3 The Bayesian Network Representation

Our goal is to represent a joint distribution P over some set of random variables $\mathcal{X} = \{X_1, \ldots, X_n\}$. Even in the simplest case where these variables are binary-valued, a joint distribution requires the specification of $2^n - 1$ numbers — the probabilities of the 2^n different assignments of values $x_1, \ldots, x_n$. For all but the smallest n, **the explicit representation of the joint distribution is unmanageable from every perspective. Computationally, it is very expensive to manipulate and generally too large to store in memory. Cognitively, it is impossible to acquire so many numbers from a human experts; moreover, the numbers are very small and do not correspond to events that people can reasonably contemplate. Statistically, if we want to learn the distribution from data, we would need ridiculously large amounts of data to estimate this many parameters robustly. These problems were the main barrier to the adoption of probabilistic methods for expert systems until the development of the methodologies described in this book.**

In this chapter, we first show how independence properties in the distribution can be used to represent such high-dimensional distributions much more compactly. We then show how a combinatorial data structure — a directed acyclic graph — can provide us with a general-purpose modeling language for exploiting this type of structure in our representation.

3.1 Exploiting Independence Properties

The compact representations we explore in this chapter are based on two key ideas: the representation of independence properties of the distribution, and the use of an alternative parameterization that allows us to exploit these finer-grained independencies.

3.1.1 Independent Random Variables

To motivate our discussion, consider a simple setting where we know that each X_i represents the outcome of a toss of coin i. In this case, we typically assume that the different coin tosses are marginally independent (definition 2.4), so that our distribution P will satisfy $(X_i \perp X_j)$ for any i, j. More generally (strictly more generally — see exercise 3.1), we assume that the distribution satisfies $(\boldsymbol{X} \perp \boldsymbol{Y})$ for any disjoint subsets of the variables $\boldsymbol{X}$ and $\boldsymbol{Y}$. Therefore, we have that:

$$P(X_1, \ldots, X_n) = P(X_1)P(X_2)\cdots P(X_n).$$

If we use the standard parameterization of the joint distribution, this independence structure is obscured, and the representation of the distribution requires 2^n parameters. However, we can use a more natural set of parameters for specifying this distribution: If θ_i is the probability with which coin i lands heads, the joint distribution P can be specified using the n *parameters* $\theta_1, \ldots, \theta_n$. These parameters implicitly specify the 2^n probabilities in the joint distribution. For example, the probability that all of the coin tosses land heads is simply $\theta_1 \cdot \theta_2 \cdot \ldots \cdot \theta_n$. More generally, letting $\theta_{x_i} = \theta_i$ when $x_i = x_i^1$ and $\theta_{x_i} = 1 - \theta_i$ when $x_i = x_i^0$, we can define:

parameters

$$P(x_1, \ldots, x_n) = \prod_i \theta_{x_i}. \tag{3.1}$$

This representation is limited, and there are many distributions that we cannot capture by choosing values for $\theta_1, \ldots, \theta_n$. This fact is obvious not only from intuition, but also from a somewhat more formal perspective. The space of all joint distributions is a $2^n - 1$ dimensional subspace of $\mathbb{R}^{2^n}$ — the set $\{(p_1, \ldots, p_{2^n}) \in \mathbb{R}^{2^n} : p_1 + \ldots + p_{2^n} = 1\}$. On the other hand, the space of all joint distributions specified in a factorized way as in equation (3.1) is an n-dimensional manifold in $\mathbb{R}^{2^n}$.

independent parameters

A key concept here is the notion of *independent parameters* — parameters whose values are not determined by others. For example, when specifying an arbitrary multinomial distribution over a k dimensional space, we have $k - 1$ independent parameters: the last probability is fully determined by the first $k - 1$. In the case where we have an arbitrary joint distribution over n binary random variables, the number of independent parameters is $2^n - 1$. On the other hand, the number of independent parameters for distributions represented as n independent binomial coin tosses is n. Therefore, the two spaces of distributions cannot be the same. (While this argument might seem trivial in this simple case, it turns out to be an important tool for comparing the expressive power of different representations.)

As this simple example shows, certain families of distributions — in this case, the distributions generated by n independent random variables — permit an alternative parameterization that is substantially more compact than the naive representation as an explicit joint distribution. Of course, in most real-world applications, the random variables are not marginally independent. However, a generalization of this approach will be the basis for our solution.

3.1.2 The Conditional Parameterization

Let us begin with a simple example that illustrates the basic intuition. Consider the problem faced by a company trying to hire a recent college graduate. The company's goal is to hire intelligent employees, but there is no way to test intelligence directly. However, the company has access to the student's SAT scores, which are informative but not fully indicative. Thus, our probability space is induced by the two random variables *Intelligence* (I) and *SAT* (S). For simplicity, we assume that each of these takes two values: $Val(I) = \{i^1, i^0\}$, which represent the values high intelligence (i^1) and low intelligence (i^0); similarly $Val(S) = \{s^1, s^0\}$, which also represent the values *high* (score) and *low* (score), respectively.

Thus, our joint distribution in this case has four entries. For example, one possible joint

distribution P would be

I	S	$P(I,S)$
i^0	s^0	0.665
i^0	s^1	0.035
i^1	s^0	0.06
i^1	s^1	0.24.

(3.2)

There is, however, an alternative, and even more natural way of representing the same joint distribution. Using the chain rule of conditional probabilities (see equation (2.5)), we have that

$$P(I,S) = P(I)P(S \mid I).$$

Intuitively, we are representing the process in a way that is more compatible with causality. Various factors (genetics, upbringing, ...) first determined (stochastically) the student's intelligence. His performance on the SAT is determined (stochastically) by his intelligence. We note that the models we construct are not required to follow causal intuitions, but they often do. We return to this issue later on.

From a mathematical perspective, this equation leads to the following alternative way of representing the joint distribution. Instead of specifying the various joint entries $P(I,S)$, we would specify it in the form of $P(I)$ and $P(S \mid I)$. Thus, for example, we can represent the joint distribution of equation (3.2) using the following two tables, one representing the *prior distribution* over I and the other the *conditional probability distribution (CPD)* of S given I:

prior distribution

CPD

i^0	i^1
0.7	0.3

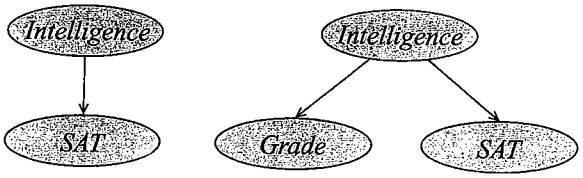
I	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

(3.3)

The CPD $P(S \mid I)$ represents the probability that the student will succeed on his SATs in the two possible cases: the case where the student's intelligence is low, and the case where it is high. The CPD asserts that a student of low intelligence is extremely unlikely to get a high SAT score ($P(s^1 \mid i^0) = 0.05$); on the other hand, a student of high intelligence is likely, but far from certain, to get a high SAT score ($P(s^1 \mid i^1) = 0.8$).

It is instructive to consider how we could parameterize this alternative representation. Here, we are using three binomial distributions, one for $P(I)$, and two for $P(S \mid i^0)$ and $P(S \mid i^1)$. Hence, we can parameterize this representation using three independent parameters, say θ_{i^1}, $\theta_{s^1 \mid i^1}$, and $\theta_{s^1 \mid i^0}$. Our representation of the joint distribution as a four-outcome multinomial also required three parameters. Thus, although the conditional representation is more natural than the explicit representation of the joint, it is not more compact. However, as we will soon see, the conditional parameterization provides a basis for our compact representations of more complex distributions.

Although we will only define Bayesian networks formally in section 3.2.2, it is instructive to see how this example would be represented as one. The Bayesian network, as shown in figure 3.1a, would have a node for each of the two random variables I and S, with an edge from I to S representing the direction of the dependence in this model.



(a) (b)

Figure 3.1 Simple Bayesian networks for the student **example**

3.1.3 The Naive Bayes Model

We now describe perhaps the simplest example where a conditional parameterization is combined with conditional independence assumptions to produce a very compact representation of a high-dimensional probability distribution. Importantly, unlike the previous example of fully independent random variables, none of the variables in this distribution are (marginally) independent.

3.1.3.1 The Student Example

Elaborating our example, we now assume that the company also has access to the student's grade G in some course. In this case, our probability space is the joint distribution over the three relevant random variables I, S, and G. Assuming that I and S are as before, and that G takes on three values g^1, g^2, g^3, representing the grades A, B, and C, respectively, then the joint distribution has twelve entries.

Before we even consider the specific numerical aspects of our distribution P in this example, we can see that independence does not help us: for any reasonable P, there are no independencies that hold. The student's intelligence is clearly correlated both with his SAT score and with his grade. The SAT score and grade are also not independent: if we condition on the fact that the student received a high score on his SAT, the chances that he gets a high grade in his class are also likely to increase. Thus, we may assume that, for our particular distribution P, $P(g^1 \mid s^1) > P(g^1 \mid s^0)$.

However, it is quite plausible that our distribution P in this case satisfies a conditional independence property. If we know that the student has high intelligence, a high grade on the SAT no longer gives us information about the student's performance in the class. More formally:

$$P(g \mid i^1, s^1) = P(g \mid i^1).$$

More generally, we may well assume that

$$P \models (S \perp G \mid I). \tag{3.4}$$

Note that this independence statement holds only if we assume that the student's intelligence is the only reason why his grade and SAT score might be correlated. In other words, it assumes that there are no correlations due to other factors, such as the student's ability to take timed exams. These assumptions are also not "true" in any formal sense of the word, and they are often only approximations of our true beliefs. (See box 3.C for some further discussion.)

As in the case of marginal independence, conditional independence allows us to provide a compact specification of the joint distribution. Again, the compact representation is based on a very natural alternative parameterization. By simple probabilistic reasoning (as in equation (2.5)), we have that

$$P(I, S, G) = P(S, G \mid I)P(I).$$

But now, the conditional independence assumption of equation (3.4) implies that

$$P(S, G \mid I) = P(S \mid I)P(G \mid I).$$

Hence, we have that

$$P(I, S, G) = P(S \mid I)P(G \mid I)P(I). \tag{3.5}$$

Thus, we have factorized the joint distribution $P(I, S, G)$ as a product of three conditional probability distributions (CPDs). This factorization immediately leads us to the desired alternative parameterization. In order to specify fully a joint distribution satisfying equation (3.4), we need the following three CPDs: $P(I)$, $P(S \mid I)$, and $P(G \mid I)$. The first two might be the same as in equation (3.3). The latter might be

I	g^1	g^2	g^3
i^0	0.2	0.34	0.46
i^1	0.74	0.17	0.09

Together, these three CPDs fully specify the joint distribution (assuming the conditional independence of equation (3.4)). For example,

$$\begin{aligned} P(i^1, s^1, g^2) &= P(i^1)P(s^1 \mid i^1)P(g^2 \mid i^1) \\ &= 0.3 \cdot 0.8 \cdot 0.17 = 0.0408. \end{aligned}$$

Once again, we note that this probabilistic model would be represented using the Bayesian network shown in figure 3.1b.

In this case, the alternative parameterization is more compact than the joint. We now have three binomial distributions — $P(I)$, $P(S \mid i^1)$ and $P(S \mid i^0)$, and two three-valued multinomial distributions — $P(G \mid i^1)$ and $P(G \mid i^0)$. Each of the binomials requires one independent parameter, and each three-valued multinomial requires two independent parameters, for a total of seven. By contrast, our joint distribution has twelve entries, so that eleven independent parameters are required to specify an arbitrary joint distribution over these three variables.

It is important to note another advantage of this way of representing the joint: modularity. When we added the new variable G, the joint distribution changed entirely. Had we used the explicit representation of the joint, we would have had to write down twelve new numbers. In the factored representation, we could reuse our local probability models for the variables I and S, and specify only the probability model for G — the CPD $P(G \mid I)$. This property will turn out to be invaluable in modeling real-world systems.

3.1.3.2 The General Model

naive Bayes

This example is an instance of a much more general model commonly called the *naive Bayes*

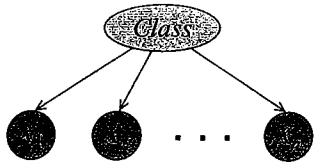


Figure 3.2 The Bayesian network graph for a naive Bayes model

model (also known as the *Idiot Bayes model*). The naive Bayes model assumes that instances fall into one of a number of mutually exclusive and exhaustive *classes*. Thus, we have a class variable C that takes on values in some set $\{c^1, \ldots, c^k\}$. In our example, the class variable is the student's intelligence I, and there are two classes of instances — students with high intelligence and students with low intelligence.

features

The model also includes some number of *features* $X_1, \ldots, X_k$ whose values are typically observed. The *naive Bayes assumption* is that the features are conditionally independent given the instance's class. In other words, within each class of instances, the different properties can be determined independently. More formally, we have that

$$(X_i \perp \boldsymbol{X}_{-i} \mid C) \quad \text{for all } i, \tag{3.6}$$

where $\boldsymbol{X}_{-i} = \{X_1, \ldots, X_k\} - \{X_i\}$. This model can be represented using the Bayesian network of figure 3.2. In this example, and later on in the book, we use a darker oval to represent variables that are always observed when the network is used.

factorization

Based on these independence assumptions, we can show that the model *factorizes* as:

$$P(C, X_1, \ldots, X_n) = P(C) \prod_{i=1}^{n} P(X_i \mid C). \tag{3.7}$$

(See exercise 3.2.) Thus, in this model, we can represent the joint distribution using a small set of factors: a prior distribution $P(C)$, specifying how likely an instance is to belong to different classes a priori, and a set of CPDs $P(X_j \mid C)$, one for each of the n finding variables. These factors can be encoded using a very small number of parameters. For example, if all of the variables are binary, the number of independent parameters required to specify the distribution is $2n+1$ (see exercise 3.6). Thus, the number of parameters is linear in the number of variables, as opposed to exponential for the explicit representation of the joint.

Box 3.A — Concept: The Naive Bayes Model. *The naive Bayes model, despite the strong assumptions that it makes, is often used in practice, because of its simplicity and the small number of parameters required. The model is generally used for* classification *— deciding, based on the values of the evidence variables for a given instance, the class to which the instance is most likely to belong. We might also want to compute our confidence in this decision, that is, the extent to which our model favors one class c^1 over another c^2. Both queries can be addressed by the following ratio:*

classification

$$\frac{P(C = c^1 \mid x_1, \ldots, x_n)}{P(C = c^2 \mid x_1, \ldots, x_n)} = \frac{P(C = c^1)}{P(C = c^2)} \prod_{i=1}^{n} \frac{P(x_i \mid C = c^1)}{P(x_i \mid C = c^2)}; \tag{3.8}$$

see exercise 3.2). This formula is very natural, since it computes the posterior probability ratio of c^1 *versus* c^2 *as a product of their prior probability ratio (the first term), multiplied by a set of terms* $\frac{P(x_i|C=c^1)}{P(x_i|C=c^2)}$ *that measure the relative support of the finding* x_i *for the two classes.*

medical diagnosis

This model was used in the early days of medical diagnosis, *where the different values of the class variable represented different diseases that the patient could have. The evidence variables represented different symptoms, test results, and the like. Note that the model makes several strong assumptions that are not generally true, specifically that the patient can have at most one disease, and that, given the patient's disease, the presence or absence of different symptoms, and the values of different tests, are all independent. This model was used for medical diagnosis because the small number of interpretable parameters made it easy to elicit from experts. For example, it is quite natural to ask of an expert physician what the probability is that a patient with pneumonia has high fever. Indeed, several early medical diagnosis systems were based on this technology, and some were shown to provide better diagnoses than those made by expert physicians.*

However, later experience showed that the strong assumptions underlying this model decrease its diagnostic accuracy. In particular, the model tends to overestimate the impact of certain evidence by "overcounting" it. For example, both hypertension (high blood pressure) and obesity are strong indicators of heart disease. However, because these two symptoms are themselves highly correlated, equation (3.8), which contains a multiplicative term for each of them, double-counts the evidence they provide about the disease. Indeed, some studies showed that the diagnostic performance of a naive Bayes model degraded as the number of features increased; this degradation was often traced to violations of the strong conditional independence assumption. This phenomenon led to the use of more complex Bayesian networks, with more realistic independence assumptions, for this application (see box 3.D).

Nevertheless, the naive Bayes model is still useful in a variety of applications, particularly in the context of models learned from data in domains with a large number of features and a relatively small number of instances, such as classifying documents into topics using the words in the documents as features; see box 17.E).

3.2 Bayesian Networks

Bayesian networks build on the same intuitions as the naive Bayes model by exploiting conditional independence properties of the distribution in order to allow a compact and natural representation. However, they are not restricted to representing distributions satisfying the strong independence assumptions implicit in the naive Bayes model. They allow us the flexibility to tailor our representation of the distribution to the independence properties that appear reasonable in the current setting.

The core of the Bayesian network representation is a directed acyclic graph (DAG) $\mathcal{G}$, whose nodes are the random variables in our domain and whose edges correspond, intuitively, to direct influence of one node on another. **This graph $\mathcal{G}$ can be viewed in two very different ways:**

- **as a data structure that provides the skeleton for representing a joint distribution compactly in a factorized way;**

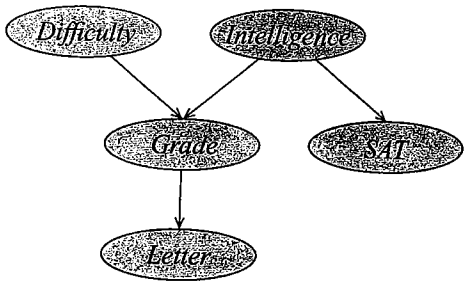


Figure 3.3 The Bayesian Network graph for the Student **example**

- **as a compact representation for a set of conditional independence assumptions about a distribution.**

As we will see, these two views are, in a strong sense, equivalent.

3.2.1 The Student Example Revisited

We begin our discussion with a simple toy example, which will accompany us, in various versions, throughout much of this book.

3.2.1.1 The Model

Consider our student from before, but now consider a slightly more complex scenario. The student's grade, in this case, depends not only on his intelligence but also on the difficulty of the course, represented by a random variable D whose domain is $Val(D) = \{easy, hard\}$. Our student asks his professor for a recommendation letter. The professor is absentminded and never remembers the names of her students. She can only look at his grade, and she writes her letter for him based on that information alone. The quality of her letter is a random variable L, whose domain is $Val(L) = \{strong, weak\}$. The actual quality of the letter depends stochastically on the grade. (It can vary depending on how stressed the professor is and the quality of the coffee she had that morning.)

We therefore have five random variables in this domain: the student's intelligence (I), the course difficulty (D), the grade (G), the student's SAT score (S), and the quality of the recommendation letter (L). All of the variables except G are binary-valued, and G is ternary-valued. Hence, the joint distribution has 48 entries.

As we saw in our simple illustrations of figure 3.1, a Bayesian network is represented using a directed graph whose nodes represent the random variables and whose edges represent direct influence of one variable on another. We can view the graph as encoding a generative sampling process executed by nature, where the value for each variable is selected by nature using a distribution that depends only on its parents. In other words, each variable is a stochastic function of its parents.

Based on this intuition, perhaps the most natural network structure for the distribution in this example is the one presented in figure 3.3. The edges encode our intuition about

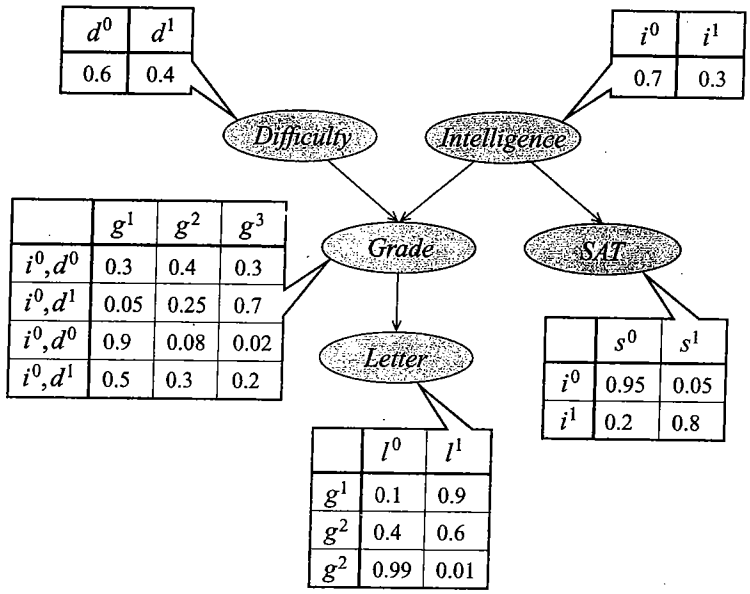


Figure 3.4 Student **Bayesian network $\mathcal{B}^{student}$ with CPDs**

the way the world works. The course difficulty and the student's intelligence are determined independently, and before any of the variables in the model. The student's grade depends on both of these factors. The student's SAT score depends only on his intelligence. The quality of the professor's recommendation letter depends (by assumption) only on the student's grade in the class. Intuitively, each variable in the model depends directly only on its parents in the network. We formalize this intuition later.

local probability model

The second component of the Bayesian network representation is a set of *local probability models* that represent the nature of the dependence of each variable on its parents. One such model, $P(I)$, represents the distribution in the population of intelligent versus less intelligent student. Another, $P(D)$, represents the distribution of difficult and easy classes. The distribution over the student's grade is a conditional distribution $P(G \mid I, D)$. It specifies the distribution over the student's grade, inasmuch as it depends on the student's intelligence and the course difficulty. Specifically, we would have a different distribution for each assignment of values i, d. For example, we might believe that a smart student in an easy class is 90 percent likely to get an A, 8 percent likely to get a B, and 2 percent likely to get a C. Conversely, a smart student in a hard class may only be 30 percent likely to get an A. In general, each variable X in the model is associated with a *conditional probability distribution (CPD)* that specifies a distribution over the values of X given each possible joint assignment of values to its parents in the model. For a node with no parents, the CPD is conditioned on the empty set of variables. Hence, the CPD turns into a marginal distribution, such as $P(D)$ or $P(I)$. One possible choice of CPDs for this domain is shown in figure 3.4. The network structure together with its CPDs is a *Bayesian network* $\mathcal{B}$; we use $\mathcal{B}^{student}$ to refer to the Bayesian network for our student example.

