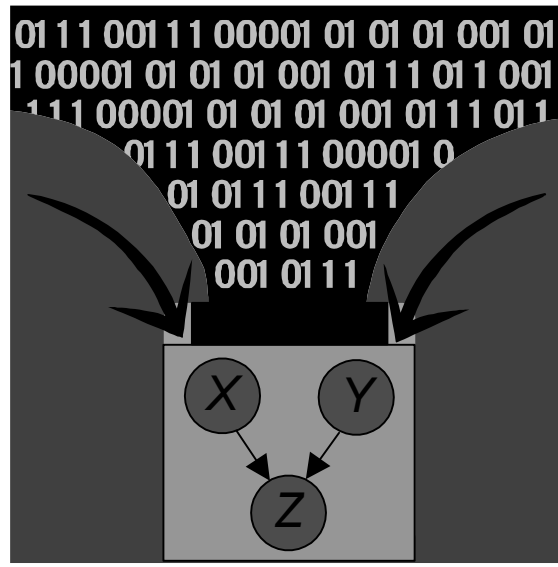
Chapter 8

Learning Bayesian Network Structure



In a Bayesian network, the DAG is called the **structure** and, as mentioned at the beginning of the previous chapter, the conditional probability distributions are called the *parameters*. In the previous chapter we discussed learning the parameters. In this chapter we address learning the structure. In Section 8.1, we formalize the notion of structure learning. Section 8.2 concerns score-based structure learning. Constraint-based structure learning is the focus of Section 8.3. In Section 8.4 we apply structure learning to inferring causal influences from data. When we do not have a large amount of data, ordinarily a unique DAG cannot be learned. In such cases, we can often still learn something of interest concerning the variables in the network by model averaging. We cover model averaging in Section 8.5. When there is a large number of variables, it

Case	Sex	Height (inches)	Wage ($)
1	female	64	30,000
2	male	64	30,000
3	female	64	40,000
4	female	64	40,000
5	female	68	30,000
6	female	68	40,000
7	male	64	40,000
8	male	64	50,000
9	male	68	40,000
10	male	68	50,000
11	male	70	40,000
12	male	70	50,000

Table 8.1: Data on 12 workers.

is necessary to do approximate structure learning. This idea is discussed in Section 8.6. Finally, Section 8.7 presents learning packages that implement the methods discussed in earlier sections.

8.1 Model Selection

Structure learning consists of learning the DAG in a Bayesian network from data. To accomplish this, we need to learn a DAG that satisfies the Markov condition with the probability distribution P that is generating the data. Note that we do not know P; all we know are the data. The process of learning such a DAG is called **model selection**.

Example 8.1 *Suppose we want to model the probability distribution P of sex, height, and wage for American workers. We may obtain the data on 12 workers appearing in Table 8.1. We don't know the probability distribution P. However, from these data we want to learn a DAG that is likely to satisfy the Markov condition with P.*

If our only goal was simply learning some DAG that satisfied the Markov condition with P, our task would be trivial because, as discussed in Section 6.2.1, a probability distribution P satisfies the Markov condition with every complete DAG containing the variables over which P is defined. Recall that our goal with a Bayesian network is to represent a probability distribution succinctly. A complete DAG does not accomplish this because, if there are n binomial variables, the last variable in a complete DAG would require 2^{n-1} conditional distributions. To represent a distribution P succinctly, we need to find a sparse DAG (one containing few edges) that satisfies the Markov condition with P. Next we present methods for doing this.

8.2 Score-Based Structure Learning

In **score-based structure learning**, we assign a score to a DAG based on how well the DAG fits the data. Here we discuss two such scores.

8.2.1 The Bayesian Score

The most straightforward score, called the **Bayesian score**, is the probability of the data D given the DAG. We present this score shortly. However, first we need to discuss the probability of data.

Probability of Data

Suppose that we are going to toss the same coin two times in a row. Let X_1 be a random variable whose value is the result of the first toss, and let X_2 be a random variable whose value is the result of the second toss. If we know that the probability of heads for this coin is .5 and make the usual assumption that the outcomes of the two tosses are independent, we have

$$P(X_1 = heads, X_2 = heads) = \frac{1}{2} \times \frac{1}{2} = \frac{1}{4}.$$

This is a standard result. Suppose now that we are going to toss a thumbtack two times in a row. Suppose further we represent our prior belief concerning the probability of heads using a Dirichlet distribution with parameters a and b (as discussed in Section 7.1.1), and we represent prior ignorance of the probability of heads by taking $a = b = 1$. If the outcome of the first toss is heads, using the notation developed in Section 7.1.1, our updated probability of heads is

$$P(heads|1, 0) = \frac{a+1}{a+b+1} = \frac{1+1}{1+1+1} = \frac{2}{3}.$$

Heads is more probable for the second toss because our belief has changed owing to heads occurring on the first toss. So, using our current notation in which we have articulated two random variables, we have that

$$P(X_2 = heads|X_1 = heads) = P(heads|1, 0) = \frac{2}{3},$$

and

$$\begin{aligned} P(X_1 = heads, X_2 = heads) &= P(X_2 = heads|X_1 = heads)P(X_1 = heads) \\ &= \frac{2}{3} \times \frac{1}{2} = \frac{1}{3} \end{aligned}$$

$$\begin{aligned} P(X_1 = heads, X_2 = tails) &= P(X_2 = tails|X_1 = heads)P(X_1 = heads) \\ &= \frac{1}{3} \times \frac{1}{2} = \frac{1}{6} \end{aligned}$$

$$\begin{aligned} P(X_1 = tails, X_2 = heads) &= P(X_2 = heads|X_1 = tails)P(X_1 = tails) \\ &= \frac{1}{3} \times \frac{1}{2} = \frac{1}{6} \end{aligned}$$

$$\begin{aligned} P(X_1 = tails, X_2 = tails) &= P(X_2 = tails|X_1 = tails)P(X_1 = tails) \\ &= \frac{2}{3} \times \frac{1}{2} = \frac{1}{3}. \end{aligned}$$

It might seem odd to you that the four outcomes do not have the same probability. However, recall that we do not know the probability of heads. Therefore, we learn something about the probability of heads from the result of the first toss. If heads occurs on the first toss, that probability goes up; if tails occurs, it goes down. So, two consecutive heads or two consecutive tails are more probable *a priori* than a head followed by a tail or a tail followed by a head.

This result extends readily to a sequence of tosses. For example, suppose we toss the thumbtack three times. Then, owing to the chain rule,

$$\begin{aligned} & P(X_1 = heads, X_2 = tails, X_3 = tails) \\ &= P(X_3 = tails|X_2 = tails, X_1 = heads)P(X_2 = tails|X_1 = heads) \\ &\quad P(X_1 = heads) \\ &= \frac{b+1}{a+b+2} \times \frac{b}{a+b+1} \times \frac{a}{a+b} \\ &= \frac{1+1}{1+1+2} \times \frac{1}{1+1+1} \times \frac{1}{1+1} = .0833. \end{aligned}$$

We get the same probability regardless of the order of the outcomes as long as the number of heads and tails is the same. For example,

$$\begin{aligned} & P(X_1 = tails, X_2 = tails, X_3 = heads) \\ &= P(X_3 = heads|X_2 = tails, X_1 = tails)P(X_2 = tails|X_1 = tails) \\ &\quad P(X_1 = tails) \\ &= \frac{a}{a+b+2} \times \frac{b+1}{a+b+1} \times \frac{b}{a+b} \\ &= \frac{1}{1+1+2} \times \frac{2}{1+1+1} \times \frac{1}{1+1} = .0833. \end{aligned}$$

We now have the following theorem.

Theorem 8.1 *Suppose that we are about to repeatedly toss a thumbtack (or perform any repeatable experiment with two outcomes). Suppose further that we assume exchangeability, and we represent our prior belief concerning the probability of heads using a Dirichlet distribution with parameters a and b, where a and b are positive integers and $m = a + b$. Let* D *be data that consist of s heads and t tails in n trials. Then*

$$P(\mathsf{D}) = \frac{(m-1)!}{(m+n-1)!} \times \frac{(a+s-1)!(b+t-1)!}{(a-1)!(b-1)!}.$$

Proof. *Since the probability is the same regardless of the order in which the heads and tails occur, we can assume all the heads occur first. We therefore have (as before, the notation s, t on the right side of the conditioning bar means that we have seen s heads and t tails) the following:*

$$
\begin{aligned}
&P(\mathsf{D}) \\
&= P(X_{s+t} = tails|s, t-1) \cdots P(X_{s+2} = tails|s, 1) P(X_{s+1} = tails|s, 0) \\
&\quad P(X_s = heads|s-1, 0) \cdots P(X_2 = heads|1, 0) P(X_1 = heads) \\
&= \frac{b+t-1}{a+b+s+t-1} \times \cdots \frac{b+1}{a+b+s+1} \times \frac{b}{a+b+s} \times \\
&\quad \frac{a+s-1}{a+b+s-1} \times \cdots \frac{a+1}{a+b+1} \times \frac{a}{a+b} \\
&= \frac{(a+b-1)!}{(a+b+s+t-1)!} \times \frac{(a+s-1)!}{(a-1)!} \times \frac{(b+t-1)!}{(b-1)!} \\
&= \frac{(m-1)!}{(m+n-1)!} \times \frac{(a+s-1)!(b+t-1)!}{(a-1)!(b-1)!}.
\end{aligned}
$$

The first equality is due to Theorem 7.1. This completes the proof. ■

Example 8.2 *Suppose that, before tossing a thumbtack, we assign $a = 3$ and $b = 5$ to model the slight belief that tails is more probable than heads. We then toss the coin ten times and obtain four heads and six tails. Owing to the preceding theorem, the probability of obtaining these data D is given by*

$$
\begin{aligned}
P(\mathsf{D}) &= \frac{(m-1)!}{(m+n-1)!} \times \frac{(a+s-1)!(b+t-1)!}{(a-1)!(b-1)!} \\
&= \frac{(8-1)!}{(8+10-1)!} \times \frac{(3+4-1)!(5+6-1)!}{(3-1)!(5-1)!} = .00077.
\end{aligned}
$$

Note that the probability of these data is very small. This is because there are many possible outcomes (namely 2^{10}) of tossing a thumbtack ten times.

Theorem 8.1 holds, even if a and b are not integers. We merely state the result here.

Theorem 8.2 *Suppose we are about to repeatedly toss a thumbtack (or perform any repeatable experiment with two outcomes), we assume exchangeability, and we represent our prior belief concerning the probability of heads using a Dirichlet distribution with parameters a and b, where a and b are positive real numbers, and $m = a + b$. Let D be data that consist of s heads and t tails in n trials. Then*

$$
P(\mathsf{D}) = \frac{\Gamma(m)}{\Gamma(m+n)} \times \frac{\Gamma(a+s)\Gamma(b+t)}{\Gamma(a)\Gamma(b)}. \tag{8.1}
$$

Proof. *The proof can be found in [Neapolitan, 2004].* ■

In the preceding theorem Γ denotes the gamma function. When n is an integer ≥ 1, we have that

$$\Gamma(n) = (n-1)! \ .$$

So, the preceding theorem generalizes Theorem 8.1.

Example 8.3 *Recall that in Example 7.7 we set $a = 1/360$ and $b = 19/360$. Then after seeing three windows containing NiS, our updated values of a and b became as follows:*

$$\begin{aligned} a &= \frac{1}{360} + 3 = \frac{1081}{360} \\ b &= \frac{19}{360} + 0 = \frac{19}{360}. \end{aligned}$$

We then wanted the probability that the next 36 windows would contain NiS. This is the probability of obtaining data with $s = 36$ and $t = 0$. Owing to the previous theorem, the probability of these data D is given by

$$\begin{aligned} P(\mathsf{D}) &= \frac{\Gamma(m)}{\Gamma(m+n)} \times \frac{\Gamma(a+s)\Gamma(b+t)}{\Gamma(a)\Gamma(b)} \\ &= \frac{\Gamma(\frac{1100}{360})}{\Gamma(\frac{1100}{360}+36)} \times \frac{\Gamma(\frac{1081}{360}+36)\Gamma(\frac{19}{360}+0)}{\Gamma(\frac{1081}{360})\Gamma(\frac{19}{360})} \\ &= .866. \end{aligned}$$

Recall that we obtained this same result by direct computation at the end of Example 7.7.

We developed the method for computing the probability of data for the case of binomial variables. It extends readily to multinomial variables. See [Neapolitan, 2004] for that extension.

Learning DAG Models Using the Bayesian Score

Next we show how we score a DAG model by computing the probability of the data given the model and how we use that score to learn a DAG model.

Suppose we have a Bayesian network for learning, as discussed in Section 7.2. For example, we might have the network in Figure 8.1 (a). Here we call such a network a **DAG model**. We can score a DAG model $\mathbb{G}$ based on data D by determining how probable the data are given the DAG model. That is, we compute $P(\mathsf{D}|\mathbb{G})$. The formula for this probability is the same as that developed in Theorem 8.2, except there is a term of the form in Equality 12.2 for each probability in the network. So, the probability of data D given the DAG model

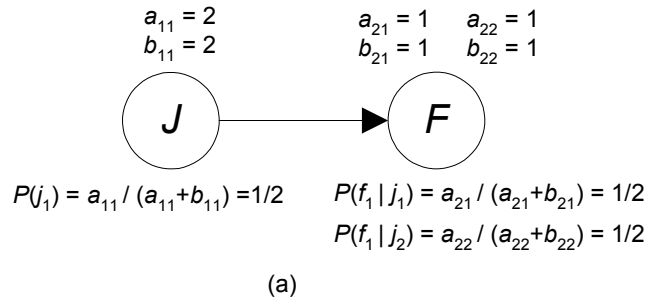


Figure 8.1: The network in (a) models that J has a causal effect on F, whereas the network in (b) models that neither variable causes the other.

