise, for each I , let $C_i = C \setminus \{\text{the } i^{\text{th}} \text{ node of } C\}$, $v_i = I(\text{node1}, \text{node2} \mid C_i)$. Find the smallest value v_m of v_1, v_2, \dots
7. If $v_m < \alpha$, return ('separated'); otherwise, if $v_m > v$ go to step 8 else let $v = v_m$, $C = C_m$, go to step 6.
8. If $N2$ has not been used, use $N2$ as condition-set C and go to step 5; otherwise return ('failed').

End of Procedure `try_to_separate_A` (current graph, node1, node2)

Procedure `try_to_separate_B` (current graph, node1, node2)

1. Find the neighbours of node1 and node2 that are on the adjacency paths between node1 and node2. Put them into two sets N1 and N2 respectively.
2. Find the neighbours of node1 and node2 that are on the adjacency paths between node1 and node2 and do not belong to N1. Put them into two sets N1'.
3. Find the neighbours of node1 and node2 that are on the adjacency paths between node1 and node2 and do not belong to N2. Put them into two sets N2'.
4. If $|N1 + N1'| < |N2 + N2'|$ let set $C = N1 + N1'$ else let $C = N2 + N2'$.
5. Conduct a conditional independence test using eq. 4.5. Let $v = I(\text{node1}, \text{node2} \mid C)$. If $v < \epsilon$, return ('separated') else if C contains only one node return ('failed').
6. Let $C' = C$. For each $i \in [1, |C|]$, let $C_i = C \setminus \{\text{the } i^{\text{th}} \text{ node of } C\}$, $v_i = I(\text{node1}, \text{node2} \mid C_i)$. If $v_i < \epsilon$, return ('separated') else if $v_i \leq v + e$ then $C' = C' \setminus \{\text{the } i^{\text{th}} \text{ node of } C\}$. (e is a small value).
7. If $|C'| < |C|$ then let $C = C'$, go to step 5; otherwise, return ('failed').

End of Procedure try_to_separate_B (current graph, node1, node2)

Procedure orient_edges (current graph)

1. For any two nodes s1 and s2 that are not directly connected and where there is at least one node that is the neighbour of both s1 and s2, find the neighbours of s1 and s2 that are on the adjacency paths between s1 and s2. Put them into two sets N1 and N2 respectively.
2. Find the neighbours of the nodes in N1 that are on the adjacency paths between s1 and s2 and do not belong to N1. Put them into set N1'.
3. Find the neighbours of the nodes in N2 that are on the adjacency paths between s1 and s2 and do not belong to N2. Put them into set N2'.
4. If $|N1 + N1'| < |N2 + N2'|$ let set $C = N1 + N1'$ else let $C = N2 + N2'$.
5. Conduct a conditional independence test using eq. 4.5. Let $v = I(s1, s2 \mid C)$. If $v < \epsilon$, go to step 8; otherwise, if $|C| = 1$, let s1 and s2 be parents of the node in C, go to step 8.
6. Let $C' = C$. For each $i \in [1, |C|]$, let $C_i = C \setminus \{\text{the } i^{\text{th}} \text{ node of } C\}$, $v_i = I(s1, s2 \mid C_i)$. If $v_i \leq v + e$ then $C' = C' \setminus \{\text{the } i^{\text{th}} \text{ node of } C\}$, let s1 and s2 be parents of the i^{th} node of C if the i^{th} node is the neighbour of both s1 and s2. If $v_i < \epsilon$, go to step 8. (e is a small value).
7. If $|C'| < |C|$ then let $C = C'$, if $|C| > 0$, go to step 5.

8. Go back to step 1 and repeat until all pairs of nodes are examined.
9. For any three nodes a, b, c if a is parent of b , b and c are adjacent, and a and c are not adjacent and an edge (b,c) is not oriented, let b be the parent of c .
10. For any edge (a,b) that is not oriented, if there is a directed path from a to b , let a be a parent of b .
11. Go back to step 9 and repeat until no more edges can be oriented.

End of Procedure orient_edges (current graph)

Algorithm B is a stepwise forward procedure since it starts assuming that nothing is connected, as can be seen in step 1 of such an algorithm.

In sum, procedures Bayes9 (section 6.2 of chapter 6), Tetrad II and Power Constructor use very similar ideas to build Bayesian networks from data: they all are constraint-based algorithms. They compare their hypothesis testing principles, mutual information and conditional mutual information for Bayes9 and Power Constructor and G^2 for Tetrad II, against the χ^2 statistic in order for them to decide the addition or deletion of an arc according to which is the case. Moreover, Bayes9 and Tetrad II use basically the same set of rules, given in (Pearl 2000), to direct the arcs whereas Power Constructor uses this same set of rules and also carries out a conditional independence test for giving direction to the arcs. Bayes9 and Power Constructor are stepwise-forward algorithms while Tetrad II is a stepwise-backward algorithm. The internal selection of the ordering of the variables that are part of the conditional set (or separation set) is possibly, in addition to the differences among the three algorithms themselves, a key feature to produce different results in each one of them. In the next section, these results are presented and discussed.

7.3 Experimental results among Tetrad II, Power Constructor and Bayes9.

The same five Bayesian network structures with their respective five databases, presented in chapters 4, 5 and 6, are used in this chapter to test the performance of the three algorithms.

Database	cases var	Tetrad II		PowCons		Bayes9		Total arcs
		m.a.	e.a.	m.a.	e.a.	m.a.	e.a.	
Alarm	10000;37	0	3	3	1	3	1	46
Child	5000;20	0	3	2	2	0	2	25
Diagcar	5000;18	4	1	6	0	4	2	20
Asia	1000;8	0	6	3	2	4	0	8
Sewell & Shah	10318;5	0	0	2	0	0	0	7

Figure 7.1: Comparison of the results given by Bayes9, Tetrad II and Power Constructor from running the Alarm, Child, Diagcar, Asia and Sewell and Shah databases. Cases refer to the sample size while no. var refers to the number of variables involved in each database; m.a. and e.a. stand for missing arcs and extra arcs respectively. Finally, total of arcs corresponds to the total number of arcs in the gold-standard network

7.3.1 Discussion of the results.

Table of figure 7.1 shows a very similar qualitative performance of Bayes9, Tetrad II and Power Constructor. It is not easy to decide, under such a criterion, which one of these three algorithms represents the best solution. One sensible solution could be a trade-off between the number of both missing and extra arcs. However, it is important to take into account that although the number of these errors can be the same in some cases, that does not necessarily mean that either the missing arcs or the extra arcs are the same in the three algorithms. At this stage, there are two possible suggestions: a) to check the quantitative performance of the three algorithms using a goodness of fit measure and/or b) to ask the human experts to grade, according to their knowledge, experience and beliefs, whether there are significant differences in the respective results given by the three algorithms. As a

matter of fact, even when a goodness of fit measure can be applied, there will surely be cases where such a metric does not provide the best result. Therefore, step b) above has to be used in order for the results given by above of these algorithms to be more robust. Psychological theories, such as the Power PC Theory presented in chapter 1, have tried to isolate where this prior human expert knowledge resides in order to automate it and build systems capable of performing as well as humans in tasks involving causal reasoning. Although the Power PC Theory has shown encouraging results, it is still very limited in scope mainly to problems that include only one rather than many interactive causes of a certain given effect.

In sum, the qualitative performance of Bayes9 seems to be comparable to those of Tetrad II and Power Constructor meaning that, at least, the three of them reach similar solutions in discovering the main relationships among the variables involved in a given problem. Furthermore, as stated in chapter 1 and as will be seen more clearly in chapter 8, this kind of tool, namely, Bayesian networks, can be very useful as a support aid for diagnostic and prognostic problems as well as decision-making problems. Thus, this is also a cautionary note on how to use such graphical models; i.e., rather than trying to replace human experts, they are aimed at helping them in situations where a large number of variables and cases need to be considered.

7.4 Goodness of fit.

Table in figure 7.2 shows the quantitative comparison, in terms of the minimum description length (MDL) criterion, among Bayes9, Tetrad II and Power Constructor.

Databases	MDL for Golden Nets	MDL for Tetrad II	MDL for PowCons	MDL for Bayes9
Alarm	40827.19	40937.34	53821.66	42403.26
Child	36809.88	37052.01	37862.63	36818.71
Diagcar	24874.50	25301.88	25367.25	25305.17
Asia	979.28	1003.07	978.44	1045.04
Sewell and Shah	19779.70	19770.21	19931.82	19779.70

Figure 7.2: Comparison of the MDL results given by Bayes9, Tetrad II and Power Constructor

Let us recapitulate some of the results of figure 7.2. In the ALARM database, a peculiar situation occurs: Bayes9 and Power Constructor have the same number of missing and extra arcs. The actual differences are a missing arc from node 23 to node 35 in Bayes9 and a missing arc from node 27 to node 33 in Power Constructor (see figure 4.2 of chapter 4 and table of figure 7.1). Also, the only one arc added by the two procedures is the same with the difference that in Bayes9 is an arc from node 35 to node 13 and in Power Constructor is from node 13 to node 35. These two differences make Bayes9 behave in a better way (according to the MDL criterion). Regarding the behaviour of Tetrad II, we can see from figure 7.1 that this procedure seems to overfit the data more than Power Constructor and Bayes9 (because of the number of extra arcs). However, according to figure 4.9 of chapter 4, there is a point in the overall behaviour of the MDL score where adding arcs helps to decrease such a score. Thus, it can be argued that this is what happens with this particular example.

In the CHILD database, Bayes9 is now the best classifier. Compared to Tetrad II, Bayes9 has 2 extra arcs instead of 3 produced by Tetrad II. Bayes9 adds the arcs from node e to node m and from node o to node l while Tetrad II adds the arc from node b to node m, from node b to node s and from node o to node l. These differences make Bayes9 the best of the three procedures in this dataset (see figure 4.3 of chapter 4 and table of figure 7.1).

In the DIAGCAR database, Tetrad II has again the best performance. Bayes9 has similar performance of that of Tetrad II with only two differences: one extra arc from node f to node q, which Tetrad II does not have and the common extra arc that these two algorithms share has a direction from node k to node p in Bayes9 and from node p to node k in Tetrad II (see figure 4.4 of chapter 4 and table of figure 7.1). Such slightly changes produce only a difference of 3.29 (MDL) of Bayes9 with respect to Tetrad II. Again, it is the decision of the experts that have to be taken into account to know whether that error represents a significant different performance between the two procedures.

Finally, in the SEWELL & SHAH database, there is only difference between the structure produced by Bayes9 and that produced by Tetrad II: the arc from PE to IQ in Bayes9 is reversed in Tetrad II. Once more, this kind of divergences can be assessed and, if it is the case, modified by the human experts.

In the next chapter, Bayes9 is tested and evaluated using real databases to check whether its performance as classifier is good enough to use it for real-world applications purposes.

Chapter 8

Applications

This chapter presents the performance of Bayes9 on a real-world database coming from the area of pathology. Such a performance is measured using different tests commonly used in the medical domain. Then, it compares Bayes9's performance against the performance of five different classification methods and against Tetrad II and Power Constructor as well and discusses the advantages and disadvantages of using Bayes9 as a potential decision-support system in this particular medical area.

8.1 Background of a real-world database from medicine.

Through the previous chapters, it has been claimed that Bayesian networks give a suitable, sound and consistent framework to manage uncertainty, knowledge and beliefs in an intuitively understandable way. Such a claim has been supported by the experimental results shown in those chapters. However, most of these results have been obtained by using simulated databases. Thus, showing that such a model can also potentially work in real-world datasets can increase the utility of this graphical model.

The real-world database used here comes from the field of pathology and has to do with the cytodiagnosis of breast cancer using a technique called fine needle aspiration of the breast lesion (**FNAB**) (Cross, Dube et al. 1998; Cross, Downs et al. 2000; Cross, Stephenson et al. 2000), which is the most common confirmatory method used in the United Kingdom for this purpose (Cross, Downs et al. 2000). This dataset has been kindly given by Dr. Simon S. Cross, Clinical Senior Lecturer at the University of Sheffield and Honorary Consultant Histopathologist to Central Sheffield University Hospitals NHS Trust.

Such a technique involves a process where a syringe sucks cells from breast lesions using a fine bore needle similar to those used for blood samples. Once this is done, these cells are transferred to a transport solution and sent to the pathology laboratory for a

microscopic study carried out by a trained cytopathologist (Cross, Downs et al. 2000). The time it normally takes to a medical doctor to become an independent practising cytopathologist is about 5 years as a minimum. This fact can give an indication of the very complex learning process which medical doctors have to pass through. It is mainly for this reason that machine learning methods for decision-making may have two potential applications: to accelerate the training process of learning by providing guidance to the trainee on what features are the most important ones to look at; and to compare the final results to those of the trainee or even of the expert so that the decision (whether the sample taken indicates a benign or a malignant output) can be made on more robust criteria (qualitative and quantitative).

During the cytodiagnosis process, it is very important to avoid the diagnosis of false positives. It is not a desired characteristic to have any false positives diagnosed since this will lead to the surgical removal of healthy breast(s); a situation that has to be always avoided. This has to do with a term called **specificity**, which refers to the ability to correctly identify those patients who do not have the disease. Moreover, there is also a trade-off (see figure 8.1) between specificity and **sensitivity** (the ability to correctly identify those patients who actually have the disease) since, whenever the specificity is set high, there may be women with breast cancer that is not detected. In sum, in this case of the diagnosis of breast cancer by means of the fine needle aspirate technique, it is required to have as much as 100% specificity. The sensitivity, according to the results shown in (Cross, Dube et al. 1998; Cross, Downs et al. 2000), should be around 80% or greater.

The idea is then to use an algorithm for mining the breast cancer dataset previously mentioned in order to possibly identify the main cytological features that may cause breast cancer. This database contains 692 consecutive specimens of FNAB received at the Department of Pathology, Royal Hallamshire Hospital in Sheffield during 1992-1993 (Cross, Downs et al. 2000). 11 independent variables and 1 dependent variable form part of such a dataset. The independent variables are: age, cellular dyshesion, intracytoplasmic lumina, “three-dimensionality” of epithelial cells clusters, bipolar “naked” nuclei, foamy macrophages, nucleoli, nuclear pleomorphism, nuclear size, necrotic epithelial cells and

apocrine change. All these variables, except age, are dichotomous taking the values of “true” or “false” indicating the presence or absence of a diagnostic feature. Variable age was actually sorted into three different categories: 1 (up to 50 years old), 2 (51 to 70 years old) and 3 (above 70 years old). The dependent variable “outcome” can take on two different values: benign or malignant. In the case of a malignant outcome, such a result was confirmed by a biopsy (where available). For more details about the variables of this dataset, the reader is referred to (Cross, Dube et al. 1998; Walker, Cross et al. 1999; Cross, Downs et al. 2000).

8.2 Tests for measuring accuracy.

The main idea of using Bayes9 as an induction algorithm is, as said in chapter 1, to build a **classifier** capable of discovering some relevant information hidden in this dataset that helps understand the domain better and possibly reveal new knowledge. Bayes9 is an unsupervised learning algorithm because such a procedure is not explicitly told which variable represents the class or dependent variable (Friedman, Geiger et al. 1997; Frey 1998; Han and Kamber 2001).

If the Bayesian network resultant from running Bayes9 on the breast cancer dataset is built accurately, then the output (malignant or benign) of each new case (new patient) will be **classified** correctly most of the time. In the data mining terminology, it is commonly accepted that the term **classification** refers to the prediction of discrete, nominal or categorical variables’ values while the term **prediction** refers to the prediction of continuous variables’ values (Han and Kamber 2001). In this particular case, the class variable “outcome” is a categorical one (malignant or benign) so that the prediction task is referred to as classification.

There exist different methods to test the performance, in terms of accuracy, of any classifier such as **holdout**, **cross-validation** and **bootstrap** (Kohavi 1995a; Han and Kamber 2001). In this chapter, the holdout and cross-validation methods are used to check the accuracy of the model built by Bayes9. Moreover, five more tests, commonly used in the medical domain, are presented in this analysis: sensitivity, specificity, **predictive value**

of a **positive result**, **predictive value** of a **negative result** and the so-called receiver operating characteristic (**ROC**) curves.

