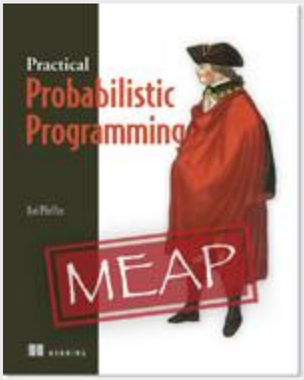

Practical Probabilistic Programming: Bayesian Networks

By Avi Pfeffer

In this article, I'll talk about Bayesian networks, which are the standard framework for encoding asymmetric relationships using directed dependencies.

Encoding relationships between variables is essential to probabilistic modeling. In this article, excerpted from my book, Practical Probabilistic Programming, I'll talk about Bayesian networks, which are the standard framework for encoding asymmetric relationships using directed dependencies. I'll provide a full definition and an explanation of the reasoning patterns you can use.

Bayesian Networks Defined

A Bayesian network is a representation of a probabilistic model consisting of three components:

1. **A set of variables with their corresponding *domains***. In the example of Figure 1, there are three variables: Subject, Size, and Brightness. The domain of a variable specifies which values are possible for that variable. The domain of Subject is { People, Landscape }, the domain of Size is { Small, Medium, Large }, while the domain of Brightness is { Dark, Bright }.

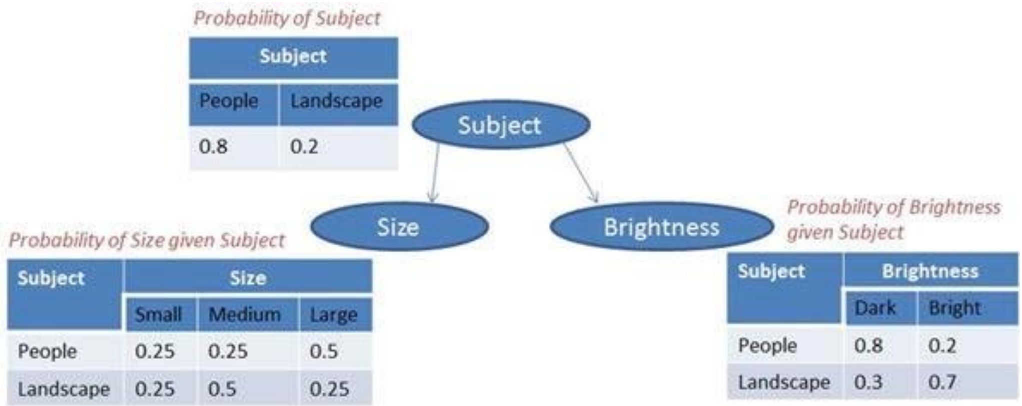

Figure **1**: A three-node Bayesian network

2. **A *directed acyclic graph* in which each variable is a node**. "Directed" means that each edge in the graph has a direction; it goes from one variable to another. The first variable is called the parent, while the second variable is called the child. In the example of Figure 1, Subject is a parent of both Size and Brightness. The word "acyclic" means that there are no cycles in the graph. This means that there are no *directed cycles* that follow the direction of the arrows. In other words, you can't start at a node, follow the arrows, and end up at the same node. However, it is allowed to have an *undirected cycle* that ignores the direction of the arrows. This concept is illustrated in Figure 2. The graph on the left has a directed cycle A-B-D-C-A. In the graph on the right, the cycle A-B-D-C-A sometimes runs counter to the direction of the arrows, so it is an undirected cycle. Therefore, the graph on the left is not allowed but the graph on the right is allowed.

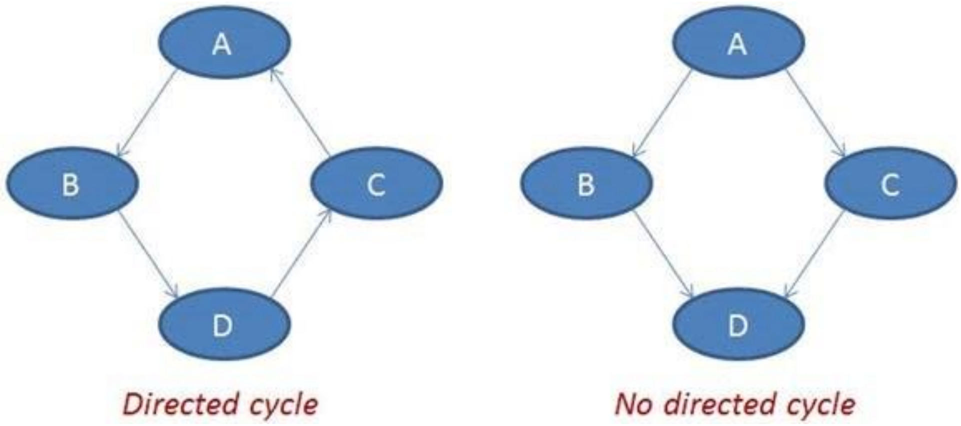

Figure 2: A directed cycle is a cycle that follows the arrows and ends up where it started.