$\mathbb{G}_1$ in Figure 8.1 (a) is given by

$$\begin{aligned} P(\mathsf{D}|\mathbb{G}_1) &= \frac{\Gamma(m_{11})}{\Gamma(m_{11}+n_{11})} \times \frac{\Gamma(a_{11}+s_{11})\Gamma(b_{11}+t_{11})}{\Gamma(a_{11})\Gamma(b_{11})} \times \\ &\quad \frac{\Gamma(m_{21})}{\Gamma(m_{21}+n_{21})} \times \frac{\Gamma(a_{21}+s_{21})\Gamma(b_{21}+t_{21})}{\Gamma(a_{21})\Gamma(b_{21})} \times \\ &\quad \frac{\Gamma(m_{22})}{\Gamma(m_{22}+n_{22})} \times \frac{\Gamma(a_{22}+s_{22})\Gamma(b_{22}+t_{22})}{\Gamma(a_{22})\Gamma(b_{22})}. \end{aligned} \tag{8.2}$$

The data used in each term include only the data relevant to the conditional probability the term represents. This is exactly the same scheme that was used to learn parameters in Section 7.2. For example, the value of s_{21} is the number of cases that have J equal to j_2 and F equal to f_1.

Similarly, the probability of data D given the DAG model $\mathbb{G}_2$ in Figure 8.1 (b) is given by

$$\begin{aligned} P(\mathsf{D}|\mathbb{G}_2) &= \frac{\Gamma(m_{11})}{\Gamma(m_{11}+n_{11})} \times \frac{\Gamma(a_{11}+s_{11})\Gamma(b_{11}+t_{11})}{\Gamma(a_{11})\Gamma(b_{11})} \times \\ &\quad \frac{\Gamma(m_{21})}{\Gamma(m_{21}+n_{21})} \times \frac{\Gamma(a_{21}+s_{21})\Gamma(b_{21}+t_{21})}{\Gamma(a_{21})\Gamma(b_{21})}. \end{aligned} \tag{8.3}$$

Note that the values of a_{11}, s_{11}, and so on in Equality 8.3 are the ones relevant to $\mathbb{G}_2$ and are not the same values as those in Equality 8.2. We have not explicitly shown their dependence on the DAG model, for the sake of notational simplicity.

Example 8.4 *Suppose we want to determine whether job status (J) has a causal effect on whether someone defaults on a loan (F). Furthermore, we articulate just two values for each variable as follows:*

Variable	Value	When the Variable Takes This Value
J	j_1	Individual is a white collar worker
	j_2	Individual is a blue collar worker
F	f_1	Individual has defaulted on a loan at least once
	f_2	Individual has never defaulted on a loan

We represent the assumption that J has a causal effect on F with the DAG model $\mathbb{G}_1$ in Figure 8.1 (a) and the assumption that neither variable has a causal effect on the other with the DAG model $\mathbb{G}_2$ in Figure 8.1 (b). We assume that F does not have a causal effect on J, so we do not model this situation. Note that in both models we used a prior equivalent sample size of four and we represented the prior belief that all probabilities are .5. In general, we can use whatever prior sample size and prior belief that best model our prior knowledge. The only requirement is that both DAG models have the same prior equivalent sample size.

Suppose that next we obtain the data D *in the following table:*

Case	J	F
1	j_1	f_1
2	j_1	f_1
3	j_1	f_1
4	j_1	f_1
5	j_1	f_2
6	j_2	f_1
7	j_2	f_2
8	j_2	f_2

Then, owing to Equality 8.2,

$$
\begin{aligned}
P(\mathsf{D}|\mathbb{G}_1) &= \frac{\Gamma(m_{11})}{\Gamma(m_{11}+n_{11})} \times \frac{\Gamma(a_{11}+s_{11})\Gamma(b_{11}+t_{11})}{\Gamma(a_{11})\Gamma(b_{11})} \times \\
&\quad \frac{\Gamma(m_{21})}{\Gamma(m_{21}+n_{21})} \times \frac{\Gamma(a_{21}+s_{21})\Gamma(b_{21}+t_{21})}{\Gamma(a_{21})\Gamma(b_{21})} \times \\
&\quad \frac{\Gamma(m_{22})}{\Gamma(m_{22}+n_{22})} \times \frac{\Gamma(a_{22}+s_{22})\Gamma(b_{22}+t_{22})}{\Gamma(a_{22})\Gamma(b_{22})} \\
&= \frac{\Gamma(4)}{\Gamma(4+8)} \times \frac{\Gamma(2+5)\,\Gamma(2+3)}{\Gamma(2)\Gamma(2)} \times \\
&\quad \frac{\Gamma(2)}{\Gamma(2+5)} \times \frac{\Gamma(1+4)\,\Gamma(1+1)}{\Gamma(1)\Gamma(1)} \times \\
&\quad \frac{\Gamma(2)}{\Gamma(2+3)} \times \frac{\Gamma(1+1)\,\Gamma(1+2)}{\Gamma(1)\Gamma(1)} \\
&= 7.2150 \times 10^{-6}.
\end{aligned}
$$

Furthermore,

$$\begin{aligned} P(\mathsf{D}|\mathbb{G}_2) &= \frac{\Gamma(m_{11})}{\Gamma(m_{11}+n_{11})} \times \frac{\Gamma(a_{11}+s_{11})\Gamma(b_{11}+t_{11})}{\Gamma(a_{11})\Gamma(b_{11})} \times \\ &\quad \frac{\Gamma(m_{21})}{\Gamma(m_{21}+n_{21})} \times \frac{\Gamma(a_{21}+s_{21})\Gamma(b_{21}+t_{21})}{\Gamma(a_{21})\Gamma(b_{21})} \times \\ &= \frac{\Gamma(4)}{\Gamma(4+8)} \times \frac{\Gamma(2+5)\,\Gamma(2+3)}{\Gamma(2)\Gamma(2)} \times \\ &\quad \frac{\Gamma(4)}{\Gamma(4+8)} \times \frac{\Gamma(2+5)\,\Gamma(2+3)}{\Gamma(2)\Gamma(2)} \\ &= 6.7465 \times 10^{-6}. \end{aligned}$$

If our prior belief is that neither model is more probable than the other, we assign

$$P(\mathbb{G}_1) = P(\mathbb{G}_2) = .5.$$

Then, owing to Bayes' Theorem,

$$\begin{aligned} P(\mathbb{G}_1|\mathsf{D}) &= \frac{P(\mathsf{D}|\mathbb{G}_1)P(\mathbb{G}_1)}{P(\mathsf{D}|\mathbb{G}_1)P(\mathbb{G}_1) + P(\mathsf{D}|\mathbb{G}_2)P(\mathbb{G}_2)} \\ &= \frac{7.2150 \times 10^{-6} \times .5}{7.2150 \times 10^{-6} \times .5 + 6.7465 \times 10^{-6} \times .5} \\ &= .517 \end{aligned}$$

and

$$\begin{aligned} P(\mathbb{G}_2|\mathsf{D}) &= \frac{P(\mathsf{D}|\mathbb{G}_2)P(\mathbb{G}_2)}{P(\mathsf{D})} \\ &= \frac{6.7465 \times 10^{-6}(.5)}{7.2150 \times 10^{-6} \times .5 + 6.7465 \times 10^{-6} \times .5} \\ &= .483. \end{aligned}$$

We select (learn) DAG $\mathbb{G}_1$ and conclude that it is more probable that job status does have a causal effect on whether someone defaults on a loan.

Example 8.5 *Suppose we are doing the same study as in Example 8.4, and we obtain the data in the following table:*

Case	J	F
1	j_1	f_1
2	j_1	f_1
3	j_1	f_1
4	j_1	f_1
5	j_2	f_2
6	j_2	f_2
7	j_2	f_2
8	j_2	f_2

Then it is left as an exercise to show

$$P(\mathsf{D}|\mathbb{G}_1) = 8.6580 \times 10^{-5}$$

$$P(\mathsf{D}|\mathbb{G}_2) = 4.6851 \times 10^{-6},$$

and if we assign the same prior probability to both DAG models,

$$\begin{aligned} P(\mathbb{G}_1|\mathsf{D}) &= .949 \\ P(\mathbb{G}_2|\mathsf{D}) &= .051. \end{aligned}$$

We select (learn) DAG $\mathbb{G}_1$. Notice that the causal model is substantially more probable. This makes sense because even though there are not many data, it exhibits complete dependence.

Example 8.6 *Suppose we are doing the same study as in Example 8.4, and we obtain the data D in the following table:*

Case	J	F
1	j_1	f_1
2	j_1	f_1
3	j_1	f_2
4	j_1	f_2
5	j_2	f_1
6	j_2	f_1
7	j_2	f_2
8	j_2	f_2

Then it is left as an exercise to show

$$P(\mathsf{D}|\mathbb{G}_1) = 2.4050 \times 10^{-6}$$

$$P(\mathsf{D}|\mathbb{G}_2) = 4.6851 \times 10^{-6},$$

and if we assign the same prior probability to both DAG models,

$$\begin{aligned} P(\mathbb{G}_1|\mathsf{D}) &= .339 \\ P(\mathbb{G}_2|\mathsf{D}) &= .661. \end{aligned}$$

We select (learn) DAG $\mathbb{G}_2$. Notice that it is somewhat more probable that the two variables are independent. This makes sense since the data exhibit complete independence.

Learning DAG Patterns

The DAG $F \rightarrow J$ is Markov equivalent to the DAG in Figure 8.1 (a). Intuitively, we would expect it to have the same score. As long as we use a prior equivalent sample size (see Section 8.2.1), they will have the same score. This is discussed in [Neapolitan, 2004]. In general, we cannot distinguish Markov-equivalent DAGs based on data. So, we are actually learning Markov equivalence classes (DAG patterns) when we learn a DAG model from data.

Scoring Larger DAG Models

We illustrated scoring using only two variables. The general formula for the score when there are n variables and the variables are binomial is as follows:

Theorem 8.3 *Suppose we have a DAG* $\mathbb{G} = (\mathsf{V}, \mathsf{E})$ *where* V *is a set of binomial random variables, we assume exchangeability, and we use a Dirichlet distribution to represent our prior belief for each conditional probability distribution of every variable in* V*. Suppose further we have data* D *consisting of a set of data items such that each data item is a vector of values of all the variables in* V*. Then*

$$P(\mathsf{D}|\mathbb{G}) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(N_{ij})}{\Gamma(N_{ij} + M_{ij})} \frac{\Gamma(a_{ij} + s_{ij})\Gamma(b_{ij} + t_{ij})}{\Gamma(a_{ij})\Gamma(b_{ij})} \tag{8.4}$$

where

1. n *is the number of variables.*
2. q_i *is the number of different instantiations of the parents of* X_i*.*
3. a_{ij} *is our ascertained prior belief concerning the number of times* X_i *took its first value when the parents of* X_i *had their* j*th instantiation.*
4. b_{ij} *is our ascertained value prior belief concerning the number of times* X_i *took its second value when the parents of* X_i *had their* j*th instantiation.*
5. s_{ij} *is the number of times in the data that* X_i *took its first value when the parents of* X_i *had their* j*th instantiation.*
6. t_{ij} *is the number of times in the data that* X_i *took its second value when the parents of* X_i *had their* j*th instantiation.*
7. N_{ij} *and* M_{ij} *are as follows:*

$$N_{ij} = a_{ij} + b_{ij}$$

$$M_{ij} = s_{ij} + t_{ij}.$$

Proof. *The proof can be found in [Neapolitan, 2004].* ■

Note that, other than n, all the variables defined in the previous theorem depend on $\mathbb{G}$, but we do not show that dependency for the sake of simplicity.

The corresponding theorem when there are n variables and the variables are multinomial is as follows:

Theorem 8.4 *Suppose we have a DAG* $\mathbb{G} = (\mathsf{V}, \mathsf{E})$ *where* V *is a set of multinomial random variables, we assume exchangeability, and we use a Dirichlet distribution to represent our prior belief for each conditional probability distribution of every variable in* V*. Suppose further we have data* D *consisting of a set of*

data items such that each data item is a vector of values of all the variables in V*. Then*

$$P(\mathsf{D}|\mathbb{G}) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(N_{ij})}{\Gamma(N_{ij} + M_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(a_{ijk} + s_{ijk})}{\Gamma(a_{ijk})} \tag{8.5}$$

where

1. n *is the number of variables.*
2. q_i *is the number of different instantiations of the parents of* X_i*.*
3. a_{ijk} *is our ascertained prior belief concerning the number of times* X_i *took its* k*th value when the parents of* X_i *had their* j*th instantiation.*
4. s_{ijk} *is the number of times in the data that* X_i *took its* k*th value when the parents of* X_i *had their* j*th instantiation.*
5. N_{ij} *and* M_{ij} *are as follows:*

$$N_{ij} = \sum_k a_{ijk}$$

$$M_{ij} = \sum_k s_{ijk}.$$

Proof. *The proof can be found in [Neapolitan, 2004].* ■

Note that Equality 8.4 is a special case of Equality 8.5. We call the $P(\mathsf{D}|\mathbb{G})$, obtained using the assumptions in the previous theorem, the **Bayesian score assuming Dirichlet priors**, but ordinarily we only say **Bayesian score**. We denote it as follows:

$$score_{Bayesian}(\mathbb{G} : \mathsf{D}) = P(\mathsf{D}|\mathbb{G}).$$

Neapolitan [2004] develops a Bayesian score for scoring Gaussian Bayesian networks. That is, each variable is assumed to be a function of its parents as shown in Equality 5.1 in Section 5.3.2.

8.2.2 The BIC Score

The **Bayesian information criterion (BIC) score** is as follows:

$$BIC\,(\mathbb{G} : \mathsf{D}) = \ln\left(P(\mathsf{D}|\hat{\mathsf{P}}, \mathbb{G})\right) - \frac{d}{2}\ln m,$$

where m is the number of data items, d is the dimension of the DAG model, and $\hat{\mathsf{P}}$ is the set of maximum likelihood values of the parameters. The dimension is the number of parameters in the model.

The BIC score is intuitively appealing because it contains (1) a term that shows how well the model predicts the data when the parameter set is equal to its ML value, and (2) a term that punishes for model complexity. Another nice feature of the BIC is that it does not depend on the prior distribution of the parameters, which means there is no need to assess one.