In the holdout method, the common practice is to randomly partition the data in two different mutually exclusive samples: the training and the test sets. The test set is also called the holdout set. The size of the training set is usually 2/3 of the data and the remaining 1/3 of the data corresponds to the size of the test set. The accuracy of the model built by the induction algorithm (in this case Bayes9) is the percentage of the test set cases that are classified to the correct category they belong by the model. In other words, the class to which each case in the test set truly belongs is compared to the prediction, made by the model, for that same instance. The reason for partitioning the data this way is to avoid overfitting the data since, if the accuracy of the model were estimated taking into account only the training set, then some possible anomalous features which are not representative of the whole data could be included in this subset and the estimate may possibly not reflect the true accuracy. Thus, a test set has to be selected and used to test the robustness of the model, in the sense of making correct classifications given noisy data.

Thus, the overall number of correct classifications divided by the size of the test set is the accuracy estimate of the holdout method. Formula 8.1 (Kohavi 1995a) shows how to calculate the accuracy using the holdout approach.

$$acc_h = \frac{1}{h} \sum_{(v_i, y_i) \in D_h} \delta(I(D_t, v_i), y_i) \quad (8.1)$$

where $I(D_t, v_i)$ denotes the instance v_i built by inducer I on data set D_t (the training set) which is assigned the label y_i and tested on the test set D_h ; h is the size of the test set.

$\delta(i,j)=1$ if $i=j$ and 0 otherwise. This means that the loss function used for calculating the accuracy in the holdout method is a 0/1 loss function, which considers equal misclassification costs.

The variance of the holdout method is estimated as follows (Kohavi 1995a):

$$Var = \frac{acc \times (1 - acc)}{h} \quad (8.2)$$

where h is the size of the test set.

Another common approach, which is also used to measure classification accuracy and which will also be used here to calculate the accuracy of the classifier induced by Bayes9, is the **k-fold** cross-validation. As Kohavi (1995a) and Han and Kamber (2001) describe, in the k-fold cross-validation method, the complete dataset is randomly partitioned in k mutually exclusive subsets (called also the folds) D_1, D_2, \dots, D_k of approximately equal size. The induction algorithm is trained and tested k times in the following way: in the first iteration, this algorithm is trained on subsets D_2, \dots, D_k and tested on subset D_1 ; in the second iteration, the algorithm is trained on subsets D_1, D_3, \dots, D_k and tested on subset D_2 and so on. The overall number of correct classifications from the k iterations divided by the size of the complete dataset is the accuracy estimate of the k-fold cross-validation method. Formula 8.3 shows how to calculate the accuracy of the cross-validation approach.

$$acc_{cv} = \frac{1}{n} \sum_{(v_i, y_i) \in D} \delta(I(D \setminus D_{(i)}, v_i), y_i) \quad (8.3)$$

where $I(D \setminus D_{(i)}, v_i)$ denotes the instance v_i built by inducer I on data set $D \setminus D_{(i)}$, which is assigned the label y_i and tested on the test set $D_{(i)}$; n is the size of the complete dataset D.

$\delta(i, j) = 1$ if $i = j$ and 0 otherwise. As in equation 8.1, this means that the loss function used for calculating the accuracy in the cross-validation method is a 0/1 loss function, which considers equal misclassification costs.

Equation 8.4 shows the formula for the estimation of the variance in this method (Kohavi 1995a):

$$Var_{cv} = \frac{acc_{cv} \times (1 - acc_{cv})}{n} \quad (8.4)$$

where n is the size of the complete dataset D .

The remaining five tests commonly used in medicine, sensitivity, specificity, predictive value of a positive result, predictive value of a negative result and receiver operating characteristic (ROC) curves, will be briefly explained below.

Most medical test cannot distinguish between a patient with disease and a patient without disease with 100% of accuracy, as figure 8.1 shows in the overlap area.

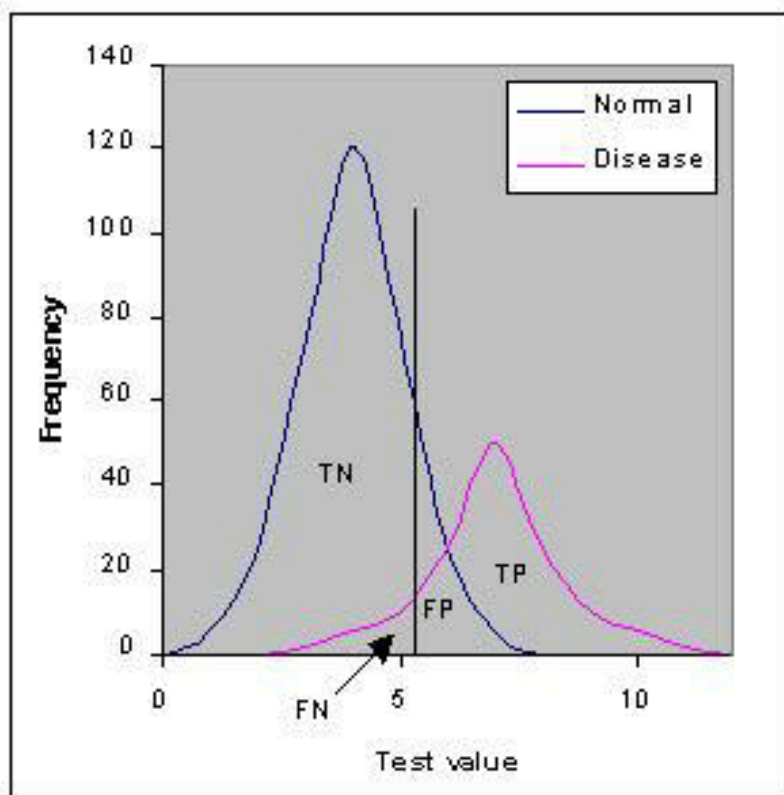


Figure 8.1: Graph showing the number of patients with and without disease according to the value of the diagnostic test

In our specific case, the fine needle aspiration of the breast lesion (FNAB) is used as a test to diagnose if a patient has breast cancer. The abbreviations TN, TP, FN and FP in

this figure stand for true negatives (not_cancer samples correctly classified as such), true positives (cancer samples correctly classified as such), false negatives (cancer samples incorrectly classified as not_cancer) and false positives (not_cancer samples incorrectly classified as cancer) respectively.

The sensitivity determines the percentage of false negatives results and specificity refers to the percentage of false positives results. The predictive value of a positive result indicates the probability that a patient with a positive result has cancer and the predictive value of a negative result indicates the probability that a patient with a negative result does not have cancer. As can be seen from figure 8.1, the vertical line represents a cut point between the curves representing both true negatives and true positives cases. If this cut point is moved to either direction, sensitivity and specificity values will change (while one decreases the other increases and vice versa). Thus, as said in section 8.1, the trade-off between sensitivity and specificity can be clearly observed.

Equations 8.5, 8.6, 8.7 and 8.8 show the formulas to calculate sensitivity, specificity, predictive value of a positive result and predictive value of a negative result respectively.

$$sensitivity = \frac{t_pos}{t_pos + f_neg} \quad (8.5)$$

$$specificity = \frac{t_neg}{t_neg + f_pos} \quad (8.6)$$

$$PV+ = \frac{t_pos}{t_pos + f_pos} \quad (8.7)$$

$$PV- = \frac{t_neg}{t_neg + f_neg} \quad (8.8)$$

where t_pos , t_neg , f_pos and f_neg stand for true positives, true negatives, false positives and false negatives respectively (all of which were explained above).

Having presented the formulas for sensitivity and specificity, the accuracy of the test can also be calculated based on those measures as formula of equation 8.9 shows.

$$accuracy = sensitivity \frac{pos}{(pos + neg)} + specificity \frac{neg}{(pos + neg)} \quad (8.9)$$

where pos is the number of positive cancer samples ($t_pos + f_neg$) and neg is the number of negative cancer samples ($t_neg + f_pos$).

It is also possible to represent the degree of the accuracy or the **power** of a certain test (in this case the FNAB test) using a well-known technique called receiver operating characteristic (ROC) curves. A ROC curve represents graphically the trade-off between the false positive rate ($1 - specificity$) on the X-axis and the true positive rate (sensitivity) on the Y-axis for every possible cut point (represented in figure 8.1 by the vertical line). Figure 8.2 shows an example of three ROC curves.

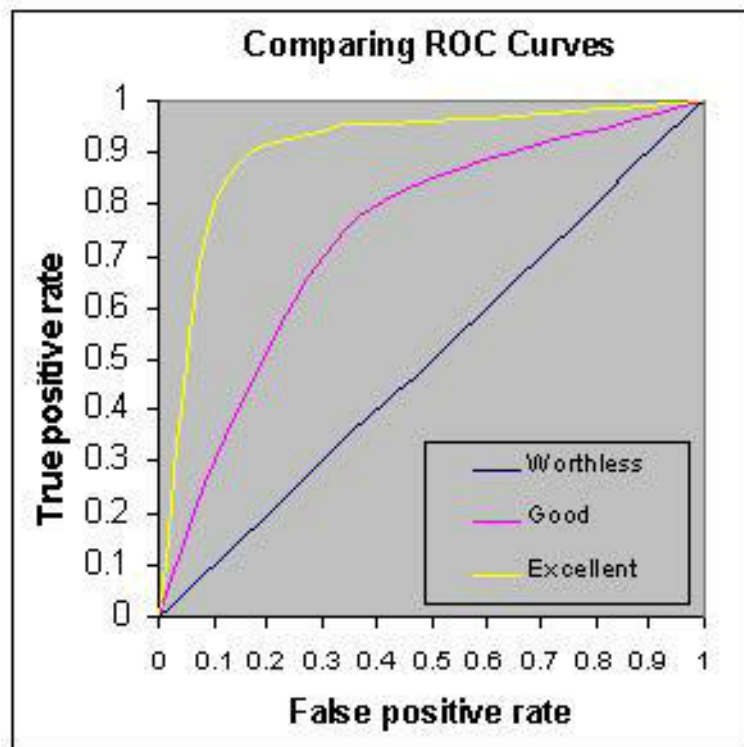


Figure 8.2: Comparing the power of three diagnostic tests

Figure 8.2 shows how accurate three different tests are. The area under the ROC curve measures this accuracy; an area of 0.5 represents a worthless test while an area of 1 represents a perfect test. Thus, in general, the **larger** the area under the curve, the **better** the diagnostic test. The diagonal line (blue) represents a worthless test in which between the 50% and the 60% of the cases are classified correctly. In other words, this same percentage range of correct classifications could be perfectly obtained by tossing a coin; this comparison indicates the poor performance, power or accuracy of that test. The curve in the middle (pink) represents a good test in which between the 80% and the 90% of the cases are classified correctly. Finally, the curve with the biggest area (yellow) represents an excellent test in which between the 90% and the 100% of the cases are classified correctly. In sum, the area under the curve measures **discrimination**, which refers to the ability of the test to classify correctly subjects with disease and subjects without disease.

8.3 Experimental results of Bayes9.

For the holdout approach, the breast cancer dataset (692 cases) was randomly partitioned into 462 cases (approx. 2/3 of the data) for training and the remaining 230 cases (approx. 1/3 of the data) for testing. The result of running Bayes9 on the 462 case training set is shown in figure 8.3.

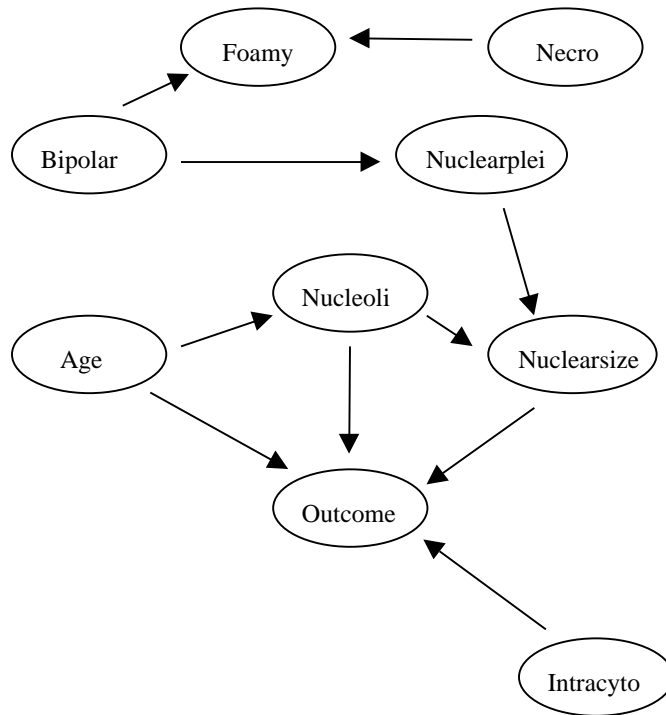


Figure 8.3: The result of running Bayes9 on the 462 case training set of the breast cancer database

As figure 8.3 shows, only 4 variables out of 11 are considered directly relevant by Bayes9 to explain the variable outcome: age, intracytoplasmic lumina, nucleoli and nuclear size. Furthermore, there are three variables, three dimensional cell clusters, cellular dyshesion and apocrine change, that are not relevant according to the information they provide (equations 4.4 and 4.5 of chapter 4) so they do not even appear in figure 8.3.

Table of figure 8.4 shows the results for accuracy, sensitivity, specificity, predictive value of a positive result (PV+) and predictive value of a negative result (PV-) for the model of figure 8.3. For all the tests, except for accuracy, their respective 95% confidence intervals (CI) are also shown in parentheses. For the accuracy test, the standard deviation is also shown.

Tests	Model proposed by Bayes9 (figure 8.3)
Accuracy	92% \pm 1.78%
Sensitivity	88% (81-95)
Specificity	94% (90-98)
PV+	89% (82-96)
PV-	93% (89-97)

Figure 8.4: Results of accuracy, sensitivity, specificity, predictive value of a positive result and predictive value of a negative result given by Bayes9 for the holdout method. 95% confidence intervals are shown in parentheses

For the cross-validation approach, following the recommendations of Friedman et al. (Friedman, Geiger et al. 1997) regarding the number of variables and the sample size of the dataset, the number of folds chosen was 5. Thus, we refer to such a cross-validation as 5-fold cross-validation. The breast cancer dataset (692 cases) was randomly partitioned in 5 folds of approximately equal size (3 folds of 138 cases and 2 folds of 139 cases). Bayes9 was trained and tested according to the cross-validation method explained in section 8.2. The results of running Bayes9 on the 5 different folds are shown in figures 8.5, 8.6, 8.7, 8.8 and 8.9.