3. **For each variable, a conditional probability distribution (CPD) over the variable given its parents.** A CPD specifies a probability distribution over the child variable given the values of its parents. This CPD considers every possible assignment of values to the parents, when the value of a parent can be any value in its domain. For each such assignment, it defines a probability distribution over the child. In Figure 1, each variable has a CPD. Subject is a root of the network, so it has no parents. When a variable has no parents, the CPD just specifies a single probability distribution over the variable. In this example, Subject takes the value People with probability 0.8 and Landscape with probability 0.2. Size has a parent, which is Subject, so its CPD has a row for each value of Subject. The CPD says that when Subject has value People, the distribution over Size makes Size has value Small with probability 0.25, Medium with probability 0.25, and Large with probability 0.5. When Subject has value Landscape, Size has a different distribution. Finally, Brightness also has parent Subject, and its CPD also has a row for each value of Subject.

How a Bayesian Network Defines a Probability Distribution

That's all there is to the definition of Bayesian networks. Now, let's see how a Bayesian network defines a probability distribution. The first thing we need to do is define the possible worlds. For a Bayesian network, a possible world consists of an assignment of values to each of the variables, making sure the value of each variable is in its domain. For example, <Subject = People, Size = Small, Brightness = Bright> is a possible world.

Next, we have to define the probability of a possible world. This is quite simple. All we have to do is identify the entry in the CPD of each variable that matches the values of the parents and child in the possible world. The process is illustrated in Figure 3. For example, for the possible world <Subject = People, Size = Small, Brightness = Bright>, the entry for Subject is 0.8, which is from the column labeled Subject. For Size, we look at the row corresponding to Subject = People and the column corresponding to Size = Small and get the entry 0.25. Finally, for Brightness, we again look at the row corresponding to Subject = People and this time take the column labeled Bright to get the entry 0.2. Finally, we multiply all these entries together to get the probability of the possible world, which is 0.8 * 0.25 * 0.2 = 0.04.

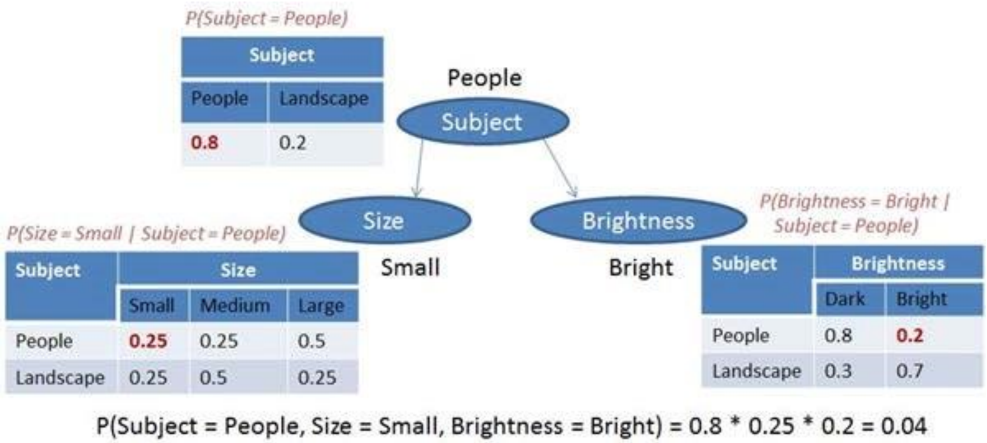

Figure 3: Computing the probability of a possible world by multiplying the appropriate entries in each CPD.

If you go through this process for every possible world, the probabilities will add up to one, just as they are supposed to. This is always the case for a Bayesian network. So, we have seen how a Bayesian network defines a valid probability distribution. Now that we understand exactly what a Bayesian network consists of and what it means, let's see how you can use one to derive beliefs about some variables given knowledge of other variables.

Reasoning with Bayesian Networks

A Bayesian network encodes a lot of independencies that hold between variables. Recall that independence between two variables means that learning something about one of the variables doesn't tell us anything new about the other variable. From the above example, we can see that Print Result Summary and Printer Power Button On are *not* independent. When we learned that the print result was empty, that reduced the probability that the power button was on.

Conditional independence is similar. Two variables are conditionally independent given a third variable if, *once the third variable is known*, learning something about the first variable doesn't tell us anything new about the second. There is a criterion, called d-separation, for determining when two variables in a Bayesian network are conditionally independent of a third set of variables. The criterion is a little involved, so I'm not going to provide a formal definition. Instead, I'll describe the basic principles and show you a few examples.

The basic idea is that reasoning *flows along a path* from one variable to another. In the example of Figure 1, reasoning can flow from Size to Brightness along the path Size-SubjectBrightness. In an indirect dependency, reasoning flows from one variable to another variable via other intermediary variables. In this example, Subject is the intermediary variable between Size and Brightness. In a Bayesian network, reasoning can flow along a path as long as the path is not *blocked* at some variable.