Example 8.7 *Suppose we have the DAG models in Figure 8.1 and the data in Example 8.4. That is, we have the data* D *in the following table:*

Case	J	F
1	j_1	f_1
2	j_1	f_1
3	j_1	f_1
4	j_1	f_1
5	j_1	f_2
6	j_2	f_1
7	j_2	f_2
8	j_2	f_2

For the DAG model in Figure 8.1 (a) we have that

$$\begin{aligned}\hat{P}(j_1) &= \frac{5}{8}\\ \hat{P}(f_1|j_1) &= \frac{4}{5}\\ \hat{P}(f_1|j_2) &= \frac{1}{3}.\end{aligned}$$

Since there are three parameters in the model, $d = 3$*. We then have that*

$$\begin{aligned}&P(\mathsf{D}|\hat{\mathsf{P}}, \mathbb{G}_1)\\ &= \left[\hat{P}(f_1|j_1)\hat{P}(j_1)\right]^4 \left[\hat{P}(f_2|j_1)\hat{P}(j_1)\right] \left[\hat{P}(f_1|j_2)\hat{P}(j_2)\right] \left[\hat{P}(f_2|j_2)\hat{P}(j_2)\right]^2\\ &= \left(\frac{4}{5}\frac{5}{8}\right)^4 \left(\frac{1}{5}\frac{5}{8}\right) \left(\frac{1}{3}\frac{3}{8}\right) \left(\frac{2}{3}\frac{3}{8}\right)^2\\ &= 6.103\,5 \times 10^{-5},\end{aligned}$$

and therefore

$$\begin{aligned}BIC\,(\mathbb{G}_1 : \mathsf{D}) &= \ln\left(P(\mathsf{D}|\hat{\mathsf{P}}, \mathbb{G}_1)\right) - \frac{d}{2}\ln m\\ &= \ln\left(6.103\,5 \times 10^{-5}\right) - \frac{3}{2}\ln 8\\ &= -12.823.\end{aligned}$$

For the DAG model in Figure 8.1 (b) we have that

$$\begin{aligned}\hat{P}(j_1) &= \frac{5}{8}\\ \hat{P}(f_1) &= \frac{5}{8}.\end{aligned}$$

Since there are three parameters in the model, $d = 2$*. We then have that*

$$
\begin{aligned}
& P(\mathsf{D}|\hat{\mathsf{P}}, \mathbb{G}_2) \\
&= \left[\hat{P}(f_1)\hat{P}(j_1)\right]^4 \left[\hat{P}(f_2|)\hat{P}(j_1)\right] \left[\hat{P}(f_1)\hat{P}(j_2)\right] \left[\hat{P}(f_2)\hat{P}(j_2)\right]^2 \\
&= \left(\frac{5}{8}\frac{5}{8}\right)^4 \left(\frac{3}{8}\frac{5}{8}\right) \left(\frac{5}{8}\frac{3}{8}\right) \left(\frac{3}{8}\frac{3}{8}\right)^2 \\
&= 2.529\,2 \times 10^{-5},
\end{aligned}
$$

and therefore

$$
\begin{aligned}
BIC\,(\mathbb{G}_2 : \mathsf{D}) &= \ln\left(P(\mathsf{D}|\hat{\mathsf{P}}, \mathbb{G}_2)\right) - \frac{d}{2}\ln m \\
&= \ln\left(2.529\,2 \times 10^{-5}\right) - \frac{2}{2}\ln 8 \\
&= -12.644.
\end{aligned}
$$

Note that although the data were more probable given $\mathbb{G}_1$, G_2 won because it is less complex.

Looking at Examples 8.4 and 8.7, we see that the Bayesian score and the BIC can choose different DAG models. The reason is that the data set is small. In the limit they will both choose the same DAG model because the BIC is asymptotically correct. A scoring criterion for DAG models is called **asymptotically correct** if for a sufficiently large data set it chooses the DAG that maximizes $P(\mathsf{D}|\mathbb{G})$.

8.2.3 Consistent Scoring Criteria

We've presented two methods for scoring DAG models. There are others, several of which are discussed in [Neapolitan, 2004]. The question remains as to the quality of these scores. The probability distribution generating the data is called the **generative distribution**. Our goal with a Bayesian network is to represent the generative distribution succinctly. A consistent scoring criterion will almost certainly do this if the data set is large. Specifically, we say a DAG **includes** a probability distribution P if every conditional independency entailed by the DAG is in P. A **consistent scoring criterion** for DAG models has the following two properties:

1. As the size of the data set approaches infinity, the probability approaches one that a DAG that includes P will score higher than a DAG that does not include P.

2. As the size of the data set approaches infinity, the probability approaches one that a smaller DAG that includes P will score higher than a larger DAG that includes P.

Both the Bayesian score and the BIC are consistent scoring criteria.

8.2.4 How Many DAGs Must We Score?

When there are not many variables, we can exhaustively score all possible DAGs. We then select the DAG(s) with the highest score. However, when the number of variables is not small, it is computationally unfeasible to find the maximizing DAGs by exhaustively considering all DAG patterns. That is, Robinson [1977] showed that the number of DAGs containing n nodes is given by the following recurrence:

$$\begin{aligned} f(n) &= \sum_{i=1}^{n} (-1)^{i+1} \binom{n}{i} 2^{i(n-i)} f(n-i) \qquad n > 2 \\ f(0) &= 1 \\ f(1) &= 1. \end{aligned}$$

It is left as an exercise to show $f(2) = 3$, $f(3) = 25$, $f(5) = 29{,}000$, and $f(10) = 4.2 \times 10^{18}$. Furthermore, Chickering [1996] proved that for certain classes of prior distributions, the problem of finding a highest-scoring DAG is NP-hard. So, researchers developed heuristic DAG search algorithms. We discuss such algorithms in Section 8.6.1.

8.2.5 Using the Learned Network to Do Inference

Once we learn a DAG from data, we can then learn the parameters. The result will be a Bayesian network that we can use to do inference for the next case. The next example illustrates the technique.

Example 8.8 *Suppose we have the situation and data in Example 8.4. That is, we have the data* D *in the following table:*

Case	J	F
1	j_1	f_1
2	j_1	f_1
3	j_1	f_1
4	j_1	f_1
5	j_1	f_2
6	j_2	f_1
7	j_2	f_2
8	j_2	f_2

Then, as shown in Example 8.4, we would learn the DAG in Figure 8.1 (a). Next we can update the conditional probabilities in the Bayesian network for learning in Figure 8.1 (a) using the preceding data and the parameter learning technique discussed in Section 7.2. The result is the Bayesian network in Figure 8.2.

Suppose now that we find out that Sam has $F = f_2$. That is, Sam has never defaulted on a loan. We can use the Bayesian network to compute the

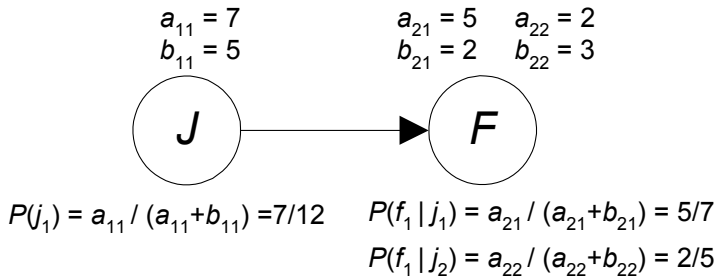


Figure 8.2: An updated Bayesian network for learning based on the data in Example 8.8.

probability that Sam is a white-collar worker. For this simple network we can just use Bayes' Theorem as follows:

$$\begin{aligned} P(j_1|f_1) &= \frac{P(f_1|j_1)P(j_1)}{P(f_1|j_1)P(j_1) + Pf_1|j_2)P(j_2)} \\ &= \frac{(5/7)\,(7/12)}{(5/7)\,(7/12) + (2/5)\,(5/12)} = .714. \end{aligned}$$

The probabilities in the previous calculation are all conditional on the data D *and the DAG that we select. However, once we select a DAG and learn the parameters, we do not bother to show this dependence.*

8.3 Constraint-Based Structure Learning

Next we discuss a quite different structure learning technique called **constraint-based learning**. In this approach, we try to learn a DAG from the conditional independencies in the generative probability distribution P. First, we illustrate the constraint-based approach by showing how to learn a DAG faithful to a probability distribution. This is followed by a discussion of embedded faithfulness. Finally, we present causal learning.

8.3.1 Learning a DAG Faithful to P

Recall that $(\mathbb{G}, P)$ satisfies the faithfulness condition if all and only the conditional independencies in P are entailed by $\mathbb{G}$. After discussing why we would want to learn a DAG faithful to a probability distribution P, we illustrate learning such a DAG.

Why We Want a Faithful DAG

Consider again the objects in Figure 2.1. In Example 2.23, we let P assign 1/13 to each object in the figure, and we defined these random variables on the set containing the objects:

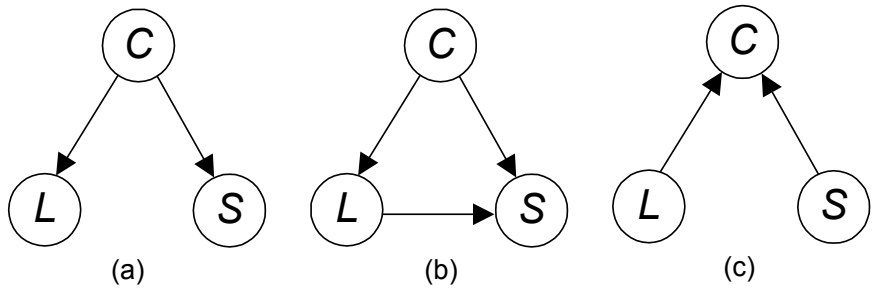


Figure 8.3: If the only conditional independency in P is $I_P(L, S|C)$, then P satisfies the Markov condition with the DAGs in (a) and (b), and P satisfies the faithfulness condition only with the DAG in (a).

Variable	Value	Outcomes Mapped to This Value
L	l_1	All objects containing an A
	l_2	All objects containing a B
S	s_1	All square objects
	s_2	All diamond-shaped objects
C	c_1	All black objects
	c_2	All white objects

We then showed that L and S are conditionally independent given C. That is, using the notation established in Section 2.2.2, we showed that

$$I_P(L, S|C).$$

In Example 5.1, we showed that this implies that P satisfies the Markov condition with the DAG in Figure 8.3 (a). However, P also satisfies the Markov condition with the complete DAG in Figure 8.3 (b). P does not satisfy the Markov condition with the DAG in Figure 8.3 (c) because that DAG entails $I_P(L, S)$ and this independency is not in P. The DAG in Figure 8.3 (b) does not represent P very well because it does not entail a conditional independency that is in P, namely $I_P(L, S|C)$. This is a violation of the faithfulness condition. Of the DAGs in Figure 8.3, only the one in Figure 8.3 (a) is faithful to P.

If we can find a DAG that is faithful to a probability distribution P, we have achieved our goal of representing P succinctly. That is, if there are DAGs faithful to P, then those DAGs are the smallest DAGs that include P (see [Neapolitan, 2004]). We say *DAGs* because if a DAG is faithful to P, then clearly any Markov-equivalent DAG is also faithful to P. For example, the DAGs $L \rightarrow C \rightarrow S$ and $S \leftarrow C \leftarrow L$, which are Markov equivalent to the DAG $L \leftarrow C \rightarrow S$, are also faithful to the probability distribution P concerning the objects in Figure 2.1. As we shall see, not every probability distribution has a DAG that is faithful to it. However, if there are DAGs faithful to a probability distribution, it is relatively easy to discover them. We discuss learning a faithful DAG next.

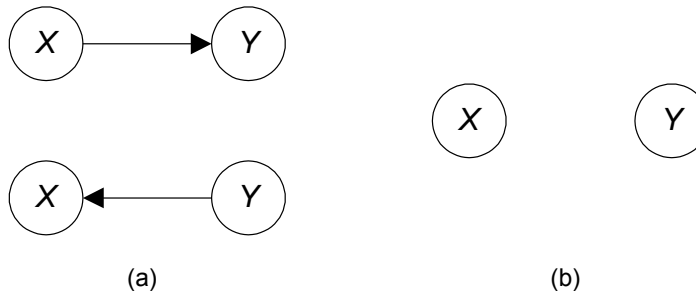


Figure 8.4: If the set of conditional independencies is $\{I_P(X, Y)\}$, we must have the DAG in (b), whereas if it is $\varnothing$, we must have one of the DAGs in (a).

Learning a Faithful DAG

We assume that we have a sample of entities from the population over which the random variables are defined, and we know the values of the variables of interest for the entities in the sample. The sample could be a random sample, or it could be obtained from passive data. From this sample, we have deduced the conditional independencies among the variables. A method for deducing conditional independencies and obtaining a measure of our confidence in them is described in [Spirtes et al., 1993; 2000] and [Neapolitan, 2004]. Our confidence in the DAG we learn is no greater than our confidence in these conditional independencies.

Example 8.9 *It is left as an exercise to show that the data shown in Example 8.1 exhibit this conditional independency:*

$$I_P(Height, Wage|Sex).$$

Therefore, from these data we can conclude, with a certain measure of confidence, that this conditional independency exists in the population at large.

Next we give a sequence of examples in which we learn a DAG that is faithful to the probability distribution of interest. These examples illustrate how a faithful DAG can be learned from the conditional independencies if one exists. We stress again that the DAG is faithful to the conditional independencies we have learned from the data. We are not certain that these are the conditional independencies in the probability distribution for the entire population.