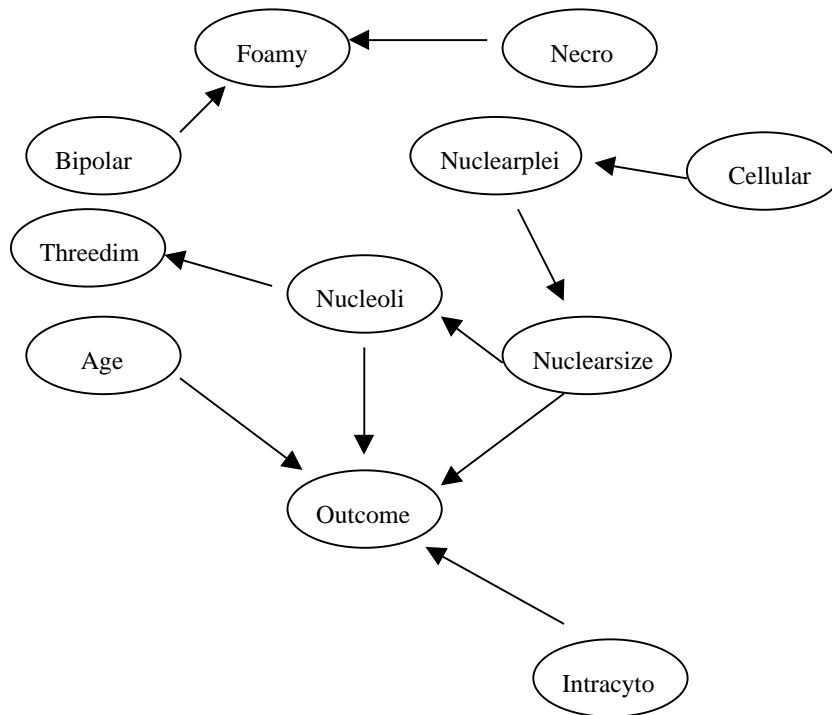


Figure 8.5: The result of running Bayes9 on the first fold of the breast cancer database (total number of cases is 554)

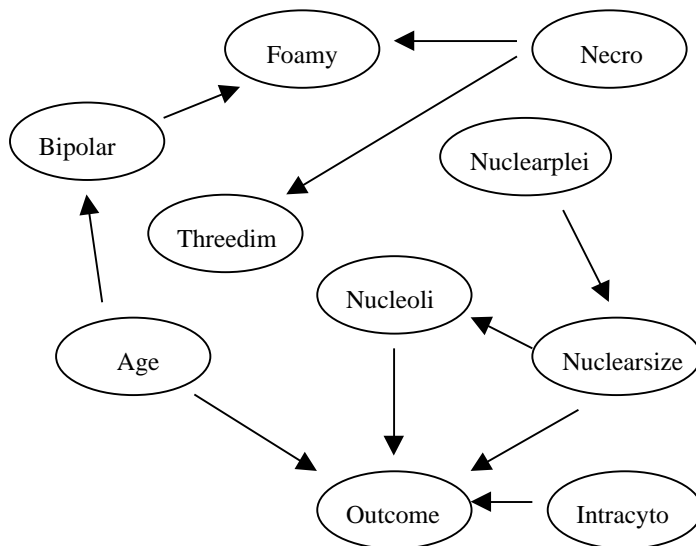


Figure 8.6: The result of running Bayes9 on the second fold of the breast cancer database (total number of cases is 554)

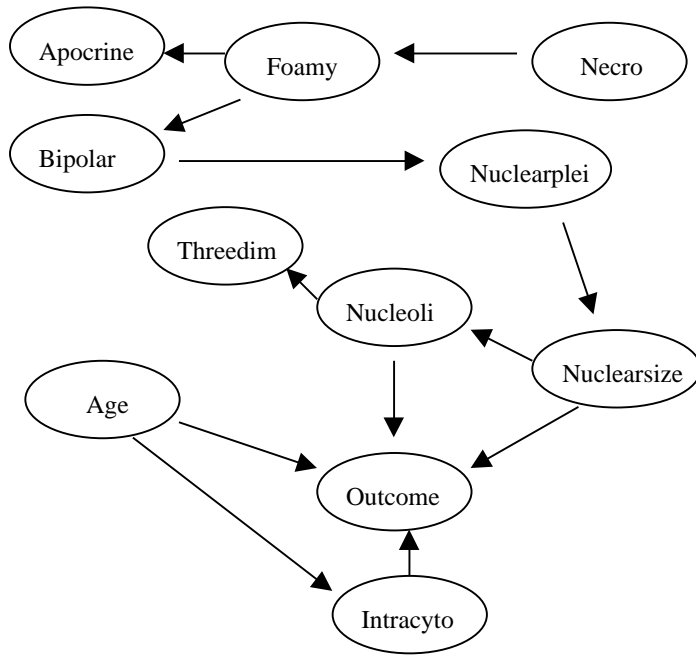


Figure 8.7: The result of running Bayes9 on the third fold of the breast cancer database (total number of cases is 554)

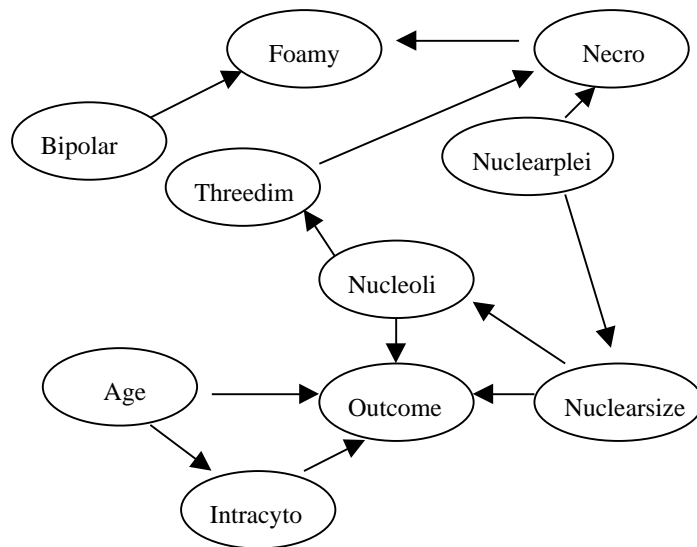


Figure 8.8: The result of running Bayes9 on the fourth fold of the breast cancer database (total number of cases is 553)

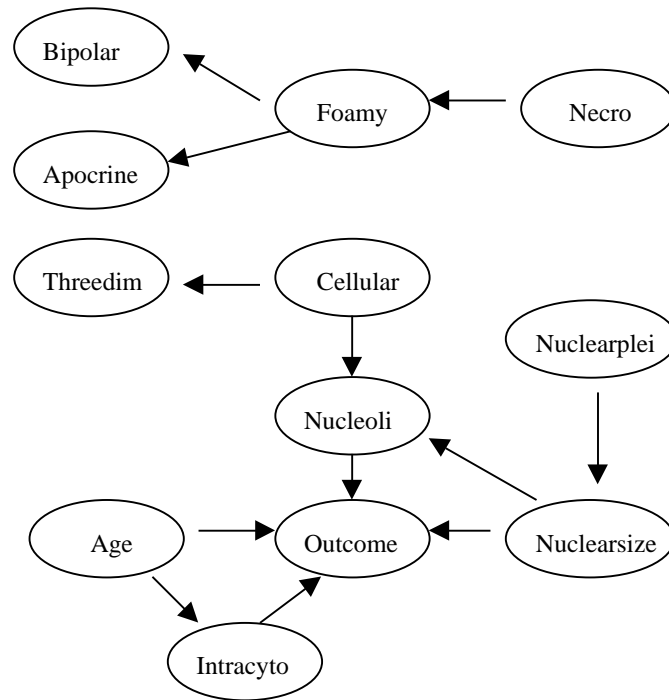


Figure 8.9: The result of running Bayes9 on the fifth fold of the breast cancer database (total number of cases is 553)

Figures 8.5, 8.6, 8.7, 8.8 and 8.9 show consistent results regarding the variables with a direct arc to the variable outcome; in all these Bayesian networks these variables are the same: age, nucleoli, intracytoplasmic lumina and nuclear size. This important finding suggests that it is these variables which carry the most relevant information to determine if the patient has or does not have breast cancer.

Table of figure 8.10 shows the overall performance of Bayes9 as a classifier using the cross-validation method. For all the tests, except for accuracy, their respective 95% confidence intervals (CI) are also shown in parentheses. For the accuracy test, its standard deviation is also shown.

Tests	Average of models proposed by Bayes9 (figures 8.5-8.9)
Accuracy	95% \pm 0.82
Sensitivity	90% (80-98)
Specificity	98% (95-100)
PV+	96% (91-100)
PV-	95% (90-99)

Figure 8.10: Overall performance of accuracy, sensitivity, specificity, predictive value of a positive result and predictive value of a negative result given by Bayes9 for the 5-fold cross-validation method. 95% confidence intervals are shown in parentheses

Table of figure 8.11 presents the values of the different tests given by Bayes9 trained on $D \setminus D_t$ and tested on D_t (where $t \in \{1, 2, 3, 4, 5\}$).

Tests	CV1	CV2	CV3	CV4	CV5
Accuracy	92% \pm 2.30	93% \pm 2.17	98% \pm 1.19	96% \pm 1.66	94% \pm 2.01
Sensitivity	86% (77-95)	85% (75-95)	92% (83-100)	94% (88-100)	89% (79-98)
Specificity	96% (92-100)	98% (95-100)	100%	98% (95-100)	97% (93-100)
PV+	94% (88-100)	95% (89-100)	100%	96% (91-100)	93% (85-100)
PV-	91% (85-97)	93% (87-98)	97% (94-100)	97% (93-100)	95% (90-99)

Figure 8.11: Results of accuracy, sensitivity, specificity, predictive value of a positive result and predictive value of a negative result given by Bayes9 for the cross-validation method for each of the 5 folds. 95% confidence intervals are shown in parentheses

Finally, once having presented the results for accuracy, sensitivity, specificity, PV+ and PV- for both the holdout and the 5-fold cross-validation methods, the result of the accuracy given by the ROC curve (which is the same for these both approaches; see figures 8.3 and 8.5 to 8.9) is presented in figure 8.12.

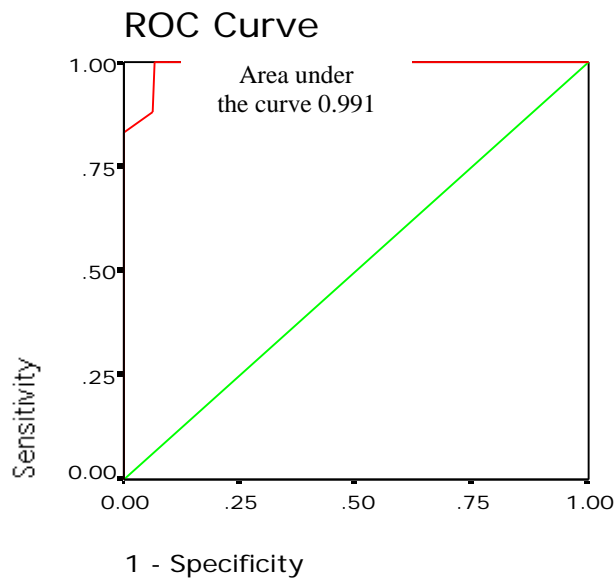


Figure 8.12: ROC curve for the holdout and 5-fold cross-validation methods

8.4 Discussion of the results.

According to the medical literature, all 10 defined observations mentioned in section 8.1 are considered relevant for the cytodiagnosis of FNAB (Cross, Downs et al. 2000). Furthermore, variable age is also considered relevant for such a purpose as it provides useful information for making the final diagnosis (Cross, Downs et al. 2000). Because of the very nature of procedure Bayes9, in the sense of parsimony, it tends to reduce the number of relevant variables to explain the class variable (outcome), as figures 8.3, 8.5, 8.6, 8.7, 8.8 and 8.9 show. According to the results shown in table of figure 8.4, it seems that this choice of variables, using the holdout approach, produces a reasonably acceptable overall performance of the classifier induced by Bayes9. While it is true that a value of 100% was not achieved either in the specificity test or in the PV+ (only 94% and 89% were obtained respectively), good values of sensitivity (88%), PV- (93%) and accuracy (92%) were accomplished by this same classifier. Hence, the trade-off between sensitivity and specificity appears to be more or less adequate.

Now, using the 5-fold cross-validation approach, the results (see table of figure 8.10) given by the average performance of the classifiers induced by Bayes9 (figures 8.5 to 8.9) seem to be more accurate than those produced by applying the holdout method (see table of figure 8.4). Table of figure 8.13 can help to visualise such an improvement. However, because the true accuracies of a real-world dataset cannot be determined due to the lack of knowledge about the target concept, the values of the tests computed by the holdout method are taken as the “true” accuracies (Kohavi 1995a; Kohavi 1995b). The target concept refers to the selection of relevant features that determine a specific value of the dependent variable (in our case whether the patient has cancer or not). Because in this medical domain (as in every other real-world domain) the target concept is not deterministic, then it is said that the target concept is unknown and its accuracy value can only be approximated. Thus, under this view, the results given by using the 5-fold cross-validation approach are somewhat optimistically biased rather than more accurate: from 92% to 95% in accuracy, from 88% to 90% in sensitivity, from 94% to 98% in specificity, from 89% to 96% in PV+ and from 93% to 95% in PV-. This comparison is quite useful as it helps us to bear in mind the behaviour of cross-validation when used to measure accuracy or for model selection.

Tests	Holdout	5-fold CV
Accuracy	92% \pm 1.78%	95% \pm 0.82
Sensitivity	88% (81-95)	90% (80-98)
Specificity	94% (90-98)	98% (95-100)
PV+	89% (82-96)	96% (91-100)
PV-	93% (89-97)	95% (90-99)

Figure 8.13: Comparison of the overall performance between the classifiers induced by Bayes9 using the holdout and 5-fold cross-validation methods

What is also very important to notice is that, although the classifier produced by the holdout method (figure 8.3) and those produced by the 5-fold cross-validation method (figures 8.5 to 8.9) have in common the same 4 variables (age, intracytoplasmic lumina, nucleoli and nuclear size) to explain the class variable (outcome), the classifiers induced by

using the latter approach seem to overfit the data. This is because training and test cases are interchanged, as equation 8.3 indicates, which leads to the selection of a model that overfits the data (Heckerman 1998).

In order to evaluate more deeply the performance of the classifier induced by Bayes9 using the holdout approach (figure 8.3), such a classifier is compared to the performances of other methods of classification: human performance, logistic regression, decision trees, multilayer perceptron neural networks and adaptive resonance theory mapping neural networks (Cross, Downs et al. 2000). In fact, Cross et al. (Cross, Downs et al. 2000) report results on all these classification methods using the holdout approach (except for the human performance) so that the comparison among them can be made against a more robust and uniform criterion.

8.4.1 Human performance vs. Bayes9

Table of figure 8.14 shows a summary of the results given by Bayes9 and the Human performance. Human performance refers to that of the expert human pathologist. The results of the 4 tests (sensitivity, specificity, PV+ and PV-) for the human performance were obtained by using the whole 692 case database.

Tests	Bayes9	Human performance
Sensitivity	88% (81-95)	82% (77-87)
Specificity	94% (90-98)	100%
PV+	89% (82-96)	100%
PV-	93% (89-97)	92% (89-94)

Figure 8.14: Results of sensitivity, specificity, predictive value of a positive result and predictive value of a negative result given by Bayes9 and Human performance. 95% confidence intervals are shown in parentheses

As figure 8.14 clearly shows, the requirement of not having any false positives diagnosed is completely achieved by the human expert. This overspecialisation is not

explicitly programmed in Bayes9 but it can be argued that this system could probably be taught to achieve 100% of specificity by manually adding some other variables that might prove useful to better classify every case. Thus, a combination of prior knowledge and data may be used to that purpose so the resultant system becomes more accurate (Heckerman, Geiger et al. 1994; Kohavi 1995b; Chickering 1996).

For the values of sensitivity and PV-, Bayes9 seems to improve upon the performance of the human expert. However, this probably happens because variable age was not provided to the human expert when he was making the final diagnosis. Therefore, it can be argued that if such a variable had been provided, then the expert would have produced better results for these tests.

In order to make their final diagnoses, cytopathologists follow a process which is not very well known to date and can only be partially explained in terms of pattern recognition with occasional use of heuristic logic (Cross, Downs et al. 2000). As said in section 1.1 of chapter 1, the idea when building an expert system is to have a computational tool that reaches conclusions similar of those reached by human experts no matter which process the expert system is following. Hence, Bayes9 tries to imitate the human performance by using the theory behind learning Bayesian networks from data. Besides, as Cross et al. (Cross, Downs et al. 2000) point out, all the features coded in the breast cancer dataset used by Bayes9 were coded by the expert cytopathologist who carried out most of the processing that is probably required to solve the diagnostic problem. So, if this is true, then there is little work left to any classifier that uses this dataset. In other words, the information provided in such a database is subjective rather than objective. Thus, it would be necessary to use image analysis techniques that could extract objective measures from the sample raw digitalised images. Of course, doing this would involve having sophisticated and expensive equipment that probably some hospitals would not want to spend on.

In this respect, an algorithm such as Bayes9 that builds Bayesian networks from data, could easily be implemented in a palmtop computer. This could be carried in the

cytopathologist pocket helping them to make virtually instantaneous predictions on new arriving cases. This practice can also be recommended to keep the consistency of the results to a maximum since such consistency can be diminished due to an overworked or stressed human.

In sum, although the performance of Bayes9 does not equal that of the human expert, in the sense of specificity, it still may provide some useful insight and help us to understand the data better as it represents graphically the relationships among all the variables involved perhaps providing new knowledge about the domain. Bayes9 might also be useful as a support tool reinforcing the decisions made by the human expert.

8.4.2 Logistic regression vs. Bayes9

The second method to compare Bayes9 with is that of logistic regression, reported also in Cross et al. (Cross, Downs et al. 2000). The sizes of the training set and the test set used by the logistic regression are 462 and 230 cases respectively. Table of figure 8.15 shows a summary of the results given by Bayes9 and the logistic regression.