In most cases, a path is blocked at a variable if the variable is observed. So, if Subject is observed, the path Size-Subject-Brightness is blocked. This means that if we observe Size, it will not change our beliefs about Brightness if Subject is also observed. Another way of saying this is that Size is conditionally independent of Brightness given Subject. In our model, the painter's choice of subject determines the size and the brightness, but after choosing the subject, the size and brightness are generated independently.

CONVERGING ARROWS AND INDUCED DEPENDENCIES

However, there is one other case that may seem counterintuitive at first. This is a case where a path is blocked at a variable in the variable is unobserved, and becomes unblocked if the variable is observed. I'll illustrate this situation by extending the example as in Figure 4. There's a new variable called Material, which could be oil or watercolor or some such. Naturally, Material is a cause of Brightness (perhaps oil paintings are brighter than watercolors), so the network has a directed edge from Material to Brightness. Here, we have two parents of the same child. This is called a *converging arrows* pattern, because edges from Subject and Material converge at Brightness.

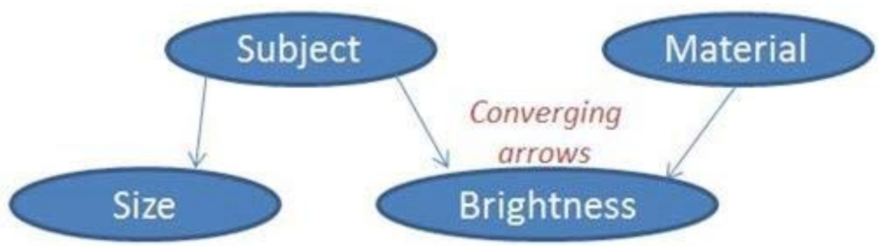

Figure 4: Extended painting example, including converging arrows between two parents of the same child

Now, let's think about reasoning between Subject and Material. According to the model, Subject and Material are generated independently. So:

(1) Subject and Material are independent when nothing is observed.

But, what happens when we observe that the painting is bright? According to our model, landscapes tend to be brighter than people paintings. So, after observing that the painting is bright, we will infer that the painting is more likely to be a landscape. Let's say we then observe that the painting is an oil painting, which also tends to be bright. This observation provides an alternative explanation of the brightness of our painting. Therefore, the probability the painting is a landscape is discounted somewhat compared to what it was after we observed that the painting is bright but before we observed that it's an oil painting. We can see that reasoning is flowing from Material to Subject along the path Material-Brightness-Subject. So, we get the following statement:

(2) Subject and Material are not conditionally independent given Brightness.

What we have here is the opposite pattern from the usual. We have a path that is blocked when the intermediary variable is unobserved, and becomes unblocked when the variable is observed. This kind of situation is called an *induced dependency*. An induced dependency is a dependency between two variables that is induced by observing a third variable. Any converging arrows pattern in a Bayesian network can lead to an induced dependency.

In general, a path between two variables can include both ordinary patterns and converging arrows. Reasoning only flows along the path if it is not blocked at any node on the path. Figure 5 shows four examples for the path Size-Subject-Brightness-Material. In the top left example, neither Subject nor Brightness is observed, and the path is blocked at Brightness because it has converging arrows. In the top right, Subject is now observed, so the path is blocked at both Subject and Brightness. In the bottom left, Subject is unobserved and Brightness is observed, which is precisely the condition required for the path not to be blocked at either Subject or Brightness. Finally, in the bottom right, Subject is observed, so the path is blocked there.

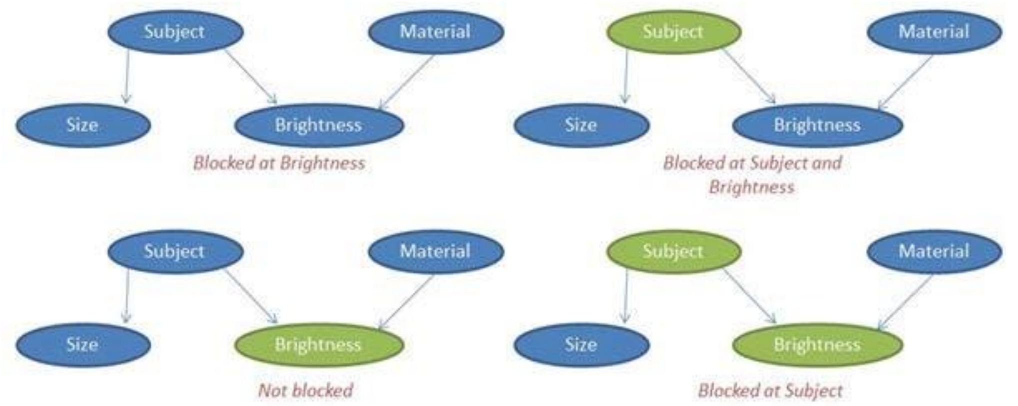

Figure 5: Examples of blocked and unblocked paths that combine an ordinary pattern with a converging arrows pattern. Each figure shows the path from Size to Material with Subject and Brightness either unobserved or observed.