Example 8.10 *Suppose* V *is our set of observed variables,* $\mathsf{V} = \{X, Y\}$*, and the set of conditional independencies in* P *is*

$$\{I_P(X, Y)\}.$$

We want to find a DAG faithful to P*. We cannot have either of the DAGs in Figure 8.4 (a). The reason is that these DAGs do not entail that* X *and* Y

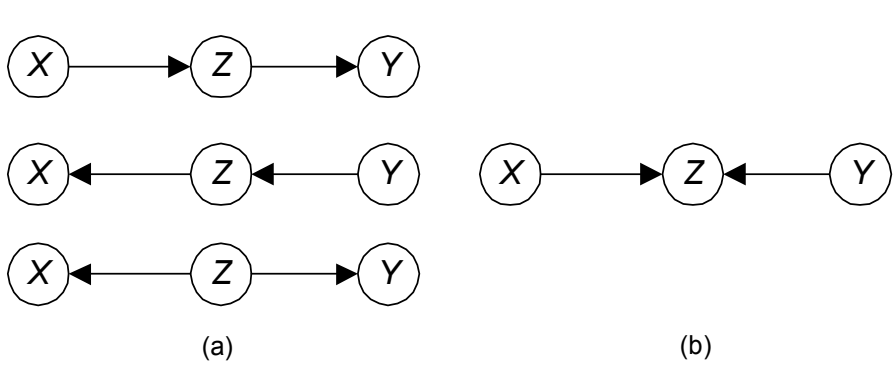


Figure 8.5: If the set of conditional independencies is $\{I_P(X, Y)\}$, we must have the DAG in (b); if it is $\{I_P(X, Y|Z)\}$, we must have one of the DAGs in (a).

are independent, which means the faithfulness condition is not satisfied. So, we must have the DAG in Figure 8.4 (b). We conclude that P is faithful to the DAG in Figure 8.4 (b).

Example 8.11 *Suppose $\mathsf{V} = \{X, Y\}$ and the set of conditional independencies in P is the empty set*

$$\varnothing.$$

That is, there are no independencies. We want to find a DAG faithful to P. We cannot have the DAG in Figure 8.4 (b). The reason is that this DAG entails that X and Y are independent, which means that the Markov condition is not satisfied. So, we must have one of the DAGs in Figure 8.4 (a). We conclude that P is faithful to both the DAGs in Figure 8.4 (a). Note that these DAGs are Markov equivalent.

Example 8.12 *Suppose $\mathsf{V} = \{X, Y, Z\}$, and the set of conditional independencies in P is*

$$\{I_P(X, Y)\}.$$

We want to find a DAG faithful to P. There can be no edge between X and Y in the DAG owing to the reason given in Example 8.10. Furthermore, there must be edges between X and Z and between Y and Z owing to the reason given in Example 8.11. We cannot have any of the DAGs in Figure 8.5 (a). The reason is that these DAGs entail $I_P(X, Y|Z)$, and this conditional independency is not present. So, the Markov condition is not satisfied. Furthermore, the DAGs do not entail $I_P(X, Y)$. So, the DAG must be the one in Figure 8.5 (b). We conclude that P is faithful to the DAG in Figure 8.5 (b).

Example 8.13 *Suppose $\mathsf{V} = \{X, Y, Z\}$ and the set of conditional independencies in P is*

$$\{I_P(X, Y|Z)\}.$$

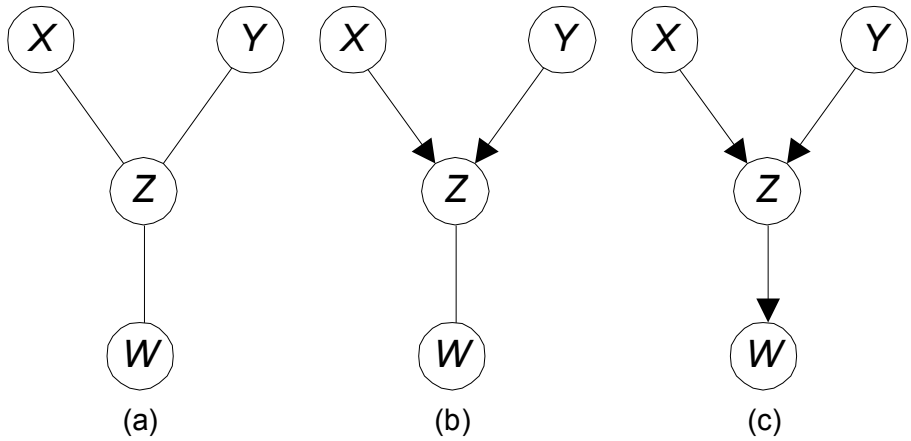


Figure 8.6: If the set of conditional independencies is $\{I_p(X, \{Y, W\}), I_P(Y, \{X, Z\})\}$, we must have the DAG in (c).

We want to find a DAG faithful to P. Owing to reasons similar to those given before, the only edges in the DAG must be between X and Z and between Y and Z. We cannot have the DAG in Figure 8.5 (b). The reason is that this DAG entails $I(X, Y)$, and this conditional independency is not present. So, the Markov condition is not satisfied. So, we must have one of the DAGs in Figure 8.5 (a). We conclude that P is faithful to all the DAGs in Figure 8.5 (a).

We now state a theorem whose proof can be found in [Neapolitan, 2004]. At this point your intuition should suspect that it is true.

Theorem 8.5 *If $(\mathbb{G}, P)$ satisfies the faithfulness condition, then there is an edge between X and Y if and only if X and Y are not conditionally independent given any set of variables.*

Example 8.14 *Suppose $\mathsf{V} = \{X, Y, Z, W\}$ and the set of conditional independencies in P is*

$$\{I_p(X, Y), \quad I_P(W, \{X, Y\}|Z)\}.$$

We want to find a DAG faithful to P. Owing to Theorem 8.5, the links (edges without regard for direction) must be as shown in Figure 8.6 (a). We must have the directed edges shown in Figure 8.6 (b) because we have $I_p(X, Y)$. Therefore, we must also have the directed edge shown in Figure 8.6 (c) because we do not have $I_P(W, X)$. We conclude that P is faithful to the DAG in Figure 8.6 (c).

Example 8.15 *Suppose $\mathsf{V} = \{X, Y, Z, W\}$ and the set of conditional independencies in P is*

$$\{I_P(X, \{Y, W\}), \quad I_P(Y, \{X, Z\})\}.$$

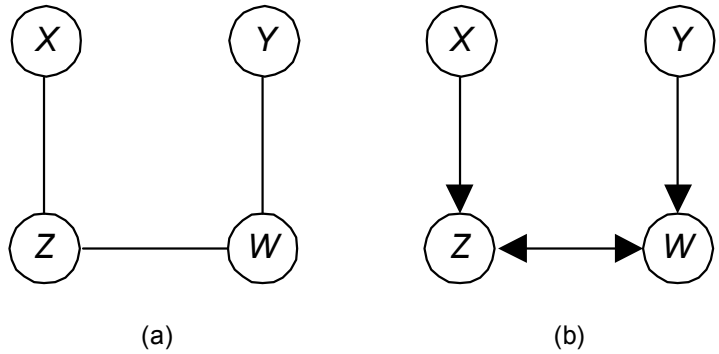


Figure 8.7: If the set of conditional independencies is $\{I_P(X, \{Y, W\}), I_P(Y, \{X, Z\})\}$ and we try to find a DAG faithful to P, we obtain the graph in (b), which is not a DAG.

We want to find a DAG faithful to P. Owing to Theorem 8.5, we must have the links shown in Figure 8.7 (a). Now, if we have the chain $X \rightarrow Z \rightarrow W$, $X \leftarrow Z \leftarrow W$, or $X \leftarrow Z \rightarrow W$, then we do not have $I_P(X, W)$. So, we must have the chain $X \rightarrow Z \leftarrow W$. Similarly, we must have the chain $Y \rightarrow W \leftarrow Z$. So, our graph must be the one in Figure 8.7 (b). However, this graph is not a DAG. The problem here is that there is no DAG faithful to P.

Example 8.16 *Again suppose $\mathsf{V} = \{X, Y, Z, W\}$ and the set of conditional independencies in P is*

$$\{I_P(X, \{Y, W\}), \quad I_P(Y, \{X, Z\})\}.$$

As shown in the previous example, there is no DAG faithful to P. However, this does not mean we cannot find a more succinct way to represent P than using a complete DAG. P satisfies the Markov condition with each of the DAGs in Figure 8.8. That is, the DAG in Figure 8.8 (a) entails

$$\{I_P(X, Y), \quad I_P(Y, Z)\}$$

and these conditional independencies are both in P, whereas the DAG in Figure 8.8 (b) entails

$$\{I_P(X, Y), \quad I_P(X, W)\}$$

and these conditional independencies are both in P. However, P does not satisfy the faithfulness condition with either of these DAGs because the DAG in Figure 8.8 (a) does not entail $I_P(X, W)$, whereas the DAG in Figure 8.8 (b) does not entail $I_P(Y, Z)$.

Each of these DAGs is as succinct as we can represent the probability distribution. So, when there is no DAG faithful to a probability distribution P, we can still represent P much more succinctly than we would by using the complete DAG. A structure learning algorithm tries to find the most succinct

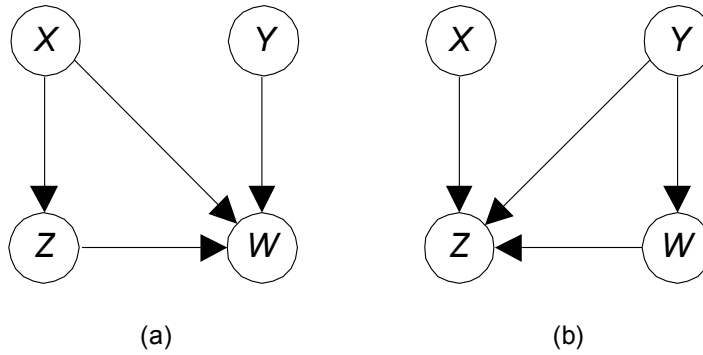


Figure 8.8: If the set of conditional independencies is $\{I_P(X, \{Y, W\}), I_P(Y, \{X, Z\})\}$, P satisfies the Markov condition with both these DAGs.

representation. Depending on the number of alternatives of each variable, one of the DAGs in Figure 8.8 may actually be a more succinct representation than the other because it contains fewer parameters. A constraint-based learning algorithm could not distinguish between the two, but a score-based one could. See [Neapolitan, 2004] for a complete discussion of this matter.

8.3.2 Learning a DAG in Which P Is Embedded Faithfully

In a sense we compromised in Example 8.16 because the DAG we learned did not entail all the conditional independencies in P. This is fine if our goal is to learn a Bayesian network that will later be used to do inference. However, another application of structure learning is *causal learning*, which is discussed in the next subsection. When we're learning causes it would be better to find a DAG in which P is embedded faithfully. We discuss embedded faithfulness next.

Definition 8.1 *Suppose we have a joint probability distribution P of the random variables in some set V and a DAG $\mathbb{G} = (\mathsf{W}, \mathsf{E})$ such that $\mathsf{V} \subseteq \mathsf{W}$. We say that $(\mathbb{G}, P)$ satisfy the* ***embedded faithfulness condition*** *if all and only the conditional independencies in P are entailed by $\mathbb{G}$, restricted to variables in V. Furthermore, we say that P is embedded faithfully in $\mathbb{G}$.*

Example 8.17 *Again suppose $\mathsf{V} = \{X, Y, Z, W\}$ and the set of conditional independencies in P is*

$$\{I_P(X, \{Y, W\}), \quad I_P(Y, \{X, Z\})\}.$$

Then P is embedded faithfully in the DAG in Figure 8.9. It is left as an exercise to show this. By including the variable H in the DAG, we are able to entail all and only the conditional independencies in P restricted to variables in V.

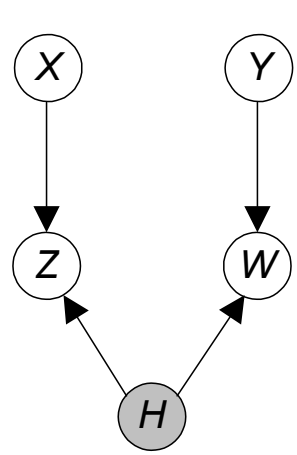


Figure 8.9: If the set of conditional independencies in P is $\{I_P(X, \{Y, W\}), I_P(Y, \{X, Z\})\}$, then P is embedded faithfully in this DAG.

Variables such as H are called **hidden variables** because they are not among the observed variables.

8.4 Causal Learning

In many, if not most, applications the variables of interest have causal relationships to each other. For example, the variables in Example 8.1 are causally related in that sex has a causal effect on height and may have a causal effect on wage. If the variables are causally related, we can learn something about their causal relationships when we learn the structure of the DAG from data. However, we must make certain assumptions to do this. We discuss these assumptions and causal learning next.

8.4.1 Causal Faithfulness Assumption

Recall from Section 5.3.2 that if we assume the observed probability distribution P of a set of random variables $\mathbf{V}$ satisfies the Markov condition with the causal DAG $\mathbb{G}$ containing the variables, we say we are making the **causal Markov assumption**, and we call $(\mathbb{G}, P)$ a **causal network**. Furthermore, we concluded that the causal Markov assumption is justified for a causal graph if the following conditions are satisfied:

1. There are no hidden common causes. That is, all common causes are represented in the graph.

2. There are no causal feedback loops. That is, our graph is a DAG.

3. Selection bias is not present.

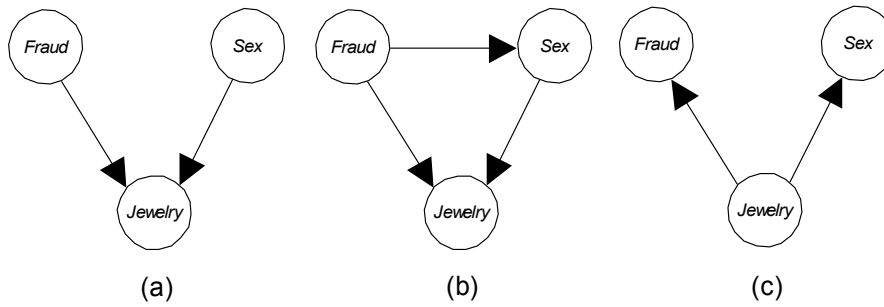


Figure 8.10: If the only causal relationships are that $Fraud$ and Sex have causal influences on $Jewelry$, then the causal DAG is the one in (a). If we make the causal Markov assumption, only the DAG in (c) is ruled out if we observe $I_P(Fraud, Sex)$.

Recall the discussion concerning credit card fraud in Section 5.1. Suppose that both fraud and sex do indeed have a causal effect on whether jewelry is purchased, and there are no other causal relationships among the variables. Then the causal DAG containing these variables is the one in Figure 8.10 (a). If we make the causal Markov assumption, we must have $I_P(Fraud, Sex)$.