Tests	Bayes9	Logistic Regression
Sensitivity	88% (81-95)	94% (89-99)
Specificity	94% (90-98)	95% (90-97)
PV+	89% (82-96)	87% (80-95)
PV-	93% (89-97)	97% (95-99)

Figure 8.15: Results of sensitivity, specificity, predictive value of a positive result and predictive value of a negative result given by Bayes9 and logistic regression. 95% confidence intervals are shown in parentheses

According to Cross et al. (Cross, Downs et al. 2000), because of its ease of use, implementation and interpretability, logistic regression is the standard statistical technique to which the performance of other classifiers or support systems should be compared. In the results presented there (Cross, Downs et al. 2000), the main variables to predict the

malignancy of the outcome are increasing age, presence of intracytoplasmic lumina, presence of three dimensional cell clusters, multiple nucleoli and nuclear pleiomorphism. As can be seen in figure 8.3, three of those variables, age, intracytoplasmic lumina and nucleoli, are common in both approaches to explain the outcome. Furthermore, logistic regression identifies one variable as being relevant for predicting a benign diagnosis: apocrine metaplasia. Again, comparing this result to that of figure 8.3, it can be seen that apocrine metaplasia does not even appear in the identification of relevant variables by Bayes9. Thus, it can be argued that the overall worse results given by Bayes9, compared to those by the logistic regression, might be because of two reasons: in a Bayesian network, **different value** combinations of the same set of variables that have direct arc to the outcome determine whether the result is benign or malignant whereas in the logistic regression **different** sets of **variables** determine this malignancy or benignancy; the second reason being that Bayes9 makes extensive use of conditional independence, therefore its very nature is to keep the least number of variables with a direct arc to any other. For instance, in figure 8.3, nuclearpleiomorphism does not have an arc to outcome because its influence is mediated by the variable nuclear size, i.e., nuclearpleiomorphism is conditionally independent of outcome given nuclear size, as the d-separation criterion dictates. Hence, this independence criterion stops variable nuclearpleiomorphism from directly influencing the behaviour of the outcome while this situation does not happen with the logistic regression. Furthermore, the specificity value in the logistic regression is only 1% more than in the Bayesian network built by Bayes9. PV+, which is also very important in this domain, is better in Bayes9 for 2%. So, in terms of these two tests, it can be said that the performance of both classifiers is roughly the same.

But once again, prior knowledge can be incorporated so that the Bayesian network constructed by Bayes9 can be explicitly told which variables need to be considered relevant to explain the behaviour of the outcome. Also, since logistic regression is a supervised method (in the sense that it needs explicitly to be told what the class variable is), it does not show the interactions among the independent variables whereas the Bayesian network model of figure 8.3 indeed does. As said previously, this significant advantage might serve in gaining some better understanding of the problem being modelled.

An important disadvantage that may be related to interobserver variability appears when using logistic regression (Cross, Stephenson et al. 2000). In one study carried out by this group of researchers, the same dataset that we used in this experiment (692 cases), which was collected by a single observer, was also used in their study to train and test two different decision support systems; one of them being logistic regression. In order to test the robustness of the system, in terms of classification accuracy, they also used a 323 test case dataset collected not by one but by multiple observers. Thus, 462 cases of the 692 case dataset were used to train the system, then the remaining 230 cases of these 692 cases were used to test the system and then 323 case dataset was also used to test the system. What they found is very interesting: the specificity and PV+ values when using the test set of 230 are 96% and 91% respectively while these same values when using the 323 test set reduced dramatically to 85% and 78% respectively. Moreover, the sensitivity and PV- values when using this latter dataset dropped too: from 91% to 87% for the sensitivity and from 96% to 92% for the PV-. They argue (Cross, Dube et al. 1998; Cross, Downs et al. 2000; Cross, Stephenson et al. 2000) that the main reason might well be the interobserver variability regarding the identification and codification of the 10 defined observations used in this study. We tested Bayes9 with this same 323 case dataset to evaluate such a claim. Bayes9 presents a similar situation but this time regarding only the sensitivity and PV- instead of specificity and PV+: the sensitivity and PV- when using the 230 case test set are 88% and 93% respectively whereas these same values when using the 323 case test set reduced dramatically to 72% and 82% respectively. However, in contrast with logistic regression, the values for specificity and PV+ in the 230 case test set and in the 323 case test set are practically the same: specificity of 94% and PV+ of 89% in the former dataset and specificity of 94% and PV+ of 90% in the latter dataset. In sum, the values of the four different tests given by logistic regression change considerably when using the 323 case test set while the values of two tests (sensitivity and PV-) are the only values that change considerably when using this same 323 case test dataset. This fact seems to indicate that, because of the nature of the Bayesian network approach presented here, such an approach is less vulnerable to this interobserver variability.

In any case, this variability may be the result of a forced dichotomization of such features, which might very probably lead to some information loss when converting continuous variables into discrete ones. Thus, the codification of the data might vary from observer to observer because of the subjectivistic nature of feature identification such as different internal threshold settings for each of these observations. Furthermore, the recognition of these features may depend on the level of diligence when looking at the specimens and, because of that, there could be some problems with the reproducibility of results among different observers and even within the same single observer at different times (Hamilton, Anderson et al. 1994; Cross, Dube et al. 1998). Thus, it also can be argued that this interobserver variability may be reduced if using a measure scale that allows codifying the features with more power and richness than that of a binary coding.

In favour of the Bayesian network approach, it seems that such a graphical model is less vulnerable to this variability (Hamilton, Anderson et al. 1994; Cross, Dube et al. 1998). This could be perhaps because of the way, when using either the classic approach or a combination of prior knowledge and data, Bayesian networks are built and because of the very nature of such networks to handle and combine knowledge and uncertainty. In either approach, the prior marginal and conditional probabilities matrices associated to each node are initialised by the human experts who need to agree most of time, according to their knowledge and beliefs, in how to fill the probability values for such matrices. Moreover, as Hamilton et al. describe in their experiment (Hamilton, Anderson et al. 1994), they had a single cytopathologist enter the evidence of the observations in the form of likelihood ratios. They tested the reproducibility of the values for these ratios relabelling the 40 case set of breast aspirates and presenting them again to the same cytopathologist who was blind to the previous likelihood ratio values and the original diagnosis. Although there were some differences between the two sets of likelihood vectors, this did not affect significantly the final classification (which was highly reproducible). Therefore, they showed in that study that it is not a single feature which gives decisive evidence for the outcome but cumulative evidence provided by a combination of them. Such an important finding suggests the higher stability of a decision system based on Bayesian networks, which leads of course to a less vulnerable system due to interobserver variability. In order to further reduce this variability,

it is also possible to build Bayesian networks and their associated probability tables not from the cytopathologists themselves but by using more objective measures such as those coming from raw digitalised images, as mentioned above.

8.4.3 Decision trees vs. Bayes9

The third method to compare Bayes9 with is that of decision trees, built by the well-known C4.5 program (Winston 1992; Kohavi 1995a; Kohavi 1995b; Friedman, Geiger et al. 1997), which is considered one of the state-of-the-art classifiers (Friedman, Geiger et al. 1997), and which is reported in Cross et al. (Cross, Downs et al. 2000). The sizes of the training set and the test set used by C4.5 are the same, 462 and 230 cases respectively. Table of figure 8.16 shows a summary of the results given by Bayes9 and C4.5.

Tests	Bayes9	C4.5
Sensitivity	88% (81-95)	95% (89-99)
Specificity	94% (90-98)	93% (90-97)
PV+	89% (82-96)	87% (80-95)
PV-	93% (89-97)	98% (95-99)

Figure 8.16: Results of sensitivity, specificity, predictive value of a positive result and predictive value of a negative result given by Bayes9 and C4.5. 95% confidence intervals are shown in parentheses

Figure 8.16 shows divided results: while Bayes9 gives more accurate results on specificity and PV+, C4.5 does on sensitivity and PV-. However, the better results given by Bayes9 over C4.5 are not as significant as those when C4.5 performs better, as can be seen from this figure as well. These two approaches, namely, Bayes9 and C4.5, have in common 4 variables to explain the outcome: nuclear size, intracytoplasmic lumina, nucleoli and age. Three more variables, apocrine change, bipolar naked nuclei and three dimensional cell clusters, are also used by C4.5 to determine the value of the outcome. Two of them, apocrine change and three dimensional cell clusters are not even considered by Bayes9 as being relevant to determine the outcome (see figure 8.3). Moreover, variable bipolar naked nuclei does not have a direct influence over the outcome because is mediated by two other

variables: nuclearpleiomorphism and nuclear size. Thus, bipolar naked nuclei variable is conditionally independent of outcome given nuclearpleiomorphism and nuclear size. Again, Bayes9 proposes a simpler model, regarding the number of variables, to explain the output and it seems to get results comparable to those obtained by the C4.5 algorithm. This latter classification procedure, as in the case of logistic regression, is a supervised method so the interactions among the independent variables cannot be shown or taken into account. As said before, due to the capability of Bayes9 in representing such interactions, doctors can gain a better understanding of the data and the domain.

As noted by Cross et al. (Cross, Downs et al. 2000), a good decision support system in a working clinical environment would be that which is easily used, implemented and which provides a good degree of explanation about how and why it reaches certain conclusions. A common advantage of both approaches may be that, either a decision tree or a Bayesian network, can be easily reproduced on a piece of paper, pinned to the wall in the work environment and could serve as a support tool to guide the trainees or even the experts in making their decisions. However, the same situation of interobserver variability mentioned previously affects the decision tree methodology (Cross, Dube et al. 1998; Cross, Downs et al. 2000). Cross et al. (Cross, Dube et al. 1998) found this variability when using the C4.5 program to induce a classification tree from a database collected by a single cytopathologist. In their study, a 200 case dataset (of a total of 600 cases) was used to train the algorithm and then the remaining 400 cases were used to test the performance of this same procedure. The values for sensitivity, specificity and PV+ are 95%, 93% and 89% respectively.

Then other 100 cases (50 positive and 50 negative) were used to test the performance of four trainees cytopathologists with the aid of the derived decision tree and without it. This group of researchers found something very interesting: the four trainees, with the help of the decision tree, did never get the 100% of specificity and PV+ while, without the aid of such a tree, three of them were able to get this 100% for both tests. This remarkable finding suggests that the trainees were not using the knowledge from the single observer from which the database was constructed. Hence, this phenomenon of

interobserver variability seems to be one of the causes that produced the poor performance by the four trainees. Thus, they conclude that this methodology (decision trees) might not be suitable as a decision support tool for this specific domain (Cross, Downs et al. 2000).

8.4.4 MLPs vs. Bayes9

The fourth method against which Bayes9 can be compared is the well-known framework of multilayer perceptron (**MLP**) neural networks (Cross, Downs et al. 2000; Han and Kamber 2001). For this method, an additional verification/optimisation set to avoid the overfitting of the data was used from the whole 692 dataset. Thus, the sizes of the training set, the verification/optimisation set and the test set used by this method are 231, 231 and 230 cases respectively. Table of figure 8.17 shows a summary of the results given by Bayes9 and the MLP after 14 training epochs. Table of figure 8.18 shows a summary of the results given by Bayes9 and the MLP after 50 training epochs.

Tests	Bayes9	MLP (14 epochs)
Sensitivity	88% (81-95)	85% (76-93)
Specificity	94% (90-98)	100%
PV+	89% (82-96)	100%
PV-	93% (89-97)	93% (90-97)

Figure 8.17: Results of sensitivity, specificity, predictive value of a positive result and predictive value of a negative result given by Bayes9 and the MLP after 14 training epochs. 95% confidence intervals are shown in parentheses

Tests	Bayes9	MLP (50 epochs)
Sensitivity	88% (81-95)	88% (80-95)
Specificity	94% (90-98)	98% (96-99)
PV+	89% (82-96)	95% (90-99)
PV-	93% (89-97)	95% (91-98)

Figure 8.18: Results of sensitivity, specificity, predictive value of a positive result and predictive value of a negative result given by Bayes9 and the MLP after 50 training epochs. 95% confidence intervals are shown in parentheses

The number of epochs refers to the number of times that the complete training set is used by the MLP. Thus, 14 epochs means that the whole training set is taken 14 times as input by the MLP. About the number of epochs, although the results given by the MLP using 14 training epochs are very similar to those by the human performance (table of figure 8.14), those obtained with 50 training epochs represent a more realistic performance by this MLP (Cross, Downs et al. 2000). Thus, as can be noticed looking at tables of figures 8.17 and 8.18, there is a decrease in the latter situation in both specificity and PV+ which degrades the excellent performance of such a classifier significantly.

Figures 8.17 and 8.18 show a very good performance by the MLP method. Although the performance of this method, either using 14 or 50 epochs, generally overcomes that of Bayes9, there are a number of important points to discuss here about this comparison beyond those that are easily perceivable and highlighted by the previous tables (figures 8.17 and 8.18). An appropriate decision support system in a medical domain has to be able to provide a good degree of explanation about its decisions, i.e., how and why it arrived to a particular conclusion so that the physician can trust in such a conclusion and decide with a more secure criterion what to do. The intrinsic nature of MLPs makes them look like a “black box” where the intermediate calculations are not visible to the external user but only the output is visible. That is to say, MLPs provide poor interpretations on how they arrived at particular conclusions because what the symbolic meaning of the learned weights is not very easily interpretable by humans (Han and Kamber 2001). On the other hand, the probabilities handled by Bayesian networks, despite their numerical nature, can be traced from beginning to end so it is possible to know why and how the Bayesian network arrived at a specific conclusion. In sum, MLPs can be extremely accurate in their classification but they lack the power of inducing comprehensible structures that might help one understand the domain better creating new knowledge (Kohavi 1995b), whereas it can be argued that Bayesian networks do indeed have this power (Pearl 1988).

MLPs also require a number of parameters to be tuned so as to work properly. These parameters are embedded in the determination of the network topology and the most important ones are the number of hidden layers and the number of neurons in each hidden

layer. As can be inferred, MLPs could involve the very time-consuming and well-known task of knowledge elicitation (see section 1.2.2 of chapter 1). The MLP used in the studies by Cross et al. (Cross, Downs et al. 2000) has the following architecture: 11 input neurons (corresponding to the 10 defined observations plus the age), 6 neurons that form the hidden layer and 1 output neuron, which corresponds to the variable outcome (the dependent variable). The best selection of the parameters that define any neural network has, in general, to be determined empirically so that one can realise the big difficulties when constructing such an MLP. Thus, this indicates that there exist no mathematically proved procedures to choose the optimal architecture for this MLP (Cross, Downs et al. 2000). In this respect, Bayesian networks can be induced from knowledge alone, data alone or a combination of both (see sections 3.2, 3.3, 3.4 and 3.5 of chapter 3). Bayes9 discovers 4 relevant variables to explain the variable outcome only from data; so it can be argued that including prior expert knowledge in such a procedure would indeed be very likely to improve the accuracy of the classification it generates.

Regarding the number of hidden layers in an MLP, it has been proven that using a bigger number of them can represent some functions that cannot be represented by using, say, two hidden layers. However, increasing the number of such layers does not always solve this problem so some functions can still go unrepresented (Marcus 1998). Furthermore, as in the cases of logistic regression and decision trees, MLPs do not represent the interactions among the input variables. Hence this kind of model cannot generalise outside the training space meaning that if novel instances appear in the test set which did not appear in the training set, then the MLP will not be able to classify them correctly regardless the number of neurons in the hidden layers, the number of hidden layers or the sequence of training examples (Marcus 1998). This is an important limitation that should be taken into account when using this kind of classifier. More important, Marcus (1998) argues that symbol manipulation systems could probably avoid that limitation; Bayesian networks fall into this category.

Finally, as in the case when using the logistic regression, the interobserver variability was again detected by Cross et al. (Cross, Stephenson et al. 2000). In the same

study, they also found a similar phenomenon: the specificity and PV+ values when using the test set of 230 are 99% and 98% respectively while these same values when using the 323 test set reduced dramatically to 94% and 90% respectively. Moreover, the sensitivity and PV- values when using this latter dataset dropped too: from 82% to 71% for the sensitivity and from 93% to 82% for the PV-. As mentioned before, the strength of Bayesian networks seems to be that they are more robust when this variability is present. This would make them more suitable for the specific cytodiagnosis problem presented here.