CPD

How do we use this data structure to specify the joint distribution? Consider some particular state in this space, for example, i^1, d^0, g^2, s^1, l^0. Intuitively, the probability of this event can be computed from the probabilities of the basic events that comprise it: the probability that the student is intelligent; the probability that the course is easy; the probability that a smart student gets a B in an easy class; the probability that a smart student gets a high score on his SAT; and the probability that a student who got a B in the class gets a weak letter. The total probability of this state is:

$$\begin{aligned} P(i^1, d^0, g^2, s^1, l^0) &= P(i^1)P(d^0)P(g^2 \mid i^1, d^0)P(s^1 \mid i^1)P(l^0 \mid g^2) \\ &= 0.3 \cdot 0.6 \cdot 0.08 \cdot 0.8 \cdot 0.4 = 0.004608. \end{aligned}$$

Clearly, we can use the same process for any state in the joint probability space. In general, we will have that

$$P(I, D, G, S, L) = P(I)P(D)P(G \mid I, D)P(S \mid I)P(L \mid G). \tag{3.9}$$

chain rule for Bayesian networks

This equation is our first example of the *chain rule for Bayesian networks* which we will define in a general setting in section 3.2.3.2.

3.2.1.2 Reasoning Patterns

A joint distribution $P_{\mathcal{B}}$ specifies (albeit implicitly) the probability $P_{\mathcal{B}}(\boldsymbol{Y} = \boldsymbol{y} \mid \boldsymbol{E} = \boldsymbol{e})$ of any event $\boldsymbol{y}$ given any observations $\boldsymbol{e}$, as discussed in section 2.1.3.3: We condition the joint distribution on the event $\boldsymbol{E} = \boldsymbol{e}$ by eliminating the entries in the joint inconsistent with our observation $\boldsymbol{e}$, and renormalizing the resulting entries to sum to 1; we compute the probability of the event $\boldsymbol{y}$ by summing the probabilities of all of the entries in the resulting posterior distribution that are consistent with $\boldsymbol{y}$. To illustrate this process, let us consider our $\mathcal{B}^{student}$ network and see how the probabilities of various events change as evidence is obtained.

Consider a particular student, George, about whom we would like to reason using our model. We might ask how likely George is to get a strong recommendation (l^1) from his professor in Econ101. Knowing nothing else about George or Econ101, this probability is about 50.2 percent. More precisely, let $P_{\mathcal{B}^{student}}$ be the joint distribution defined by the preceding BN; then we have that $P_{\mathcal{B}^{student}}(l^1) \approx 0.502$. We now find out that George is not so intelligent (i^0); the probability that he gets a strong letter from the professor of Econ101 goes down to around 38.9 percent; that is, $P_{\mathcal{B}^{student}}(l^1 \mid i^0) \approx 0.389$. We now further discover that Econ101 is an easy class (d^0). The probability that George gets a strong letter from the professor is now $P_{\mathcal{B}^{student}}(l^1 \mid i^0, d^0) \approx 0.513$. Queries such as these, where we predict the "downstream" effects of various factors (such as George's intelligence), are instances of *causal reasoning* or *prediction*.

causal reasoning

Now, consider a recruiter for Acme Consulting, trying to decide whether to hire George based on our previous model. A priori, the recruiter believes that George is 30 percent likely to be intelligent. He obtains George's grade record for a particular class Econ101 and sees that George received a C in the class (g^3). His probability that George has high intelligence goes down significantly, to about 7.9 percent; that is, $P_{\mathcal{B}^{student}}(i^1 \mid g^3) \approx 0.079$. We note that the probability that the class is a difficult one also goes up, from 40 percent to 62.9 percent.

Now, assume that the recruiter fortunately (for George) lost George's transcript, and has only the recommendation letter from George's professor in Econ101, which (not surprisingly) is

weak. The probability that George has high intelligence still goes down, but only to 14 percent: $P_{\mathcal{B}^{student}}(i^1 \mid l^0) \approx 0.14$. Note that if the recruiter has both the grade and the letter, we have the same probability as if he had only the grade: $P_{\mathcal{B}^{student}}(i^1 \mid g^3, l^0) \approx 0.079$; we will revisit this issue. Queries such as this, where we reason from effects to causes, are instances of *evidential reasoning* or *explanation*.

evidential reasoning

Finally, George submits his SAT scores to the recruiter, and astonishingly, his SAT score is high. The probability that George has high intelligence goes up dramatically, from 7.9 percent to 57.8 percent: $P_{\mathcal{B}^{student}}(i^1 \mid g^3, s^1) \approx 0.578$. Intuitively, the reason that the high SAT score outweighs the poor grade is that students with low intelligence are extremely unlikely to get good scores on their SAT, whereas students with high intelligence can still get C's. However, smart students are much more likely to get C's in hard classes. Indeed, we see that the probability that Econ101 is a difficult class goes up from the 62.9 percent we saw before to around 76 percent.

This last pattern of reasoning is a particularly interesting one. The information about the SAT gave us information about the student's intelligence, which, in conjunction with the student's grade in the course, told us something about the difficulty of the course. In effect, we have one causal factor for the *Grade* variable — *Intelligence* — giving us information about another — *Difficulty*.

Let us examine this pattern in its pure form. As we said, $P_{\mathcal{B}^{student}}(i^1 \mid g^3) \approx 0.079$. On the other hand, if we now discover that Econ101 is a hard class, we have that $P_{\mathcal{B}^{student}}(i^1 \mid g^3, d^1) \approx 0.11$. In effect, we have provided at least a partial explanation for George's grade in Econ101. To take an even more striking example, if George gets a B in Econ 101, we have that $P_{\mathcal{B}^{student}}(i^1 \mid g^2) \approx 0.175$. On the other hand, if Econ101 is a hard class, we get $P_{\mathcal{B}^{student}}(i^1 \mid g^2, d^1) \approx 0.34$. In effect we have *explained away* the poor grade via the difficulty of the class. **Explaining away is an instance of a general reasoning pattern called *intercausal reasoning*, where different causes of the same effect can interact. This type of reasoning is a very common pattern in human reasoning.** For example, when we have fever and a sore throat, and are concerned about mononucleosis, we are greatly relieved to be told we have the flu. Clearly, having the flu does not prohibit us from having mononucleosis. Yet, having the flu provides an alternative explanation of our symptoms, thereby reducing substantially the probability of mononucleosis.

explaining away

intercausal reasoning

This intuition of providing an alternative explanation for the evidence can be made very precise. As shown in exercise 3.3, if the flu deterministically causes the symptoms, the probability of mononucleosis goes down to its prior probability (the one prior to the observations of any symptoms). On the other hand, if the flu might occur without causing these symptoms, the probability of mononucleosis goes down, but it still remains somewhat higher than its base level. Explaining away, however, is not the only form of intercausal reasoning. The influence can go in any direction. Consider, for example, a situation where someone is found dead and may have been murdered. The probabilities that a suspect has motive and opportunity both go up. If we now discover that the suspect has motive, the probability that he has opportunity goes up. (See exercise 3.4.)

It is important to emphasize that, although our explanations used intuitive concepts such as cause and evidence, there is nothing mysterious about the probability computations we performed. They can be replicated simply by generating the joint distribution, as defined in equation (3.9), and computing the probabilities of the various events directly from that.

3.2.2 Basic Independencies in Bayesian Networks

As we discussed, a Bayesian network graph $\mathcal{G}$ can be viewed in two ways. In the previous section, we showed, by example, how it can be used as a skeleton data structure to which we can attach local probability models that together define a joint distribution. In this section, we provide a formal semantics for a Bayesian network, starting from the perspective that the graph encodes a set of conditional independence assumptions. We begin by understanding, intuitively, the basic conditional independence assumptions that we want a directed graph to encode. We then formalize these desired assumptions in a definition.

3.2.2.1 Independencies in the Student Example

In the Student example, we used the intuition that edges represent direct dependence. For example, we made intuitive statements such as "the professor's recommendation letter depends only on the student's grade in the class"; this statement was encoded in the graph by the fact that there are no direct edges into the L node except from G. This intuition, that "a node depends directly only on its parents," lies at the heart of the semantics of Bayesian networks.

We give formal semantics to this assertion using conditional independence statements. For example, the previous assertion can be stated formally as the assumption that L is conditionally independent of all other nodes in the network given its parent G:

$$(L \perp I, D, S \mid G). \tag{3.10}$$

In other words, once we know the student's grade, our beliefs about the quality of his recommendation letter are not influenced by information about any other variable. Similarly, to formalize our intuition that the student's SAT score depends only on his intelligence, we can say that S is conditionally independent of all other nodes in the network given its parent I:

$$(S \perp D, G, L \mid I). \tag{3.11}$$

Now, let us consider the G node. Following the pattern blindly, we may be tempted to assert that G is conditionally independent of all other variables in the network given its parents. However, this assumption is false both at an intuitive level and for the specific example distribution we used earlier. Assume, for example, that we condition on i^1, d^1; that is, we have a smart student in a difficult class. In this setting, is G independent of L? Clearly, the answer is no: if we observe l^1 (the student got a strong letter), then our probability in g^1 (the student received an A in the course) should go up; that is, we would expect

$$P(g^1 \mid i^1, d^1, l^1) > P(g^1 \mid i^1, d^1).$$

Indeed, if we examine our distribution, the latter probability is 0.5 (as specified in the CPD), whereas the former is a much higher 0.712.

Thus, we see that we do not expect a node to be conditionally independent of all other nodes given its parents. In particular, even given its parents, it can still depend on its descendants. Can it depend on other nodes? For example, do we expect G to depend on S given I and D? Intuitively, the answer is no. Once we know, say, that the student has high intelligence, his SAT score gives us no additional information that is relevant toward predicting his grade. Thus, we

would want the property that:

$$(G \perp S \mid I, D). \tag{3.12}$$

It remains only to consider the variables I and D, which have no parents in the graph. Thus, in our search for independencies given a node's parents, we are now looking for marginal independencies. As the preceding discussion shows, in our distribution $P_{\mathcal{B}^{student}}$, I is not independent of its descendants G, L, or S. Indeed, the only nondescendant of I is D. Indeed, we assumed implicitly that *Intelligence* and *Difficulty* are independent. Thus, we expect that:

$$(I \perp D). \tag{3.13}$$

This analysis might seem somewhat surprising in light of our earlier examples, where learning something about the course difficulty drastically changed our beliefs about the student's intelligence. In that situation, however, we were reasoning in the presence of information about the student's grade. In other words, we were demonstrating the dependence of I and D *given* G. This phenomenon is a very important one, and we will return to it.

For the variable D, both I and S are nondescendants. Recall that, if $(I \perp D)$ then $(D \perp I)$. The variable S increases our beliefs in the student's intelligence, but knowing that the student is smart (or not) does not influence our beliefs in the difficulty of the course. Thus, we have that

$$(D \perp I, S). \tag{3.14}$$

We can see a pattern emerging. Our intuition tells us that the parents of a variable "shield" it from probabilistic influence that is causal in nature. In other words, once I know the value of the parents, no information relating directly or indirectly to its parents or other ancestors can influence my beliefs about it. However, information about its descendants *can* change my beliefs about it, via an evidential reasoning process.

3.2.2.2 Bayesian Network Semantics

We are now ready to provide the formal definition of the semantics of a Bayesian network structure. We would like the formal definition to match the intuitions developed in our example.

Definition 3.1 Bayesian network structure; local independencies

A Bayesian network structure $\mathcal{G}$ *is a directed acyclic graph whose nodes represent random variables* $X_1, \ldots, X_n$. *Let* $\mathrm{Pa}_{X_i}^{\mathcal{G}}$ *denote the parents of* X_i *in* $\mathcal{G}$, *and* $\mathrm{NonDescendants}_{X_i}$ *denote the variables in the graph that are not descendants of* X_i. *Then* $\mathcal{G}$ *encodes the following set of conditional independence assumptions, called the* local independencies, *and denoted by* $\mathcal{I}_\ell(\mathcal{G})$:

For each variable X_i: $(X_i \perp \mathrm{NonDescendants}_{X_i} \mid \mathrm{Pa}_{X_i}^{\mathcal{G}})$. ■

In other words, the local independencies state that each node X_i is conditionally independent of its nondescendants given its parents.

Returning to the Student network $G_{student}$, the local Markov independencies are precisely the ones dictated by our intuition, and specified in equation (3.10) – equation (3.14).

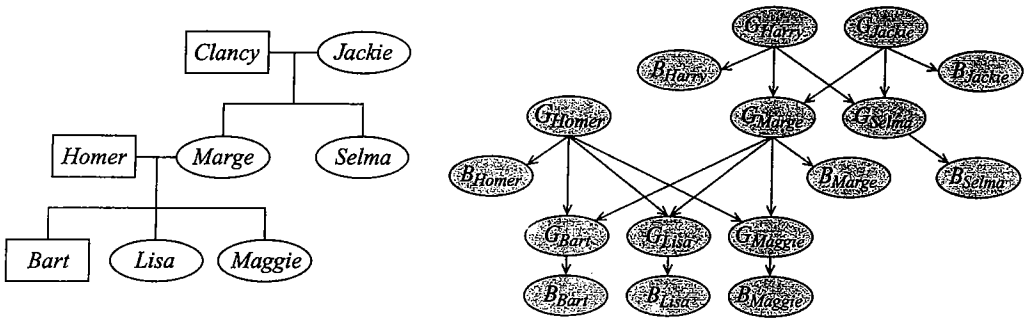


Figure 3.B.1 — Modeling Genetic Inheritance (a) A small family tree. (b) A simple BN for genetic inheritance in this domain. The G variables represent a person's genotype, and the B variables the result of a blood-type test.

Box 3.B — Case Study: The Genetics **Example.** *One of the very earliest uses of a Bayesian network model (long before the general framework was defined) is in the area of genetic pedigrees. In this setting, the local independencies are particularly intuitive. In this application, we want to model the transmission of a certain property, say blood type, from parent to child. The blood type of a person is an observable quantity that depends on her genetic makeup. Such properties are called* phenotypes. *The genetic makeup of a person is called* genotype.

To model this scenario properly, we need to introduce some background on genetics. The human genetic material consists of 22 pairs of autosomal *chromosomes and a pair of the* sex *chromosomes (X and Y). Each chromosome contains a set of genetic material, consisting (among other things) of genes that determine a person's properties. A region of the chromosome that is of interest is called a* locus*; a locus can have several variants, called* alleles.

For concreteness, we focus on autosomal chromosome pairs. In each autosomal pair, one chromosome is the paternal *chromosome, inherited from the father, and the other is the* maternal *chromosome, inherited from the mother. For genes in an autosomal pair, a person has two copies of the gene, one on each copy of the chromosome. Thus, one of the gene's alleles is inherited from the person's mother, and the other from the person's father. For example, the region containing the gene that encodes a person's blood type is a locus. This gene comes in three variants, or alleles: A, B, and O. Thus, a person's genotype is denoted by an ordered pair, such as $\langle A, B \rangle$; with three choices for each entry in the pair, there are 9 possible genotypes. The blood type phenotype is a function of both copies of the gene. For example, if the person has an A allele and an O allele, her observed blood type is "A." If she has two O alleles, her observed blood type is "O."*

To represent this domain, we would have, for each person, two variables: one representing the person's genotype, and the other her phenotype. We use the name $G(p)$ to represent person p's genotype, and $B(p)$ to represent her blood type.

In this example, the independence assumptions arise immediately from the biology. Since the

blood type is a function of the genotype, once we know the genotype of a person, additional evidence about other members of the family will not provide new information about the blood type. Similarly, the process of genetic inheritance implies independence assumption. Once we know the genotype of both parents, we know what each of them can pass on to the offspring. Thus, learning new information about ancestors (or nondescendants) does not provide new information about the genotype of the offspring. These are precisely the local independencies in the resulting network structure, shown for a simple family tree in figure 3.B.1. The intuition here is clear; for example, Bart's blood type is correlated with that of his aunt Selma, but once we know Homer's and Marge's genotype, the two become independent.

To define the probabilistic model fully, we need to specify the CPDs. There are three types of CPDs in this model:

- The penetrance model $P(B(c) \mid G(c))$, which describes the probability of different variants of a particular phenotype (say different blood types) given the person's genotype. In the case of the blood type, this CPD is a deterministic function, but in other cases, the dependence can be more complex.
- The transmission model $P(G(c) \mid G(p), G(m))$, where c is a person and p, m her father and mother, respectively. Each parent is equally likely to transmit either of his or her two alleles to the child.
- Genotype priors $P(G(c))$, used when person c has no parents in the pedigree. These are the general genotype frequencies within the population.

Our discussion of blood type is simplified for several reasons. First, some phenotypes, such as late-onset diseases, are not a deterministic function of the genotype. Rather, an individual with a particular genotype might be more likely to have the disease than an individual with other genotypes. Second, the genetic makeup of an individual is defined by many genes. Some phenotypes might depend on multiple genes. In other settings, we might be interested in multiple phenotypes, which (naturally) implies a dependence on several genes. Finally, as we now discuss, the inheritance patterns of different genes are not independent of each other.

Recall that each of the person's autosomal chromosome is inherited from one of her parents. However, each of the parents also has two copies of each autosomal chromosome. These two copies, within each parent, recombine to produce the chromosome that is transmitted to the child. Thus, the maternal chromosome inherited by Bart is a combination of the chromosomes inherited by his mother Marge from her mother Jackie and her father Clancy. The recombination process is stochastic, but only a handful recombination events take place within a chromosome in a single generation. Thus, if Bart inherited the allele for some locus from the chromosome his mother inherited from her mother Jackie, he is also much more likely to inherit Jackie's copy for a nearby locus. Thus, to construct an appropriate model for multilocus inheritance, we must take into consideration the probability of a recombination taking place between pairs of adjacent loci.

We can facilitate this modeling by introducing selector variables that capture the inheritance pattern along the chromosome. In particular, for each locus ℓ and each child c, we have a variable $S(\ell, c, m)$ that takes the value 1 if the locus ℓ in c's maternal chromosome was inherited from c's maternal grandmother, and 2 if this locus was inherited from c's maternal grandfather. We have a similar selector variable $S(\ell, c, p)$ for c's paternal chromosome. We can now model correlations induced by low recombination frequency by correlating the variables $S(\ell, c, m)$ and $S(\ell', c, m)$ for adjacent loci ℓ, ℓ'.

This type of model has been used extensively for many applications. In genetic counseling and prediction, one takes a phenotype with known loci and a set of observed phenotype and genotype data for some individuals in the pedigree to infer the genotype and phenotype for another person in the pedigree (say, a planned child). The genetic data can consist of direct measurements of the relevant disease loci (for some individuals) or measurements of nearby loci, which are correlated with the disease loci.

In linkage analysis, the task is a harder one: identifying the location of disease genes from pedigree data using some number of pedigrees where a large fraction of the individuals exhibit a disease phenotype. Here, the available data includes phenotype information for many individuals in the pedigree, as well as genotype information for loci whose location in the chromosome is known. Using the inheritance model, the researchers can evaluate the likelihood of these observations under different hypotheses about the location of the disease gene relative to the known loci. By repeated calculation of the probabilities in the network for different hypotheses, researchers can pinpoint the area that is "linked" to the disease. This much smaller region can then be used as the starting point for more detailed examination of genes in that area. This process is crucial, for it can allow the researchers to focus on a small area (for example, $1/10,000$ of the genome).

As we will see in later chapters, the ability to describe the genetic inheritance process using a sparse Bayesian network provides us the capability to use sophisticated inference algorithms that allow us to reason about large pedigrees and multiple loci. It also allows us to use algorithms for model learning to obtain a deeper understanding of the genetic inheritance process, such as recombination rates in different regions or penetrance probabilities for different diseases.

3.2.3 Graphs and Distributions

The formal semantics of a Bayesian network graph is as a set of independence assertions. On the other hand, our Student BN was a graph annotated with CPDs, which defined a joint distribution via the chain rule for Bayesian networks. In this section, we show that these two definitions are, in fact, equivalent. A distribution P satisfies the local independencies associated with a graph $\mathcal{G}$ if and only if P is representable as a set of CPDs associated with the graph $\mathcal{G}$. We begin by formalizing the basic concepts.

3.2.3.1 I-Maps

We first define the set of independencies associated with a distribution P.

Definition 3.2 (independencies in P)

Let P be a distribution over $\mathcal{X}$. We define $\mathcal{I}(P)$ to be the set of independence assertions *of the form $(\boldsymbol{X} \perp \boldsymbol{Y} \mid \boldsymbol{Z})$ that hold in P.* ■

We can now rewrite the statement that "P satisfies the local independencies associated with $\mathcal{G}$" simply as $\mathcal{I}_\ell(\mathcal{G}) \subseteq \mathcal{I}(P)$. In this case, we say that $\mathcal{G}$ is an *I-map* (independency map) for P. However, it is useful to define this concept more broadly, since different variants of it will be used throughout the book.