Suppose now that we do not know the causal relationships among the variables, we make the causal Markov assumption, and we learn only the conditional independency $I_P(Fraud, Sex)$ from data. Can we conclude that the causal DAG must be the one in Figure 8.10 (a)? No, we cannot because P also satisfies the Markov condition with the DAG in Figure 8.10 (b). This concept is a bit tricky to understand. However, recall that we are assuming that we do not know the causal relationships among the variables. As far as we know, they could be the ones in Figure 8.10 (b). If the DAG in Figure 8.10 (b) were the causal DAG, the causal Markov assumption would still be satisfied when the only conditional independency is $I_P(Fraud, Sex)$, because that DAG satisfies the Markov condition with P. So, if we make only the causal Markov assumption, we cannot distinguish the causal DAGs in Figures 8.10 (a) and 8.10 (b) based on the conditional independency $I_P(Fraud, Sex)$. The causal Markov assumption only enables us to rule out causal DAGs that contain conditional independencies that are not in P.

One such DAG is the one in Figure 8.10 (c). We need to make the causal faithfulness assumption to conclude that the causal DAG is the one in Figure 8.10 (a). That assumption is as follows: If we assume that the observed probability distribution P of a set of random variables $\mathbf{V}$ satisfies the faithfulness condition with the causal $\mathbb{G}$ containing the variables, we say we are making the **causal faithfulness assumption**. If we make the causal faithfulness assumption, then if we find a unique DAG that is faithful to P, the edges in that DAG must represent causal influences. This is illustrated by the following examples.

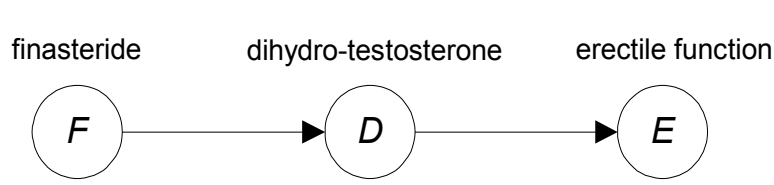


Figure 8.11: Finasteride and erectile function are independent.

Example 8.18 *Recall that in Example 8.12 we showed that if* $\mathsf{V} = \{X, Y, Z\}$ *and the set of conditional independencies in* P *is*

$$\{I_P(X, Y)\},$$

the only DAG faithful to P *is the one in Figure 8.5 (b). If we make the causal faithfulness assumption, this DAG must be the causal DAG, which means we can conclude that* X *and* Y *each cause* Z*. This is the exact same situation as that illustrated earlier concerning fraud, sex, and jewelry. Therefore, if we make the causal faithfulness assumption, we can conclude that the causal DAG is the one in Figure 8.10 (a) based on the conditional independency* $I_P(Fraud, Sex)$*.*

Example 8.19 *In Example 8.13 we showed that if* $\mathsf{V} = \{X, Y, Z\}$ *and the set of conditional independencies in* P *is*

$$\{I_P(X, Y|Z)\},$$

all the DAGs in Figure 8.5 (a) are faithful to P*. So, if we make the causal faithfulness assumption, we can conclude that one of these DAGs is the causal DAG, but we do not know which one.*

Example 8.20 *In Example 8.14 we showed that if* $\mathsf{V} = \{X, Y, Z, W\}$ *and the set of conditional independencies in* P *is*

$$\{I_p(X, Y), \quad I_P(W, \{X, Y\}|Z)\},$$

the only DAG faithful to P *is the one in Figure 8.6 (c). So, we can conclude that* X *and* Y *each cause* Z *and* Z *causes* W*.*

When is the causal faithfulness assumption justified? It requires the three conditions mentioned previously for the causal Markov assumption plus one more, which we discuss next. Recall from Section 6.2.1 that the causal relationships among finasteride (F), dihydro-testosterone (D), and erectile dysfunction (E) have clearly been found to be those depicted in Figure 8.11. However, as discussed in that section, we have $I_P(F, E|D)$. We would expect a causal mediary to transmit an effect from its antecedent to its consequence, but in this case it does not. As also discussed in Section 6.2.1, the explanation is that finasteride cannot lower dihydro-testosterone levels beyond a certain threshold level, and that level is all that is needed for erectile function. So, we have $I_P(F, E)$.

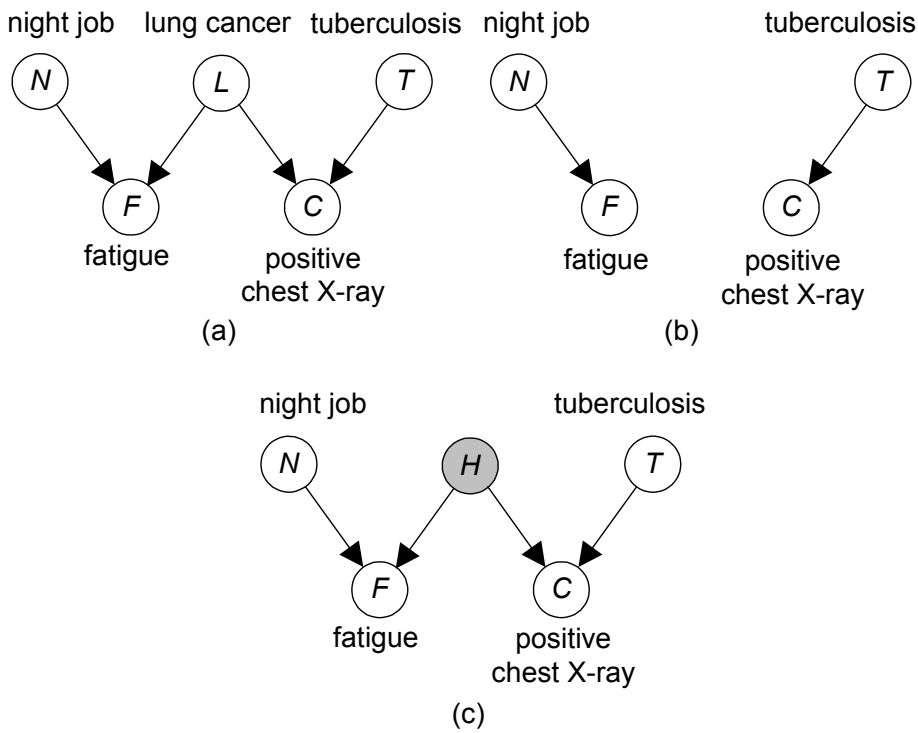


Figure 8.12: If the causal relationships are those shown in (a), P is not faithful to the DAG in (b), but P is embedded faithfully in the DAG in (c).

The Markov condition does not entail $I_P(F, E)$ for the causal DAG in Figure 8.11. It only entails $I_P(F, E|D)$. So, the causal faithfulness assumption is not justified. If we learned the conditional independencies in the probability distribution of these variables from data, we would learn the following set of independencies:

$$\{I_P(F, E), \quad I_P(F, E|D)\}.$$

There is no DAG that entails both these conditional independencies, so no DAG could be learned from such data.

The causal faithfulness assumption is usually justified when the three conditions listed previously for the causal Markov assumption are satisfied and when we do not have unusual causal relationships, as in the finasteride example. So, the causal faithfulness assumption is ordinarily justified for a causal graph if the following conditions are satisfied:

1. There are no hidden common causes. That is, all common causes are represented in the graph.

2. There are no causal feedback loops. That is, our graph is a DAG.

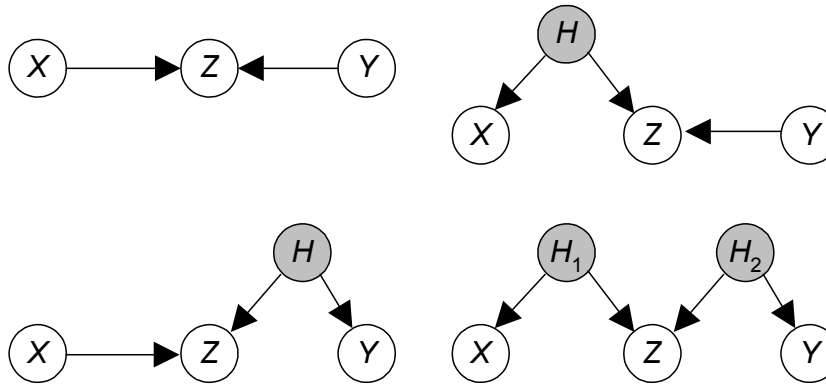


Figure 8.13: If we make the causal embedded faithfulness assumption and our set of conditional independencies is $\{I_P(X, Y)\}$, the causal relationships could be the ones in any of these DAGs.

3. Selection bias is not present.

4. All intermediate causes transmit influences from their antecedents to their consequences.

8.4.2 Causal Embedded Faithfulness Assumption

It seems that the main exception to the causal faithfulness assumption (and the causal Markov assumption) is the presence of hidden common causes. Even in the example concerning sex, height, and wage (Example 8.1), perhaps there is a genetic trait that makes people grow taller and also gives them some personality trait that helps them compete better in the job market. Our next assumption eliminates the requirement that there are no hidden common causes. If we assume that the observed probability distribution P of a set of random variables V is embedded faithfully in a causal DAG containing the variables, we say that we are making the **causal embedded faithfulness assumption**. The causal embedded faithfulness assumption is usually justified when the conditions for the causal faithfulness assumption are satisfied, except that hidden common causes may be present.

Next we illustrate the causal embedded faithfulness assumption. Suppose that the causal DAG in Figure 8.12 (a) satisfies the causal faithfulness assumption. However, we only observe $\mathsf{V} = \{N, F, C, T\}$. Then the causal DAG containing the observed variables is the one in Figure 8.12 (b). The DAG in Figure 8.12 (b) entails $I_P(F, C)$, and this conditional independency is not entailed by the DAG in Figure 8.12 (a). Therefore, the observed distribution $P(\mathsf{V})$ does not satisfy the Markov condition with the causal DAG in Figure 8.12 (b), which means the causal faithfulness assumption is not warranted. However, $P(\mathsf{V})$ is embedded faithfully in the DAG in Figure 8.12 (c). So, the causal embedded

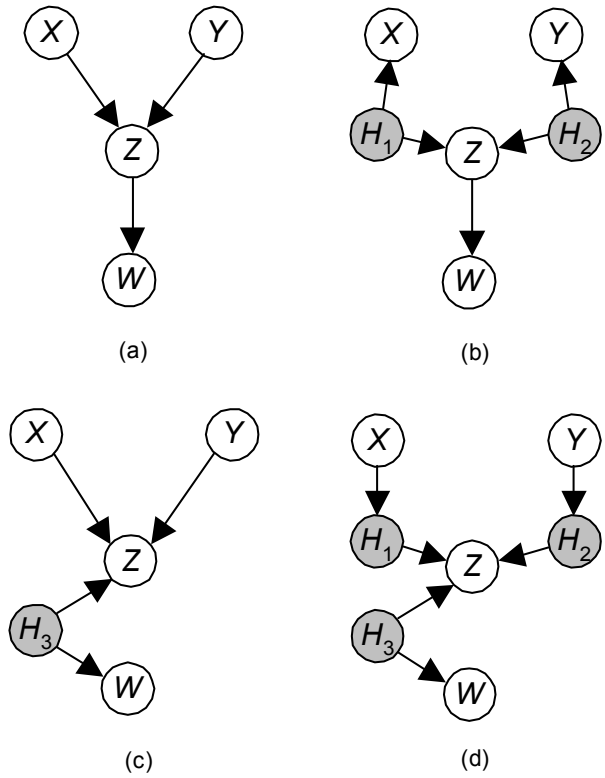


Figure 8.14: If our set of conditional independencies is $\{I_P(X,Y), I_P(W,\{X,Y\}|Z)\}$, then P is embedded faithfully in the DAGs in (a) and (b) but not in the DAGs in (c) and (d).

faithfulness assumption is warranted. Note that this example illustrates a situation in which we identify four variables and two of them have a hidden common cause. That is, we have not identified lung cancer as a feature of humans.

Let's see how much we can learn about causal influences when we make only the causal embedded faithfulness assumption.

Example 8.21 *Recall that in Example 8.18,* $\mathsf{V} = \{X, Y, Z\}$*, our set of conditional independencies was*

$$\{I_P(X,Y)\},$$

and we concluded that X *and* Y *each caused* Z *while making the causal faithfulness assumption. However, the probability distribution is embedded faithfully in all the DAGs in Figure 8.13. So, if we make only the causal embedded faithfulness assumption, it could be that* X *causes* Z*, or it could be that* X *and* Z *have a hidden common cause. The same holds for* Y *and* Z*.*

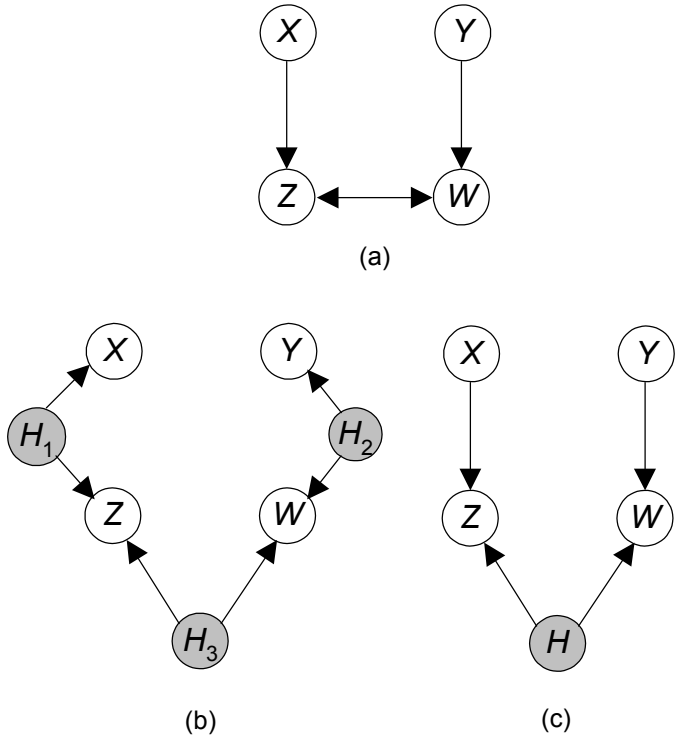


Figure 8.15: If our set of conditional independencies is $\{I_P(X, \{Y, W\}), I_P(Y, \{X, Z\})$, we can conclude that Z and W have a hidden common cause.