8.4.5 ARTMAPs vs. Bayes9

The fifth and final method against which Bayes9 can be compared is another type of neural networks known as adaptive resonance theory mapping (**ARTMAP**) neural networks (Cross, Downs et al. 2000). For this method, as well as for the MLP methodology, an additional verification/optimisation set to avoid the overfitting of the data was used from the whole 692 dataset. Thus, the sizes of the training set, the verification/optimisation set and the test set used by this method are 231, 231 and 230 cases respectively. Table of figure 8.19 shows a summary of the results given by Bayes9 and ARTMAP.

Tests	Bayes9	ARTMAP
Sensitivity	88% (81-95)	90% (84-96)
Specificity	94% (90-98)	96% (93-99)
PV+	89% (82-96)	94% (89-99)
PV-	93% (89-97)	94% (90-98)

Figure 8.19: Results of sensitivity, specificity, predictive value of a positive result and predictive value of a negative result given by Bayes9 and ARTMAP. 95% confidence intervals are shown in parentheses

Figure 8.19 shows that ARTMAP outdoes the performance of Bayes9 in all the tests. The limitations of the MLP approach mentioned above such as the number of adjustable parameters and the difficulty in presenting transparent explanations of the results (the so-called black box problem) seem to be eliminated by the ARTMAP methodology

because of its architecture: it does not contain hidden layers with implicit meaning (Cross, Downs et al. 2000). About the first limitation, ARTMAPs only have one adjustable parameter, called the vigilance factor, which reduces the complexity of having a combinatory explosion to choose the best parameters in an MLP; therefore, the optimisation of the ARTMAP network is simpler. With respect to the second limitation, it is possible to extract automatically, once the ARTMAP is trained, a set of rules similar to those used in rule-based expert systems so that these rules can give a good account of why the ARTMAP reached a certain conclusion, in contrast with the MLP methodology. However, in comparison with the results of specificity and PV+ given by MLP either using 14 epochs or 50 epochs (figures 8.17 and 8.18), ARTMAP's results of specificity and PV+ (figure 8.19) are less accurate: 100% specificity and 100% PV+ with 14 epochs in MLP against 96% specificity and 94% PV+ in ARTMAP or 98% specificity and 95% PV+ with 50 epochs in MLP against 96% specificity and 94% PV+ in ARTMAP. Hence, although ARTMAPs can overcome, in general, the intrinsic problems of MLPs, they still show a poorer performance than that of MLPs in terms of classification accuracy in this specific cytopathological problem.

Regarding the problem of the MLPs about their limitation to generalise outside the training space, ARTMAPs could possibly have the potential to cope with that problem in a simple and beautiful way: if a case can be instantiated significantly enough to a predefined category then this case is assigned to such a category, otherwise a new category node is created to classify that case. This is achieved by adjusting the threshold of the vigilance factor parameter. The problem with this approach is that if the value of the threshold is set very high, then there may be many new categories created leading to the poor compression of the data, poor explanation of the output and overall, a poor classification performance (Cross, Downs et al. 2000). Another related problem about the formation of new categories comes when the data items, with which the ARTMAP is fed, are presented in different order. Thus, different order in item presentation can lead to the formation of different clusters. A possible solution to this problem is to use something known as voting strategy. The way this strategy works is to train a certain number of ARTMAPs with different item

presentation orders and take the final prediction from the majority decision of all these networks.

In favour of the Bayesian network approach presented in this dissertation (Bayes9), we can say that it provides significantly good estimates if we take into account that it learns the classifier structure only from data without any external supervision. In the case of ARTMAPs, it can still be perceived that a time-consuming knowledge elicitation process involves the construction of such an ARTMAP. Thus, it can be argued that if Bayesian network classifiers are built from a combination of expert knowledge and data, they could provide much better estimates comparable to those given by the best of the approaches shown in this chapter. As Cross et al. show (Cross, Downs et al. 2000), some if-then rules from the ARTMAP in their study can be extracted. A key point about this is that some of these rules that predict a malignant outcome have common variables with some rules that predict a benign outcome. Hence, it is the incorporation of different variables in those rules, apart from the common ones, which determines if the output is either malignant or benign. In contrast, in the Bayesian network approach, the same set of variables (same structure) with different combinations for this set is responsible for determining whether the outcome is malignant or benign. Therefore, it also can be argued that if the structure given by Bayes9 were modified by expert knowledge, then it would give very accurate overall classification estimates.

Cross et al. (Cross, Downs et al. 2000) do not discuss the interobserver variability problem within the ARTMAP framework. Thus, it would be necessary and very interesting to test the latter methodology against this variability to make sure that ARTMAPs can potentially be a suitable and robust classifier for the FNAB cytodiagnosis task. Moreover, ARTMAPs do not represent the interactions among the independent variables, which could prevent these networks from discovering some important relationships and gaining a deeper understanding of the problem being modelled. Bayes9 does indeed represent these interactions and also provide a more parsimonious model (only 4 relevant variables to explain the outcome against 11 of the ARTMAP model) with acceptable estimates. As said

above, if Bayes9 were combined with some expert knowledge, then it would probably show a much better performance.

8.4.6 ROC curves by logistic regression, MLPs and Bayes9

Finally, we present a comparison of ROC curves produced by logistic regression (Cross, Downs et al. 2000; Cross, Stephenson et al. 2000), MLP (Cross, Stephenson et al. 2000) and Bayes9. A summary table of this comparison is shown in figure 8.20.

Method	Area under curve
Logistic Regression	0.979 (0.958-0.999)
MLP	0.970 (0.948-0.992)
Bayes9	0.991 (0.983-0.998)

Figure 8.20: Results of the area under the ROC curves given by the logistic regression, MLP and Bayes9. 95% confidence intervals are shown in parentheses

Figure 8.20 shows an excellent overall classification performance given by Bayes9, which overcomes those given by both logistic regression and MLP. The area under such a curve (see figure 8.12) represents a very small rate of false positive as well as false negatives. Because a ROC curve determines the trade-off between sensitivity and specificity (if one test is improved the other will worsen and vice versa), it is possible to set a different cut point to improve the specificity given by the classifier induced by Bayes9 because of the unusual requirement of high specificity for the FNAB cytodiagnosis. Based on the results of figure 8.20 and because the logistic regression performance represents the current available level of performance against which other classifiers have to be compared (Cross, Downs et al. 2000), it can be concluded that Bayes9 performs really well as an unsupervised classifier. Comparing Bayesian networks and decision trees, an advantage of the former approach over the latter is that, because of the dichotomous nature of the decision trees, ROC curves cannot be built (Cross, Downs et al. 2000) so that it is not possible to set any threshold to reduce the number of false positives. Because, by definition

Bayesian networks handle random variables naturally, this threshold can be adjusted reducing the rate of false positives as mentioned above.

8.4.7 Performance of Tetrad II, Power Constructor and Bayes9 on the breast cancer dataset

We used the same 462 cases to train procedures Tetrad II and Power Constructor and the same 230 remaining cases to test the performance of such algorithms, as in the case for Bayes9. Figures 8.21 and 8.22 show the results of running Tetrad II and Power Constructor on the 462 case training set respectively.

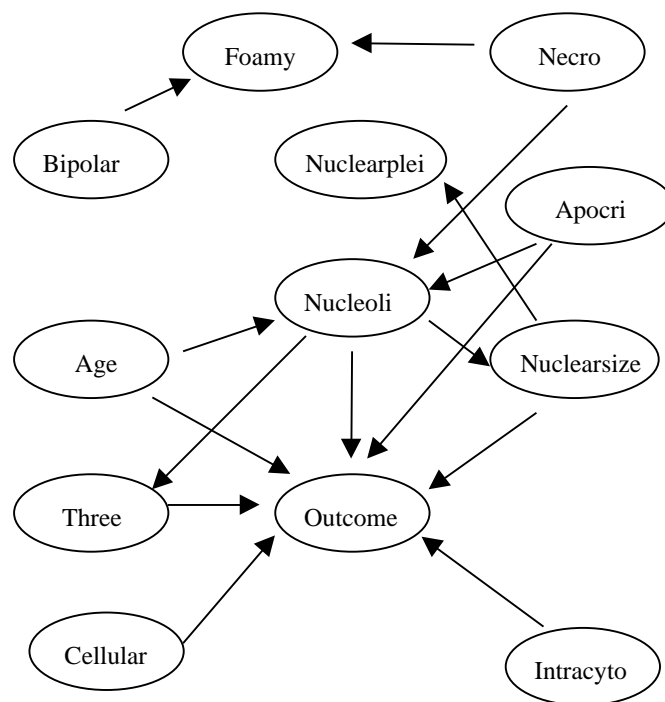


Figure 8.21: The result of running Tetrad II on the 462 case training set of the breast cancer database

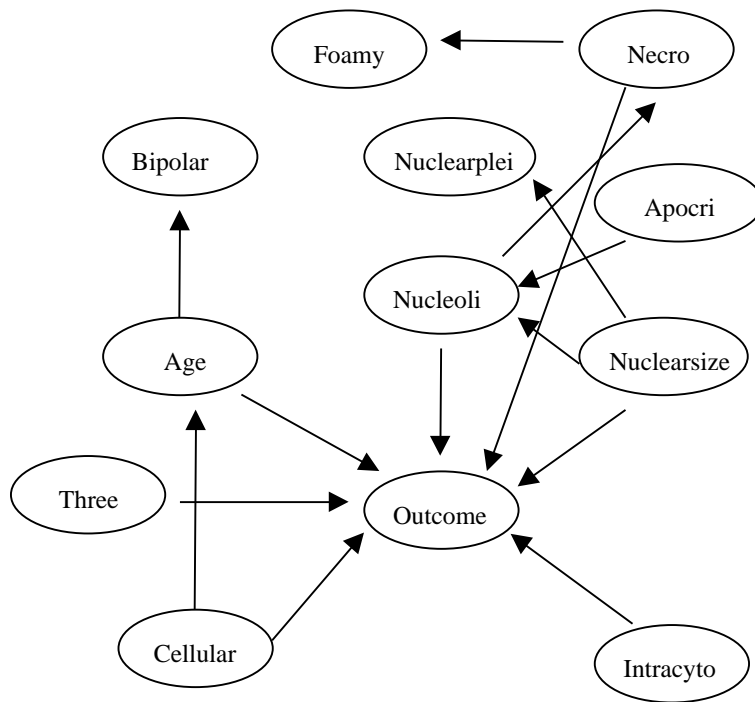


Figure 8.22: The result of running Power Constructor on the 462 case training set of the breast cancer database

From figures 8.3, 8.21 and 8.22, it can be seen that Bayes9, Tetrad II and Power Constructor share three variables to explain the outcome: age, intracytoplasmic lumina, nucleoli and nuclear size. Moreover, Tetrad II and Power Constructor have also two more variables in common to explain the class variable: cellular dyshesion and the “three-dimensionality” of epithelial cells clusters. Tetrad II also considers variable apocrine change as being relevant to explain the outcome whereas Power Constructor considers variable necrotic epithelial cells as being relevant for the prediction of the output.

Table of figure 8.23 presents the results of the six different tests (including the MDL score) by these three algorithms on the 230 case test set.

Tests	Tetrad II	Pow Cons	Bayes9
Accuracy	93% \pm 1.68%	92% \pm 1.78%	92% \pm 1.78%
Sensitivity	87% (79-94)	88% (81-95)	88% (81-95)
Specificity	97% (94-100)	95% (91-98)	94% (90-98)
PV+	94% (88-99)	90% (84-97)	89% (82-96)
PV-	93% (89-97)	93% (89-97)	93% (89-97)
MDL	982.12	998.58	1066.40

Figure 8.23: Results of accuracy, sensitivity, specificity, predictive value of a positive result, predictive value of a negative result and MDL given by Tetrad II, Power Constructor and Bayes9 on the 230 case test set for the holdout method. 95% confidence intervals are shown in parentheses

According to the results of table of figure 8.23, the best overall performance among the three procedures is that of Tetrad II. Compared to the performance of Bayes9, it seems that the three variables that are not shared by both procedures are the key to produce much better results in Tetrad II for specificity and PV+. In comparison to Power Constructor, it seems that variable apocrine change is the variable that makes the difference in Tetrad II to produce better results in these same tests. Finally, according to the MDL criterion, the Bayesian network produced by Tetrad II seems to compress the data in the best way. This last result supports what the other tests indicate: that Tetrad II is the best of the three classifiers on the 230 case dataset.

To press the limits of the performance of these three procedures, we decided to test them on the 323 case dataset, which was collected by multiple observers (see section 8.4.2), in order to evaluate their behaviour regarding the previously mentioned interobserver variability problem. Thus, as explained in section 8.4.2, we used the 462 case test set to train the three algorithms (figures 8.3, 8.21 and 8.22). Then, we used the 323 case test set to assess the robustness of these algorithms when tested with a multiple observer collected database. The results are summarised in table of figure 8.24.

Tests	Tetrad II	Pow Cons	Bayes9
Accuracy	80% \pm 2.22%	77% \pm 2.34%	85% \pm 1.98%
Sensitivity	56% (48-64)	50% (42-58)	72% (64-79)
Specificity	96% (94-99)	96% (94-99)	94% (91-98)
PV+	91% (85-98)	91% (84-97)	90% (84-95)
PV-	75% (70-81)	73% (67-79)	82% (77-87)
MDL	814.22	819.06	854.81

Figure 8.24: Results of accuracy, sensitivity, specificity, predictive value of a positive result, predictive value of a negative result and MDL given by Tetrad, Power Constructor and Bayes9 on the 323 case test set for the holdout method. 95% confidence intervals are shown in parentheses

Figure 8.24 shows now divided results: while Bayes9 has the best performance in accuracy, sensitivity and PV-, Tetrad II has the best performance in specificity, PV+ and MDL. However, it can be argued that the tests where Bayes9 behaves in the best way are considerably much better than those where Tetrad II does show the best behaviour. It seems that the conservative nature of Bayes9, regarding the addition of arcs (underfitting), makes it less vulnerable to the important interobserver variability problem. That is to say, because the number of arcs produced by Tetrad II and Power Constructor (7 arcs) to explain the variable outcome is bigger than that produced by Bayes9 (4 arcs), it can be argued that whereas the Bayesian networks given by Tetrad II and Power Constructor match the training data (462 cases) better than the Bayesian network given by Bayes9, they have, in general, a poorer performance (overfitting) on the test data (323 cases) than that by Bayes9, as shown in figure 8.24. Moreover, comparing the results of figures 8.23 and 8.24 for each procedure, one can easily see that the difference within the same procedure using the 230 and 323 case test sets is far bigger in Tetrad II and Power Constructor than in Bayes9. The only test that cannot be compared in this way is the MDL criterion because, by definition, it gives different scores when the sizes of the samples are different. In sum, the results given by Bayes9 (figure 8.3) seem to support Occam's razor: simpler models are, in general, more useful and adequate than complex models (Grunwald 2000).

The previous experiments using the Bayesian network approach seem also to unveil interesting characteristics in the experts' behaviour that avoid the reproducibility of results

among different observers. To support this hypothesis, we decided to train the three procedures presented here with the 323 case database to measure the individual differences among the cytopathologists in the codification of the variables. Figures 8.25, 8.26 and 8.27 will help us visualise these differences.

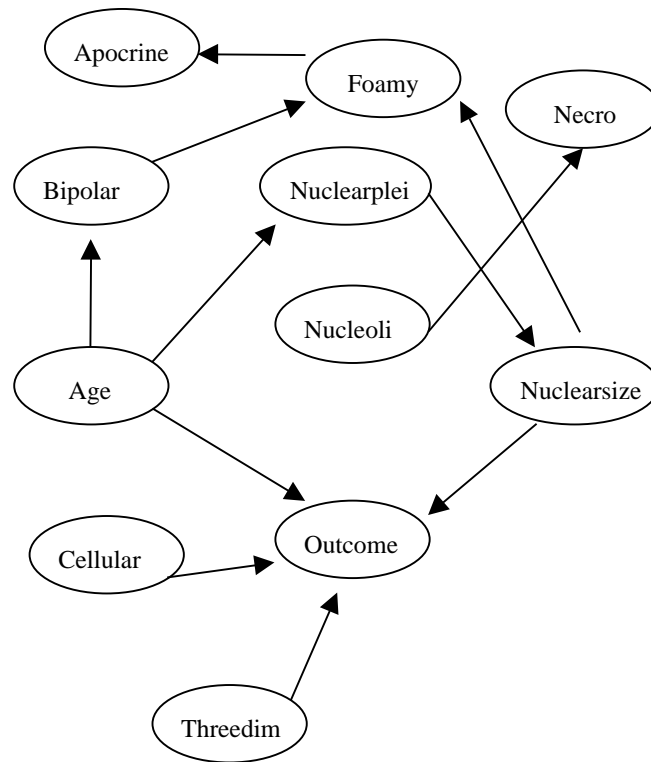


Figure 8.25: The result of running Bayes9 on the 323 case test set

When classifying the variables, a single observer cannot be blind to the values they have assigned to earlier variables. When a variable has an ambiguous value, the expert may then be biased by their knowledge about an earlier decision. The observed values are thus not strictly independent (see assumption 2 in section 4.2 of chapter 4). In fact, the expert's own implicit knowledge about the relationships between the variables may be affecting the

values recorded in the dataset. The results of the Bayesian model will thus reflect the expert's knowledge rather than (as well as) the real relationships.