Definition 3.3 (I-map)

Let $\mathcal{K}$ be any graph object associated with a set of independencies $\mathcal{I}(\mathcal{K})$. We say that $\mathcal{K}$ is an I-map *for a set of independencies $\mathcal{I}$ if $\mathcal{I}(\mathcal{K}) \subseteq \mathcal{I}$.* ■

We now say that $\mathcal{G}$ is an I-map for P if $\mathcal{G}$ is an I-map for $\mathcal{I}(P)$.

As we can see from the direction of the inclusion, for $\mathcal{G}$ to be an I-map of P, it is necessary that $\mathcal{G}$ does not mislead us regarding independencies in P: any independence that $\mathcal{G}$ asserts must also hold in P. Conversely, P may have additional independencies that are not reflected in $\mathcal{G}$.

Let us illustrate the concept of an I-map on a very simple example.

Example 3.1

Consider a joint probability space over two independent random variables X and Y. There are three possible graphs over these two nodes: $\mathcal{G}_\emptyset$, which is a disconnected pair $X \quad Y$; $\mathcal{G}_{X \to Y}$, which has the edge $X \to Y$; and $\mathcal{G}_{Y \to X}$, which contains $Y \to X$. The graph $\mathcal{G}_\emptyset$ encodes the assumption that $(X \perp Y)$. The latter two encode no independence assumptions.

Consider the following two distributions:

X	Y	$P(X,Y)$
x^0	y^0	0.08
x^0	y^1	0.32
x^1	y^0	0.12
x^1	y^1	0.48

X	Y	$P(X,Y)$
x^0	y^0	0.4
x^0	y^1	0.3
x^1	y^0	0.2
x^1	y^1	0.1

In the example on the left, X and Y are independent in P; for example, $P(x^1) = 0.48 + 0.12 = 0.6$, $P(y^1) = 0.8$, and $P(x^1, y^1) = 0.48 = 0.6 \cdot 0.8$. Thus, $(X \perp Y) \in \mathcal{I}(P)$, and we have that $\mathcal{G}_\emptyset$ is an I-map of P. In fact, all three graphs are I-maps of P: $\mathcal{I}_\ell(\mathcal{G}_{X \to Y})$ is empty, so that trivially P satisfies all the independencies in it (similarly for $\mathcal{G}_{Y \to X}$). In the example on the right, $(X \perp Y) \notin \mathcal{I}(P)$, so that $\mathcal{G}_\emptyset$ is not an I-map of P. Both other graphs are I-maps of P. ■

3.2.3.2 I-Map to Factorization

A BN structure $\mathcal{G}$ encodes a set of conditional independence assumptions; every distribution for which $\mathcal{G}$ is an I-map must satisfy these assumptions. This property is the key to allowing the compact factorized representation that we saw in the Student example in section 3.2.1. The basic principle is the same as the one we used in the naive Bayes decomposition in section 3.1.3.

Consider any distribution P for which our Student BN $G_{student}$ is an I-map. We will decompose the joint and show that it factorizes into local probabilistic models, as in section 3.2.1. Consider the joint distribution $P(I, D, G, L, S)$; from the chain rule for probabilities (equation (2.5)). We can decompose this joint in the following way:

$$P(I, D, G, L, S) == P(I)P(D \mid I)P(G \mid I, D)P(L \mid I, D, G)P(S \mid I, D, G, L). \quad (3.15)$$

This transformation relies on no assumptions; it holds for any joint distribution P. However, it is also not very helpful, since the conditional probabilities in the factorization on the right-hand side are neither natural nor compact. For example, the last factor requires the specification of 24 conditional probabilities: $P(s^1 \mid i, d, g, l)$ for every assignment of values i, d, g, l.

This form, however, allows us to apply the conditional independence assumptions induced from the BN. Let us assume that $G_{student}$ is an I-map for our distribution P. In particular, from equation (3.14), we have that $(I \perp D) \in \mathcal{I}(P)$. From that, we can conclude that $P(D \mid I) = P(D)$, allowing us to simplify the second factor on the right-hand side. Similarly, we know from

equation (3.10) that $(L \perp I, D \mid G) \in \mathcal{I}(P)$. Hence, $P(L \mid I, D, G) = P(L \mid G)$, allowing us to simplify the third term. Using equation (3.11) in a similar way, we obtain that

$$P(I, D, G, L, S) = P(I)P(D)P(G \mid I, D)P(L \mid G)P(S \mid I). \tag{3.16}$$

This factorization is precisely the one we used in section 3.2.1.

This result tells us that any entry in the joint distribution can be computed as a product of factors, one for each variable. Each factor represents a conditional probability of the variable given its parents in the network. This factorization applies to *any* distribution P for which $G_{student}$ is an I-map.

We now state and prove this fundamental result more formally.

Definition 3.4 factorization

Let $\mathcal{G}$ be a BN graph over the variables $X_1, \ldots, X_n$. We say that a distribution P over the same space factorizes *according to $\mathcal{G}$ if P can be expressed as a product*

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid \mathrm{Pa}_{X_i}^{\mathcal{G}}). \tag{3.17}$$

chain rule for Bayesian networks

This equation is called the chain rule for Bayesian networks. *The individual factors $P(X_i \mid \mathrm{Pa}_{X_i}^{\mathcal{G}})$ are called* conditional probability distributions (CPDs) *or* local probabilistic models. ■

CPD

Definition 3.5 Bayesian network

A Bayesian network *is a pair $\mathcal{B} = (\mathcal{G}, P)$ where P factorizes over $\mathcal{G}$, and where P is specified as a set of CPDs associated with $\mathcal{G}$'s nodes. The distribution P is often annotated $P_{\mathcal{B}}$.* ■

We can now prove that the phenomenon we observed for $G_{student}$ holds more generally.

Theorem 3.1

Let $\mathcal{G}$ be a BN structure over a set of random variables $\mathcal{X}$, and let P be a joint distribution over the same space. If $\mathcal{G}$ is an I-map for P, then P factorizes according to $\mathcal{G}$.

topological ordering

PROOF Assume, without loss of generality, that $X_1, \ldots, X_n$ is a *topological ordering* of the variables in $\mathcal{X}$ relative to $\mathcal{G}$ (see definition 2.19). As in our example, we first use the chain rule for probabilities:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid X_1, \ldots, X_{i-1}).$$

Now, consider one of the factors $P(X_i \mid X_1, \ldots, X_{i-1})$. As $\mathcal{G}$ is an I-map for P, we have that $(X_i \perp \mathrm{NonDescendants}_{X_i} \mid \mathrm{Pa}_{X_i}^{\mathcal{G}}) \in \mathcal{I}(P)$. By assumption, all of X_i's parents are in the set $X_1, \ldots, X_{i-1}$. Furthermore, none of X_i's descendants can possibly be in the set. Hence,

$$\{X_1, \ldots, X_{i-1}\} = \mathrm{Pa}_{X_i} \cup \boldsymbol{Z}$$

where $\boldsymbol{Z} \subseteq \mathrm{NonDescendants}_{X_i}$. From the local independencies for X_i and from exercise 2.9(1), it follows that $(X_i \perp \boldsymbol{Z} \mid \mathrm{Pa}_{X_i})$. Hence, we have that

$$P(X_i \mid X_1, \ldots, X_{i-1}) = P(X_i \mid \mathrm{Pa}_{X_i}).$$

Applying this transformation to all of the factors in the chain rule decomposition, the result follows. ■

Thus, the conditional independence assumptions implied by a BN structure $\mathcal{G}$ allow us to factorize a distribution P for which $\mathcal{G}$ is an I-map into small CPDs. Note that the proof is constructive, providing a precise algorithm for constructing the factorization given the distribution P and the graph $\mathcal{G}$.

The resulting factorized representation can be substantially more compact, particularly for sparse structures.

Example 3.2

In our Student *example, the number of independent parameters is fifteen: we have two binomial distributions $P(I)$ and $P(D)$, with one independent parameter each; we have four multinomial distributions over $\mathcal{G}$ — one for each assignment of values to I and D — each with two independent parameters; we have three binomial distributions over L, each with one independent parameter; and similarly two binomial distributions over S, each with an independent parameter. The specification of the full joint distribution would require $48 - 1 = 47$ independent parameters.* ■

More generally, in a distribution over n binary random variables, the specification of the joint distribution requires $2^n - 1$ independent parameters. If the distribution factorizes according to a graph $\mathcal{G}$ where each node has at most k parents, the total number of independent parameters required is less than $n \cdot 2^k$ (see exercise 3.6). In many applications, we can assume a certain locality of influence between variables: although each variable is generally correlated with many of the others, it often depends *directly* on only a small number of other variables. Thus, in many cases, k will be very small, even though n is large. As a consequence, the number of parameters in the Bayesian network representation is typically exponentially smaller than the number of parameters of a joint distribution. This property is one of the main benefits of the Bayesian network representation.

3.2.3.3 Factorization to I-Map

Theorem 3.1 shows one direction of the fundamental connection between the conditional independencies encoded by the BN structure and the factorization of the distribution into local probability models: that the conditional independencies imply factorization. The converse also holds: factorization according to $\mathcal{G}$ implies the associated conditional independencies.

Theorem 3.2

Let $\mathcal{G}$ be a BN structure over a set of random variables $\mathcal{X}$ and let P be a joint distribution over the same space. If P factorizes according to $\mathcal{G}$, then $\mathcal{G}$ is an I-map for P.

We illustrate this theorem by example, leaving the proof as an exercise (exercise 3.9). Let P be some distribution that factorizes according to $G_{student}$. We need to show that $\mathcal{I}_\ell(G_{student})$ holds in P. Consider the independence assumption for the random variable S — $(S \perp D, G, L \mid I)$. To prove that it holds for P, we need to show that

$$P(S \mid I, D, G, L) = P(S \mid I).$$

By definition,

$$P(S \mid I, D, G, L) = \frac{P(S, I, D, G, L)}{P(I, D, G, L)}.$$

By the chain rule for BNs equation (3.16), the numerator is equal to $P(I)P(D)P(G \mid I, D)P(L \mid G)P(S \mid I)$. By the process of marginalizing over a joint distribution, we have that the denominator is:

$$\begin{aligned} P(I, D, G, L) &= \sum_S P(I, D, G, L, S) \\ &= \sum_S P(I)P(D)P(G \mid I, D)P(L \mid G)P(S \mid I) \\ &= P(I)P(D)P(G \mid I, D)P(L \mid G)\sum_S P(S \mid I) \\ &= P(I)P(D)P(G \mid I, D)P(L \mid G), \end{aligned}$$

where the last step is a consequence of the fact that $P(S \mid I)$ is a distribution over values of S, and therefore it sums to 1. We therefore have that

$$\begin{aligned} P(S \mid I, D, G, L) &= \frac{P(S, I, D, G, L)}{P(I, D, G, L)} \\ &= \frac{P(I)P(D)P(G \mid I, D)P(L \mid G)P(S \mid I)}{P(I)P(D)P(G \mid I, D)P(L \mid G)} \\ &= P(S \mid I). \end{aligned}$$

Box 3.C — Skill: Knowledge Engineering. *Our discussion of Bayesian network construction focuses on the process of going from a given distribution to a Bayesian network. Real life is not like that. We have a vague model of the world, and we need to crystallize it into a network structure and parameters. This task breaks down into several components, each of which can be quite subtle. Unfortunately, modeling mistakes can have significant consequences for the quality of the answers obtained from the network, or to the cost of using the network in practice.*

Picking variables *When we model a domain, there are many possible ways to describe the relevant entities and their attributes. Choosing which random variables to use in the model is often one of the hardest tasks, and this decision has implications throughout the model. A common problem is using ill-defined variables. For example, deciding to include the variable Fever to describe a patient in a medical domain seems fairly innocuous. However, does this random variable relate to the internal temperature of the patient? To the thermometer reading (if one is taken by the medical staff)? Does it refer to the temperature of the patient at a specific moment (for example, the time of admission to the hospital) or to occurrence of a fever over a prolonged period? Clearly, each of these might be a reasonable attribute to model, but the interaction of Fever with other variables depends on the specific interpretation we use.*

clarity test

As this example shows, we must be precise in defining the variables in the model. The clarity test *is a good way of evaluating whether they are sufficiently well defined. Assume that we are a million years after the events described in the domain; can an omniscient being, one who saw everything, determine the value of the variable? For example, consider a Weather variable with a value sunny. To be absolutely precise, we must define where we check the weather, at what time,*

and what fraction of the sky must be clear in order for it to be sunny. For a variable such as Heart-attack, we must specify how large the heart attack has to be, during what period of time it has to happen, and so on. By contrast, a variable such as Risk-of-heart-attack is meaningless, as even an omniscient being cannot evaluate whether a person had high risk or low risk, only whether the heart attack occurred or not. Introducing variables such as this confounds actual events and their probability. Note, however, that we can *use a notion of "risk group," as long as it is defined in terms of clearly specified attributes such as age or lifestyle.*

If we are not careful in our choice of variables, we will have a hard time making sure that evidence observed and conclusions made are coherent.

hidden variable

Generally speaking, we want our model to contain variables that we can potentially observe or that we may want to query. However, sometimes we want to put in a hidden variable *that is neither observed nor directly of interest. Why would we want to do that? Let us go back to the example of the cholesterol test. For the answers to be accurate, the subject has to have eaten nothing after 10:00 PM the previous evening. If the person eats (having no willpower), the results are consistently off. We do not really care about a Willpower variable, nor can we observe it. However, without it, all of the different cholesterol tests become correlated. To avoid graphs where all the tests are correlated, it is better to put in this additional hidden variable, rendering them conditionally independent given the true cholesterol level and the person's willpower.*

On the other hand, it is not necessary to add every variable that might be relevant. In our Student *example, the student's SAT score may be affected by whether he goes out for drinks on the night before the exam. Is this variable important to represent? The probabilities already account for the fact that he may achieve a poor score despite being intelligent. It might not be worthwhile to include this variable if it cannot be observed.*

It is also important to specify a reasonable domain of values for our variables. In particular, if our partition is not fine enough, conditional independence assumptions may be false. For example, we might want to construct a model where we have a person's cholesterol level, and two cholesterol tests that are conditionally independent given the person's true cholesterol level. We might choose to define the value normal to correspond to levels up to 200, and high to levels above 200. But it may be the case that both tests are more likely to fail if the person's cholesterol is marginal (200–240). In this case, the assumption of conditional independence given the value (high/normal) of the cholesterol test is false. It is only true if we add a marginal value.

Picking structure *As we saw, there are many structures that are consistent with the same set of independencies. One successful approach is to choose a structure that reflects the causal order and dependencies, so that causes are parents of the effect. Such structures tend to work well. Either because of some real locality of influence in the world, or because of the way people perceive the world, causal graphs tend to be sparser. It is important to stress that the causality is in the world, not in our inference process. For example, in an automobile insurance network, it is tempting to put Previous-accident as a parent of Good-driver, because that is how the insurance company thinks about the problem. This is not the causal order in the world, because being a bad driver causes previous (and future) accidents. In principle, there is nothing to prevent us from directing the edges in this way. However, a noncausal ordering often requires that we introduce many additional edges to account for induced dependencies (see section 3.4.1).*

One common approach to constructing a structure is a backward construction process. We begin with a variable of interest, say Lung-Cancer. We then try to elicit a prior probability for that

variable. If our expert responds that this probability is not determinable, because it depends on other factors, that is a good indication that these other factors should be added as parents for that variable (and as variables into the network). For example, we might conclude using this process that Lung-Cancer really should have Smoking as a parent, and (perhaps not as obvious) that Smoking should have Gender and Age as a parent. This approach, called extending the conversation, *avoids probability estimates that result from an average over a heterogeneous population, and therefore leads to more precise probability estimates.*

When determining the structure, however, we must also keep in mind that approximations are inevitable. For many pairs of variables, we can construct a scenario where one depends on the other. For example, perhaps Difficulty depends on Intelligence, because the professor is more likely to make a class difficult if intelligent students are registered. In general, **there are many weak influences that we might choose to model, but if we put in all of them, the network can become very complex.** *Such networks are problematic from a representational perspective: they are hard to understand and hard to debug, and eliciting (or learning) parameters can get very difficult. Moreover, as reasoning in Bayesian networks depends strongly on their connectivity (see section 9.4), adding such edges can make the network too expensive to use.*

This final consideration may lead us, in fact, to make approximations that we know to be wrong. For example, in networks for fault or medical diagnosis, the correct approach is usually to model each possible fault as a separate random variable, allowing for multiple failures. However, such networks might be too complex to perform effective inference in certain settings, and so we may sometimes resort to a single fault approximation, *where we have a single random variable encoding the primary fault or disease.*

Picking probabilities *One of the most challenging tasks in constructing a network manually is eliciting probabilities from people. This task is somewhat easier in the context of causal models, since the parameters tend to be natural and more interpretable. Nevertheless, people generally dislike committing to an exact estimate of probability.*

One approach is to elicit estimates qualitatively, using abstract terms such as "common," "rare," and "surprising," and then assign these to numbers using a predefined scale. This approach is fairly crude, and often can lead to misinterpretation. There are several approaches developed for assisting in eliciting probabilities from people. For example, one can visualize the probability of the event as an area (slice of a pie), or ask people how they would compare the probability in question to certain predefined lotteries. Nevertheless, probability elicitation is a long, difficult process, and one whose outcomes are not always reliable: the elicitation method can often influence the results, and asking the same question using different phrasing can often lead to significant differences in the answer. For example, studies show that people's estimates for an event such as "Death by disease" are significantly lower than their estimates for this event when it is broken down into different possibilities such as "Death from cancer," "Death from heart disease," and so on.

How important is it that we get our probability estimates exactly right? In some cases, small errors have very little effect. For example, changing a conditional probability of 0.7 *to* 0.75 *generally does not have a significant effect. Other errors, however, can have a significant effect:*

- **Zero probabilities**: *A common mistake is to assign a probability of zero to an event that is extremely unlikely, but not impossible. The problem is that* **one can never condition away a zero probability, no matter how much evidence we get. When an event is unlikely**

but not impossible, giving it probability zero is guaranteed to lead to irrecoverable errors. *For example, in one of the early versions of the the Pathfinder system (box 3.D), 10 percent of the misdiagnoses were due to zero probability estimates given by the expert to events that were unlikely but not impossible. As a general rule, very few things (except definitions) have probability zero, and we must be careful in assigning zeros.*

- **Orders of magnitude:** *Small differences in very low probability events can make a large difference to the network conclusions. Thus, a (conditional) probability of* 10^{-4} *is very different from* 10^{-5}.
- **Relative values:** *The qualitative behavior of the conclusions reached by the network — the value that has the highest probability — is fairly sensitive to the relative sizes of* $P(x \mid y)$ *for different values* y *of* Pa_X. *For example, it is important that the network encode correctly that the probability of having a high fever is greater when the patient has pneumonia than when he has the flu.*

sensitivity analysis

A very useful tool for estimating network parameters is sensitivity analysis, *which allows us to determine the extent to which a given probability parameter affects the outcome. This process allows us to evaluate whether it is important to get a particular CPD entry right. It also helps us figure out which CPD entries are responsible for an answer to some query that does not match our intuitions.*

Box 3.D — Case Study: Medical Diagnosis Systems. *One of the earliest applications of Bayesian networks was to the task of* medical diagnosis. *In the 1980s, a very active area of research was the construction of* expert systems *— computer-based systems that replace or assist an expert in performing a complex task. One such task that was tackled in several ways was medical diagnosis. This task, more than many others, required a treatment of uncertainty, due to the complex, nondeterministic relationships between findings and diseases. Thus, it formed the basis for experimentation with various formalisms for uncertain reasoning.*

medical diagnosis

expert system

Pathfinder

The Pathfinder *expert system was designed by Heckerman and colleagues (Heckerman and Nathwani 1992a; Heckerman et al. 1992; Heckerman and Nathwani 1992b) to help a pathologist diagnose diseases in lymph nodes. Ultimately, the model contained more than sixty different diseases and around a hundred different features. It evolved through several versions, including some based on nonprobabilistic formalisms, and several that used variants of Bayesian networks. Its diagnostic ability was evaluated over real pathological cases and compared to the diagnoses of pathological experts.*

One of the first models used was a simple naive Bayes model, which was compared to the models based on alternative uncertainty formalisms, and judged to be superior in its diagnostic ability. It therefore formed the basis for subsequent development of the system.

The same evaluation pointed out important problems in the way in which parameters were elicited from the expert. First, it was shown that 10 percent of the cases were diagnosed incorrectly, because the correct disease was ruled out by a finding that was unlikely, but not impossible, to manifest in that disease. Second, in the original construction, the expert estimated the probabilities $P(\mathit{Finding} \mid \mathit{Disease})$ *by fixing a single disease and evaluating the probabilities of all its findings. It was found that the expert was more comfortable considering a single finding and evaluating its*

probability across all diseases. This approach allows the expert to compare the relative values of the same finding across multiple diseases, as described in box 3.C.

With these two lessons in mind, another version of Pathfinder — Pathfinder III — was constructed, still using the naive Bayes model. Finally, Pathfinder IV used a full Bayesian network, with a single disease hypothesis but with dependencies between the features. Pathfinder IV was constructed using a similarity network (see box 5.B), significantly reducing the number of parameters that must be elicited. Pathfinder IV, viewed as a Bayesian network, had a total of around 75,000 parameters, but the use of similarity networks allowed the model to be constructed with fewer than 14,000 distinct parameters. Overall, the structure of Pathfinder IV took about 35 hours to define, and the parameters 40 hours.

A comprehensive evaluation of the performance of the two models revealed some important insights. First, the Bayesian network performed as well or better on most cases than the naive Bayes model. In most of the cases where the Bayesian network performed better, the use of richer dependency models was a contributing factor. As expected, these models were useful because they address the strong conditional independence assumptions of the naive Bayes model, as described in box 3.A. Somewhat more surprising, they also helped in allowing the expert to condition the probabilities on relevant factors other than the disease, using the process of extending the conversation described in box 3.C, leading to more accurate elicited probabilities. Finally, the use of similarity networks led to more accurate models, for the smaller number of elicited parameters reduced irrelevant fluctuations in parameter values (due to expert inconsistency) that can lead to spurious dependencies.