While making only the more reasonable causal embedded faithfulness assumption, we were not able to learn any causal influences in the previous example. Can we ever learn a causal influence while making only this assumption? The next example shows that we can.

Example 8.22 *Recall that in Example 8.20,* $\mathsf{V} = \{X, Y, Z, W\}$ *and our set of conditional independencies was*

$$\{I_P(X, Y), \quad I_P(W, \{X, Y\}|Z)\}.$$

In this case the probability distribution P *is embedded faithfully in the DAGs in Figures 8.14 (a) and 8.14 (b). However, it is not embedded faithfully in the DAGs in Figure 8.14 (c) or 8.14 (d). The reason is that these latter DAGs entail* $I_P(X, W)$*, and we do not have this conditional independency. That is, the Markov condition says* X *must be independent of its nondescendents conditional on its parents. Since* X *has no parents, this means that* X *must simply be independent of its nondescendents, and* W *is one of its nondescendents. If we make the causal embedded faithfulness assumption, we conclude that* Z *causes* W*.*

Example 8.23 *Recall that in Example 8.16,* $\mathsf{V} = \{X, Y, Z, W\}$*, our set of conditional independencies was*

$$\{I_P(X, \{Y, W\}), \quad I_P(Y, \{X, Z\}),$$

and we obtained the graph in Figure 8.15 (a) when we tried to learn a DAG faithful to P*. We concluded that there is no DAG faithful to* P*. Then in Example 8.17 we showed that* P *is embedded faithfully in the DAG in Figure 8.15 (c).* P *is also embedded faithfully in the DAG in Figure 8.15 (b). If we make the causal embedded faithfulness assumption, we conclude that* Z *and* W *have a hidden common cause.*

8.5 Model Averaging

Heckerman et al. [1999] illustrate that when the number of variables is small and the amount of data is large, one structure can be orders of magnitude more likely than any other. In such cases, model selection yields good results. However, recall that in Example 8.4 we had few data, we obtained $P(\mathbb{G}_1|\mathsf{D}) = .517$ and $P(\mathbb{G}_2|\mathsf{D}) = .483$, and we chose (learned) DAG $\mathbb{G}_1$ because it was most probable. Then in Example 8.8 we used a Bayesian network containing DAG $\mathbb{G}_1$ to do inference for Sam. Since the probabilities of the two models are so close, it seems somewhat arbitrary to choose $\mathbb{G}_1$. So, model selection does not seem appropriate. Next, we describe another approach.

Instead of choosing a single DAG and then using it to do inference, we could use the Law of Total Probability to do the inference as follows: We perform the inference using each DAG and multiply the result (a probability value) by the posterior probability of the DAG. This is called **model averaging**.

Example 8.24 *Recall that based on the data in Example 8.4 we learned that*

$$P(\mathbb{G}_1|\mathsf{D}) = .517$$

and

$$P(\mathbb{G}_2|\mathsf{D}) = .483.$$

In Example 8.8 we updated a Bayesian network containing $\mathbb{G}_1$ *based on the data to obtain the Bayesian network in Figure 8.16 (a). If in the same way we update a Bayesian network containing* $\mathbb{G}_2$*, we obtain the Bayesian network in Figure 8.16 (b). Given that Sam has never defaulted on a loan (*$F = f_2$*), we can then use model averaging to compute the probability that Sam is a white-collar worker, as follows:*[1]

$$\begin{aligned} P(j_1|f_1, \mathsf{D}) &= P(j_1|f_1, \mathbb{G}_1)P(\mathbb{G}_1|\mathsf{D}) + P(j_1|f_1, \mathbb{G}_2)P(\mathbb{G}_2|\mathsf{D}) \\ &= (.714)\,(.517) + (7/12)(.483) = .651. \end{aligned}$$

[1] Note that we substitute $P(\mathbb{G}_1|\mathsf{D})$ for $P(\mathbb{G}_1|f_1, \mathsf{D})$. They are not exactly equal, but we are assuming that the data set is sufficiently large that the dependence of the DAG models on the current case can be ignored.

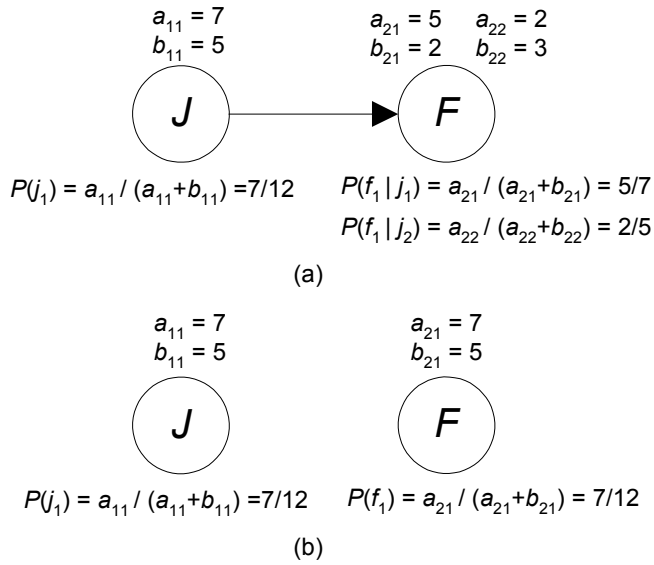


Figure 8.16: Updated Bayesian network for learning based on the data in Examples 8.4 and 8.8.

The result that $P(j_1|f_1, \mathbb{G}_1) = .714$ was obtained in Example 8.8, although in that example we did not show the dependence on $\mathbb{G}_1$ because that DAG was the only DAG considered. The result that $P(j_1|f_1, \mathbb{G}_2) = 7/12$ is obtained directly from the Bayesian network in Figure 8.16 (b) because J and F are independent in that network.

Example 8.24 illustrated using model averaging to do inference. The following example illustrates using it to learn partial structure.

Example 8.25 *Suppose we have three random variables X_1, X_2, and X_3. Then the possible DAG patterns are the ones in Figure 8.17. We might be interested in the probability that a feature of the DAG pattern is present. For example, we might be interested in the probability that there is an edge between X_1 and X_2. Given the five DAG patterns in which there is an edge, this probability is 1, and given the six DAG pattern in which there is no edge, this probability is 0. Let gp denote a DAG pattern. If we let F be a random variable whose value is present if a feature is present,*

$$\begin{aligned} P(F = present|\mathsf{D}) &= \sum_{gp} P(F = present|gp, \mathsf{D}) P(gp|\mathsf{D}) \\ &= \sum_{gp} P(F = present|gp) P(gp|\mathsf{D}), \end{aligned}$$

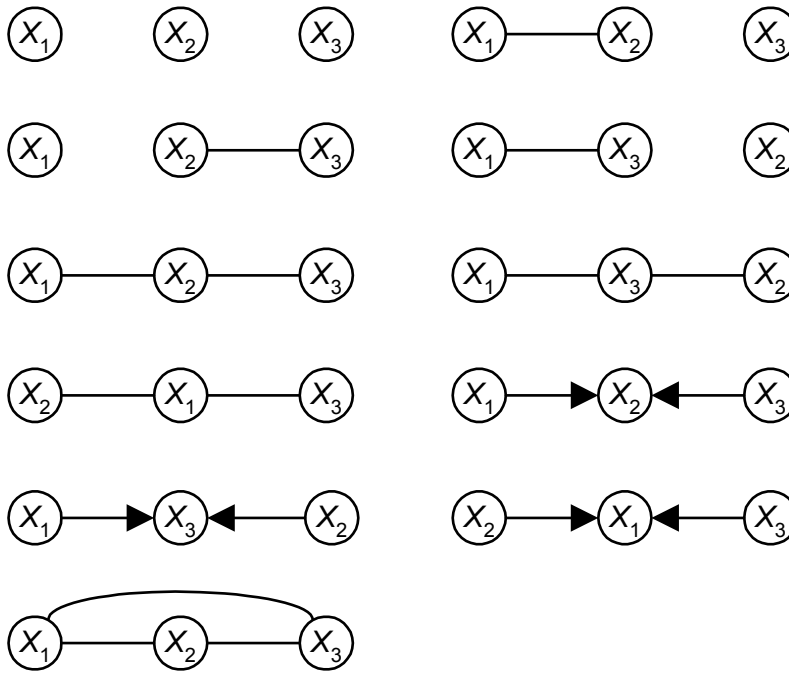


Figure 8.17: The 11 DAG patterns when there are three nodes.

where

$$P(F = present|gp) = \begin{cases} 1 & \textit{if the feature is present in } gp \\ 0 & \textit{if the feature is not present in } gp. \end{cases}$$

You may wonder what event a feature represents. For example, what event does an edge between X_1 and X_2 represent? This event is the event that X_1 and X_2 are not independent and are not conditionally independent given X_3 in the actual relative frequency distribution of the variables. Another possible feature is that there is a directed edge from X_1 to X_2. This feature is the event that, assuming that the relative frequency distribution admits a faithful DAG representation, there is a directed edge from X_1 to X_2 in the DAG pattern faithful to that distribution. Similarly, the feature that there is a directed path from X_1 to X_2 represents the event that there is a directed path from X_1 to X_2 in the DAG pattern faithful to that distribution.

Given that we are only discussing the relative frequency distribution, these events are ordinarily not of great interest. However, if we are discussing causality, they tell us something about the causal relationships among the variables. For example, recall that in Example 5.16 we mentioned that Chapter 12 discusses learning from data how the proteins produced by one gene have a causal effect on the level of mRNA (called the *gene expression level*) of another gene. Ordinarily there are thousands of genes (variables), but typically we have at

most only a few thousand data items. In such cases, there are often many structures that are equally likely. So, choosing one particular structure is somewhat arbitrary. However, in these cases we are not always interested in learning the entire structure. That is, rather than needing the structure for inference and decision making, we are only interested in learning relationships among some of the variables. In particular, in the gene expression example, we are interested in the dependence and causal relationships between the expression levels of certain genes (see [Lander and Shenoy, 1999]).

As is the case for model selection, when the number of possible DAGs is large, we cannot average over all DAGs. In these situations we heuristically search for high-probability DAGs, and then we average over them. In particular, in the gene expression example, since there are thousands of variables, we could not average over all of them. We discuss approximate model averaging in Section 8.6.2.

8.6 Approximate Structure Learning

Recall from Section 8.2.4 that when the number of variables is not small, it is computationally unfeasible to find the maximizing DAGs by exhaustively considering all DAGs. Therefore, researchers have developed heuristic search algorithms. We discuss such algorithms next.

8.6.1 Approximate Model Selection

Here we discuss heuristic search algorithms for model selection. First we present algorithms that search over DAGs.

Algorithms That Search Over DAGs

We present two algorithms in which the search space consists of DAGs. Specifically, the search space is the set of all DAGs containing n nodes, where n is our number of random variables. In these algorithms, our goal is find a DAG with maximum score, where our scoring criterion could be the Bayesian score, the BIC score, or some other score. Therefore, we will simply refer to the score as $score\,(\mathbb{G} : \mathsf{D})$, where D is our data.

The K2 Algorithm If we use either the Bayesian score or the BIC score, the score for the entire DAG is a product of local scores for each node. For example, Theorem 8.4 obtains the result that the Bayesian score, in the case of multinomial variables, is given by

$$score_{Bayesian}(\mathbb{G} : \mathsf{D}) = P(\mathsf{D}|\mathbb{G}) = \prod_{i=1}^{n} \prod_{j=1}^{q_i^{\mathbb{G}}} \frac{\Gamma(N_{ij}^{\mathbb{G}})}{\Gamma(N_{ij}^{\mathbb{G}} + M_{ij}^{\mathbb{G}})} \prod_{k=1}^{r_i} \frac{\Gamma(a_{ijk}^{\mathbb{G}} + s_{ijk}^{\mathbb{G}})}{\Gamma(a_{ijk}^{\mathbb{G}})}.$$

See Theorem 8.4 for the definition of the variables in this formula. Note that we have now explicitly shown their dependence on $\mathbb{G}$. Let $\mathsf{PA}_i^{\mathbb{G}}$ denote the parents

of X_i in $\mathbb{G}$. For each node X_i, the value of

$$\prod_{j=1}^{q_i^{\mathbb{G}}} \frac{\Gamma(N_{ij}^{\mathbb{G}})}{\Gamma(N_{ij}^{\mathbb{G}} + M_{ij}^{\mathbb{G}})} \prod_{k=1}^{r_i} \frac{\Gamma(a_{ijk}^{\mathbb{G}} + s_{ijk}^{\mathbb{G}})}{\Gamma(a_{ijk}^{\mathbb{G}})}$$

depends only on parameter values stored locally at X_i, and data values of X_i and nodes in $\mathsf{PA}_i^{\mathbb{G}}$. Define

$$score(X_i, \mathsf{PA}_i^{\mathbb{G}} : \mathsf{D}) = \prod_{j=1}^{q_i^{\mathbb{G}}} \frac{\Gamma(N_{ij}^{\mathbb{G}})}{\Gamma(N_{ij}^{\mathbb{G}} + M_{ij}^{\mathbb{G}})} \prod_{k=1}^{r_i} \frac{\Gamma(a_{ijk}^{\mathbb{G}} + s_{ijk}^{\mathbb{G}})}{\Gamma(a_{ijk}^{\mathbb{G}})}.$$

Cooper and Herskovits [1992] developed a greedy search algorithm that tries to maximize the score of the DAG by maximizing these local scores. That is, for each variable X_i they locally find a value PA_i that approximately maximizes $score(X_i, \mathsf{PA}_i^{\mathbb{G}} : \mathsf{D})$. The single operation in this search algorithm is the addition of a parent to a node. The algorithm proceeds as follows: We assume an ordering of the nodes such that if X_i precedes X_j in the order, an arc from X_j to X_i is not allowed. Let $Pred(X_i)$ be the set of nodes that precede X_i in the ordering, We initially set the parents $\mathsf{PA}_i^{\mathbb{G}}$ of X_i to empty and compute $score(X_i, \mathsf{PA}_i^{\mathbb{G}} : \mathsf{D})$. Next we visit the nodes in sequence according to the ordering. When we visit X_i, we determine the node in $Pred(X_i)$ that most increases $score(X_i, \mathsf{PA}_i^{\mathbb{G}} : \mathsf{D})$. We "greedily" add this node to PA_i. We continue doing this until the addition of no node increases $score(X_i, \mathsf{PA}_i^{\mathbb{G}} : \mathsf{D})$. Pseudocode for this algorithm follows. The algorithm is called K2 because it evolved from a system named Kutató [Herskovits and Cooper, 1990].