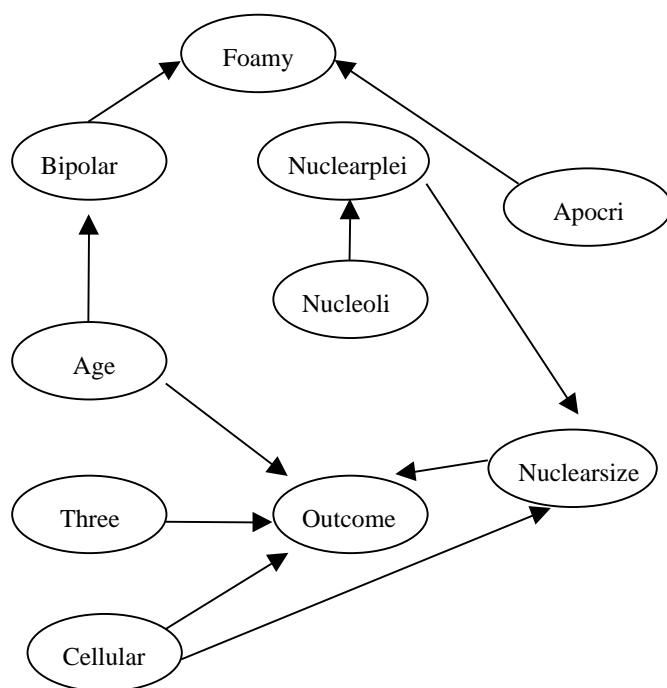


Figure 8.26: The result of running Tetrad II on the 323 case test set

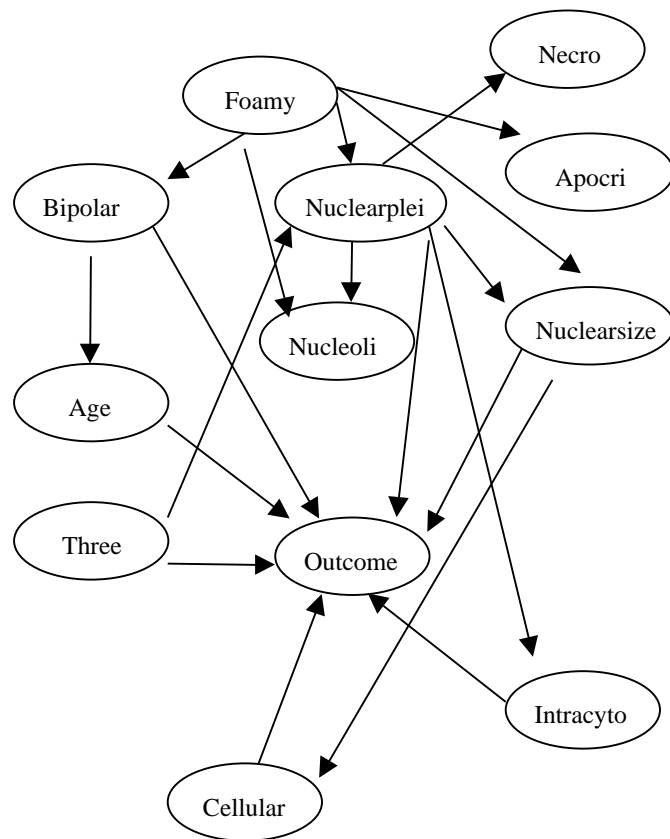


Figure 8.27: The result of running Power Constructor on the 323 case test set

It is now easy to see why the solutions of the three procedures do not generalise well when using the database collected by a single observer (462 cases) for training purposes and the database collected by multiple observers (323 cases) for test purposes. For the case of Bayes9, if we compare figure 8.3 against 8.25, we can notice that it is only two variables (age and nuclear size) which are common for the Bayesian networks of both figures. According to these results and the results of tables of figures 8.23 and 8.24, it seems that both the single observer and the multiple observers agree that variables age and nuclear size are the most informative variables to determine specificity and PV+, since it is the values of these tests which remain practically constant within the two datasets. Recall that cytopathologists are specially trained to keep the values of these two tests at maximum

(100%). Thus, these results seem to support such specialized behaviour. Furthermore, the values for sensitivity and PV- drop considerably from one database (462 cases) to another (323 cases). These results also seem to support this mentioned overspecialisation.

For the case of Tetrad II, if we compare figure 8.21 against 8.26, we can notice that it is now four variables (age, three dimensionality, cellular dyshesion and nuclear size) which are common for the Bayesian networks of both figures. According to these results and the results of tables of figures 8.23 and 8.24, it seems less clear (with respect to the results of Bayes9) in what variables the single observer and the multiple observers agree this time as being the most informative variables to determine specificity and PV+, since the specificity remains almost constant within the two datasets but PV+ does not. Because of the overfitting produced by Tetrad II, this algorithm seems to be more vulnerable to the interobserver variability since the values for sensitivity and PV- drop much more dramatically for this procedure than for Bayes9 from one database (462 cases) to another (323 cases): from 87% to 56% in sensitivity and from 93% to 75% in PV- for Tetrad II; from 88% to 72% in sensitivity and from 93% to 82% in PV- for Bayes9.

For the case of Power Constructor, if we compare figure 8.22 against 8.27, we can notice that it is five variables (age, three dimensionality, cellular dyshesion, intracytoplasmic lumina and nuclear size) which are common for the Bayesian networks of both figures. In contrast with Tetrad II, and according to these results and the results of tables of figures 8.23 and 8.24, it seems that the additional variable found by Power Constructor to explain the outcome (intracytoplasmic lumina) is the key to produce almost identical values of specificity and PV+ within the two datasets. However, with the selection of these five variables, Power Constructor overfits the data. This overfitting can be perceived in the values of sensitivity and PV-, which drop even more dramatically than those given by Tetrad II from one database (462 cases) to another (323 cases): from 88% to 50% in sensitivity and from 93% to 73% in PV- for Power Constructor; from 87% to 56% in sensitivity and from 93% to 75% in PV- for Tetrad II. Hence, this algorithm seems to be also more vulnerable to the interobserver variability than Bayes9.

From all the results shown in this chapter, it can be argued that the Bayesian networks suggested by Bayes9 have the potential to help cytopathologist trainees become experts in the area, help expert cytopathologists make more robust, consistent and objective diagnosis and also serve as a part of a bigger automated diagnosis system which can collect the data from a machine vision module. Bayes9 can also provide cytopathologists with a support decision tool that helps them improve on the reproducibility in the results, i.e., Bayes9 procedure can show robustness when confronted with the important interobserver variability problem (at least for the case of specificity and PV+). Bayes9 can give insight about how the diagnoses are made as well. Based on those results, it is also possible to argue that Bayes9 algorithm could be used to help recognise the most important interactions among the features that help diagnose the presence or absence of breast cancer.

Chapter 9

Discussion

The intrinsically growing nature of information requires new and more reliable methods to collect and analyse data more efficiently, soundly and consistently. It can be argued that the approach taken and presented in this dissertation, namely, that of Bayesian networks, provides a powerful way to reveal the meaning of the data, discovering patterns, rules and, in general, knowledge implicitly contained in the data so that we can gain a more profound understanding of the phenomenon under investigation. Bayes9 has proven a very useful tool to save enormous amounts of time during the knowledge elicitation process, known as the knowledge acquisition bottleneck. It can also provide some significant and important insight about how the variables of a particular problem interact each other leading possibly to the constitution of new knowledge about a specific domain. That is to say, Bayes9 can lead to an improvement upon expert performance suggesting new relationships that might not have been taken into account before by human experts. Moreover, under the experts' supervision, it is possible to consider these relationships of causal nature so as we could manipulate some of the causes to produce a desired effect or prevent it, according to which is the case. This is called analysis under interventions, which gives causal models the power of doing prediction given this set of interventions. Causal models are also easier to understand and prior knowledge can easily be incorporated into them.

The typical kinds of problems that we can solve using the Bayesian network approach are those of classification, prediction, diagnosis, decision-making and control, among some others. In fact, Bayesian networks can be potentially applied in any problem where uncertainty comes to the scene. Such a graphical model provides an intelligent tool, from the Artificial Intelligence perspective, that is able to manipulate uncertainty in a sound and consistent way using a powerful combination of a qualitative part (the structure of the

network) and a quantitative part (the marginal and conditional probability tables attached to each node in that network). The structure of the network by itself provides a means to compactly represent a joint distribution by factorising it in a product of marginal and conditional probabilities. This will reduce dramatically, in many cases, the number of parameters needed to compute such joint probability distribution so that we can perform probability propagation and know any value of interest for prediction, diagnosis or decision-making purposes. This structure also gives suitable means to expressing dependence / independence assumptions and making efficient inference from observations. The presence of arcs as well as the absence of them provides useful information for such purposes. Furthermore, the high dimensionality represented by the interactions among the variables taking part in a determined problem will be reduced to only two dimensions represented by the Bayesian network structure. This in turn will make much easier the understanding of such interactions presenting them in an easy and very intuitively understandable way.

The approach presented in this dissertation is part of the field known as Knowledge Discovery in Databases or Data Mining, which has taken ideas and principles from many scientific disciplines: database technology, artificial intelligence, machine learning, neural networks, statistics and some others. This assures that a sound theoretical background supports it. The combination of those ideas has proved useful in many cases and conforms an active research area worth it to explore. But we do not have to forget that, to date, it is the correct selection of the variables in the first instance, made by the human experts, that makes any of the algorithms within the KDD approach perform well. In other words, a model selection carried out by, say, Bayes9, will be accurate as long as the variables used as input for this algorithm are relevant, according to the experts. Thus, Bayes9 does not intend, by any means, to serve as a replacement of the good judgments of human experts but to serve as a support tool, which such experts can rely on and make more sound inferences.

Although this dissertation has to do only with the automatic extraction of knowledge from data in the form of Bayesian networks, a sensible exploration and future

work could be the use of methods from Bayesian statistics in order to take advantage of the prior knowledge at hand as well as the data. Such a combination (knowledge and data) seems very promising and it is not only supported by Bayesian statistics but also by psychological theories such as the Power PC Theory.

A very important advantage that could not be easily noticed is that Bayes9 carries out the process of model selection among an enormous number of possible structures in contrast with the traditional statistical methods that only test a specific model or select one from a small list. Hence, Bayes9 saves time for the experts in performing this repetitive task of selection and testing by finding the probable most correct model or set of models once these experts have preselected the relevant variables. However, this enormous number of structures is by no means exhaustive since an exhaustive enumeration of them represents a very complex problem, in the sense of an explosive combination that not even computers can handle efficiently. To avoid this problem, it is imperative to use heuristic methods, such as Bayes9. Thus, the framework of learning Bayesian networks from data uses old ideas and combines them with new ones to achieve impressive results, as presented in this dissertation.

The performance of any method that learns Bayesian networks from data has to be qualitatively evaluated by an expert or group of them according to their knowledge and experience. Also, there exist quantitative methods that give a measure of the goodness of fit of the proposed model given the data. In this research, the minimum description length (MDL) criterion was used to measure how well the model proposed by Bayes9 fits the data. It is very important to explain here why I believe such a metric represents a sensible solution to the selection model problem. First of all, as the MDL definition states, this criterion incorporates naturally Occam's razor: it prefers simpler models than complex ones. It can be argued that, by seeking among the simpler models, it is more likely to find models that make more sense and provide useful, accurate and adequate results which can account very well for new data; i.e., models useful for prediction, diagnosis and decision-making tasks. The MDL score needs no specification of any prior knowledge; hence, it can be claimed, it is objective. The way this criterion works is very intuitive: it

selects the most probable model given the data. It is also very closely related to other reliable goodness of fit criteria such as BIC (Bayesian information criterion), MML (minimum message length) and CV (cross-validation). MDL does not search for the true model but for a simple model that fits the data reasonably well. In other words, MDL does not decide whether a model is a good one; rather it chooses the best model among a list of competitive models. Thus, the final decision about what model is to be taken to explain the phenomenon under investigation must involve human judgement. MDL is a method used for inductive inference; hence, it is subject to many controversies: is it valid to go from particular facts or observations to the consolidation of a general law or theory that explains a wide range of situations? To the best of my knowledge, nobody has given a final word on this matter. But MDL can help to continue exploring on this issue.

The MDL results shown in chapters 6, 7 and 8 indicate what was said above: that such a criterion selects the model that obtains the minimum score but such a model must be ultimately reviewed and validated by the human expert. There is a special relationship between the measure used by Bayes9 to build Bayesian networks from data (entropy) and the MDL score: the former can be seen as a special case of the latter. As the results of procedures Bayes2, Bayes5 and Bayes9 in chapters 4, 5 and 6 respectively suggest, entropy is a powerful and successful measure yet it is very controversial. Entropy proved not very useful in choosing the ancestral ordering needed by Bayes2 and Bayes5 to construct accurate Bayesian networks but it proved a very good metric for Bayes9 for the same purpose. There are a number of studies which indicate that MDL performs well both in theory and practice. In the studies carried out in this dissertation, I argue that MDL performs, in general, very well. The tendency or bias of MDL to underfit the data can be clearly seen specially in chapter 6 where the performances of Bayes2, Bayes5 and Bayes9 are compared against each other: MDL chooses simpler models because the more complex the model, the more likely the learned model will be unreliable.

A surprising feature of MDL is that it allows comparison among different model classes (say, polynomials, Bayesian networks, neural networks) since, in the end, we compare the code length given by those models. It is important to say that MDL represents

only an approximation because, as it needs a suggested class of models and observed data, a finite amount of these data cannot completely represent the underlying probability distribution. Some people argue that the advantage of MDL in not requiring the specification of any prior knowledge can be regarded as a disadvantage as well. This is because, they claim, it denies the fact that prior knowledge can be useful in finding a sensible solution. But, if we look carefully, such prior knowledge is indeed used to guide the selection of the classes of models to solve the problem at hand. Moreover, we should use it within the MDL principle only if it helps reduce the code length. That is to say, it can be the case that this prior knowledge could create conflict with the MDL principle. There exists a controversy about the role of MDL in model selection: it selects the best model class but does not estimate the best combination of the parameters belonging to such class. For instance, it might select the best order polynomial rather than the estimation of its coefficients. In the context of Bayesian networks, MDL may choose the best network regarding the number of arcs but it would not say anything about the best fitting of the parameters given this number of arcs. Since MDL only gives a yardstick to measure the code length of a model, the task of selecting the best combination of its parameters lies beyond the MDL principle. In sum, the very purpose of this principle is to extract or discover regular and learnable patterns from the data and compress, in the best possible way, such regularities. In other words, it tries to unveil the algorithm responsible for generating those data.

Returning to the results given by Bayes9, they show that this procedure produces accurate and close estimates, under the MDL principle, with respect to those of the gold-standard models. It also produces similar results of those given by well-established procedures such as Tetrad II and Power Constructor. All of them are constraint-based methods that use the χ^2 statistics to test the conditional independence hypothesis in order to include or not an arc between a pair of variables. We argue that because of the internal programming details of each of them, their results are not exactly the same. It is also important to mention that neither of these procedures needs an ancestral ordering of the variables to work properly, as is the case of some of the search and score methods.