Overall, the Bayesian network model agreed with the predictions of an expert pathologist in $50/53$ *cases, as compared with* $47/53$ *cases for the naive Bayes model, with significant therapeutic implications. A later evaluation showed that the diagnostic accuracy of Pathfinder IV was at least as good as that of the expert used to design the system. When used with less expert pathologists, the system significantly improved the diagnostic accuracy of the physicians alone. Moreover, the system showed greater ability to identify important findings and to integrate these findings into a correct diagnosis.*

Unfortunately, multiple reasons prevent the widespread adoption of Bayesian networks as an aid for medical diagnosis, including legal liability issues for misdiagnoses and incompatibility with the physicians' workflow. However, several such systems have been fielded, with significant success. Moreover, similar technology is being used successfully in a variety of other diagnosis applications (see box 23.C).

3.3 Independencies in Graphs

Dependencies and independencies are key properties of a distribution and are crucial for understanding its behavior. As we will see, independence properties are also important for answering queries: they can be exploited to reduce substantially the computation cost of inference. Therefore, it is important that our representations make these properties clearly visible both to a user and to algorithms that manipulate the BN data structure.

As we discussed, a graph structure $\mathcal{G}$ encodes a certain set of conditional independence assumptions $\mathcal{I}_\ell(\mathcal{G})$. Knowing only that a distribution P factorizes over $\mathcal{G}$, we can conclude

that it satisfies $\mathcal{I}_\ell(\mathcal{G})$. An immediate question is whether there are other independencies that we can "read off" directly from $\mathcal{G}$. That is, are there other independencies that hold for *every* distribution P that factorizes over $\mathcal{G}$?

3.3.1 D-separation

Our aim in this section is to understand when we can *guarantee* that an independence $(\boldsymbol{X} \perp \boldsymbol{Y} \mid \boldsymbol{Z})$ holds in a distribution associated with a BN structure $\mathcal{G}$. To understand when a property is guaranteed to hold, it helps to consider its converse: "Can we imagine a case where it does not?" Thus, we focus our discussion on analyzing when it is *possible* that $\boldsymbol{X}$ can influence $\boldsymbol{Y}$ given $\boldsymbol{Z}$. If we construct an example where this influence occurs, then the converse property $(\boldsymbol{X} \perp \boldsymbol{Y} \mid \boldsymbol{Z})$ cannot hold for all of the distributions that factorize over $\mathcal{G}$, and hence the independence property $(\boldsymbol{X} \perp \boldsymbol{Y} \mid \boldsymbol{Z})$ cannot follow from $\mathcal{I}_\ell(\mathcal{G})$.

We therefore begin with an intuitive case analysis: Here, we try to understand when an observation regarding a variable X can possibly change our beliefs about Y, in the presence of evidence about the variables $\boldsymbol{Z}$. Although this analysis will be purely intuitive, we will show later that our conclusions are actually provably correct.

Direct connection We begin with the simple case, when X and Y are directly connected via an edge, say $X \rightarrow Y$. For any network structure $\mathcal{G}$ that contains the edge $X \rightarrow Y$, it is possible to construct a distribution where X and Y are correlated regardless of any evidence about any of the other variables in the network. In other words, if X and Y are directly connected, we can always get examples where they influence each other, regardless of $\boldsymbol{Z}$.

In particular, assume that $Val(X) = Val(Y)$; we can simply set $X = Y$. That, by itself, however, is not enough; if (given the evidence $\boldsymbol{Z}$) X deterministically takes some particular value, say 0, then X and Y both deterministically take that value, and are uncorrelated. We therefore set the network so that X is (for example) uniformly distributed, regardless of the values of any of its parents. This construction suffices to induce a correlation between X and Y, regardless of the evidence.

Indirect connection Now consider the more complicated case when X and Y are not directly connected, but there is a trail between them in the graph. We begin by considering the simplest such case: a three-node network, where X and Y are not directly connected, but where there is a trail between them via Z. It turns out that this simple case is the key to understanding the whole notion of indirect interaction in Bayesian networks.

There are four cases where X and Y are connected via Z, as shown in figure 3.5. The first two correspond to causal chains (in either direction), the third to a common cause, and the fourth to a common effect. We analyze each in turn.

Indirect causal effect (figure 3.5a). To gain intuition, let us return to the Student example, where we had a causal trail $I \rightarrow G \rightarrow L$. Let us begin with the case where G is not observed. Intuitively, if we observe that the student is intelligent, we are more inclined to believe that he gets an A, and therefore that his recommendation letter is strong. In other words, the probability of these latter events is higher conditioned on the observation that the student is intelligent. In fact, we saw precisely this behavior in the distribution of figure 3.4. Thus, in this case, we believe that X can influence Y via Z.

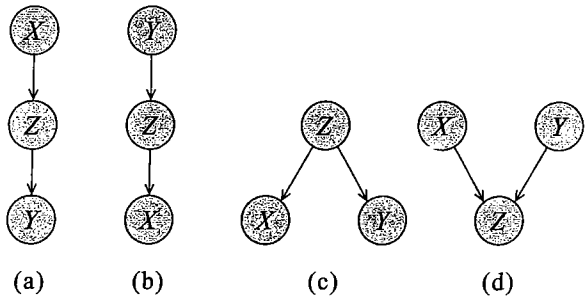


Figure 3.5 The four possible two-edge trails from X to Y via Z: (a) An indirect causal effect; (b) An indirect evidential effect; (c) A common cause; (d) A common effect.

Now assume that Z is observed, that is, $Z \in \boldsymbol{Z}$. As we saw in our analysis of the Student example, if we observe the student's grade, then (as we assumed) his intelligence no longer influences his letter. In fact, the local independencies for this network tell us that $(L \perp I \mid G)$. Thus, we conclude that X cannot influence Y via Z if Z is observed.

Indirect evidential effect (figure 3.5b). Returning to the Student example, we have a chain $I \rightarrow G \rightarrow L$. We have already seen that observing a strong recommendation letter for the student changes our beliefs in his intelligence. Conversely, once the grade is observed, the letter gives no additional information about the student's intelligence. Thus, our analysis in the case $Y \rightarrow Z \rightarrow X$ here is identical to the causal case: X can influence Y via Z, but only if Z is not observed. The similarity is not surprising, as dependence is a symmetrical notion. Specifically, if $(X \perp Y)$ does not hold, then $(Y \perp X)$ does not hold either.

Common cause (figure 3.5c). This case is one that we have analyzed extensively, both within the simple naive Bayes model of section 3.1.3 and within our Student example. Our example has the student's intelligence I as a parent of his grade G and his SAT score S. As we discussed, S and G are correlated in this model, in that observing (say) a high SAT score gives us information about a student's intelligence and hence helps us predict his grade. However, once we observe I, this correlation disappears, and S gives us no additional information about G. Once again, for this network, this conclusion follows from the local independence assumption for the node G (or for S). Thus, our conclusion here is identical to the previous two cases: X can influence Y via Z if and only if Z is not observed.

Common effect (figure 3.5d). In all of the three previous cases, we have seen a common pattern: X can influence Y via Z if and only if Z is not observed. Therefore, we might expect that this pattern is universal, and will continue through this last case. Somewhat surprisingly, this is not the case. Let us return to the Student example and consider I and D, which are parents of G. When G is not observed, we have that I and D are independent. In fact, this conclusion follows (once again) from the local independencies from the network. Thus, in this case, influence cannot "flow" along the trail $X \rightarrow Z \leftarrow Y$ if the intermediate node Z is not observed.

On the other hand, consider the behavior when Z is observed. In our discussion of the Student example, we analyzed precisely this case, which we called intercausal reasoning; we showed, for example, that the probability that the student has high intelligence goes down

dramatically when we observe that his grade is a C ($G = g^3$), but then goes up when we observe that the class is a difficult one $D = d^1$. Thus, in presence of the evidence $G = g^3$, we have that I and D are correlated.

Let us consider a variant of this last case. Assume that we do not observe the student's grade, but we do observe that he received a weak recommendation letter ($L = l^0$). Intuitively, the same phenomenon happens. The weak letter is an indicator that he received a low grade, and therefore it suffices to correlate I and D.

When influence can flow from X to Y via Z, we say that the trail $X \rightleftharpoons Z \rightleftharpoons Y$ is *active*. The results of our analysis for active two-edge trails are summarized thus:

- **Causal trail** $X \rightarrow Z \rightarrow Y$: active if and only if Z is not observed.
- **Evidential trail** $X \leftarrow Z \leftarrow Y$: active if and only if Z is not observed.
- **Common cause** $X \leftarrow Z \rightarrow Y$: active if and only if Z is not observed.
- **Common effect** $X \rightarrow Z \leftarrow Y$: active if and only if either Z or one of Z's descendants is observed.

v-structure

A structure where $X \rightarrow Z \leftarrow Y$ (as in figure 3.5d) is also called a *v-structure*.

It is useful to view probabilistic influence as a flow in the graph. Our analysis here tells us when influence from X can "flow" through Z to affect our beliefs about Y.

General Case Now consider the case of a longer trail $X_1 \rightleftharpoons \cdots \rightleftharpoons X_n$. Intuitively, for influence to "flow" from X_1 to X_n, it needs to flow through every single node on the trail. In other words, X_1 can influence X_n if every two-edge trail $X_{i-1} \rightleftharpoons X_i \rightleftharpoons X_{i+1}$ along the trail allows influence to flow.

We can summarize this intuition in the following definition:

Definition 3.6

observed variable

active trail

Let $\mathcal{G}$ be a BN structure, and $X_1 \rightleftharpoons \ldots \rightleftharpoons X_n$ a trail in $\mathcal{G}$. Let $\boldsymbol{Z}$ be a subset of observed variables. *The trail $X_1 \rightleftharpoons \ldots \rightleftharpoons X_n$ is* active *given $\boldsymbol{Z}$ if*

- *Whenever we have a v-structure $X_{i-1} \rightarrow X_i \leftarrow X_{i+1}$, then X_i or one of its descendants are in $\boldsymbol{Z}$;*
- *no other node along the trail is in $\boldsymbol{Z}$.* ■

Note that if X_1 or X_n are in $\boldsymbol{Z}$ the trail is not active.

In our Student BN, we have that $D \rightarrow G \leftarrow I \rightarrow S$ is not an active trail for $\boldsymbol{Z} = \emptyset$, because the v-structure $D \rightarrow G \leftarrow I$ is not activated. That same trail is active when $\boldsymbol{Z} = \{L\}$, because observing the descendant of G activates the v-structure. On the other hand, when $\boldsymbol{Z} = \{L, I\}$, the trail is not active, because observing I blocks the trail $G \leftarrow I \rightarrow S$.

What about graphs where there is more than one trail between two nodes? Our flow intuition continues to carry through: one node can influence another if there is any trail along which influence can flow. Putting these intuitions together, we obtain the notion of *d-separation*, which provides us with a notion of separation between nodes in a directed graph (hence the term d-separation, for directed separation):

d-separation

Definition 3.7

Let $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$ be three sets of nodes in $\mathcal{G}$. We say that $\boldsymbol{X}$ and $\boldsymbol{Y}$ are d-separated given $\boldsymbol{Z}$, *denoted* $\text{d-sep}_{\mathcal{G}}(\boldsymbol{X}; \boldsymbol{Y} \mid \boldsymbol{Z})$, *if there is no active trail between any node $X \in \boldsymbol{X}$ and $Y \in \boldsymbol{Y}$ given $\boldsymbol{Z}$.*

We use $\mathcal{I}(\mathcal{G})$ to denote the set of independencies that correspond to d-separation:

$$\mathcal{I}(\mathcal{G}) = \{(\mathbf{X} \perp \mathbf{Y} \mid \mathbf{Z}) : \text{d-sep}_{\mathcal{G}}(\mathbf{X}; \mathbf{Y} \mid \mathbf{Z})\}.$$ ■

global Markov independencies

This set is also called the set of *global Markov independencies*. The similarity between the notation $\mathcal{I}(\mathcal{G})$ and our notation $\mathcal{I}(P)$ is not coincidental: As we discuss later, the independencies in $\mathcal{I}(\mathcal{G})$ are precisely those that are guaranteed to hold for every distribution over $\mathcal{G}$.

3.3.2 Soundness and Completeness

So far, our definition of d-separation has been based on our intuitions regarding flow of influence, and on our one example. As yet, we have no guarantee that this analysis is "correct." Perhaps there is a distribution over the BN where X can influence Y despite the fact that all trails between them are blocked.

soundness of d-separation

Hence, the first property we want to ensure for d-separation as a method for determining independence is *soundness*: if we find that two nodes X and Y are d-separated given some $\mathbf{Z}$, then we are guaranteed that they are, in fact, conditionally independent given $\mathbf{Z}$.

Theorem 3.3

If a distribution P factorizes according to $\mathcal{G}$, then $\mathcal{I}(\mathcal{G}) \subseteq \mathcal{I}(P)$.

In other words, any independence reported by d-separation is satisfied by the underlying distribution. The proof of this theorem requires some additional machinery that we introduce in chapter 4, so we defer the proof to that chapter (see section 4.5.1.1).

completeness of d-separation

A second desirable property is the complementary one — *completeness*: d-separation detects *all* possible independencies. More precisely, if we have that two variables X and Y are independent given $\mathbf{Z}$, then they are d-separated. A careful examination of the completeness property reveals that it is ill defined, inasmuch as it does not specify the distribution in which X and Y are independent.

To formalize this property, we first define the following notion:

Definition 3.8

faithful

A distribution P is faithful *to $\mathcal{G}$ if, whenever $(X \perp Y \mid \mathbf{Z}) \in \mathcal{I}(P)$, then* $\text{d-sep}_{\mathcal{G}}(X; Y \mid \mathbf{Z})$. *In other words, any independence in P is reflected in the d-separation properties of the graph.* ■

We can now provide one candidate formalization of the completeness property is as follows:

- For any distribution P that factorizes over $\mathcal{G}$, we have that P is faithful to $\mathcal{G}$; that is, if X and Y are *not d-separated* given $\mathbf{Z}$ in $\mathcal{G}$, then X and Y are *dependent in all* distributions P that factorize over $\mathcal{G}$.

This property is the obvious converse to our notion of soundness: If true, the two together would imply that, for any P that factorizes over $\mathcal{G}$, we have that $\mathcal{I}(P) = \mathcal{I}(\mathcal{G})$. Unfortunately, this highly desirable property is easily shown to be false: Even if a distribution factorizes over $\mathcal{G}$, it can still contain additional independencies that are not reflected in the structure.

Example 3.3

Consider a distribution P over two variables A and B, where A and B are independent. One possible I-map for P is the network $A \rightarrow B$. For example, we can set the CPD for B to be

	b^0	b^1
a^0	0.4	0.6
a^1	0.4	0.6

This example clearly violates the first candidate definition of completeness, because the graph $\mathcal{G}$ is an I-map for the distribution P, yet there are independencies that hold for this distribution but do not follow from d-separation. In fact, these are not independencies that we can hope to discover by examining the network structure. ■

Thus, the completeness property does not hold for this candidate definition of completeness. We therefore adopt a weaker yet still useful definition:

- If $(X \perp Y \mid \mathbf{Z})$ in *all* distributions P that factorize over $\mathcal{G}$, then *d-sep*$_{\mathcal{G}}(X; Y \mid \mathbf{Z})$. And the contrapositive: If X and Y are *not d-separated* given $\mathbf{Z}$ in $\mathcal{G}$, then X and Y are *dependent in some* distribution P that factorizes over $\mathcal{G}$.

Using this definition, we can show:

Theorem 3.4

Let $\mathcal{G}$ be a BN structure. If X and Y are not d-separated *given $\mathbf{Z}$ in $\mathcal{G}$, then X and Y are* dependent *given $\mathbf{Z}$ in* some distribution P *that factorizes over $\mathcal{G}$.*

PROOF The proof constructs a distribution P that makes X and Y correlated. The construction is roughly as follows. As X and Y are not d-separated, there exists an active trail $U_1, \ldots, U_k$ between them. We define CPDs for the variables on the trail so as to make each pair U_i, U_{i+1} correlated; in the case of a v-structure $U_i \rightarrow U_{i+1} \leftarrow U_{i+2}$, we define the CPD of U_{i+1} so as to ensure correlation, and also define the CPDs of the path to some downstream evidence node, in a way that guarantees that the downstream evidence activates the correlation between U_i and U_{i+2}. All other CPDs in the graph are chosen to be uniform, and thus the construction guarantees that influence only flows along this single path, preventing cases where the influence of two (or more) paths cancel out. The details of the construction are quite technical and laborious, and we omit them. ■

We can view the completeness result as telling us that our definition of $\mathcal{I}(\mathcal{G})$ is the maximal one. For any independence assertion that is not a consequence of d-separation in $\mathcal{G}$, we can always find a counterexample distribution P that factorizes over $\mathcal{G}$. In fact, this result can be strengthened significantly:

Theorem 3.5

For almost all distributions P that factorize over $\mathcal{G}$, that is, for all distributions except for a set of measure zero in the space of CPD parameterizations, we have that $\mathcal{I}(P) = \mathcal{I}(\mathcal{G})$.[1]

This result strengthens theorem 3.4 in two distinct ways: First, whereas theorem 3.4 shows that any dependency in the graph can be found in some distribution, this new result shows that there exists a single distribution that is faithful to the graph, that is, where all of the dependencies in the graph hold simultaneously. Second, not only does this property hold for a single distribution, but it also holds for almost all distributions that factorize over $\mathcal{G}$.

PROOF At a high level, the proof is based on the following argument: Each conditional independence assertion is a set of polynomial equalities over the space of CPD parameters (see

1. A set has measure zero if it is infinitesimally small relative to the overall space. For example, the set of all rationals has measure zero within the interval $[0, 1]$. A straight line has measure zero in the plane. This intuition is defined formally in the field of *measure theory*.

exercise 3.13). A basic property of polynomials is that a polynomial is either identically zero or it is nonzero almost everywhere (its set of roots has measure zero). Theorem 3.4 implies that polynomials corresponding to assertions outside $\mathcal{I}(\mathcal{G})$ cannot be identically zero, because they have at least one counterexample. Thus, the set of distributions P, which exhibit any one of these "spurious" independence assertions, has measure zero. The set of distributions that do not satisfy $\mathcal{I}(P) = \mathcal{I}(\mathcal{G})$ is the union of these separate sets, one for each spurious independence assertion. The union of a finite number of sets of measure zero is a set of measure zero, proving the result. ■

☞ **These results state that for almost all parameterizations P of the graph $\mathcal{G}$ (that is, for almost all possible choices of CPDs for the variables), the d-separation test precisely characterizes the independencies that hold for P.** In other words, even if we have a distribution P that satisfies more independencies than $\mathcal{I}(\mathcal{G})$, a slight perturbation of the CPDs of P will almost always eliminate these "extra" independencies. This guarantee seems to state that such independencies are always accidental, and we will never encounter them in practice. However, as we illustrate in example 3.7, there are cases where our CPDs have certain local structure that is not accidental, and that implies these additional independencies that are not detected by d-separation.

3.3.3 An Algorithm for d-Separation

The notion of d-separation allows us to infer independence properties of a distribution P that factorizes over $\mathcal{G}$ simply by examining the connectivity of $\mathcal{G}$. However, in order to be useful, we need to be able to determine d-separation effectively. Our definition gives us a constructive solution, but a very inefficient one: We can enumerate all trails between X and Y, and check each one to see whether it is active. The running time of this algorithm depends on the number of trails in the graph, which can be exponential in the size of the graph.

Fortunately, there is a much more efficient algorithm that requires only linear time in the size of the graph. The algorithm has two phases. We begin by traversing the graph bottom up, from the leaves to the roots, marking all nodes that are in $\mathbf{Z}$ or that have descendants in $\mathbf{Z}$. Intuitively, these nodes will serve to enable v-structures. In the second phase, we traverse breadth-first from X to Y, stopping the traversal along a trail when we get to a blocked node. A node is blocked if: (a) it is the "middle" node in a v-structure and unmarked in phase I, or (b) is not such a node and is in $\mathbf{Z}$. If our breadth-first search gets us from X to Y, then there is an active trail between them.