Algorithm 8.1: K2

Problem: Find a DAG that approximates maximizing $score\,(\mathbb{G} : \mathsf{D})$.

Inputs: A set V of n random variables; an upper bound u on the number of parents a node may have; data D.

Outputs: n sets of parent nodes PA_i, where $1 \leq i \leq n$, in a DAG that approximates maximizing $score\,(\mathbb{G} : \mathsf{D})$.

```
void K2  (set_of_variables V, int u,
          data D, for 1 ≤ i ≤ n parent_set& PA_i)
{
  for (i = 1; i <= n; i++) {      // n is the number of variables.
    PA_i^G = ∅;
    P_old = score(X_i, PA_i^G : D);
    findmore = true;
```

```
        while (findmore && |PA_i^G| < u) {
          Z = node in Pred(X_i) − PA_i that maximizes
                score(X_i, PA_i^G ∪ {Z} : D);
          P_new = score(X_i, PA_i^G ∪ {Z} : D) ;
          if (P_new > P_old) {
            P_old = P_new;
            PA_i^G = PA_i^G ∪ {Z};
          }
          else
            findmore = false;
        }
      }
    }
```

Neapolitan [2004] analyzes the algorithm. Furthermore, he shows an example in which the algorithm was provided with a prior order and learned a DAG from 10,000 cases sampled at random from the ALARM Bayesian network [Beinlich et al., 1989]. The DAG learned was identical to the one in the ALARM network except that one edge was missing.

You might wonder where we could obtain the ordering required by Algorithm 8.1. Such an ordering could possibly be obtained from domain knowledge such as a time ordering of the variables. For example, we might know that in patients, smoking precedes bronchitis and lung cancer and that each of these conditions precedes fatigue and a positive chest X-ray.

When a model searching algorithm need only locally recompute a few scores to determine the score for the next model under consideration, we say the algorithm has **local scoring updating**. A model with local scoring updating is considerably more efficient than one without it. Clearly, the K2 algorithm has local scoring updating.

An Algorithm without a Prior Ordering We present a straightforward greedy search algorithm that does not require a time ordering. The search space is again the set of all DAGs containing the n variables, and the set DAGOPS of operations is as follows:

1. If two nodes are not adjacent, add an edge between them in either direction.
2. If two nodes are adjacent, remove the edge between them.
3. If two nodes are adjacent, reverse the edge between them.

All operations are subject to the constraint that the resultant graph does not contain a cycle. The set of all DAGs that can be obtained from $\mathbb{G}$ by applying one of the operations is called $\mathsf{Nbhd}(\mathbb{G})$. If $\mathbb{G}' \in \mathsf{Nbhd}(\mathbb{G})$, we say $\mathbb{G}'$ is in the **neighborhood** of $\mathbb{G}$. Clearly, this set of operations is **complete** for the

search space. That is, for any two DAGs $\mathbb{G}$ and $\mathbb{G}'$ there exists a sequence of operations that transforms $\mathbb{G}$ to $\mathbb{G}'$. The reverse edge operation is not needed for the operations to be complete, but it increases the connectivity of the space without adding too much complexity, which typically leads to a better search. Furthermore, when we use a greedy search algorithm, including edge reversals often seems to lead to a better local maximum.

The algorithm proceeds as follows: We start with a DAG with no edges. At each step of the search, of all those DAGs in the neighborhood of our current DAG, we "greedily" choose the one that maximizes $score\,(\mathbb{G} : \mathsf{D})$. We halt when no operation increases this score.

Note that in each step, if an edge to X_i is added or deleted, we need only re-evaluate $score(X_i, \mathsf{PA}_i : \mathsf{D})$. If an edge between X_i and X_j is reversed, we need only reevaluate $score(X_i, \mathsf{PA}_i : \mathsf{D})$ and $score(X_j, \mathsf{PA}_j : \mathsf{D})$. Therefore, this algorithm has local scoring updating. The algorithm follows:

Algorithm 8.2: DAG Search

Problem: Find a DAG that approximates maximizing $score\,(\mathbb{G} : \mathsf{D})$.

Inputs: A set V of n random variables; data D.

Outputs: A set of edges E in a DAG that approximates maximizing $score\,(\mathbb{G} : \mathsf{D})$.

```
void DAG_search (set_of_variables V, data D,
                 set_of_edges& E)
{
  E = ∅; G = (V, E);
  do
    if (any DAG in the neighborhood of our current DAG
    increases score (G : D))
      modify E according to the one that increases score (G : D) the most;
  while (some operation increases score (G : D));
}
```

A problem with a greedy search algorithm is that it could halt at a candidate solution that locally maximizes the objective function rather than globally maximizes it (see [Xiang et al., 1996]). One way for dealing with this problem is iterated hill-climbing. In iterated hill-climbing, local search is done until a local maximum is obtained. Then the current structure is randomly perturbed, and the process is repeated. Finally, the maximum over local maxima is used. Other methods for attempting to avoid local maxima include simulated annealing [Metropolis et al., 1953], best-first search [Korf, 1993], and Gibb's sampling [Neapolitan, 2004].

Searching over DAG Patterns

We present an algorithm that searches over DAG patterns. First, we discuss why we might want to do this.

Why Search over DAG Patterns? Although Algorithms 8.1 and 8.2 find a DAG $\mathbb{G}$ rather than a DAG pattern, we can use them to find a DAG pattern by determining the DAG pattern gp representing the Markov equivalence class to which $\mathbb{G}$ belongs. Since $score(gp : \mathsf{D}) = score(\mathbb{G} : \mathsf{D})$, we have approximated maximizing $score(\mathsf{D}, gp)$. However, as discussed in [Anderson et al., 2007], there are a number of potential problems in searching for a DAG instead of a DAG pattern. Briefly, we discuss two of the problems. The first is efficiency. By searching over DAGs, the algorithm can waste time encountering and rescoring DAGs in the same Markov equivalence class. A second problem has to do with priors. If we search over DAGs, we are implicitly assigning equal priors to all DAGs, which means that DAG patterns containing more DAGs will have higher prior probability. For example, if there are n nodes, the complete DAG pattern (representing no conditional independencies) contains $n!$ DAGs, whereas the pattern with no edges (representing that all variables are mutually independent) contains just one DAG. On the other hand, Gillispie and Pearlman [2001] show that an asymptotic ratio of the number of DAGs to DAG patterns equal to about 3.7 is reached when the number of nodes is only 10. Therefore, on average the number of DAGs in a given equivalence class is small, and perhaps our concern about searching over DAGs is not necessary. Contrariwise, in simulations performed by Chickering [2001] the average number of DAGs in the equivalence classes over which his algorithm searched were always greater than 8.5 and in one case was 9.7×10^{19}.

When performing model selection, assigning equal priors to DAGs is not necessarily a serious problem, since we will finally select a high-scoring DAG that corresponds to a high-scoring DAG pattern. However, as discussed in Section 8.6.2, it can be a more serious problem in the case of model averaging.

The GES Algorithm In 1997 Meek developed an algorithm called **Greedy Equivalent Search (GES)**, which searches over DAG patterns and has the following property: If there is a DAG pattern faithful to P, as the size of the data set approaches infinity, the limit of the probability of finding a DAG pattern faithful to P is equal to 1. In 2002 Chickering proved this is the case. We describe the algorithm next.

In what follows we denote the equivalence class represented by DAG pattern gp by gp. GES is a two-phase algorithm that searches over DAG patterns. In the first phase, DAG pattern gp' is in the neighborhood of DAG pattern gp, denoted $\mathsf{Nbhd}^+(gp)$, if there is some DAG $\mathbb{G} \in \mathsf{gp}$ for which a single edge addition results in a DAG $\mathbb{G}' \in \mathsf{gp}'$. Starting with the DAG pattern containing no edges, we repeatedly replace the current DAG pattern gp by the DAG pattern in $\mathsf{Nbhd}^+(gp)$ that has the highest score of all DAG patterns in $\mathsf{Nbhd}^+(gp)$. We do this until there is no DAG pattern in $\mathsf{Nbhd}^+(gp)$ that increases the score.

The second phase is completely analogous to the first phase. In this phase, DAG pattern gp' is in the neighborhood of DAG pattern gp, denoted $\mathsf{Nbhd}^-(gp)$, if there is some DAG $\mathbb{G} \in \mathsf{gp}$ for which a single edge deletion results in a DAG $\mathbb{G}' \in \mathsf{gp}'$. Starting with the DAG pattern obtained in the first phase, we repeatedly replace the current DAG pattern gp by the DAG pattern in $\mathsf{Nbhd}^-(gp)$ that has the highest score of all DAG patterns in $\mathsf{Nbhd}^-(gp)$. We do this until there is no DAG pattern in $\mathsf{Nbhd}^-(gp)$ that increases the score.

It is left as an exercise to write this algorithm.

Neapolitan [2004] discusses other algorithms that search over DAG patterns.

8.6.2 Approximate Model Averaging

As mentioned previously, Heckerman et al. [1999] illustrate that when the number of variables is small and the amount of data is large, one structure can be orders of magnitude more likely than any other. In such cases, approximate model selection yields good results. However, if the amount of data is not large, it seems more appropriate to do inference by approximately averaging over models as illustrated in Example 8.24. Another application of approximate model averaging would be to learn partial structure when the amount of data is small relative to the number of variables. In Example 8.25, we discussed how we might do this to learn how the protein transcription factors produced by one gene have a causal effect on the level of mRNA of another gene.

Approximate Model Averaging Using MCMC

Next we discuss how we can heuristically search for high-probability structures and then average over them using the Markov Chain Monte Carlo (MCMC) method.

Recall our two examples of model averaging (Examples 8.24 and 8.25). The first involved computing a conditional probability over all possible DAGs as follows:

$$P(x|y,\mathsf{D}) = \sum_{\mathbb{G}} P(x|y,\mathbb{G},\mathsf{D})P(\mathbb{G}|\mathsf{a},\mathsf{D}).$$

The second involved computing the probability a feature is present as follows:

$$P(F = present|\mathsf{D}) = \sum_{gp} P(F = present|gp)P(gp|\mathsf{D}).$$

In general, these problems involve some function of the DAG or DAG pattern and possibly the data and a probability distribution of the DAGs or DAG patterns conditional on the data. So, we can represent the general problem to be the determination of

$$\sum_{gp} f(gp,\mathsf{D})P(gp|\mathsf{D}), \tag{8.6}$$

where f is some function of gp and possibly D, and P is some probability distribution of the DAG patterns. Although we represented the problem in terms of DAG patterns, we could sum over DAGs instead.

To approximate the value of Expression 8.6 using MCMC, our stationary distribution $\mathbf{r}$ is $P(gp|\mathsf{D})$. Ordinarily we can compute $P(\mathsf{D}|gp)$ but not $P(gp|\mathsf{D})$. However, if we assume that the prior probability $P(gp)$ is the same for all DAG patterns,

$$\begin{aligned} P(gp|\mathsf{D}) &= \frac{P(\mathsf{D}|gp)P(gp)}{P(\mathsf{D})} \\ &= kP(\mathsf{D}|gp)P(gp), \end{aligned}$$

where k does not depend on gp. If we use Equality 3.14 or 3.15 as our expression for $\boldsymbol{\alpha}$, k cancels out of the expression, which means that we can use $P(\mathsf{D}|gp)$ in the expression for $\boldsymbol{\alpha}$. Note that we do not have to assign equal prior probabilities to all DAG patterns. That is, we could use $P(\mathsf{D}|gp)P(gp)$ in the expression for $\boldsymbol{\alpha}$ also.

If we average over DAGs instead of DAG patterns, the problem is the determination of

$$\sum_{\mathbb{G}} f(\mathbb{G}, \mathsf{D})P(\mathbb{G}|\mathsf{D}),$$

where f is some function of $\mathbb{G}$ and possibly D, and P is some probability distribution of the DAGs. As is the case for DAG patterns, if we assume that the prior probability $P(\mathbb{G})$ is the same for all DAGs, then $P(\mathbb{G}|\mathsf{D}) = kP(\mathsf{D}|\mathbb{G})$, and we can use $P(\mathsf{D}|\mathbb{G})$ in the expression for $\boldsymbol{\alpha}$. However, we must realize what this assumption entails. If we assign equal prior probabilities to all DAGs, DAG patterns containing more DAGs will have higher prior probability.

As noted previously, when we perform model selection, assigning equal prior probabilities to DAGs is not necessarily a serious problem, since we will finally select a high-scoring DAG that corresponds to a high-scoring DAG pattern. However, in performing model averaging, a given DAG pattern will be included in the average according to the number of DAGs in the pattern. For example, there are three DAGs corresponding to the DAG pattern $X - Y - Z$ but only one corresponding to DAG pattern $X \rightarrow Y \leftarrow Z$. So, by assuming that all DAGs have the same prior probability, we are assuming that the prior probability that the actual relative frequency distribution has the set of conditional independencies $\{I_P(X, Z|Y)\}$ is three times the prior probability that it has the set of conditional independencies $\{I_P(X, Z)\}$. Even more dramatic, there are $n!$ DAGs corresponding to the complete DAG pattern and only one corresponding to the DAG pattern with no edges. So, we are assuming that the prior probability that there are no conditional independencies is far greater than the prior probability that the variables are mutually independent.

This assumption has consequences as follows: Suppose, for example, that the correct DAG pattern is $X : Y$, which denotes the DAG pattern with no edge, and the feature of interest is $I_P(X, Y)$. Since the feature is present, our results are better if we confirm it. Therefore, averaging over DAG patterns has a better result because, by averaging over DAG patterns, we are assigning a prior probability of 1/2 to the feature, whereas by averaging over DAGs, we are only assigning a prior probability of 1/3 to the feature. On the other hand, if the correct DAG pattern is $X - Y$, the feature is not present, which means

that our results are better if we disconfirm. Therefore, averaging over DAGs is better.