Regarding this ancestral ordering, a strong assumption made by Bayes2 was that the mutual and conditional mutual information measures could provide a way to establish such an ordering correctly. The results show that Bayes2 performs in general very poorly because these measures proved unable to do so. Moreover, Bayes2's lack of performing higher conditional independence tests prevents it from producing accurate results too. Although its immediate successor, procedure Bayes5, is capable of carrying out conditional independence tests of higher order, it makes this strong assumption about the ancestral ordering too, which leads to inaccurate results as well. The empirical results obtained by running these two algorithms with different databases, led us to propose, design and implement another algorithm, namely, Bayes9, which is capable of overcoming these intrinsic limitations shown by its predecessors. The plausible and sensible results obtained by running Bayes9 on the simulated and real-world databases suggest that this procedure has the potential to be applied in real-world problems serving as a decision support system. Thus, procedures Bayes2, Bayes5 and Bayes9 have given us a good insight about the possibility of extracting causal knowledge from raw data.

Bayes9 also takes the important role played by the Occam's razor in model selection into account. Its tendency to underfit the data, rather than overfitting the data, is an example of that. As the algorithm and the results show, its very nature is to keep at minimum the number of arcs arriving to every node in the network. Thus, it is more common to have errors of missing arcs than to added arcs.

The fundamental problems of Artificial Intelligence, namely, efficient manipulation of uncertainty, knowledge representation and learning from data, seem to be solved in a beautiful way, under certain conditions, by the Bayesian network approach, as the results given by Bayes9 suggest. Bayesian networks have the capability of providing a reasonable explanation of the outputs and interpretability and modifiability of the results. They also allow belief revision and belief update, qualitative inference through their structure and analysis under interventions (in the sense of causality), as mentioned previously.

There are still many things to do. Bayes9 needs more development in the sense of extending its capabilities to make it able of handling a range of situations under different

conditions. Such situations can perfectly be the manipulation of missing values and hidden variables, the combination of discrete and continuous variables or the discretization of the latter ones, the inclusion of modules for probability propagation, data visualisation, Monte Carlo data generator and classification methods, among others.

Regarding the model selection problem, it would be very interesting to explore the combination of different methodologies, such as genetic algorithms, neural networks and Bayesian networks to build a tool that could take advantage of the best of each approach. It would be also interesting and worthwhile to explore the combination of constraint-based algorithms and search and score algorithms to learn Bayesian networks from data. This may be done, for instance, by combining the MDL criterion and procedure Bayes9 so that Bayes9 can provide a useful and close-to-the-truth Bayesian network structure to a search and score algorithm that might well use MDL as the score metric. Such a combination would provide a good approximation of the gold-standard network by Bayes9 and refined by the “greedy” algorithm. Furthermore, the results of this exploration could be refined by the experts’ knowledge, which probably will make the results more robust and sound. I argue here that these extended features (greedy algorithm + prior knowledge) will produce more beautiful results (in the sense of accuracy) than those produced by Tetrad II and Power Constructor. This combination of prior knowledge and data has been one of the successes of Bayesian statistics and it seems that, under restricted conditions, the Power PC Theory can incorporate naturally such knowledge to capture causality from data. However, the work by Hume, Kant and researches from Psychology, Philosophy, Statistics and Computer Science still does not have a final word about how causal knowledge is acquired. It seems that, to date, this causal knowledge escapes mechanization in a wide range of situations. A very nice future work would then be to continue exploring and investigating the roots of such a problem by finding a link or a close relationship between the Power PC Theory and Bayesian networks so that we can also take advantage of the best that each approach can offer. If successful, such a work would make Bayesian networks become a useful psychological model to explain different phenomena and make psychologists become more interested in them.

Finally, a good and sensible exploration would be that of the application of Bayes9 to real-world problems. The medical domain asks urgently for the development of robust decision-support systems that make this task much easier for doctors. In fact, Bayes9 is the first step to build a real-world decision-making tool. To be considered as such, Bayes9 would need to incorporate utility nodes showing the cost of taking a particular decision. If we want to use a Bayesian network as a decision-support tool, then we have to add some features, i.e., additional types of nodes, to convert the former into the latter. This special kind of Bayesian networks is called influence diagrams. Thus, an influence diagram consists of three different types of nodes: chances nodes (circles representing random variables), decision nodes (squared nodes which contain different choices that the decision maker can choose from) and one value node (a diamond representing the random variable whose value is the utility of the decision). The new advances of technology will surely provide more powerful tools that will permit us to collect and extract more richly the knowledge contained in data. Such tools may be well combined with procedures such as Bayes9 to construct more accurate, exact and precise decision-making systems.

In sum, Bayesian networks have begun a revolution in the development of intelligent systems, managing uncertainty and complexity. The combination of simpler parts from probability and graph theories has allowed us to build complex systems capable of giving very good results. Because of this synergy, graphical models, such as Bayesian networks, might be able to explain how to acquire causal relationships from raw data. But more investigation and experimentation are required in order to reveal whether or not computational procedures can embody causality soundly and efficiently.

Appendix

A detailed description of Bayes9

In this appendix, we use an example to describe more precisely the way procedure Bayes9 works.

12 A.1 Procedure Bayes9 (step by step).

We will use the example presented in chapters 4, 5 and 6 to explain in detail how procedure Bayes9 builds Bayesian networks from data. Imagine we have the structure of a gold-standard network represented in figure A.1, from which we generate a database.

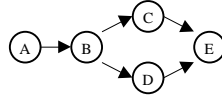


Figure A.1: A gold-standard Bayesian network

The random variables taking part in this problem are then A, B, C, D, E. So, for the first step of procedure Bayes9, the number $n = 5$.

Step 1 of procedure Bayes9

1. For $a = 1$ to $a = n$

 For $b = a$ to $b = n$

 If $a \neq b$

 Compute the value of the mutual information (equation 4.4) for each pair of variables $Y_{(a)}$, $Y_{(b)}$. Then use formula 4.8 to check whether the null hypothesis H_0 (two variables are independent from each other) holds or not. If H_0 does not hold then draw an **undirected** arc from $Y_{(a)}$ to $Y_{(b)}$ and form a queue **W** such that $\mathbf{W} = \bigcup (Y_{(a)}, Y_{(b)})$

end for b
end for a

End of step 1 of procedure Bayes9

The independence tests to be performed for these 5 variables are: $I(A, \cdot, B)$, $I(A, \cdot, C)$, $I(A, \cdot, D)$, $I(A, \cdot, E)$, $I(B, \cdot, C)$, $I(B, \cdot, D)$, $I(B, \cdot, E)$, $I(C, \cdot, D)$, $I(C, \cdot, E)$ and $I(D, \cdot, E)$. For this example, it is the case that, for all pairs of variables, the null hypothesis H_0 does not hold, as figure A.2 shows. The significance level () used to carry out all the marginal and conditional independence tests is 0.05. Hence, $\mathbf{W} = \{(A,B), (A,C), (A,D), (A,E), (B,C), (B,D), (B,E), (C,D), (C,E), (D,E)\}$.

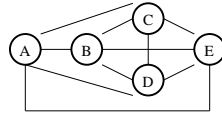


Figure A.2: Step 1 of procedure Bayes9

Step 2 of procedure Bayes9

2. Let $(\mathbf{W}, Y_{(a)})$ be the set of vertices adjacent to $Y_{(a)}$ in the undirected acyclic graph \mathbf{W} while $\mathbf{W} \neq \emptyset$

Select the pair of variables $Y_{(a)}, Y_{(b)}$ from the beginning of \mathbf{W} ; then form the adjacency **power** set of $Y_{(a)}$ called $\mathbf{Z} = \{(\mathbf{W}, Y_{(a)}) \setminus Y_{(b)}\}$. Compute the value of the conditional mutual information (eq. 4.5) between each pair of variables $Y_{(a)}$ and $Y_{(b)}$ given \mathbf{Z} . Then use formula 4.10 to check whether the null hypothesis H_0 (two variables are independent from each other given a set of variables in \mathbf{Z}) holds or not. If H_0 holds then remove this pair of variables from \mathbf{W} . If H_0 does not hold, form a new set $\mathbf{X} = \bigcup (Y_{(a)}, Y_{(b)})$ and remove this same pair of variables from \mathbf{W}

end while \mathbf{W}

End of step 2 of procedure Bayes9

The first pair of variables is (A,B) and the adjacency power of A (not including B), i.e., $\mathbf{Z} = \{ \emptyset, (C), (D), (E), (C,D), (C,E), (D,E), (C,D,E) \}$. For this example, the null hypothesis does not hold; thus, there is no variable or combination of them that can d-separate A from B. So, we delete this pair of variables from \mathbf{W} and form a new set $\mathbf{X} = \{(A,B)\}$.

The second pair of variables is (A,C) and the adjacency power of A (not including C), i.e., $\mathbf{Z} = \{ \emptyset, (B), (D), (E), (B,D), (B,E), (D,E), (C,D,E) \}$. For this example, the null hypothesis does hold since B can d-separate A from C. So, we delete this pair of variables from \mathbf{W} . $\mathbf{X} = \{(A,B)\}$.

The third pair of variables is (A,D) and the adjacency power of A (not including D), i.e., $\mathbf{Z} = \{ \emptyset, (B), (C), (E), (B,C), (B,E), (C,E), (B,C,E) \}$. For this example, the null hypothesis does hold since B can d-separate A from D. So, we delete this pair of variables from \mathbf{W} . $\mathbf{X} = \{(A,B)\}$.

The fourth pair of variables is (A,E) and the adjacency power of A (not including E), i.e., $\mathbf{Z} = \{ \emptyset, (B), (C), (D), (B,C), (B,D), (C,D), (B,C,D) \}$. For this example, the null hypothesis does hold since B can d-separate A from E. So, we delete this pair of variables from \mathbf{W} . $\mathbf{X} = \{(A,B)\}$.

The fifth pair of variables is (B,C) and the adjacency power of B (not including C), i.e., $\mathbf{Z} = \{ \emptyset, (D), (E), (D,E) \}$. For this example, the null hypothesis does not hold; thus, there is no variable or combination of them that can d-separate B from C. So, we delete this pair of variables from \mathbf{W} and add this pair to the set \mathbf{X} . Now $\mathbf{X} = \{(A,B), (B,C)\}$.

The sixth pair of variables is (B,D) and the adjacency power of B (not including D), i.e., $\mathbf{Z} = \{ \emptyset, (C), (E), (C,E) \}$. For this example, the null hypothesis does not hold; thus, there is no variable or combination of them that can d-separate B from D. So, we delete this pair of variables from \mathbf{W} and add this pair to the set \mathbf{X} . Now $\mathbf{X} = \{(A,B), (B,C), (B,D)\}$.

The seventh pair of variables is (B,E) and the adjacency power of B (not including C), i.e., $\mathbf{Z} = \{ \emptyset, (C), (D), (C,D) \}$. For this example, the null hypothesis does hold since (C,D)

can d-separate B from E. So, we delete this pair of variables from \mathbf{W} . $\mathbf{X} = \{(A,B), (B,C),(B,D)\}$.

The eighth pair of variables is (C,D) and the adjacency power of C (not including D), i.e., $\mathbf{Z} = \{ \text{ }, (E) \}$. For this example, the null hypothesis does hold since (E) can d-separate C from D. So, we delete this pair of variables from \mathbf{W} . $\mathbf{X} = \{(A,B), (B,C),(B,D)\}$.

The ninth pair of variables is (C,E) and the adjacency power of C (not including E), i.e., $\mathbf{Z} = \{ \text{ }, (D) \}$. For this example, the null hypothesis does not hold; thus, there is no variable or combination of them that can d-separate C from E. So, we delete this pair of variables from \mathbf{W} and add this pair to the set \mathbf{X} . Now $\mathbf{X} = \{(A,B), (B,C),(B,D),(C,E)\}$.

The last pair of variables is (D,E) and the adjacency power of D (not including E), i.e., $\mathbf{Z} = \{ \text{ } \}$. For this example, the null hypothesis does not hold; thus, there is no variable or combination of them that can d-separate D from E. So, we delete this pair of variables from \mathbf{W} and add this pair to the set \mathbf{X} . Now $\mathbf{X} = \{(A,B), (B,C),(B,D),(C,E),(D,E)\}$.

Figure A.3 shows the result given by step 2 of procedure Bayes9.

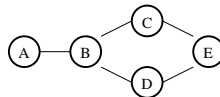


Figure A.3: Step 2 of procedure Bayes9

Step 3 of procedure Bayes9

3. Let $Y_{(a)} \in \mathbf{X}$, $Y_{(b)} \in \mathbf{X}$ and $Y_{(c)} \in \mathbf{X}$ be three different variables. If $Y_{(a)} - Y_{(b)} \in \mathbf{X}$, $Y_{(b)} - Y_{(c)} \in \mathbf{X}$, $Y_{(a)} - Y_{(c)} \in \mathbf{X}$ and $Y_{(b)}$ is not in the d-separation set of (a,c), then orient the triplet $Y_{(a)} - Y_{(b)} - Y_{(c)}$ as the directed triplet $Y_{(a)} \rightarrow Y_{(b)} \rightarrow Y_{(c)}$

End of step 3 of procedure Bayes9

For this example, rule in step 3 can be perfectly applied to the pattern produced by procedure Bayes9. In this case, $(A,B) \perp\!\!\!\perp X$, $(B,C) \perp\!\!\!\perp X$, $(A,C) \perp\!\!\!\perp X$ but (B) is in the d-separation set of (A,C) ; then it is not possible to direct this triplet. However, two arcs can be oriented: from C to E and from D to E . This is because $(B,C) \perp\!\!\!\perp X$, $(C,E) \perp\!\!\!\perp X$, $(B,E) \perp\!\!\!\perp X$ and (C) alone is not in the d-separation set of (B,E) whereas $(B,D) \perp\!\!\!\perp X$, $(D,E) \perp\!\!\!\perp X$, $(B,E) \perp\!\!\!\perp X$ and (D) alone is not in the d-separation set of (B,E) .

Now, figure A.4 shows the result given by step 3 of procedure Bayes9.

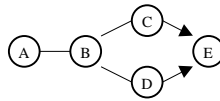


Figure A.4: Step 3 of procedure Bayes9

Step 4 of procedure Bayes9

4. Do

If $Y_{(b)} - Y_{(c)} \perp\!\!\!\perp X$ and $Y_{(a)} - Y_{(b)}$ exist and $Y_{(a)} - Y_{(c)} \perp\!\!\!\perp X$, then orient $Y_{(b)} - Y_{(c)}$ as $Y_{(b)} \rightarrow Y_{(c)}$

If $Y_{(a)} - Y_{(b)} \perp\!\!\!\perp X$ and $Y_{(a)} - Y_{(c)} \rightarrow Y_{(b)}$ exists, then orient $Y_{(a)} - Y_{(b)}$ as $Y_{(a)} \rightarrow Y_{(b)}$

while no more arcs can be oriented

End of step 4 of procedure Bayes9

The only step that cannot be applied for this particular example is step 4. However, the example presented in this appendix provides a very good indication regarding the way procedure Bayes9 builds Bayesian networks from data.

Bibliography

- Bouckaert, R. R. (1994). Probabilistic Network Construction using the Minimum Description Length Principle, Technical Report RUU-CS-94-27, Utrecht University.
- Bozdogan, H. (2000). "Akaike's Information Criterion and Recent Developments in Information Complexity." Journal of Mathematical Psychology **44**: 62-91.
- Bryant, P. and Squire, S. (2001). Children's Mathematics: Lost and Found in Space. Spatial Schemas and Abstract Thought. M. Gattis, MIT Press: 175-200.
- Browne, M. W. (2000). "Cross-Validation Methods." Journal of Mathematical Psychology **44**: 108-132.
- Bullock, M., R. Gelman, et al. (1982). The Development of Causal Reasoning. The Developmental Psychology of Time. W. J. Friedman. New York, Academic Press: 209-254.
- Buntine, W. (1996). "A guide to the literature on learning Probabilistic Networks from data." IEEE Transactions on Knowledge and Data Engineering **8**: 195-210.
- Busemeyer, J. R. and Y.-M. Wang (2000). "Model Comparisons and Model Selection Based on Generalization Criterion Methodology." Journal of Mathematical Psychology **44**: 171-189.
- Chaitin, G. J. (1975). Randomness and Mathematical Proof. Scientific American. **232**: 47-52.