The precise algorithm is shown in algorithm 3.1. The first phase is straightforward. The second phase is more subtle. For efficiency, and to avoid infinite loops, the algorithm must keep track of all nodes that have been visited, so as to avoid visiting them again. However, in graphs with loops (multiple trails between a pair of nodes), an intermediate node Y might be involved in several trails, which may require different treatment within the algorithm:

Example 3.4 *Consider the Bayesian network of figure 3.6, where our task is to find all nodes reachable from X. Assume that Y is observed, that is, $Y \in \mathbf{Z}$. Assume that the algorithm first encounters Y via the direct edge $Y \rightarrow X$. Any extension of this trail is blocked by Y, and hence the algorithm stops the traversal along this trail. However, the trail $X \leftarrow Z \rightarrow Y \leftarrow W$ is not blocked by Y. Thus, when we encounter Y for the second time via the edge $Z \rightarrow Y$, we should not ignore it. Therefore, after*

Algorithm 3.1 Algorithm for finding nodes reachable from X given Z via active trails

```
    Procedure Reachable (
      G,   // Bayesian network graph
      X,   // Source variable
      Z    // Observations
    )
1     // Phase I: Insert all ancestors of Z into V
2     L ← Z    // Nodes to be visited
3     A ← ∅    // Ancestors of Z
4     while L ≠ ∅
5       Select some Y from L
6       L ← L − {Y}
7       if Y ∉ A then
8         L ← L ∪ Pa_Y    // Y's parents need to be visited
9       A ← A ∪ {Y}    // Y is ancestor of evidence
10
11    // Phase II: traverse active trails starting from X
12    L ← {(X, ↑)}    // (Node,direction) to be visited
13    V ← ∅    // (Node,direction) marked as visited
14    R ← ∅    // Nodes reachable via active trail
15    while L ≠ ∅
16      Select some (Y, d) from L
17      L ← L − {(Y, d)}
18      if (Y, d) ∉ V then
19        if Y ∉ Z then
20          R ← R ∪ {Y}    // Y is reachable
21        V ← V ∪ {(Y, d)}    // Mark (Y, d) as visited
22        if d =↑ and Y ∉ Z then    // Trail up through Y active if Y not in Z
23          for each Z ∈ Pa_Y
24            L ← L ∪ {(Z, ↑)}    // Y's parents to be visited from bottom
25          for each Z ∈ Ch_Y
26            L ← L ∪ {(Z, ↓)}    // Y's children to be visited from top
27        else if d =↓ then    // Trails down through Y
28          if Y ∉ Z then
29              // Downward trails to Y's children are active
30            for each Z ∈ Ch_Y
31              L ← L∪{(Z, ↓)}    // Y's children to be visited from top
32          if Y ∈ A then    // v-structure trails are active
33            for each Z ∈ Pa_Y
34              L ← L∪{(Z, ↑)}    // Y's parents to be visited from bottom
35    return R
```

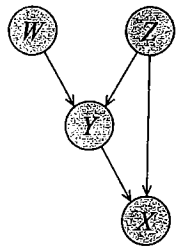


Figure 3.6 A simple example for the d-separation algorithm

the first visit to Y, we can mark it as visited for the purpose of trails coming in from children of Y, but not for the purpose of trails coming in from parents of Y. ■

In general, we see that, for each node Y, we must keep track separately of whether it has been visited from the top and whether it has been visited from the bottom. Only when both directions have been explored is the node no longer useful for discovering new active trails.

Based on this intuition, we can now show that the algorithm achieves the desired result:

Theorem 3.6

The algorithm Reachable$(\mathcal{G}, X, \mathbf{Z})$ *returns the set of all nodes reachable from X via trails that are active in $\mathcal{G}$ given $\mathbf{Z}$.*

The proof is left as an exercise (exercise 3.14).

3.3.4 I-Equivalence

The notion of $\mathcal{I}(\mathcal{G})$ specifies a set of conditional independence assertions that are associated with a graph. This notion allows us to abstract away the details of the graph structure, viewing it purely as a specification of independence properties. In particular, one important implication of this perspective is the observation that very different BN structures can actually be equivalent, in that they encode precisely the same set of conditional independence assertions. Consider, for example, the three networks in figure 3.5a,(b),(c). All three of them encode precisely the same independence assumptions: $(X \perp Y \mid Z)$.

Definition 3.9
I-equivalence

Two graph structures $\mathcal{K}_1$ and $\mathcal{K}_2$ over $\mathcal{X}$ are I-equivalent *if $\mathcal{I}(\mathcal{K}_1) = \mathcal{I}(\mathcal{K}_2)$. The set of all graphs over $\mathcal{X}$ is partitioned into a set of mutually exclusive and exhaustive* I-equivalence classes, *which are the set of equivalence classes induced by the I-equivalence relation.* ■

Note that the v-structure network in figure 3.5d induces a very different set of d-separation assertions, and hence it does not fall into the same I-equivalence class as the first three. Its I-equivalence class contains only that single network.

I-equivalence of two graphs immediately implies that any distribution P that can be factorized over one of these graphs can be factorized over the other. Furthermore, **there is no intrinsic property of P that would allow us to associate it with one graph rather than an equivalent one. This observation has important implications with respect to our ability to determine the directionality of influence.** In particular, although we can determine, for a distribution

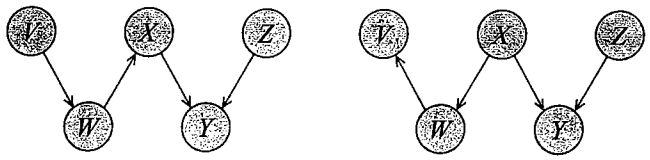


Figure 3.7 Skeletons and v-structures in a network. The two networks shown have the same skeleton and v-structures ($X \rightarrow Y \leftarrow Z$).

$P(X, Y)$, whether X and Y are correlated, there is nothing in the distribution that can help us determine whether the correct structure is $X \rightarrow Y$ or $Y \rightarrow X$. We return to this point when we discuss the causal interpretation of Bayesian networks in chapter 21.

The d-separation criterion allows us to test for I-equivalence using a very simple graph-based algorithm. We start by considering the trails in the networks.

Definition 3.10 skeleton

The skeleton *of a Bayesian network graph $\mathcal{G}$ over $\mathcal{X}$ is an undirected graph over $\mathcal{X}$ that contains an edge $\{X, Y\}$ for every edge (X, Y) in $\mathcal{G}$.* ■

In the networks of figure 3.7, the networks (a) and (b) have the same skeleton.

If two networks have a common skeleton, then the set of trails between two variables X and Y is same in both networks. If they do not have a common skeleton, we can find a trail in one network that does not exist in the other and use this trail to find a counterexample for the equivalence of the two networks.

Ensuring that the two networks have the same trails is clearly not enough. For example, the networks in figure 3.5 all have the same skeleton. Yet, as the preceding discussion shows, the network of figure 3.5d is not equivalent to the networks of figure 3.5a–(c). The difference, is of course, the v-structure in figure 3.5d. Thus, it seems that if the two networks have the same skeleton and exactly the same set of v-structures, they are equivalent. Indeed, this property provides a sufficient condition for I-equivalence:

Theorem 3.7

Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be two graphs over $\mathcal{X}$. If $\mathcal{G}_1$ and $\mathcal{G}_2$ have the same skeleton and the same set of v-structures then they are I-equivalent.

The proof is left as an exercise (see exercise 3.16).

Unfortunately, this characterization is not an equivalence: there are graphs that are I-equivalent but do not have the same set of v-structures. As a counterexample, consider *complete* graphs over a set of variables. Recall that a complete graph is one to which we cannot add additional arcs without causing cycles. Such graphs encode the empty set of conditional independence assertions. Thus, any two complete graphs are I-equivalent. Although they have the same skeleton, they invariably have different v-structures. Thus, by using the criterion on theorem 3.7, we can conclude (in certain cases) only that two networks are I-equivalent, but we cannot use it to guarantee that they are not.

We can provide a stronger condition that does correspond exactly to I-equivalence. Intuitively, the unique independence pattern that we want to associate with a v-structure $X \rightarrow Z \leftarrow Y$ is that X and Y are independent (conditionally on their parents), but dependent given Z. If there

is a direct edge between X and Y, as there was in our example of the complete graph, the first part of this pattern is eliminated.

Definition 3.11 *A v-structure $X \rightarrow Z \leftarrow Y$ is an* immorality *if there is no direct edge between X and Y. If there is such an edge, it is called a* covering edge *for the v-structure.* ■

immorality

covering edge

Note that not every v-structure is an immorality, so that two networks with the same immoralities do not necessarily have the same v-structures. For example, two different complete directed graphs always have the same immoralities (none) but different v-structures.

Theorem 3.8 *Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be two graphs over $\mathcal{X}$. Then $\mathcal{G}_1$ and $\mathcal{G}_2$ have the same skeleton and the same set of immoralities if and only if they are I-equivalent.*

The proof of this (more difficult) result is also left as an exercise (see exercise 3.17).

We conclude with a final characterization of I-equivalence in terms of local operations on the graph structure.

Definition 3.12 *An edge $X \rightarrow Y$ in a graph $\mathcal{G}$ is said to be* covered *if* $\mathrm{Pa}_Y^{\mathcal{G}} = \mathrm{Pa}_X^{\mathcal{G}} \cup \{X\}$. ■

covered edge

Theorem 3.9 *Two graphs $\mathcal{G}$ and $\mathcal{G}'$ are I-equivalent if and only if there exists a sequence of networks $\mathcal{G} = \mathcal{G}_1, \ldots, \mathcal{G}_k = \mathcal{G}'$ that are all I-equivalent to $\mathcal{G}$ such that the only difference between $\mathcal{G}_i$ and $\mathcal{G}_{i+1}$ is a single reversal of a covered edge.*

The proof of this theorem is left as an exercise (exercise 3.18).

3.4 From Distributions to Graphs

In the previous sections, we showed that, if P factorizes over $\mathcal{G}$, we can derive a rich set of independence assertions that hold for P by simply examining $\mathcal{G}$. This result immediately leads to the idea that we can use a graph as a way of revealing the structure in a distribution. In particular, we can test for independencies in P by constructing a graph $\mathcal{G}$ that represents P and testing d-separation in $\mathcal{G}$. As we will see, having a graph that reveals the structure in P has other important consequences, in terms of reducing the number of parameters required to specify or learn the distribution, and in terms of the complexity of performing inference on the network.

In this section, we examine the following question: Given a distribution P, to what extent can we construct a graph $\mathcal{G}$ whose independencies are a reasonable surrogate for the independencies in P? It is important to emphasize that we will never actually take a fully specified distribution P and construct a graph $\mathcal{G}$ for it: As we discussed, a full joint distribution is much too large to represent explicitly. However, answering this question is an important conceptual exercise, which will help us later on when we try to understand the process of constructing a Bayesian network that represents our model of the world, whether manually or by learning from data.

3.4.1 Minimal I-Maps

One approach to finding a graph that represents a distribution P is simply to take any graph that is an I-map for P. The problem with this naive approach is clear: As we saw in example 3.3, the

Algorithm 3.2 Procedure to build a minimal I-map given an ordering

```
Procedure Build-Minimal-I-Map (
  X_1, ..., X_n    // an ordering of random variables in 𝒳
  ℐ    // Set of independencies
)
1   Set 𝒢 to an empty graph over 𝒳
2   for i = 1, ..., n
3     U ← {X_1, ..., X_{i-1}}    // U is the current candidate for parents of X_i
4     for U' ⊆ {X_1, ..., X_{i-1}}
5       if U' ⊂ U and (X_i ⊥ {X_1, ..., X_{i-1}} − U' | U') ∈ ℐ then
6         U ← U'
7       // At this stage U is a minimal set satisfying (X_i ⊥
          {X_1, ..., X_{i-1}} − U | U)
8       // Now set U to be the parents of X_i
9     for X_j ∈ U
10      Add X_j → X_i to 𝒢
11  return 𝒢
```

complete graph is an I-map for any distribution, yet it does not reveal any of the independence structure in the distribution. However, examples such as this one are not very interesting. The graph that we used as an I-map is clearly and trivially unrepresentative of the distribution, in that there are edges that are obviously redundant. This intuition leads to the following definition, which we also define more broadly:

Definition 3.13 minimal I-map

A graph $\mathcal{K}$ is a minimal I-map *for a set of independencies $\mathcal{I}$ if it is an I-map for $\mathcal{I}$, and if the removal of even a single edge from $\mathcal{K}$ renders it not an I-map.* ■

This notion of an I-map applies to multiple types of graphs, both Bayesian networks and other types of graphs that we will encounter later on. Moreover, because it refers to a set of independencies $\mathcal{I}$, it can be used to define an I-map for a distribution P, by taking $\mathcal{I} = \mathcal{I}(P)$, or to another graph $\mathcal{K}'$, by taking $\mathcal{I} = \mathcal{I}(\mathcal{K}')$.

Recall that definition 3.5 defines a Bayesian network to be a distribution P that factorizes over $\mathcal{G}$, thereby implying that $\mathcal{G}$ is an I-map for P. It is standard to restrict the definition even further, by requiring that $\mathcal{G}$ be a minimal I-map for P.

How do we obtain a minimal I-map for the set of independencies induced by a given distribution P? The proof of the factorization theorem (theorem 3.1) gives us a procedure, which is shown in algorithm 3.2. We assume we are given a predetermined *variable ordering*, say, $\{X_1, \ldots, X_n\}$. We now examine each variable X_i, $i = 1, \ldots, n$ in turn. For each X_i, we pick some minimal subset $\boldsymbol{U}$ of $\{X_1, \ldots, X_{i-1}\}$ to be X_i's parents in $\mathcal{G}$. More precisely, we require that $\boldsymbol{U}$ satisfy $(X_i \perp \{X_1, \ldots, X_{i-1}\} - \boldsymbol{U} \mid \boldsymbol{U})$, and that no node can be removed from $\boldsymbol{U}$ without violating this property. We then set $\boldsymbol{U}$ to be the parents of X_i.

variable ordering

The proof of theorem 3.1 tells us that, if each node X_i is independent of $X_1, \ldots, X_{i-1}$ given its parents in $\mathcal{G}$, then P factorizes over $\mathcal{G}$. We can then conclude from theorem 3.2 that $\mathcal{G}$ is an I-map for P. By construction, $\mathcal{G}$ is minimal, so that $\mathcal{G}$ is a minimal I-map for P.

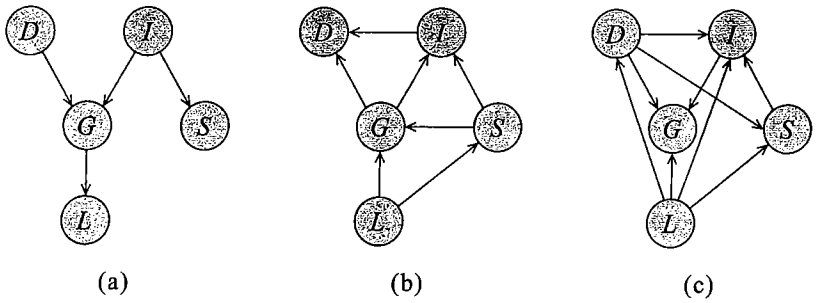


Figure 3.8 Three minimal I-maps for $P_{\mathcal{B}^{student}}$, induced by different orderings: (a) D, I, S, G, L; (b) L, S, G, I, D; (C) L, D, S, I, G.

Note that our choice of $\boldsymbol{U}$ may not be unique. Consider, for example, a case where two variables A and B are logically equivalent, that is, our distribution P only gives positive probability to instantiations where A and B have the same value. Now, consider a node C that is correlated with A. Clearly, we can choose either A or B to be a parent of C, but having chosen the one, we cannot choose the other without violating minimality. Hence, the minimal parent set $\boldsymbol{U}$ in our construction is not necessarily unique. However, one can show that, if the distribution is positive (see definition 2.5), that is, if for any instantiation ξ to all the network variables $\mathcal{X}$ we have that $P(\xi) > 0$, then the choice of parent set, given an ordering, is unique. Under this assumption, algorithm 3.2 can produce all minimal I-maps for P: Let $\mathcal{G}$ be any minimal I-map for P. If we give call Build-Minimal-I-Map with an ordering $\prec$ that is topological for $\mathcal{G}$, then, due to the uniqueness argument, the algorithm must return $\mathcal{G}$.

At first glance, the minimal I-map seems to be a reasonable candidate for capturing the structure in the distribution: It seems that if $\mathcal{G}$ is a minimal I-map for a distribution P, then we should be able to "read off" all of the independencies in P directly from $\mathcal{G}$. Unfortunately, this intuition is false.

Example 3.5

Consider the distribution $P_{\mathcal{B}^{student}}$, as defined in figure 3.4, and let us go through the process of constructing a minimal I-map for $P_{\mathcal{B}^{student}}$. We note that the graph G_{student} precisely reflects the independencies in this distribution $P_{\mathcal{B}^{student}}$ (that is, $\mathcal{I}(P_{\mathcal{B}^{student}}) = \mathcal{I}(G_{\text{student}})$), so that we can use G_{student} to determine which independencies hold in $P_{\mathcal{B}^{student}}$.

Our construction process starts with an arbitrary ordering on the nodes; we will go through this process for three different orderings. Throughout this process, it is important to remember that we are testing independencies relative to the distribution $P_{\mathcal{B}^{student}}$. We can use G_{student} (figure 3.4) to guide our intuition about which independencies hold in $P_{\mathcal{B}^{student}}$, but we can always resort to testing these independencies in the joint distribution $P_{\mathcal{B}^{student}}$.

The first ordering is a very natural one: D, I, S, G, L. We add one node at a time and see which of the possible edges from the preceding nodes are redundant. We start by adding D, then I. We can now remove the edge from D to I because this particular distribution satisfies $(I \perp D)$, so I is independent of D given its other parents (the empty set). Continuing on, we add S, but we can remove the edge from D to S because our distribution satisfies $(S \perp D \mid I)$. We then add G, but we can remove the edge from S to G, because the distribution satisfies $(G \perp S \mid I, D)$.

Finally, we add L, but we can remove all edges from D, I, S. Thus, our final output is the graph in figure 3.8a, which is precisely our original network for this distribution.

Now, consider a somewhat less natural ordering: L, S, G, I, D. In this case, the resulting I-map is not quite as natural or as sparse. To see this, let us consider the sequence of steps. We start by adding L to the graph. Since it is the first variable in the ordering, it must be a root. Next, we consider S. The decision is whether to have L as a parent of S. Clearly, we need an edge from L to S, because the quality of the student's letter is correlated with his SAT score in this distribution, and S has no other parents that help render it independent of L. Formally, we have that $(S \perp L)$ does not *hold in the distribution. In the next iteration of the algorithm, we introduce G. Now, all possible subsets of $\{L, S\}$ are potential parents set for G. Clearly, G is dependent on L. Moreover, although G is independent of S given I, it is not independent of S given L. Hence, we must add the edge between S and G. Carrying out the procedure, we end up with the graph shown in figure 3.8b.*

Finally, consider the ordering: L, D, S, I, G. In this case, a similar analysis results in the graph shown in figure 3.8c, which is almost a complete graph, missing only the edge from S to G, which we can remove because G is independent of S given I. ■

Note that the graphs in figure 3.8b,c really are minimal I-maps for this distribution. However, they fail to capture some or all of the independencies that hold in the distribution. Thus, they show that the fact that $\mathcal{G}$ is a minimal I-map for P is far from a guarantee that $\mathcal{G}$ captures the independence structure in P.

3.4.2 Perfect Maps

We aim to find a graph $\mathcal{G}$ that precisely captures the independencies in a given distribution P.

Definition 3.14 perfect map

We say that a graph $\mathcal{K}$ is a perfect map (P-map) *for a set of independencies $\mathcal{I}$ if we have that $\mathcal{I}(\mathcal{K}) = \mathcal{I}$. We say that $\mathcal{K}$ is a perfect map for P if $\mathcal{I}(\mathcal{K}) = \mathcal{I}(P)$.* ■

If we obtain a graph $\mathcal{G}$ that is a P-map for a distribution P, then we can (by definition) read the independencies in P directly from $\mathcal{G}$. By construction, our original graph $G_{student}$ is a P-map for $P_{\mathcal{B}^{student}}$.

If our goal is to find a perfect map for a distribution, an immediate question is whether every distribution has a perfect map. Unfortunately, the answer is no, and for several reasons. The first type of counterexample involves regularity in the parameterization of the distribution that cannot be captured in the graph structure.

Example 3.6

Consider a joint distribution P over 3 random variables X,Y,Z such that:

$$P(x, y, z) = \begin{cases} 1/12 & x \oplus y \oplus z = \textit{false} \\ 1/6 & x \oplus y \oplus z = \textit{true} \end{cases}$$

where $\oplus$ is the XOR (exclusive OR) function. A simple calculation shows that $(X \perp Y) \in \mathcal{I}(P)$, and that Z is not independent of X given Y or of Y given X. Hence, one minimal I-map for this distribution is the network $X \rightarrow Z \leftarrow Y$, using a deterministic XOR for the CPD of Z. However, this network is not a perfect map; a precisely analogous calculation shows that $(X \perp Z) \in \mathcal{I}(P)$, but this conclusion is not supported by a d-separation analysis. ■

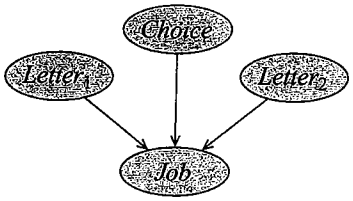


Figure 3.9 Network for the OneLetter **example**

Thus, we see that deterministic relationships can lead to distributions that do not have a P-map. Additional examples arise as a consequence of other regularities in the CPD.

Example 3.7

Consider a slight elaboration of our Student *example. During his academic career, our student George has taken both Econ101 and CS102. The professors of both classes have written him letters, but the recruiter at Acme Consulting asks for only a single recommendation. George's chance of getting the job depends on the quality of the letter he gives the recruiter. We thus have four random variables: $L1$ and $L2$, corresponding to the quality of the recommendation letters for Econ101 and CS102 respectively; C, whose value represents George's choice of which letter to use; and J, representing the event that George is hired by Acme Consulting.*

The obvious minimal I-map for this distribution is shown in figure 3.9. Is this a perfect map? Clearly, it does not reflect independencies that are not at the variable level. In particular, we have that $(L1 \perp J \mid C = 2)$. However, this limitation is not surprising; by definition, a BN structure makes independence assertions only at the level of variables. (We return to this issue in section 5.2.2.) However, our problems are not limited to these finer-grained independencies. Some thought reveals that, in our target distribution, we also have that $(L1 \perp L2 \mid C, J)$! This independence is not implied by d-separation, because the v-structure $L1 \rightarrow J \leftarrow L2$ is enabled. However, we can convince ourselves that the independence holds using reasoning by cases. If $C = 1$, then there is no dependence of J on $L2$. Intuitively, the edge from $L2$ to J disappears, eliminating the trail between $L1$ and $L2$, so that $L1$ and $L2$ are independent in this case. A symmetric analysis applies in the case that $C = 2$. Thus, in both cases, we have that $L1$ and $L2$ are independent. This independence assertion is not captured by our minimal I-map, which is therefore not a P-map. ■

A different class of examples is not based on structure within a CPD, but rather on symmetric variable-level independencies that are not naturally expressed within a Bayesian network.