We see then that we need look at the ensemble of all relative frequency distributions rather than any one to discuss which method might be "correct." If relative frequency distributions are distributed uniformly in nature and we assign equal prior probabilities to all DAG patterns, then $P(F = present|\mathsf{D})$, obtained by averaging over DAG patterns, is the relative frequency with which we are investigating a relative frequency distribution with this feature when we are observing these data. So, averaging over DAG patterns is "correct." On the other hand, if relative frequency distributions are distributed in nature according to the number of DAGs in DAG patterns and we assign equal prior probabilities to all DAGs, then $P(F = present|\mathsf{D})$, obtained by averaging over DAGs, is the relative frequency with which we are investigating a relative frequency distribution with this feature when we are observing these data. So, averaging over DAGs is "correct." Although it seems reasonable to assume that relative frequency distributions are distributed uniformly in nature, some feel that a relative frequency distribution, represented by a DAG pattern containing a larger number of DAGs, may occur more often because there are more causal relationships that can give rise to it.

Approximate Averaging over DAGs

After presenting a straightforward algorithm, we simplify it.

A Straightforward Algorithm We show how to use MCMC to approximate averaging over DAGs. Our set of states is the set of all possible DAGs containing the variables in the application, and our stationary distribution is $P(\mathbb{G}|\mathsf{D})$, but as noted previously, we can use $P(\mathsf{D}|\mathbb{G})$ in our expression for $\boldsymbol{\alpha}$. Recall from Section 8.6.1 that $\mathsf{Nbhd}(\mathbb{G})$ is the set of all DAGs that differ from $\mathbb{G}$ by one edge addition, one edge deletion, or one edge reversal. Clearly $\mathbb{G}_j \in \mathsf{Nbhd}(\mathbb{G}_i)$ if and only if $\mathbb{G}_i \in \mathsf{Nbhd}(\mathbb{G}_j)$. However, since adding or reversing an edge can create a cycle, if $\mathbb{G}_j \in \mathsf{Nbhd}(\mathbb{G}_i)$ it is not necessarily true that $\mathsf{Nbhd}(\mathbb{G}_i)$ and $\mathsf{Nbhd}(\mathbb{G}_j)$ contain the same number of elements. For example, if $\mathbb{G}_i$ and $\mathbb{G}_j$ are the DAGs in Figures 8.18 (a) and (b) respectively, then $\mathbb{G}_j \in \mathsf{Nbhd}(\mathbb{G}_i)$. However, $\mathsf{Nbhd}(\mathbb{G}_i)$ contains five elements because adding the edge $X_3 \rightarrow X_1$ would create a cycle, whereas $\mathsf{Nbhd}(\mathbb{G}_j)$ contains six elements. We create our transition matrix $\mathbf{Q}$ as follows: For each pair of states $\mathbb{G}_i$ and $\mathbb{G}_j$ we set

$$q_{ij} = \begin{cases} \dfrac{1}{|\mathsf{Nbhd}(\mathbb{G}_i)|} & \mathbb{G}_j \in \mathsf{Nbhd}(\mathbb{G}_i) \\ \\ 0 & \mathbb{G}_j \notin \mathsf{Nbhd}(\mathbb{G}_i) \end{cases},$$

where $|\mathsf{Nbhd}(\mathbb{G}_i)|$ returns the number of elements in the set. Since $\mathbf{Q}$ is not symmetric, we use Equality 3.14 rather than Equality 3.15 to compute α_{ij}. Specifically, our steps are as follows:

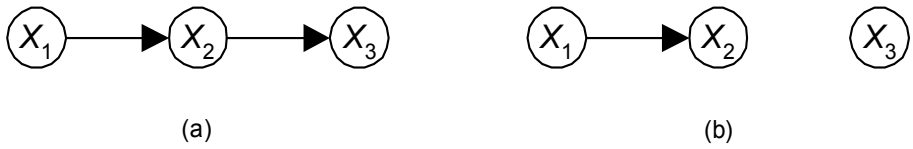


Figure 8.18: These DAGs are in each other's neighborhoods, but their neighborhoods do not contain the same number of elements.

1. If the DAG at the trial k is $\mathbb{G}_i$ choose a DAG uniformly from $\mathsf{Nbhd}(\mathbb{G}_i)$. Suppose that DAG is $\mathbb{G}_j$.
2. Choose the DAG for trial $k+1$ to be $\mathbb{G}_j$ with probability

$$\alpha_{ij} = \begin{cases} 1 & \frac{P(\mathbf{D}|\mathbb{G}_j) \times |\mathsf{Nbhd}(\mathbb{G}_i)|}{P(\mathbf{D}|\mathbb{G}_i) \times |\mathsf{Nbhd}(\mathbb{G}_j)|} \geq 1 \\ \frac{P(\mathbf{D}|\mathbb{G}_j)\,|\mathsf{Nbhd}(\mathbb{G}_i)|}{P(\mathbf{D}|\mathbb{G}_i)\,|\mathsf{Nbhd}(\mathbb{G}_j)|} & \frac{P(\mathbf{D}|\mathbb{G}_j) \times |\mathsf{Nbhd}(\mathbb{G}_i)|}{P(\mathbf{D}|\mathbb{G}_i) \times |\mathsf{Nbhd}(\mathbb{G}_j)|} \leq 1 \end{cases},$$

   and to be $\mathbb{G}_i$ with probability $1 - \alpha_{ij}$.

A Simplification It is burdensome to compute the sizes of the neighborhoods of the DAGs in each step. Alternatively, we could include DAGs with cycles in the neighborhoods. That is, $\mathsf{Nbhd}(\mathbb{G}_i)$ is the set of all graphs (including ones with cycles) that differ from $\mathbb{G}_i$ by one edge addition, one edge deletion, or one edge reversal. It is not hard to see that then the size of every neighborhood is equal $n(n-1)$. We therefore define

$$q_{ij} = \begin{cases} \frac{1}{n(n-1)} & \mathbb{G}_j \in \mathsf{Nbhd}(\mathbb{G}_i) \\ 0 & \mathbb{G}_j \notin \mathsf{Nbhd}(\mathbb{G}_i) \end{cases}.$$

If we are currently in state $\mathbb{G}_i$ and we obtain a graph $\mathbb{G}_j$ that is not a DAG, we set $P(\mathbf{D}|\mathbb{G}_j) = 0$ (effectively making r_j zero). In this way α_{ij} is zero, the graph is not chosen, and we stay at $\mathbb{G}_i$ in this step. Since $\mathbf{Q}$ is now symmetric, we can use Equality 3.15 to compute α_{ij}. Notice that our theory was developed by assuming that all values in the stationary distribution are positive, which is not currently the case. However, Tierney [1996] shows that convergence also follows if we allow 0 values as discussed here.

Neapolitan [2004] develops a similar method that averages over DAG patterns.

8.7 Software Packages for Learning

Based on considerations such as those illustrated in Section 8.3.1, Spirtes et al. [1993, 2000] developed an algorithm that finds the DAG faithful to P from the

conditional independencies in P when there is a DAG faithful to P. Spirtes et al. [1993, 2000] further developed an algorithm that learns a DAG in which P is embedded faithfully from the conditional independencies in P when such a DAG exists. These algorithms have been implemented in the Tetrad software package [Scheines et al., 1994], which can be downloaded for free from www.phil.cmu.edu/projects/tetrad/.

The Tetrad software package also has a module that uses the GES algorithm along with the BIC score to learn a Bayesian network from data.

Other Bayesian network learning packages include the following:

- Belief Network Power Constructor (constraint-based approach), www.cs.ualberta.ca/~jcheng/bnpc.htm.
- Bayesware (structure and parameters), www.bayesware.com/.
- Bayes Net Toolbox, bnt.sourceforge.net/.
- Probabilistic Net Library, www.eng.itlab.unn.ru/?dir=139.

EXERCISES

Section 8.2

Exercise 8.1 Suppose we have the two models in Figure 8.1 and the following data:

Case	J	F
1	j_1	f_1
2	j_1	f_1
3	j_1	f_1
4	j_1	f_1
5	j_1	f_1
6	j_1	f_2
7	j_2	f_2
8	j_2	f_2
9	j_2	f_2
10	j_2	f_1

1. Score each DAG model using the Bayesian score and compute their posterior probabilities, assuming that the prior probability of each model is .5.
2. Create a data set containing 20 records by duplicating the data in the table one time, and score the models using this 20-record data set. How have the scores changed?

3. Create a data set containing 200 records by duplicating the data in the table 19 times, and score the models using this 200-record data set. How have the scores changed?

Exercise 8.2 Assume that we have the models and data set discussed in Exercise 8.1. Using model averaging, compute the following:

1. $P(j_1|f_1, \mathsf{D})$ when D consists of our original 10 records.
2. $P(j_1|f_1, \mathsf{D})$ when D consists of our original 20 records.
3. $P(j_1|f_1, \mathsf{D})$ when D consists of our original 200 records.

Section 8.3

Exercise 8.3 Suppose $\mathsf{V} = \{X, Y, Z, U, W\}$ and the set of conditional independencies in P is

$$\{I_P(X, Y) \qquad I_P(\{W, U\}, \{X, Y\}|Z) \qquad I_P(U, \{X, Y, Z\}|W)\}.$$

Find all DAGs faithful to P.

Exercise 8.4 Suppose $\mathsf{V} = \{X, Y, Z, U, W\}$ and the set of conditional independencies in P is

$$\{I_P(X, Y) \qquad I_P(X, Z) \qquad I_P(Y, Z) \qquad I_P(U, \{X, Y, Z\}|W)\}.$$

Find all DAGs faithful to P.

Exercise 8.5 Suppose $\mathsf{V} = \{X, Y, Z, U, W\}$ and the set of conditional independencies in P is

$$\{I_P(X, Y|U) \qquad I_P(U, \{Z, W\}|\{X, Y\}) \qquad I_P(\{X, Y, U\}, W|Z)\}.$$

Find all DAGs faithful to P.

Exercise 8.6 Suppose $\mathsf{V} = \{X, Y, Z, W, T, V, R\}$ and the set of conditional independencies in P is

$$\{I_P(X, Y|Z) \qquad I_P(T, \{X, Y, Z, V\}|W)$$
$$I_P(V, \{X, Z, W, T\}|Y) \qquad I_P(R, \{X, Y, Z, W\}|\{T, V\})\}.$$

Find all DAGs faithful to P.

Exercise 8.7 Suppose $\mathsf{V} = \{X, Y, Z, W, U\}$ and the set of conditional independencies in P is

$$\{I_P(X, \{Y, W\}|U) \qquad I_P(Y, \{X, Z\}|U)\}.$$

1. Is there any DAG faithful to P?
2. Find DAGs in which P is embedded faithfully.

Section 8.4

Exercise 8.8 If we make the causal faithfulness assumption, determine what causal influences we can learn in each of the following cases:

1. Given the conditional independencies in Exercise 8.3
2. Given the conditional independencies in Exercise 8.4
3. Given the conditional independencies in Exercise 8.5
4. Given the conditional independencies in Exercise 8.6

Exercise 8.9 If we make only the causal embedded faithfulness assumption, determine what causal influences we can learn in each of the following cases:

1. Given the conditional independencies in Exercise 8.3
2. Given the conditional independencies in Exercise 8.4
3. Given the conditional independencies in Exercise 8.5
4. Given the conditional independencies in Exercise 8.6
5. Given the conditional independencies in Exercise 8.7

Section 8.5

Exercise 8.10 In Example 8.24, we computed $P(j_1|f_1, \mathsf{D})$ using model averaging. Use the same technique to compute $P(j_1|f_2, \mathsf{D})$.

Exercise 8.11 Assume that there are three variables X_1, X_2, and X_3, and that all DAG patterns have the same posterior probability (1/11) given the data. Compute the probability of the following features being present given the data (assuming faithfulness):

1. $I_p(X_1, X_2)$
2. $\neg I_p(X_1, X_2)$
3. $I_p(X_1, X_2|X_3)$ and $\neg I_p(X_1, X_2)$
4. $\neg I_p(X_1, X_2|X_3)$ and $I_p(X_1, X_2)$

Section 8.7

Exercise 8.12 Using Tetrad (or some other Bayesian network learning algorithm), learn a DAG from the data in Table 8.1. Next learn the parameters for the DAG. Can you suspect any causal influences from the learned DAG?

Exercise 8.13 Create a data file containing 120 records from the data in Table 8.1 by duplicating the data nine times. Using Tetrad (or some other Bayesian network learning algorithm), learn a DAG from this larger data set. Next learn the parameters for the DAG. Compare these results to those obtained in Exercise 8.12.

Exercise 8.14 Suppose we have the following variables:

Variable	What the Variable Represents
H	Parents' smoking habits
I	Income
S	Smoking
L	Lung cancer

and the following data:

Case	H	I	S	L
1	Yes	30,000	Yes	Yes
2	Yes	30,000	Yes	Yes
3	Yes	30,000	Yes	No
4	Yes	50,000	Yes	Yes
5	Yes	50,000	Yes	Yes
6	Yes	50,000	Yes	No
7	Yes	50,000	No	No
8	No	30,000	Yes	Yes
9	No	30,000	Yes	Yes
10	No	30,000	Yes	No
11	No	30,000	No	No
12	No	30,000	No	No
14	No	50,000	Yes	Yes
15	No	50,000	Yes	Yes
16	No	50,000	Yes	No
17	No	50,000	No	No
18	No	50,000	No	No
19	No	50,000	No	No

Using Tetrad (or some other Bayesian network learning algorithm) learn a DAG from these data. Next learn the parameters for the DAG. Can you suspect any causal influences from the learned DAG?

Exercise 8.15 Create a data file containing 190 records from the data in Exercise 8.14 by duplicating the data nine times. Using Tetrad (or some other Bayesian network learning algorithm), learn a DAG from this larger data set. Next learn the parameters for the DAG. Compare these results to those obtained in Exercise 8.14.