- Chase, V. M. (1999). Where to look to find out why: Rational Information search in Causal Hypothesis Testing. PhD Thesis, The University of Chicago, Chicago, Illinois, 1999.
- Cheng, J. (1998). Learning Bayesian Networks from data: An information theory based approach, PhD Thesis, Faculty of Informatics, University of Ulster, Jordanstown, United Kingdom, 1998.
- Cheng, J., D. Bell, et al. (1998). Learning Bayesian Networks from Data: An Efficient Approach based on Information Theory, 1998, Electronic source: <http://www.cs.ualberta.ca/~jcheng/Doc/report98.pdf>.
- Cheng, P. W. (1993). "Separating Causal Laws from Casual Facts: Pressing the Limits of Statistical Relevance." The Psychology of Learning and Motivation **30**: 215-264.
- Cheng, P. W. (1997). "From Covariation to Causation: A Causal Power Theory." Psychological Review **104**(2): 367-405.
- Chickering, D. M. (1996). Learning Bayesian Networks from Data, Technical Report R-245, Computer Science, Cognitive Systems Laboratory. Los Angeles, California, University of California, Los Angeles, 1996.
- Chickering, D. M., D. Heckerman, et al. (1997). A Bayesian Approach to Learning Bayesian Networks with Local Structure, Technical Report MSR-TR-97-07, Microsoft Research, Redmond, Washington, 1997.
- Cooper, G. F. (1999). An Overview of the Representation and Discovery of Causal Relationships using Bayesian Networks. Computation, Causation & Discovery. C. Glymour and G. F. Cooper, AAAI Press / MIT Press: 3-62.

- Cooper, G. F. and E. Herskovits (1992). "A Bayesian Method for the Induction of Probabilistic Networks from Data." Machine Learning **9**: 309-347.
- Cross, S. S., J. Downs, et al. (2000). Which Decision Support Technologies Are Appropriate for the Cytodiagnosis of Breast Cancer? Artificial Intelligence Techniques in Breast Cancer Diagnosis and Prognosis. A. Jain, A. Jain, S. Jain and L. Jain, World Scientific. **39**: 265-295.
- Cross, S. S., A. K. Dube, et al. (1998). "Evaluation of a statistically derived decision tree for the cytodiagnosis of fine needle aspirates of the breast (FNAB)." Cytopathology **8**: 178-187.
- Cross, S. S., T. J. Stephenson, et al. (2000). "Validation of a decision support system for the cytodiagnosis of fine needle aspirates of the breast using a prospectively collected dataset from multiple observers in a working clinical environment." Cytopathology(11): 503-512.
- Cruz-Ramirez, N. (1997). Un algoritmo para generar redes probabilistas a partir de datos estadísticos aplicadas a problemas de clasificación y/o pronóstico, Tesis de Maestría, Maestría en Inteligencia Artificial, Universidad Veracruzana, Xalapa, Veracruz, Mexico, 1997.
- Cruz-Ramirez, N. and M. Martinez-Morales (1997). Un algoritmo para generar redes Bayesianas a partir de datos estadísticos. Primer Encuentro Nacional de Computación, ENC 97, Querétaro.
- Cutting, J. E. (2000). "Accuracy, Scope and Flexibility of Models." Journal of Mathematical Psychology **44**: 3-19.
- Diez, F. J. (1996). DIAVAL, sistema experto bayesiano para ecocardiografía. V Congreso Iberoamericano de Inteligencia Artificial IBERAMIA 96, Cholula, Puebla, Mexico.

Diez, F. J. and E. Nell (1998/99). Introduccion al Razonamiento Aproximado. Madrid, España, Universidad Nacional de Educacion a Distancia, 1998/1999, Electronic Source: <http://www.dia.uned.es/~fjdiez/libros/razaprox.html>.

Edwards, D. (1995). Introduction to Graphical Modelling. New York, Springer-Verlag.

Einhorn, H. J. and R. M. Hogarth (1986). "Judging Probable Cause." Psychological Bulletin **99**(1): 3-19.

Emmorey, Karen (2001). Space on Hand: The Exploitation of Signing Space to Illustrate Abstract Thought. Spatial Schemas and Abstract Thought. M. Gattis, MIT Press: 147-174.

Forster, M. R. (2000). "Key Concepts in Model Selection: Performance and Generalizability." Journal of Mathematical Psychology **44**: 205-231.

Frey, B. J. (1998). Graphical Models for Machine Learning and Digital Communication, MIT Press.

Friedman, N., D. Geiger, et al. (1997). "Bayesian Network Classifiers." Machine Learning **29**: 131-163.

Friedman, N. and M. Goldszmidt (1998a). Learning Bayesian Networks from Data, University of California, Berkeley and Stanford Research Institute, 1998, Electronic source: <http://www.cs.berkeley.edu/~nir/Tutorial>.

Friedman, N. and M. Goldszmidt (1998b). Learning Bayesian Networks with Local Structure. Learning in Graphical Models. M. I. Jordan, Kluwer Academic: 421-459.

Friedman, N. and Z. Yakhini (1996). "On the Sample Complexity of Learning Bayesian Networks", Twelfth Conference on Uncertainty in Artificial Intelligence.

- Gattis, M., Ed. (2001). Spatial Schemas and Abstract Thought. The MIT Press.
- Geiger, D., D. Heckerman, et al. (1998). Asymptotic Model Selection for Directed Networks with Hidden Variables. Learning in Graphical Models. M. I. Jordan, Kluwer Academic: 461-477.
- Glasgow, J., Narayanan, N. Hary and Chandrasekaran, B., Eds. (1995). Diagrammatic reasoning: Cognitive and Computational Perspectives. AAAI Press / MIT Press.
- Glymour, C. and G. F. Cooper, Eds. (1999). Computation, Causation & Discovery, AAAI Press / MIT Press.
- Glymour, C., P. Spirtes, et al. (1999a). On the Possibility of Inferring Causation from Association without Background Knowledge. Computation, Causation & Discovery. C. Glymour and G. F. Cooper, AAAI Press / MIT Press: 323-331.
- Glymour, C., P. Spirtes, et al. (1999b). Response to Rejoinder. Computation, Causation & Discovery. C. Glymour and G. F. Cooper, AAAI Press / MIT Press: 343-345.
- Grunwald, P. (2000). "Model Selection Based on Minimum Description Length." Journal of Mathematical Psychology **44**: 133-152.
- Hamilton, P. W., N. Anderson, et al. (1994). "Expert system support using Bayesian belief networks in the diagnosis of fine needle aspiration biopsy specimens of the breast." Journal of Clinical Pathology **47**: 329-336.
- Hamilton, P. W., P. H. Bartels, et al. (1995). "Editorial. Improved Diagnostic Decision-Making in Pathology: Do Inference Networks hold the key?" Journal of Pathology **175**: 1-5.

- Han, J. and Kamber, M. (2001). Data Mining. Concepts and Techniques, Morgan Kaufmann.
- Heckerman, D. (1997). A Tutorial on Learning with Bayesian Networks, Technical Report MSR-TR-95-06, Microsoft Research, Redmond, Washington, 1997.
- Heckerman, D. (1998). A Tutorial on Learning with Bayesian Networks. Learning in Graphical Models. M. I. Jordan, MIT Press: 301-354.
- Heckerman, D., D. Geiger, et al. (1994). Learning Bayesian Networks: The combination of knowledge and statistical data, Technical Report MSR-TR-94-09, Microsoft Research, Redmond, Washington, 1994.
- Heckerman, D., A. Mandani, et al. (1995). Real-World Applications of Bayesian Networks. Communications of the ACM. **38**: 24-26.
- Hines, W. W. and D. C. Montgomery (1997). Probabilidad y Estadística para Ingeniería y Administración. Mexico, Mexico, Compañía Editorial Continental, S.A. de C.V.
- Hofstadter, D. R. (1999). Godel, Escher, Bach: An Eternal Golden Braid, Penguin.
- Howell, D. C. (1997). Statistical Methods for Psychology, Duxbury Press.
- Hummel, John E. and Holyoak, Keith J. (2001). A Process Model of Human Transitive Inference. Spatial Schemas and Abstract Thought. M. Gattis, MIT Press: 279-305.
- Jackson, P. (1990). Introduction to Expert Systems, Addison-Wesley.
- Jensen, F. V. (1996). An Introduction to Bayesian Networks, UCL Press.

- Jordan, M. I., Ed. (1998). Learning in Graphical Models. Adaptive Computation and Machine Learning, Kluwer Academic.
- Keim, D. A. (2001). Visual Exploration of Large Data Sets. Communications of the ACM. **44**: 39-44.
- Kohavi, R. (1995a). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. 14th International Joint Conference on Artificial Intelligence IJCAI'95, Montreal, Canada, Morgan Kaufmann.
- Kohavi, R. (1995b). Wrappers for Performance Enhancement and Oblivious Decision Graphs, PhD Thesis, Department of Computer Science. Stanford University, Palo Alto, California, 1995b.
- Kullback, S. (1959). Information Theory and Statistics. Dover, New York.
- Larkin, Jill H. and Simon, Herbert A. (1995). Why a Diagram is (sometimes) worth ten thousand words. Diagrammatic reasoning: Cognitive and Computational Perspectives. Janice Glasgow, N. Hary Narayanan and B. Chandrasekaran, AAAI Press / MIT Press: 69-109.
- Lauritzen, S. L. (1996). Graphical Models. Oxford, United Kingdom, Oxford University Press.
- Li, M. and P. Vitányi (1993). An Introduction to Kolmogorov Complexity and its Applications, Springer-Verlag.
- Liben, Lynn S. (2001). Thinking through Maps. Spatial Schemas and Abstract Thought. M. Gattis, MIT Press: 45-77.

- Lien, Y. and P. W. Cheng (2000). "Distinguishing genuine from spurious causes: a coherence hypothesis." Cognitive Psychology(40): 87-137.
- Linhart, H. and W. Zucchini (1986). Model Selection, John Wiley & Sons.
- Luger, G. F. and W. A. Stubblefield (1993). Artificial Intelligence: Structures and Strategies for Complex Problem Solving, The Benjamin/Cummings Publishing Company, Inc.
- MacKay, D. (1995). A Short Course in Information Theory. Cambridge, Cavendish Laboratory, University of Cambridge, Cambridge, United Kingdom, 1995, Electronic source: <http://131.111.48.24/pub/mackay/info-theory/course.html>
- Marcus, G. F. (1998). "Rethinking Eliminative Connectionism." Cognitive Psychology **37**: 243-282.
- Marr, D. (1982). Vision: A Computational Investigation into the Human Representation and Processing of Visual Information, W.H. Freeman & Company.
- Martinez-Morales, M. (1995). An Algorithm for the Induction of Probabilistic Networks from Data. XII Reunion Nacional de Inteligencia Artificial, ITESM, Cuernavaca, Morelos, Mexico, Limusa.
- McCarthy, J. (2000). What is Artificial Intelligence? 2000, Electronic source: <http://www-formal.stanford.edu/jmc/whatisai/whatisai.html>
- McGonigle, Brendan and Chalmers, Margaret (2001). Spatial Representation as Cause and Effect: Circular Causality comes to Cognition. Spatial Schemas and Abstract Thought. M. Gattis, MIT Press: 247-277
- Monti, S. and G. F. Cooper (1998). Learning Hybrid Bayesian Networks from Data. Learning in Graphical Models. M. I. Jordan, Kluwer Academic: 521-540.

- Myung, I. J., M. R. Forster, et al. (2000). "Guest Editors' Introduction, Special Issue on Model Selection." Journal of Mathematical Psychology **44**: 1-2.
- Neapolitan, R. E. (1990). Probabilistic Reasoning in Expert Systems. Theory and Algorithms, John Wiley & Sons, Inc.
- Neapolitan, R. E., S. B. Morris, et al. (1997). Cognitive Processing of Causal Knowledge. 13th Annual Conference on Uncertainty in Artificial Intelligence, Providence, RI, USA, Morgan Kaufmann.
- Norsys (2001) Norsys Software Corporation, 2001, Electronic source: <http://www.norsys.com>.
- Pearl, J. (1988). Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. San Mateo, California, Morgan Kaufmann.
- Pearl, J. (1996). "Structural and Probabilistic Causality." The Psychology of Learning and Motivation **34**: 393-435.
- Pearl, J. (2000). Causality. Models, Reasoning and Inference, Cambridge University Press.
- Perales, J. C. and D. R. Shanks (2001). "Normative and Descriptive Accounts of the Influence of Power and Contingency on Causal Judgment." Manuscript submitted for publication.
- Plach, M. (1997). Using Bayesian Networks to model probabilistic inferences about the likelihood of traffic congestions. Engineering Psychology and Cognitive Ergonomics. D. Harris. **1**: 363-371.
- Plach, M. (1999). "Bayesian Networks as Models of Human Judgement under Uncertainty: The case of belief updating." Kognitionswissenschaft(8): 30-39.

Rissanen, J. (1989). Stochastic Complexity in Statistical Enquiry.

Robins, J. M. and L. Wasserman (1999a). On the Impossibility of Inferring Causation from Association without Background Knowledge. Computation, Causation & Discovery. C. Glymour and G. F. Cooper, AAAI Press / MIT Press: 305-321.

Robins, J. M. and L. Wasserman (1999b). Rejoinder to Glymour, Spirtes and Richardson. Computation, Causation & Discovery. C. Glymour and G. F. Cooper, AAAI Press / MIT Press: 333-342.

Russell, S. and P. Norvig (1994). Artificial Intelligence: A Modern Approach, Prentice Hall.

Scheines, R. (1999b). Estimating Latent Causal Influences: TETRAD II Model Selection and Bayesian Parameter Estimation. Computation, Causation & Discovery. C. Glymour and G. F. Cooper, AAAI Press / MIT Press: 425-437.

Scheines, R., C. Glymour, et al. (1999a). Truth is Among the Best Explanations: Finding Causal Explanations of Conditional Independence and Dependence. Computation, Causation & Discovery. C. Glymour and G. F. Cooper, AAAI Press / MIT Press: 167-209.

Schneider, T. D. (1995). Information Theory Primer with an appendix on logarithms, 1995, Electronic source: <http://www.lecb.ncifcrf.gov/~toms/paper/primer/>

Shannon, C. E. and W. Weaver (1949). The mathematical theory of communication. Urbana, University of Illinois Press.

Spiegelhalter, D. J., A. P. Dawid, et al. (1993). "Bayesian Analysis in Expert Systems." Statistical Science 8(3): 219-247.

- Spirtes, P., C. Glymour, et al. (1993). Causation, Prediction and Search, Springer-Verlag.
- Spirtes, P. and C. Meek (1995). Learning Bayesian Networks with Discrete Variables from Data. First International Conference on Knowledge Discovery and Data Mining.
- Spirtes, P., R. Scheines, et al. (1994). Tetrad II: Tools for Causal Modeling. Hillsdale, New Jersey, United States of America, Lawrence Erlbaum Associates, Inc.
- Sucar, L. E. (1994). Structure and Parameter Learning in Probabilistic Networks. XI Reunion Nacional de Inteligencia Artificial, Universidad de Guadalajara, Guadalajara, Jalisco, Mexico, Limusa.
- Sucar, L. E. and M. Martinez-Arroyo (1998). Interactive Structural Learning of Bayesian Networks. World Congress on Expert Systems.
- Barbara, Tversky (2001). Spatial Schemas in Depictions. Spatial Schemas and Abstract Thought. M. Gattis, MIT Press: 79-112.
- Waldmann, M. R. (1996). "Knowledge-Based Causal Induction." The Psychology of Learning and Motivation **34**: 47-88.
- Waldmann, M. R. and Y. Hagmayer (2001). "Estimating Causal Strength: The Role of Structural Knowledge and Processing Effort." Manuscript submitted for publication.
- Waldmann, M. R. and L. Martignon (1998). A Bayesian Network Model of Causal Learning. Twentieth Annual Conference of the Cognitive Science Society.
- Walker, A. J., S. S. Cross, et al. (1999). "Visualisation of biomedical datasets by use of growing cell structure networks: a novel diagnostic classification technique." The Lancet **354**: 1518-1521.

Wasserman, L. (2000). "Bayesian Model Selection and Model Averaging." Journal of Mathematical Psychology **44**: 92-107.

Whittaker, J. (1990). Graphical Models in Applied Mathematical Multivariate Statistics, John Wiley & Sons.

Winston, P. H. (1992). Artificial Intelligence, Addison-Wesley.

Zucchini, W. (2000). "An Introduction to Model Selection." Journal of Mathematical Psychology **44**: 41-61.