A second class of distributions that do not have a perfect map are those for which the independence assumptions imposed by the structure of Bayesian networks is simply not appropriate.

Example 3.8

Consider a scenario where we have four students who get together in pairs to work on the homework for a class. For various reasons, only the following pairs meet: Alice and Bob; Bob and Charles; Charles and Debbie; and Debbie and Alice. (Alice and Charles just can't stand each other, and Bob and Debbie had a relationship that ended badly.) The study pairs are shown in figure 3.10a.

In this example, the professor accidentally misspoke in class, giving rise to a possible misconception among the students in the class. Each of the students in the class may subsequently have figured out the problem, perhaps by thinking about the issue or reading the textbook. In subsequent study pairs, he or she may transmit this newfound understanding to his or her study partners. We

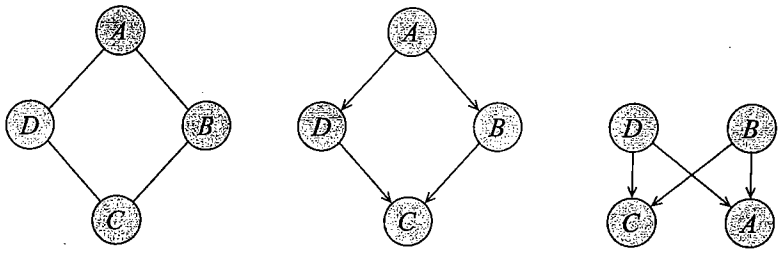


Figure 3.10 Attempted Bayesian network models for the Misconception **example:** (a) Study pairs over four students. (b) First attempt at a Bayesian network model. (c) Second attempt at a Bayesian network model.

therefore have four binary random variables, representing whether the student has the misconception or not. We assume that for each $X \in \{A, B, C, D\}$, x^1 denotes the case where the student has the misconception, and x^0 denotes the case where he or she does not.

Because Alice and Charles never speak to each other directly, we have that A and C are conditionally independent given B and D. Similarly, B and D are conditionally independent given A and C. Can we represent this distribution (with these independence properties) using a BN? One attempt is shown in figure 3.10b. Indeed, it encodes the independence assumption that $(A \perp C \mid \{B, D\})$. However, it also implies that B and D are independent given only A, but dependent given both A and C. Hence, it fails to provide a perfect map for our target distribution. A second attempt, shown in figure 3.10c, is equally unsuccessful. It also implies that $(A \perp C \mid \{B, D\})$, but it also implies that B and D are marginally independent. It is clear that all other candidate BN structures are also flawed, so that this distribution does not have a perfect map. ■

3.4.3 Finding Perfect Maps ⋆

Earlier we discussed an algorithm for finding minimal I-maps. We now consider an algorithm for finding a perfect map (P-map) of a distribution. Because the requirements from a P-map are stronger than the ones we require from an I-map, the algorithm will be more involved.

Throughout the discussion in this section, we assume that P has a P-map. In other words, there is an unknown DAG $\mathcal{G}^*$ that is P-map of P. Since $\mathcal{G}^*$ is a P-map, we will interchangeably refer to independencies in P and in $\mathcal{G}^*$ (since these are the same). We note that the algorithms we describe do fail when they are given a distribution that does not have a P-map. We discuss this issue in more detail later.

Thus, our goal is to identify $\mathcal{G}^*$ from P. One obvious difficulty that arises when we consider this goal is that $\mathcal{G}^*$ is, in general, not uniquely identifiable from P. A P-map of a distribution, if one exists, is generally not unique: As we saw, for example, in figure 3.5, multiple graphs can encode precisely the same independence assumptions. However, the P-map of a distribution *is* unique up to I-equivalence between networks. That is, a distribution P can have many P-maps, but all of them are I-equivalent.

If we require that a P-map construction algorithm return a single network, the output we get may be some arbitrary member of the I-equivalence class of $\mathcal{G}^*$. A more correct answer would be to return the entire equivalence class, thus avoiding an arbitrary commitment to a possibly incorrect structure. Of course, we do not want our algorithm to return a (possibly very large) set of distinct networks as output. Thus, one of our tasks in this section is to develop a compact representation of an entire equivalence class of DAGs. As we will see later in the book, this representation plays a useful role in other contexts as well.

This formulation of the problem points us toward a solution. Recall that, according to theorem 3.8, two DAGs are I-equivalent if they share the same skeleton and the same set of immoralities. Thus, we can construct the I-equivalence class for $\mathcal{G}^*$ by determining its skeleton and its immoralities from the independence properties of the given distribution P. We then use both of these components to build a representation of the equivalence class.

3.4.3.1 Identifying the Undirected Skeleton

At this stage we want to construct an undirected graph S that contains an edge X—Y if X and Y are adjacent in $\mathcal{G}^*$; that is, if either $X \rightarrow Y$ or $Y \rightarrow X$ is an edge in $\mathcal{G}^*$.

The basic idea is to use independence queries of the form $(X \perp Y \mid \boldsymbol{U})$ for different sets of variables $\boldsymbol{U}$. This idea is based on the observation that if X and Y are adjacent in $\mathcal{G}^*$, we cannot separate them with any set of variables.

Lemma 3.1 *Let $\mathcal{G}^*$ be a P-map of a distribution $\mathcal{P}$, and let X and Y be two variables such that $X \rightarrow Y$ is in $\mathcal{G}^*$. Then, $P \not\models (X \perp Y \mid \boldsymbol{U})$ for any set $\boldsymbol{U}$ that does not include X and Y.*

PROOF Assume that that $X \rightarrow Y \in \mathcal{G}^*$, and let $\boldsymbol{U}$ be a set of variables. According to d-separation the trail $X \rightarrow Y$ cannot be blocked by the evidence set $\boldsymbol{U}$. Thus, X and Y are not d-separated by U. Since $\mathcal{G}^*$ is a P-map of P, we have that $P \not\models (X \perp Y \mid \boldsymbol{U})$. ■

This lemma implies that if X and Y are adjacent in $\mathcal{G}^*$, all conditional independence queries that involve both of them would fail. Conversely, if X and Y are not adjacent in $\mathcal{G}$, we would hope to be able to find a set of variables that makes these two variables conditionally independent. Indeed, as we now show, we can provide a precise characterization of such a set:

Lemma 3.2 *Let $\mathcal{G}^*$ be an I-map of a distribution P, and let X and Y be two variables that are not adjacent in $\mathcal{G}^*$. Then either $P \models (X \perp Y \mid \mathrm{Pa}_X^{\mathcal{G}^*})$ or $P \models (X \perp Y \mid \mathrm{Pa}_Y^{\mathcal{G}^*})$.*

The proof is left as an exercise (exercise 3.19).

witness

Thus, if X and Y are not adjacent in $\mathcal{G}^*$, then we can find a set $\boldsymbol{U}$ so that $\mathcal{P} \models (X \perp Y \mid \boldsymbol{U})$. We call this set $\boldsymbol{U}$ a *witness* of their independence. Moreover, the lemma shows that we can find a witness of bounded size. Thus, if we assume that $\mathcal{G}^*$ has bounded indegree, say less than or equal to d, then we do not need to consider witness sets larger than d.

With these tools in hand, we can now construct an algorithm for building a skeleton of $\mathcal{G}^*$, shown in algorithm 3.3. For each pair of variables, we consider all potential witness sets and test for independence. If we find a witness that separates the two variables, we record it (we will soon see why) and move on to the next pair of variables. If we do not find a witness, then we conclude that the two variables are adjacent in $\mathcal{G}^*$ and add them to the skeleton. The list

Algorithm 3.3 Recovering the undirected skeleton for a distribution P that has a P-map

```
Procedure Build-PMap-Skeleton (
  X = {X_1, ..., X_n},   // Set of random variables
  P,   // Distribution over X
  d    // Bound on witness set
)
1   Let H be the complete undirected graph over X
2   for X_i, X_j in X
3     U_{X_i,X_j} ← ∅
4     for U ∈ Witnesses(X_i, X_j, H, d)
5         // Consider U as a witness set for X_i, X_j
6       if P ⊨ (X_i ⊥ X_j | U) then
7         U_{X_i,X_j} ← U
8         Remove X_i—X_j from H
9         break
10    return (H,{U_{X_i,X_j} : i, j ∈ {1, ..., n}})
```

Witnesses(X_i, X_j, S, d) in line 4 specifies the set of possible witness sets that we consider for separating X_i and X_j. From our earlier discussion, if we assume a bound d on the indegree, then we can restrict attention to sets $\boldsymbol{U}$ of size at most d. Moreover, using the same analysis, we saw that we have a witness that consists either of the parents of X_i or of the parents of X_j. In the first case, we can restrict attention to sets $\boldsymbol{U} \subseteq \mathcal{X} - \{X_i, X_j\} - \mathrm{Nb}_{X_i}^{\mathcal{H}}$, where $\mathrm{Nb}_{X_i}^{S}$ are the neighbors of X_i in the current graph $\mathcal{H}$; in the second, we can similarly restrict attention to sets $\boldsymbol{U} \subseteq \mathcal{X} - \{X_i, X_j\} - \mathrm{Nb}_{X_j}^{\mathcal{H}}$. Finally, we note that if $\boldsymbol{U}$ separates X_i and X_j, then also many of $\boldsymbol{U}$'s supersets will separate X_i and X_j. Thus, we search the set of possible witnesses in order of increasing size.

This algorithm will recover the correct skeleton given that $\mathcal{G}^*$ is a P-map of P and has bounded indegree d. If P does not have a P-map, then the algorithm can fail; see exercise 3.22. This algorithm has complexity of $O(n^{d+2})$ since we consider $O(n^2)$ pairs, and for each we perform $O((n-2)^d)$ independence tests. We greatly reduce the number of independence tests by ordering potential witnesses accordingly, and by aborting the inner loop once we find a witness for a pair (after line 9). However, for pairs of variables that are directly connected in the skeleton, we still need to evaluate all potential witnesses.

3.4.3.2 Identifying Immoralities

At this stage we have reconstructed the undirected skeleton S using Build-PMap-Skeleton. Now, we want to reconstruct edge direction. The main cue for learning about edge directions in $\mathcal{G}^*$ are immoralities. As shown in theorem 3.8, all DAGs in the equivalence class of $\mathcal{G}^*$ share the same set of immoralities. Thus, our goal is to consider *potential immoralities* in the skeleton and for each one determine whether it is indeed an immorality. A triplet of variables X, Z, Y is a *potential immorality* if the skeleton contains $X—Z—Y$ but does not contain an edge between X and Y. If such a triplet is indeed an immorality in $\mathcal{G}^*$, then X and Y cannot be independent

potential immorality

Algorithm 3.4 Marking immoralities in the construction of a perfect map

```
Procedure Mark-Immoralities (
  X = {X_1, ..., X_n},
  S    // Skeleton
  {U_{X_i,X_j} : 1 ≤ i, j ≤ n}    // Witnesses found by Build-PMap-Skeleton
)
1   K ← S
2   for X_i, X_j, X_k such that X_i—X_j—X_k ∈ S and X_i—X_k ∉ S
3       // X_i—X_j—X_k is a potential immorality
4     if X_j ∉ U_{X_i,X_k} then
5       Add the orientations X_i → X_j and X_j ← X_k to K
6   return K
```

given Z. Nor will they be independent given a set $\boldsymbol{U}$ that contains Z. More precisely,

Proposition 3.1 *Let $\mathcal{G}^*$ be a P-map of a distribution P, and let X, Y and Z be variables that form an immorality $X \rightarrow Z \leftarrow Y$. Then, $P \not\models (X \perp Y \mid \boldsymbol{U})$ for any set $\boldsymbol{U}$ that contains Z.*

PROOF Let $\boldsymbol{U}$ be a set of variables that contains Z. Since Z is observed, the trail $X \rightarrow Z \leftarrow Y$ is active, and so X and Y are not d-separated in $\mathcal{G}^*$. Since $\mathcal{G}^*$ is a P-map of P, we have that $P^* \not\models (X \perp Y \mid \boldsymbol{U})$. ■

What happens in the complementary situation? Suppose $X—Z—Y$ in the skeleton, but is not an immorality. This means that one of the following three cases is in $\mathcal{G}^*$: $X \rightarrow Z \rightarrow Y$, $Y \rightarrow Z \rightarrow X$, or $X \leftarrow Z \rightarrow Y$. In all three cases, X and Y are d-separated only if Z is observed.

Proposition 3.2 *Let $\mathcal{G}^*$ be a P-map of a distribution P, and let the triplet X, Y, Z be a potential immorality in the skeleton of $\mathcal{G}^*$, such that $X \rightarrow Z \leftarrow Y$ is not in $\mathcal{G}^*$. If $\boldsymbol{U}$ is such that $P \models (X \perp Y \mid \boldsymbol{U})$, then $Z \in \boldsymbol{U}$.*

PROOF Consider all three configurations of the trail $X \rightleftharpoons Z \rightleftharpoons Y$. In all three, Z must be observed in order to block the trail. Since $\mathcal{G}^*$ is a P-map of P, we have that if $P \models (X \perp Y \mid \boldsymbol{U})$, then $Z \in \boldsymbol{U}$. ■

Combining these two results, we see that a potential immorality $X—Z—Y$ is an immorality if and only if Z is not in the witness set(s) for X and Y. That is, if $X—Z—Y$ is an immorality, then proposition 3.1 shows that Z is not in any witness set $\boldsymbol{U}$; conversely, if $X—Z—Y$ is not an immorality, the Z must be in every witness set $\boldsymbol{U}$. Thus, we can use the specific witness set $\boldsymbol{U}_{X,Y}$ that we recorded for X, Y in order to determine whether this triplet is an immorality or not: we simply check whether $Z \in \boldsymbol{U}_{X,Y}$. If $Z \notin \boldsymbol{U}_{X,Y}$, then we declare the triplet an immorality. Otherwise, we declare that it is not an immorality. The Mark-Immoralities procedure shown in algorithm 3.4 summarizes this process.

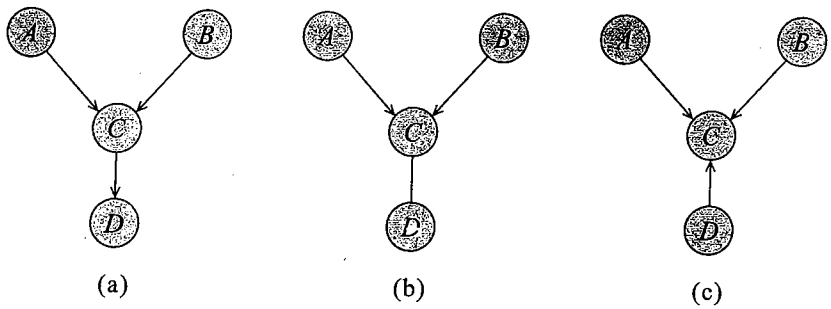


Figure 3.11 Simple example of compelled edges in the representation of an equivalence class. (a) Original DAG $\mathcal{G}^*$. (b) Skeleton of $\mathcal{G}^*$ annotated with immoralities. (c) a DAG that is not equivalent to $\mathcal{G}^*$.

3.4.3.3 Representing Equivalence Classes

Once we have the skeleton and identified the immoralities, we have a specification of the equivalence class of $\mathcal{G}^*$. For example, to test if $\mathcal{G}$ is equivalent to $\mathcal{G}^*$ we can check whether it has the same skeleton as $\mathcal{G}^*$ and whether it agrees on the location of the immoralities.

The description of an equivalence class using only the skeleton and the set of immoralities is somewhat unsatisfying. For example, we might want to know whether the fact that our network is in the equivalence class implies that there is an arc $X \rightarrow Y$. Although the definition does tell us whether there is some edge between X and Y, it leaves the direction unresolved. In other cases, however, the direction of an edge is fully determined, for example, by the presence of an immorality. To encode both of these cases, we use a graph that allows both directed and undirected edges, as defined in section 2.2. Indeed, as we show, the chain graph, or PDAG, representation (definition 2.21) provides precisely the right framework.

Definition 3.15 class PDAG

Let $\mathcal{G}$ be a DAG. A chain graph $\mathcal{K}$ is a class PDAG *of the equivalence class of $\mathcal{G}$ if shares the same skeleton as $\mathcal{G}$, and contains a directed edge $X \rightarrow Y$ if and only if all $\mathcal{G}'$ that are I-equivalent to $\mathcal{G}$ contain the edge $X \rightarrow Y$.*[2] ■

In other words, a class PDAG represents potential edge orientations in the equivalence classes. If the edge is directed, then all the members of the equivalence class agree on the orientation of the edge. If the edge is undirected, there are two DAGs in the equivalence class that disagree on the orientation of the edge.

For example, the networks in figure 3.5a–(c) are I-equivalent. The class PDAG of this equivalence class is the graph $X—Z—Y$, since both edges can be oriented in either direction in some member of the equivalence class. Note that, although both edges in this PDAG are undirected, not all joint orientations of these edges are in the equivalence class. As discussed earlier, setting the orientations $X \rightarrow Z \leftarrow Y$ results in the network of figure 3.5d, which does not belong this equivalence class. More generally, if the class PDAG has k undirected edges, the equivalence class can contain at most 2^k networks, but the actual number can be much smaller.

2. For consistency with standard terminology, we use the PDAG terminology when referring to the chain graph representing an I-equivalence class.

Can we effectively construct the class PDAG $\mathcal{K}$ for $\mathcal{G}^*$ from the reconstructed skeleton and immoralities? Clearly, edges involved in immoralities must be directed in $\mathcal{K}$. The obvious question is whether $\mathcal{K}$ can contain directed edges that are not involved in immoralities. In other words, can there be additional edges whose direction is necessarily the same in every member of the equivalence class? To understand this issue better, consider the following example:

Example 3.9

Consider the DAG of figure 3.11a. This DAG has a single immorality $A \rightarrow C \leftarrow B$. This immorality implies that the class PDAG of this DAG must have the arcs $A \rightarrow C$ and $B \rightarrow C$ directed, as shown in figure 3.11b. This PDAG representation suggests that the edge C—D can assume either orientation. Note, however, that the DAG of figure 3.11c, where we orient the edge between C and D as $D \rightarrow C$, contains additional immoralities (that is, $A \rightarrow C \leftarrow D$ and $B \rightarrow C \leftarrow D$). Thus, this DAG is not *equivalent to our original DAG.*

In this example, there is only one possible orientation of C—D that is consistent with the finding that A—C—D is not an immorality. Thus, we conclude that the class PDAG for the DAG of figure 3.11a is simply the DAG itself. In other words, the equivalence class of this DAG is a singleton.■

As this example shows, a negative result in an immorality test also provides information about edge orientation. In particular, in any case where the PDAG $\mathcal{K}$ contains a structure $X \rightarrow Y$—Z and there is no edge from X to Z, then we must orient the edge $Y \rightarrow Z$, for otherwise we would create an immorality $X \rightarrow Y \leftarrow Z$.

Some thought reveals that there are other local configurations of edges where some ways of orienting edges are inconsistent, forcing a particular direction for an edge. Each such configuration can be viewed as a local constraint on edge orientation, give rise to a rule that can be used to orient more edges in the PDAG. Three such rules are shown in figure 3.12.

Let us understand the intuition behind these rules. Rule R1 is precisely the one we discussed earlier. Rule R2 is derived from the standard acyclicity constraint: If we have the directed path $X \rightarrow Y \rightarrow Z$, and an undirected edge X—Z, we cannot direct the edge $X \leftarrow Z$ without creating a cycle. Hence, we can conclude that the edge must be directed $X \rightarrow Z$. The third rule seems a little more complex, but it is also easily motivated. Assume, by contradiction, that we direct the edge $Z \rightarrow X$. In this case, we cannot direct the edge X—Y_1 as $X \rightarrow Y_1$ without creating a cycle; thus, we must have $Y_1 \rightarrow X$. Similarly, we must have $Y_2 \rightarrow X$. But, in this case, $Y_1 \rightarrow X \leftarrow Y_2$ forms an immorality (as there is no edge between Y_1 and Y_2), which contradicts the fact that the edges X—Y_1 and X—Y_2 are undirected in the original PDAG.

These three rules can be applied constructively in an obvious way: A rule applies to a PDAG whenever the induced subgraph on a subset of variables exactly matches the graph on the left-hand side of the rule. In that case, we modify this subgraph to match the subgraph on the right-hand side of the rule. Note that, by applying one rule and orienting a previously undirected edge, we create a new graph. This might create a subgraph that matches the antecedent of a rule, enforcing the orientation of additional edges. This process, however, must terminate at some point (since we are only adding orientations at each step, and the number of edges is finite). This implies that iterated application of this local constraint to the graph (a process known as *constraint propagation*) is guaranteed to converge.

constraint propagation

Algorithm 3.5 implements this process. It builds an initial graph using Build-PMap-Skeleton and Mark-Immoralities, and then iteratively applies the three rules until convergence, that is, until we cannot find a subgraph that matches a left-hand side of any of the rules.

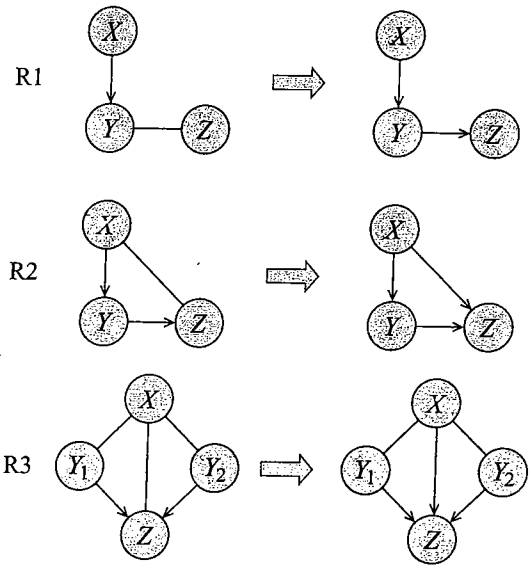


Figure 3.12 Rules for orienting edges in PDAG. Each rule lists a configuration of edges before and after an application of the rule.

Algorithm 3.5 Finding the class PDAG characterizing the P-map of a distribution P

Procedure Build-PDAG (
  $\mathcal{X} = \{X_1, \ldots, X_n\}$ // A specification of the random variables
  P // Distribution of interest
)

1 $S, \{U_{X_i,X_j}\} \leftarrow$ Build-PMap-Skeleton($\mathcal{X}, P$)
2 $\mathcal{K} \leftarrow$ Find-Immoralities($\mathcal{X}, S, \{U_{X_i,X_j}\}$)
3 **while** not converged
4 Find a subgraph in $\mathcal{K}$ matching the left-hand side of a rule R1–R3
5 Replace the subgraph with the right-hand side of the rule
6 **return** K

Example 3.10

Consider the DAG shown in figure 3.13a. After checking for immoralities, we find the graph shown in figure 3.13b. Now, we can start applying the preceding rules. For example, consider the variables B, E, and F. They induce a subgraph that matches the left-hand side of rule R1. Thus, we orient the edge between E and F to $E \rightarrow F$. Now, consider the variables C, E, and F. Their induced subgraph matches the left-hand side of rule R2, so we now orient the edge between C and F to $C \rightarrow F$. At this stage, if we consider the variables E, F, G, we can apply the rule R1, and orient the edge $F \rightarrow G$. (Alternatively, we could have arrived at the same orientation using C, F, and G.) The resulting PDAG is shown in figure 3.13c. ■

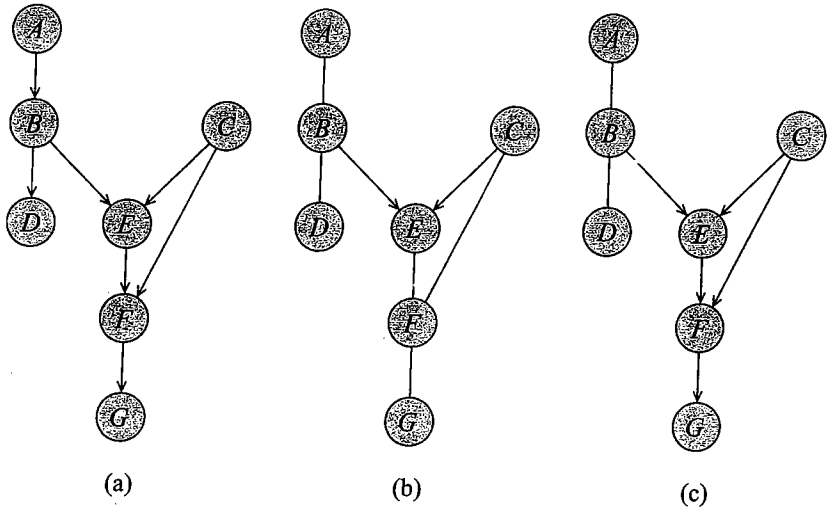


Figure 3.13 More complex example of compelled edges in the representation of an equivalence class. (a) Original DAG $\mathcal{G}^*$. (b) Skeleton of $\mathcal{G}^*$ annotated with immoralities. (c) Complete PDAG representation of the equivalence class of $\mathcal{G}^*$.

It seems fairly obvious that this algorithm is guaranteed to be sound: Any edge that is oriented by this procedure is, indeed, directed in exactly the same way in all of the members of the equivalence class. Much more surprising is the fact that it is also complete: Repeated application of these three local rules is guaranteed to capture all edge orientations in the equivalence class, without the need for additional global constraints. More precisely, we can prove that this algorithm produces the correct class PDAG for the distribution P:

Theorem 3.10

Let P be a distribution that has a P-map $\mathcal{G}^*$, and let $\mathcal{K}$ be the PDAG returned by Build-PDAG($\mathcal{X}, P$). *Then, $\mathcal{K}$ is a class PDAG of $\mathcal{G}^*$.*

The proof of this theorem can be decomposed into several aspects of correctness. We have already established the correctness of the skeleton found by Build-PMap-Skeleton. Thus, it remains to show that the directionality of the edges is correct. Specifically, we need to establish three basic facts:

- **Acyclicity:** The graph returned by Build-PDAG($\mathcal{X}$,P) is acyclic.
- **Soundness:** If $X \rightarrow Y \in \mathcal{K}$, then $X \rightarrow Y$ appears in all DAGs in $\mathcal{G}^*$'s I-equivalence class.
- **Completeness:** If $X—Y \in \mathcal{K}$, then we can find a DAG $\mathcal{G}$ that is I-equivalent to $\mathcal{G}^*$ such that $X \rightarrow Y \in \mathcal{G}$.

The last condition establishes completeness, since there is no constraint on the direction of the arc. In other words, the same condition can be used to prove the existence of a graph with $X \rightarrow Y$ and of a graph with $Y \rightarrow X$. Hence, it shows that either direction is possible within the equivalence class.

We begin with the soundness of the procedure.

Proposition 3.3 *Let P be a distribution that has a P-map $\mathcal{G}^*$, and let $\mathcal{K}$ be the graph returned by* Build-PDAG($\mathcal{X}, P$). *Then, if $X \rightarrow Y \in \mathcal{K}$, then $X \rightarrow Y$ appears in all DAGs in the I-equivalence class of $\mathcal{G}^*$.*

The proof is left as an exercise (exercise 3.23).

Next, we consider the acyclicity of the graph. We start by proving a property of graphs returned by the procedure. (Note that, once we prove that the graph returned by the procedure is the correct PDAG, it will follow that this property also holds for class PDAGs in general.)

Proposition 3.4 *Let $\mathcal{K}$ be the graph returned by* Build-PDAG. *Then, if $X \rightarrow Y \in \mathcal{K}$ and $Y - Z \in \mathcal{K}$, then $X \rightarrow Z \in \mathcal{K}$.*

The proof is left as an exercise (exercise 3.24).

Proposition 3.5 *Let $\mathcal{K}$ be the chain graph returned by* Build-PDAG. *Then $\mathcal{K}$ is acyclic.*

PROOF Suppose, by way of contradiction, that $\mathcal{K}$ contains a cycle. That is, there is a (partially) directed path $X_1 \rightleftharpoons X_2 \rightleftharpoons \ldots \rightleftharpoons X_n \rightleftharpoons X_1$. Without loss of generality, assume that this path is the shortest cycle in $\mathcal{K}$. We claim that the path cannot contain an undirected edge. To see that, suppose that the the path contains the triplet $X_i \rightarrow X_{i+1} - X_{i+2}$. Then, invoking proposition 3.4, we have that $X_i \rightarrow X_{i+2} \in \mathcal{K}$, and thus, we can construct a shorter path without X_{i+1} that contains the edge $X_i \rightarrow X_{i+2}$. At this stage, we have a directed cycle $X_1 \rightarrow X_2 \rightarrow \ldots X_n \rightarrow X_1$. Using proposition 3.3, we conclude that this cycle appears in any DAG in the I-equivalence class, and in particular in $\mathcal{G}^*$. This conclusion contradicts the assumption that $\mathcal{G}^*$ is acyclic. It follows that $\mathcal{K}$ is acyclic. ■

The final step is the completeness proof. Again, we start by examining a property of the graph $\mathcal{K}$.

Proposition 3.6 *The PDAG $\mathcal{K}$ returned by* Build-PDAG *is necessarily chordal.*

The proof is left as an exercise (exercise 3.25).

This property allows us to characterize the structure of the PDAG $\mathcal{K}$ returned by Build-PDAG. Recall that, since $\mathcal{K}$ is an undirected chain graph, we can partition $\mathcal{X}$ into chain components $\boldsymbol{K}_1, \ldots, \boldsymbol{K}_\ell$, where each chain component contains variables that are connected by undirected edges (see definition 2.21). It turns out that, in an undirected chordal graph, we can orient any edge in any direction without creating an immorality.

Proposition 3.7 *Let $\mathcal{K}$ be a undirected chordal graph over $\mathcal{X}$, and let $X, Y \in \mathcal{X}$. Then, there is a DAG $\mathcal{G}$ such that*

(a) The skeleton of $\mathcal{G}$ is $\mathcal{K}$.

(b) $\mathcal{G}$ does not contain immoralities.

(c) $X \rightarrow Y \in \mathcal{G}$.

The proof of this proposition requires some additional machinery that we introduce in chapter 4, so we defer the proof to that chapter.

Using this proposition, we see that we can orient edges in the chain component $\boldsymbol{K}_j$ without introducing immoralities within the component. We still need to ensure that orienting an edge $X—Y$ within a component cannot introduce an immorality involving edges from outside the component. To see why this situation cannot occur, suppose we orient the edge $X \rightarrow Y$, and suppose that $Z \rightarrow Y \in \mathcal{K}$. This seems like a potential immorality. However, applying proposition 3.4, we see that since $Z \rightarrow Y$ and $Y—X$ are in $\mathcal{K}$, then so must be $Z \rightarrow X$. Since Z is a parent of both X and Y, we have that $X \rightarrow Y \leftarrow Z$ is not an immorality. This argument applies to any edge we orient within an undirected component, and thus no new immoralities are introduced.

With these tools, we can complete the completeness proof of Build-PDAG.

Proposition 3.8 *Let P be a distribution that has a P-map $\mathcal{G}^*$, and let $\mathcal{K}$ be the graph returned by* Build-PDAG$(\mathcal{X}, P)$. *If $X—Y \in \mathcal{K}$, then we can find a DAG $\mathcal{G}$ that is I-equivalent to $\mathcal{G}^*$ such that $X \rightarrow Y \in \mathcal{G}$.*

PROOF Suppose we have an undirected edge $X—Y \in \mathcal{K}$. We want to show that there is a DAG $\mathcal{G}$ that has the same skeleton and immoralities as $\mathcal{K}$ such that $X \rightarrow Y \in \mathcal{G}$. If can build such a graph $\mathcal{G}$, then clearly it is in the I-equivalence class of $\mathcal{G}^*$.

The construction is simple. We start with the chain component that contains $X—Y$, and use proposition 3.7 to orient the edges in the component so that $X \rightarrow Y$ is in the resulting DAG. Then, we use the same construction to orient all other chain components. Since the chain components are ordered and acyclic, and our orientation of each chain component is acyclic, the resulting directed graph is acyclic. Moreover, as shown, the new orientation in each component does not introduce immoralities. Thus, the resulting DAG has exactly the same skeleton and immoralities as $\mathcal{K}$.

■

3.5 Summary

In this chapter, we discussed the issue of specifying a high-dimensional joint distribution compactly by exploiting its independence properties. We provided two complementary definitions of a Bayesian network. The first is as a directed graph $\mathcal{G}$, annotated with a set of conditional probability distributions $P(X_i \mid \mathrm{Pa}_{X_i})$. The network together with the CPDs define a distribution via the chain rule for Bayesian networks. In this case, we say that P factorizes over $\mathcal{G}$. We also defined the independence assumptions associated with the graph: the local independencies, the set of basic independence assumptions induced by the network structure; and the larger set of global independencies that are derived from the d-separation criterion. We showed the equivalence of these three fundamental notions: P factorizes over $\mathcal{G}$ if and only if P satisfies the local independencies of $\mathcal{G}$, which holds if and only if P satisfies the global independencies derived from d-separation. This result shows the equivalence of our two views of a Bayesian network: as a scaffolding for factoring a probability distribution P, and as a representation of a set of independence assumptions that hold for P. We also showed that the set of independencies derived from d-separation is a complete characterization of the independence properties that are implied by the graph structure alone, rather than by properties of a specific distribution over $\mathcal{G}$.

We defined a set of basic notions that use the characterization of a graph as a set of independencies. We defined the notion of a *minimal I-map* and showed that almost every distribution

has multiple I-maps, but that a minimal I-map for P does not necessarily capture all of the independence properties in P. We then defined a more stringent notion of a *perfect map*, and showed that not every distribution has a perfect map. We defined *I-equivalence*, which captures an independence-equivalence relationship between two graphs, one where they specify precisely the same set of independencies.

Finally, we defined the notion of a *class PDAG*, a partially directed graph that provides a compact representation for an entire I-equivalence class, and we provided an algorithm for constructing this graph.

These definitions and results are fundamental properties of the Bayesian network representation and its semantics. Some of the algorithms that we discussed are never used as is; for example, we never directly use the procedure to find a minimal I-map given an explicit representation of the distribution. However, these results are crucial to understanding the cases where we can construct a Bayesian network that reflects our understanding of a given domain, and what the resulting network means.

3.6 Relevant Literature

The use of a directed graph as a framework for analyzing properties of distributions can be traced back to the path analysis of Wright (1921, 1934).

influence diagram

The use of a directed acyclic graph to encode a general probability distribution (not within a specific domain) was first proposed within the context of *influence diagrams*, a decision-theoretic framework for making decisions under uncertainty (see chapter 23). Within this setting, Howard and Matheson (1984b) and Smith (1989) both proved the equivalence between the ability to represent a distribution as a DAG and the local independencies (our theorem 3.1 and theorem 3.2).

The notion of Bayesian networks as a qualitative data structure encoding independence relationships was first proposed by Pearl and his colleagues in a series of papers (for example, Verma and Pearl 1988; Geiger and Pearl 1988; Geiger et al. 1989, 1990), and in Pearl's book *Probabilistic Reasoning in Intelligent Systems* (Pearl 1988). Our presentation of I-maps, P-maps, and Bayesian networks largely follows the trajectory laid forth in this body of work.

The definition of d-separation was first set forth by Pearl (1986b), although without formal justification. The soundness of d-separation was shown by Verma (1988), and its completeness for the case of Gaussian distributions by Geiger and Pearl (1993). The measure-theoretic notion of completeness of d-separation, stating that almost all distributions are faithful (theorem 3.5), was shown by Meek (1995b). Several papers have been written exploring the yet stronger notion of completeness for d-separation (faithfulness for all distributions that are minimal I-maps), in various subclasses of models (for example, Becker et al. 2000). The *BayesBall* algorithm, an elegant and efficient algorithm for d-separation and a class of related problems, was proposed by (Shachter 1998).

BayesBall

The notion of I-equivalence was defined by Verma and Pearl (1990, 1992), who also provided and proved the graph-theoretic characterization of theorem 3.8. Chickering (1995) provided the alternative characterization of I-equivalence in terms of covered edge reversal. This definition provides an easy mechanism for proving important properties of I-equivalent networks. As we will see later in the book, the notion of I-equivalence class plays an important role in identifying networks, particularly when learning networks from data. The first algorithm for constructing

a perfect map for a distribution, in the form of an I-equivalence class, was proposed by Pearl and Verma (1991); Verma and Pearl (1992). This algorithm was subsequently extended by Spirtes et al. (1993) and by Meek (1995a). Meek also provides an algorithm for finding all of the directed edges that occur in every member of the I-equivalence class.

inclusion

A notion related to I-equivalence is that of *inclusion*, where the set of independencies $\mathcal{I}(\mathcal{G}')$ is *included* in the set of independencies $\mathcal{I}(\mathcal{G})$ (so that $\mathcal{G}$ is an I-map for any distribution that factorizes over $\mathcal{G}'$). Shachter (1989) showed how to construct a graph $\mathcal{G}'$ that includes a graph $\mathcal{G}$, but with one edge reversed. Meek (1997) conjectured that inclusion holds if and only if one can transform $\mathcal{G}$ to $\mathcal{G}'$ using the operations of edge addition and covered edge reversal. A limited version of this conjecture was subsequently proved by Kočka, Bouckaert, and Studený (2001).

The naive Bayes model, although naturally represented as a graphical model, far predates this view. It was applied with great success within expert systems in the 1960s and 1970s (de Bombal et al. 1972; Gorry and Barnett 1968; Warner et al. 1961). It has also seen significant use as a simple yet highly effective method for classification tasks in machine learning, starting as early as the 1970s (for example, Duda and Hart 1973), and continuing to this day.

The general usefulness of the types of reasoning patterns supported by a Bayesian network, including the very important pattern of intercausal reasoning, was one of the key points raised by Pearl in his book (Pearl 1988). These qualitative patterns were subsequently formalized by Wellman (1990) in his framework of *qualitative probabilistic networks*, which explicitly annotate arcs with the direction of influence of one variable on another. This framework has been used to facilitate knowledge elicitation and knowledge-guided learning (Renooij and van der Gaag 2002; Hartemink et al. 2002) and to provide verbal explanations of probabilistic inference (Druzdzel 1993).

qualitative probabilistic networks

There have been many applications of the Bayesian network framework in the context of real-world problems. The idea of using directed graphs as a model for genetic inheritance appeared as far back as the work on path analysis of Wright (1921, 1934). A presentation much closer to modern-day Bayesian networks was proposed by Elston and colleagues in the 1970s (Elston and Stewart 1971; Lange and Elston 1975). More recent developments include the development of better algorithms for inference using these models (for example, Kong 1991; Becker et al. 1998; Friedman et al. 2000) and the construction of systems for genetic linkage analysis based on this technology (Szolovits and Pauker 1992; Schäffer 1996).

Many of the first applications of the Bayesian network framework were to medical expert systems. The Pathfinder system is largely the work of David Heckerman and his colleagues (Heckerman and Nathwani 1992a; Heckerman et al. 1992; Heckerman and Nathwani 1992b). The success of this system as a diagnostic tool, including its ability to outperform expert physicians, was one of the major factors that led to the rise in popularity of probabilistic methods in the early 1990s. Several other large diagnostic networks were developed around the same period, including MUNIN (Andreassen et al. 1989), a network of over 1000 nodes used for interpreting electromyographic data, and QMR-DT (Shwe et al. 1991; Middleton et al. 1991), a probabilistic reconstruction of the QMR/INTERNIST system (Miller et al. 1982) for general medical diagnosis.

The problem of knowledge acquisition of network models has received some attention. Probability elicitation is a long-standing question in decision analysis; see, for example, Spetzler and von Holstein (1975); Chesley (1978). Unfortunately, elicitation of probabilities from humans is a difficult process, and one subject to numerous biases (Tversky and Kahneman 1974; Daneshkhah 2004). Shachter and Heckerman (1987) propose the "backward elicitation" approach for obtaining

similarity network

both the network structure and the parameters from an expert. *Similarity networks* (Heckerman and Nathwani 1992a; Geiger and Heckerman 1996) generalize this idea by allowing an expert to construct several small networks for differentiating between "competing" diagnoses, and then superimposing them to construct a single large network. Morgan and Henrion (1990) provide an overview of knowledge elicitation methods.

The difficulties in eliciting accurate probability estimates from experts are well recognized across a wide range of disciplines. In the specific context of Bayesian networks, this issue has been tackled in several ways. First, there has been both empirical (Pradhan et al. 1996) and theoretical (Chan and Darwiche 2002) analysis of the extent to which the choice of parameters affects the conclusions of the inference. Overall, the results suggest that even fairly significant changes to network parameters cause only small degradations in performance, except when the changes relate to extreme parameters — those very close to 0 and 1. Second, the concept of *sensitivity analysis* (Morgan and Henrion 1990) is used to allow researchers to evaluate the sensitivity of their specific network to variations in parameters. Largely, sensitivity has been measured using the derivative of network queries relative to various parameters (Laskey 1995; Castillo et al. 1997b; Kjærulff and van der Gaag 2000; Chan and Darwiche 2002), with the focus of most of the work being on properties of sensitivity values and on efficient algorithms for estimating them.

sensitivity analysis

As pointed out by Pearl (1988), the notion of a Bayesian network structure as a representation of independence relationships is a fundamental one, which transcends the specifics of probabilistic representations. There have been many proposed variants of Bayesian networks that use a nonprobabilistic "parameterization" of the local dependency models. Examples include various logical calculi (Darwiche 1993), Dempster-Shafer belief functions (Shenoy 1989), possibility values (Dubois and Prade 1990), qualitative (order-of-magnitude) probabilities (known as kappa rankings; Darwiche and Goldszmidt 1994), and interval constraints on probabilities (Fertig and Breese 1989; Cozman 2000).

The acyclicity constraint of Bayesian networks has led to many concerns about its ability to express certain types of interactions. There have been many proposals intended to address this limitation. Markov networks, based on undirected graphs, present a solution for certain types of interactions; this class of probability models are described in chapter 4. Dynamic Bayesian networks "stretch out" the interactions over time, therefore providing an acyclic version of feedback loops; these models are are described in section 6.2.

cyclic graphical model

There has also been some work on directed models that encode cyclic dependencies directly. *Cyclic graphical models* (Richardson 1994; Spirtes 1995; Koster 1996; Pearl and Dechter 1996) are based on distributions over systems of simultaneous linear equations. These models are a natural generalization of Gaussian Bayesian networks (see chapter 7), and are also associated with notions of d-separation or I-equivalence. Spirtes (1995) shows that this connection breaks down when the system of equations is nonlinear and provides a weaker version for the cyclic case.

dependency network

Dependency networks (Heckerman et al. 2000) encode a set of local dependency models, representing the conditional distribution of each variable on all of the others (which can be compactly represented by its dependence on its Markov blanket). A dependency network represents a probability distribution only indirectly, and is only guaranteed to be coherent under certain conditions. However, it provides a local model of dependencies that is very naturally interpreted by people.

3.7 Exercises

Exercise 3.1

Provide an example of a distribution $P(X_1, X_2, X_3)$ where for each $i \neq j$, we have that $(X_i \perp X_j) \in \mathcal{I}(P)$, but we also have that $(X_1, X_2 \perp X_3) \notin \mathcal{I}(P)$.

Exercise 3.2

a. Show that the naive Bayes factorization of equation (3.7) follows from the naive Bayes independence assumptions of equation (3.6).

b. Show that equation (3.8) follows from equation (3.7).

c. Show that, if all the variables $C, X_1, \ldots, X_n$ are binary-valued, then $\log \frac{P(C=c^1|x_1,\ldots,x_n)}{P(C=c^2|x_1,\ldots,x_n)}$ is a linear function of the value of the finding variables, that is, can be written as $\sum_{i=1}^{n} \alpha_i X_i + \alpha_0$ (where $X_i = 0$ if $X = x^0$ and 1 otherwise).

Exercise 3.3

Consider a simple example (due to Pearl), where a burglar alarm (A) can be set off by either a burglary (B) or an earthquake (E).

a. Define constraints on the CPD of $P(A \mid B, E)$ that imply the *explaining away* property.

b. Show that if our model is such that the alarm always (deterministically) goes off whenever there is a earthquake:

$$P(a^1 \mid b^1, e^1) = P(a^1 \mid b^0, e^1) = 1$$

then $P(b^1 \mid a^1, e^1) = P(b^1)$, that is, observing an earthquake provides a full explanation for the alarm.

Exercise 3.4

We have mentioned that explaining away is one type of intercausal reasoning, but that other type of intercausal interactions are also possible. Provide a realistic example that exhibits the opposite type of interaction. More precisely, consider a v-structure $X \rightarrow Z \leftarrow Y$ over three binary-valued variables. Construct a CPD $P(Z \mid X, Y)$ such that:

- X and Y both increase the probability of the effect, that is, $P(z^1 \mid x^1) > P(z^1)$ and $P(z^1 \mid y^1) > P(z^1)$,
- each of X and Y increases the probability of the other, that is, $P(x^1 \mid z^1) < P(x^1 \mid y^1, z^1)$, and similarly $P(y^1 \mid z^1) < P(y^1 \mid x^1, z^1)$.

Note that strong (rather than weak) inequality must hold in all cases.

Your example should be realistic, that is, X, Y, Z should correspond to real-world variables, and the CPD should be reasonable.

Exercise 3.5

Consider the Bayesian network of figure 3.14.

Assume that all variables are binary-valued. We do not know the CPDs, but do know how each random variable *qualitatively* affects its children. The influences, shown in the figure, have the following interpretation:

- $X \xrightarrow{+} Y$ means $P(y^1 \mid x^1, \boldsymbol{u}) > P(y^1 \mid x^0, \boldsymbol{u})$, for all values $\boldsymbol{u}$ of Y's other parents.
- $X \xrightarrow{-} Y$ means $P(y^1 \mid x^1, \boldsymbol{u}) < P(y^1 \mid x^0, \boldsymbol{u})$, for all values $\boldsymbol{u}$ of Y's other parents.

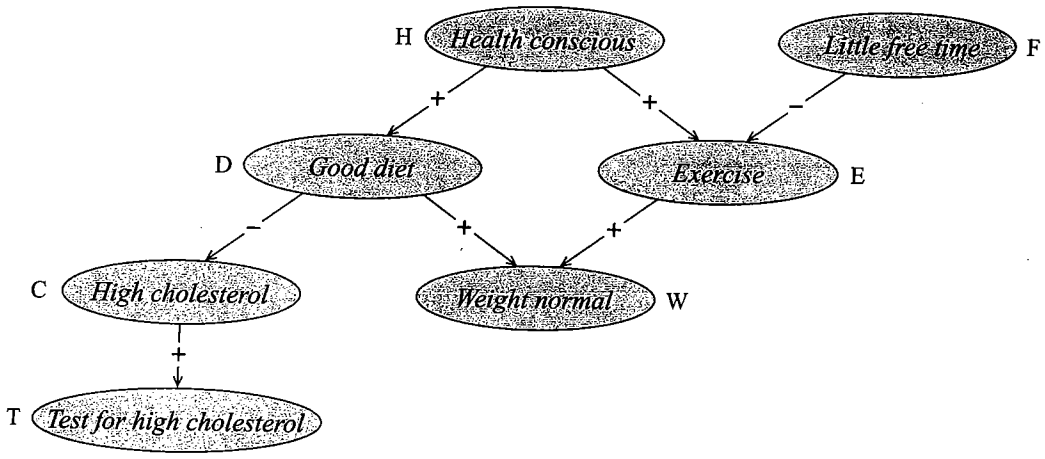


Figure 3.14 A Bayesian network with qualitative influences

We also assume explaining away as the interaction for all cases of intercausal reasoning.

For each of the following pairs of conditional probability queries, use the information in the network to determine if one is larger than the other, if they are equal, or if they are incomparable. For each pair of queries, indicate all relevant active trails, and their direction of influence.

(a)	$P(t^1 \mid d^1)$	$P(t^1)$
(b)	$P(d^1 \mid t^0)$	$P(d^1)$
(c)	$P(h^1 \mid e^1, f^1)$	$P(h^1 \mid e^1)$
(d)	$P(c^1 \mid f^0)$	$P(c^1)$
(e)	$P(c^1 \mid h^0)$	$P(c^1)$
(f)	$P(c^1 \mid h^0, f^0)$	$P(c^1 \mid h^0)$
(g)	$P(d^1 \mid h^1, e^0)$	$P(d^1 \mid h^1)$
(h)	$P(d^1 \mid e^1, f^0, w^1)$	$P(d^1 \mid e^1, f^0)$
(i)	$P(t^1 \mid w^1, f^0)$	$P(t^1 \mid w^1)$

Exercise 3.6

Consider a set of variables $X_1, \ldots, X_n$ where each X_i has $|Val(X_i)| = \ell$.

a. Assume that we have a Bayesian network over $X_1, \ldots, X_n$, such that each node has at most k parents. What is a simple upper bound on the number of independent parameters in the Bayesian network? How many independent parameters are in the full joint distribution over $X_1, \ldots, X_n$?

b. Now, assume that each variable X_i has the parents $X_1, \ldots, X_{i-1}$. How many independent parameters are there in the Bayesian network? What can you conclude about the expressive power of this type of network?

c. Now, consider a naive Bayes model where $X_1, \ldots, X_n$ are evidence variables, and we have an additional class variable C, which has k possible values $c_1, \ldots, c_k$. How many independent parameters are required to specify the naive Bayes model? How many independent parameters are required for an explicit representation of the joint distribution?

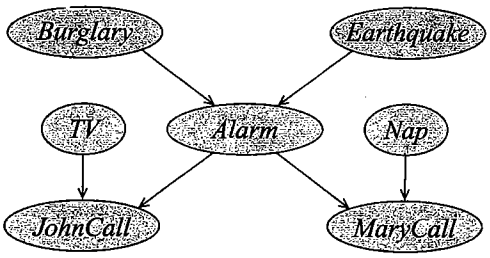


Figure 3.15 A simple network for a burglary alarm **domain**

Exercise 3.7

Show how you could efficiently compute the distribution over a variable X_i given some assignment to all the other variables in the network: $P(X_i \mid x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$. Your procedure should not require the construction of the entire joint distribution $P(X_1, \ldots, X_n)$. Specify the computational complexity of your procedure.

Exercise 3.8

Let $\mathcal{B} = (\mathcal{G}, P)$ be a Bayesian network over some set of variables $\mathcal{X}$. Consider some subset of evidence nodes $\boldsymbol{Z}$, and let $\boldsymbol{X}$ be all of the ancestors of the nodes in $\boldsymbol{Z}$. Let $\mathcal{B}'$ be a network over the induced subgraph over $\boldsymbol{X}$, where the CPD for every node $X \in \boldsymbol{X}$ is the same in $\mathcal{B}'$ as in $\mathcal{B}$. Prove that the joint distribution over $\boldsymbol{X}$ is the same in $\mathcal{B}$ and in $\mathcal{B}'$. The nodes in $\mathcal{X} - \boldsymbol{X}$ are called *barren nodes* relative to $\boldsymbol{X}$, because (when not instantiated) they are irrelevant to computations concerning $\boldsymbol{X}$.

barren node

Exercise 3.9⋆

Prove theorem 3.2 for a general BN structure $\mathcal{G}$. Your proof should not use the soundness of d-separation.

Exercise 3.10

Prove that the global independencies, derived from d-separation, imply the local independencies. In other words, prove that a node is d-separated from its nondescendants given its parents.

Exercise 3.11⋆

One operation on Bayesian networks that arises in many settings is the marginalization of some node in the network.

a. Consider the Burglary Alarm network $\mathcal{B}$ shown in figure 3.15. Construct a Bayesian network $\mathcal{B}'$ over all of the nodes except for *Alarm* that is a minimal I-map for the marginal distribution $P_{\mathcal{B}}(B, E, T, N, J, M)$. Be sure to get *all* dependencies that remain from the original network.

b. Generalize the procedure you used to solve the preceding problem into a node elimination algorithm. That is, define an algorithm that transforms the structure of $\mathcal{G}$ into $\mathcal{G}'$ such that one of the nodes X_i of $\mathcal{G}$ is not in $\mathcal{G}'$ and $\mathcal{G}'$ is an I-map of the marginal distribution over the remaining variables as defined by $\mathcal{G}$.

Exercise 3.12⋆⋆

edge reversal

Another operation on Bayesian networks that arises often is *edge reversal*. This involves transforming a Bayesian network $\mathcal{G}$ containing nodes X and Y as well as arc $X \rightarrow Y$ into another Bayesian network $\mathcal{G}'$ with reversed arc $Y \rightarrow X$. However, we want $\mathcal{G}'$ to represent the same distribution as $\mathcal{G}$; therefore, $\mathcal{G}'$ will need to be an I-map of the original distribution.

a. Consider the Bayesian network structure of figure 3.15. Suppose we wish to reverse the arc $B \rightarrow A$. What additional minimal modifications to the structure of the network are needed to ensure that the new network is an I-map of the original distribution? Your network should not reverse any additional edges, and it should differ only minimally from the original in terms of the number of edge additions or deletions. Justify your response.

b. Now consider a general Bayesian network $\mathcal{G}$. For simplicity, assume that the arc $X \rightarrow Y$ is the only directed trail from X to Y. Define a general procedure for reversing the arc $X \rightarrow Y$, that is, for constructing a graph $\mathcal{G}'$ is an I-map for the original distribution, but that contains an arc $Y \rightarrow X$ and otherwise differs minimally from $\mathcal{G}$ (in the same sense as before). Justify your response.

c. Suppose that we use the preceding method to transform $\mathcal{G}$ into a graph $\mathcal{G}'$ with a reversed arc between X and Y. Now, suppose we reverse that arc *back* to its original direction in $\mathcal{G}$ by repeating the preceding method, transforming $\mathcal{G}'$ into $\mathcal{G}''$. Are we guaranteed that the final network structure is equivalent to the original network structure ($\mathcal{G} = \mathcal{G}''$)?

Exercise 3.13⋆

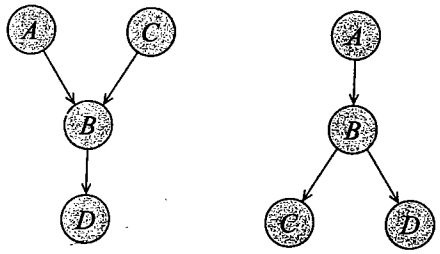
Let $\mathcal{B} = (\mathcal{G}, P)$ be a Bayesian network over $\mathcal{X}$. The Bayesian network is parameterized by a set of CPD parameters of the form $\theta_{x|u}$ for $X \in \mathcal{X}$, $\boldsymbol{U} = \mathrm{Pa}_X^{\mathcal{G}}$, $x \in Val(X)$, $\boldsymbol{u} \in Val(\boldsymbol{U})$. Consider any conditional independence statement of the form $(\boldsymbol{X} \perp \boldsymbol{Y} \mid \boldsymbol{Z})$. Show how this statement translates into a set of polynomial equalities over the set of CPD parameters $\theta_{x|u}$. (Note: A polynomial equality is an assertion of the form $a\theta_1^2 + b\theta_1\theta_2 + c\theta_2^3 + d = 0$.)

Exercise 3.14⋆

Prove theorem 3.6.

Exercise 3.15

Consider the two networks:



(a) (b)

For each of them, determine whether there can be any other Bayesian network that is I-equivalent to it.

Exercise 3.16⋆

Prove theorem 3.7.

Exercise 3.17⋆⋆

We proved earlier that two networks that have the same skeleton and v-structures imply the same conditional independence assumptions. As shown, this condition is not an if and only if. Two networks can have different v-structures, yet still imply the same conditional independence assumptions. In this problem, you will provide a condition that precisely relates I-equivalence and similarity of network structure.

minimal active trail

a. A key notion in this question is that of a *minimal active trail*. We define an active trail $X_1 \dots X_m$ to be *minimal* if there is no other active trail from X_1 to X_m that "shortcuts" some of the nodes, that is, there is no active trail $X_1 \rightleftharpoons X_{i_1} \rightleftharpoons \dots \rightleftharpoons X_{i_k} \rightleftharpoons X_m$ for $1 < i_1 < \dots < i_k < m$.

Our first goal is to analyze the types of "triangles" that can occur in a minimal active trail, that is, cases where we have $X_{i-1} \rightleftharpoons X_i \rightleftharpoons X_{i+1}$ with a direct edge between X_{i-1} and X_{i+1}. Prove that the only possible triangle in a minimal active trail is one where $X_{i-1} \leftarrow X_i \rightarrow X_{i+1}$, with an edge between X_{i-1} and X_{i+1}, and where either X_{i-1} or X_{i+1} are the center of a v-structure in the trail.

b. Now, consider two networks $\mathcal{G}_1$ and $\mathcal{G}_2$ that have the same skeleton and same immoralities. Prove, using the notion of minimal active trail, that $\mathcal{G}_1$ and $\mathcal{G}_2$ imply precisely the same conditional independence assumptions, that is, that if X and Y are d-separated given $\boldsymbol{Z}$ in $\mathcal{G}_1$, then X and Y are also d-separated given $\boldsymbol{Z}$ in $\mathcal{G}_2$.

c. Finally, prove the other direction. That is, prove that two networks $\mathcal{G}_1$ and $\mathcal{G}_2$ that induce the same conditional independence assumptions must have the same skeleton and the same immoralities.

Exercise 3.18⋆

In this exercise, you will prove theorem 3.9. This result provides an alternative reformulation of I-equivalence in terms of local operations on the graph structure.

covered edge

a. Let $\mathcal{G}$ be a directed graph with a *covered edge* $X \rightarrow Y$ (as in definition 3.12), and $\mathcal{G}'$ the graph that results by reversing the edge $X \rightarrow Y$ to produce $Y \rightarrow X$, but leaving everything else unchanged. Prove that $\mathcal{G}$ and $\mathcal{G}'$ are I-equivalent.

b. Provide a counterexample to this result in the case where $X \rightarrow Y$ is not a covered edge.

c. Now, prove that for every pair of I-equivalent networks $\mathcal{G}$ and $\mathcal{G}'$, there exists a sequence of covered edge reversal operations that converts $\mathcal{G}$ to $\mathcal{G}'$. Your proof should show how to construct this sequence.

Exercise 3.19⋆

Prove lemma 3.2.

Exercise 3.20⋆

requisite CPD

In this question, we will consider the sensitivity of a particular query $P(X \mid \boldsymbol{Y})$ to the CPD of a particular node Z. Let X and Z be nodes, and $\boldsymbol{Y}$ be a set of nodes. We say that Z has a *requisite CPD* for answering the query $P(\boldsymbol{X} \mid \boldsymbol{Y})$ if there are two networks $\mathcal{B}_1$ and $\mathcal{B}_2$ that have identical graph structure $\mathcal{G}$ and identical CPDs everywhere except at the node Z, and where $P_{\mathcal{B}_1}(X \mid \boldsymbol{Y}) \neq P_{\mathcal{B}_2}(X \mid \boldsymbol{Y})$; in other words, the CPD of Z affects the answer to this query.

This type of analysis is useful in various settings, including determining which CPDs we need to acquire for a certain query (and others that we discuss later in the book).

Show that we can test whether Z is a requisite probability node for $P(X \mid \boldsymbol{Y})$ using the following procedure: We modify $\mathcal{G}$ into a graph $\mathcal{G}'$ that contains a new "dummy" parent $\widehat{Z}$, and then test whether $\widehat{Z}$ has an active trail to $\boldsymbol{X}$ given $\boldsymbol{Y}$.

- Show that this is a sound criterion for determining whether Z is a requisite probability node for $P(X \mid \boldsymbol{Y})$ in $\mathcal{G}$, that is, for all pairs of networks $\mathcal{B}_1, \mathcal{B}_2$ as before, $P_{\mathcal{B}_1}(X \mid \boldsymbol{Y}) = P_{\mathcal{B}_2}(X \mid \boldsymbol{Y})$.
- Show that this criterion is weakly complete (like d-separation), in the sense that, if it fails to identify Z as requisite in $\mathcal{G}$, there exists some pair of networks $\mathcal{B}_1, \mathcal{B}_2$ as before, $P_{\mathcal{B}_1}(X \mid \boldsymbol{Y}) \neq P_{\mathcal{B}_2}(X \mid \boldsymbol{Y})$.

Exercise 3.21⋆

Define a set $\boldsymbol{Z}$ of nodes to be *self-contained* if, for every pair of nodes $A, B \in \boldsymbol{Z}$, and any *directed* path between A and B, all nodes along the trail are also in $\boldsymbol{Z}$.

a. Consider a self-contained set $\boldsymbol{Z}$, and let $\boldsymbol{Y}$ be the set of all nodes that are a parent of some node in $\boldsymbol{Z}$ but are not themselves in $\boldsymbol{Z}$. Let $\boldsymbol{U}$ be the set of nodes that are an ancestor of some node in $\boldsymbol{Z}$ but that are not already in $\boldsymbol{Y} \cup \boldsymbol{Z}$. (See figure 3.16.)

Prove, based on the d-separation properties of the network, that $(\boldsymbol{Z} \perp \boldsymbol{U} \mid \boldsymbol{Y})$. Make sure that your proof covers all possible cases.

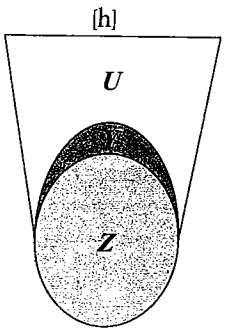


Figure 3.16 Illustration of the concept of a self-contained set

b. Provide a counterexample to this result if we retract the assumption that Z is self-contained. (Hint: 4 nodes are enough.)

Exercise 3.22⋆

We showed that the algorithm Build-PMap-Skeleton of algorithm 3.3 constructs the skeleton of the P-map of a distribution P if P has a P-map (and that P-map has indegrees bounded by the parameter d). In this question, we ask you consider what happens if P does not have a P-map.

There are two types of errors we might want to consider:

- **Missing edges**: The edge $X \rightleftharpoons Y$ appears in all the minimal I-maps of P, yet $X—Y$ is not in the skeleton S returned by Build-PMap-Skeleton.
- **Spurious edges**: The edge $X \rightleftharpoons Y$ does not appear in all of the minimal I-maps of P (but may appear in some of them), yet $X—Y$ is in the skeleton S returned by Build-PMap-Skeleton.

For each of these two types of errors, either prove that they cannot happen, or provide a counterexample (that is, a distribution P for which Build-PMap-Skeleton makes that type of an error).

Exercise 3.23⋆

In this exercise, we prove proposition 3.3. To help us with the proof, we need an auxiliary definition. We say that a partially directed graph $\mathcal{K}$ is a *partial class graph* for a DAG $\mathcal{G}^*$ if

a. $\mathcal{K}$ has the same skeleton as $\mathcal{G}^*$;
b. $\mathcal{K}$ has the same immoralities as $\mathcal{G}^*$;
c. if $X \rightarrow Y \in \mathcal{K}$, then $X \rightarrow Y \in \mathcal{G}$ for any DAG $\mathcal{G}$ that is I-equivalent to $\mathcal{G}^*$.

Clearly, the graph returned by by Mark-Immoralities is a partial class graph of $\mathcal{G}^*$.

Prove that if $\mathcal{K}$ is a partial class graph of $\mathcal{G}^*$, and we apply one of the rules R1–R3 of figure 3.12, then the resulting graph is also a partial class graph $\mathcal{G}^*$. Use this result to prove proposition 3.3 by induction.

Exercise 3.24⋆

Prove proposition 3.4. Hint: consider the different cases by which the edge $X \rightarrow Y$ was oriented during the procedure.

Exercise 3.25

Prove proposition 3.6. Hint: Show that this property is true of the graph returned by Mark-Immoralities.

Exercise 3.26

Implement an efficient algorithm that takes a Bayesian network over a set of variables $\mathcal{X}$ and a full instantiation ξ to $\mathcal{X}$, and computes the probability of ξ according to the network.

Exercise 3.27

Implement Reachable of algorithm 3.1.

Exercise 3.28⋆

Implement an efficient algorithm that determines, for a given set $\boldsymbol{Z}$ of observed variables and all pairs of nodes X and Y, whether X, Y are d-separated in $\mathcal{G}$ given $\boldsymbol{Z}$. Your algorithm should be significantly more efficient than simply running Reachable of algorithm 3.1 separately for each possible source variable X_i.