Bayesian Networks

The table given in the example of the first handout is one way to represent a probability space. But it is a computationally rather expensive way. To store the probability function $P$ for three variables that have two values each, we need to store 8 values. In general the storage grows exponentially with the number of variables, which makes this representation problematic for computational purposes. Therefore, in the 80ties people started to look for alternative ways to implement probabilistic reasoning. This led to the development of Bayes Nets.

Let’s go back to the first example. In this case $B$ depended on $A$, $C$ on $B$, but $C$ was independent of $A$ given $B$. We could represent these dependencies by drawing arrows between the variables, leading from a variable to those dependent on it. For our example, this gives us the DAG (directed acyclic graph\(^1\)) in figure 1. Let’s call for each variable $X_i$ the variables that have an arrow leading to $X_i$ the parents of $X_i$ and let $PA_i$ denote the set of all parents of $X_i$. In our example $A$ is the only parent of $B$ and $B$ is the only parent of $C$. $A$ doesn’t have any parents. If we now store only the conditional probabilities of a variable $X_i$ given its parents, we can do with significantly less storage. The less dependencies, the more economic the representation.

![Figure 1: Bayes Net for example 1](image)

This procedure associating graphs with probability distributions can be generalised. Suppose we have a probability distribution $P$ for variables $X_1, X_2, ..., X_n$. We proceed stepwise through the list of variables. For $X_1$ we do nothing. For $X_2$ we check whether it depends on $X_1$. If yes, we draw an arrow from $X_1$ to $X_2$. If not, we do nothing. We proceed to the third variable. If this variable is independent of the two first, no arrow is drawn. Otherwise check whether one screens off $X_3$ from the other. If yes, draw an arrow from the variable that does the screening off to $X_3$. If no, draw an arrow from both variables to $X_3$. In general, for each new variable $X_i$, we determine a minimal set of predecessors that screen off the new variable from all the other variables.

---

1A graph consists of a set of $V$ of vertices (or nodes) and a set $E$ of edges (or links) that connect some pairs of vertices. The vertices in our graphs will correspond to variables. Edges in the graph can be either directed or indirected. A graph with only directed edges is called directed. Furthermore, graphs may include directed cycles (e.g. $X \rightarrow Y, Y \rightarrow X$). A graph that is directed and contains no directed cycles is called a directed acyclic graph (DAG).
predecessors. This set defines the parents of $X$ (the variables that will send arrows to $X_i$). We then store for each $X_i$ only the probability of $X_i$ given its parents.

The graphic representation we obtain this way only makes sense if from it we can recompute the original probability distribution. But that is possible. Basically, it is an application of the chain rule (5) on the first handout. Let's apply it to our example: $P(a,b,c) = P(c|a,b)P(b|a)P(a)$. But given that we know that $C$ is independent of $A$ given $B$, the first factor can be simplified ($P(a,b,c) = P(c|b)P(b|a)P(a)$) and all necessary values can be directly read off the graphic representation. This can be generalised to arbitrary graphs obtained by our procedure.

\[-\ln{P(x_1,\ldots,x_n)} = \prod^n_{i=1} P(x_i|\text{pa}_i) \tag{1}\]

DAGs as in figure 1 are called Bayesian networks. They were developed to allow an efficient representation of probabilistic information. The graph tells us which conditional probabilities we have to store. It thereby facilitates economical representation of probabilistic information. The graph can do that because it encodes information about probabilistic (in-)dependence encoded in a probability distribution $P$. Which graph is suitable for the probability function we want to store depends on whether it represents correctly the conditional dependencies of the distribution. We can use the formula in (1) to define this “suitability” of a graph for a probability distribution. This is called the Markov condition.

**Definition 1** If a probability function $P$ admits the factorisation of (1) relative to a DAG $G$, we say that $G$ and $P$ are compatible, or that $P$ is Markov relative to $G$.

We said that a graph represents information about probabilistic (in-)dependence. But it doesn’t represent all the information of a probability distribution. There is a set of probability distributions that are Markov compatible with a graph. This set can be characterised using $d$-separation, a graphical criterion that allows you to read off probabilistic independencies from a graph (see Pearl [2000] for more information). Using $d$-separation one can prove theorem 1. This theorem allows us to turn around the process we used at the beginning to arrive at (1). In our example we used conditional (in-)dependence to construct the graph and then saw that we can get back the probability distribution using the formula in (1). This tells us that if a probability distribution is Markov compatible with a graph $G$, i.e. the equation in (1) holds with respect to the graph, then $P$ renders every variable is independent of all its non-descendants given its parents.

**Theorem 1** A necessary and sufficient condition for a probability distribution $P$ to be Markov relative a DAG $G$ is that every variable be independent of all its non descendants (in $G$), conditional on its parents.

**Inference in Bayesian networks**

Rule 1 also gives us an easy way to calculate total probabilities $P(x)$. Lets call $Y$ the variables that are neither $X_i$ nor in $PA_i$.

\[
P(x_i) = \sum_{PA_i} \sum_{Y} P(x_i, pa_i, y) \quad \text{summing up the joints}
\]
\[
= \sum_{PA_i} \sum_{Y} P(x_i|pa_i, y) P(y|pa_i) P(pa_i) \quad \text{using the product rule}
\]
\[
= \sum_{PA_i} \sum_{Y} P(x_i|pa_i) P(y|pa_i) P(pa_i) \quad \text{using independence in Bayesian Nets}
\]
\[
= \sum_{PA_i} P(x_i|pa_i) P(pa_i) \quad \text{the last sum sums up to 1}
\]
The inference you will want to compute given a Bayes Net is what the probability of a set of variables X is given evidence that the variables E take a certain value. In such a situation we can distinguish three types of variables: evidence variables E, query variables X and the variables Y that you don’t care about. The posterior probability of X given the evidence e can be computed using the rule for calculating conditional probabilities and determining $P(x, e)$ and $P(e)$ by summing up the relevant joints (see rule (1) on the probabilities handout). The joints are computed using the adapted chain rule for Bayesian Nets (1). On the handout we already discussed how the normalisation factor $P(e)$ (here called $\alpha$) can be calculated.

$$P(x|e) = \frac{P(x, e)}{P(e)} = \alpha P(x, e) = \alpha \Sigma Y p(x, e, y)$$

However, the involved computations take a lot of effort. In general calculating inferences for arbitrary Bayesian Nets is NP hard. Various algorithms have been developed, specifically suited to different network structures and performance requirements. For some networks exact computation is not computational feasible. For such cases approximation algorithms have been proposed. For more information on the algorithms, read, for instance, chapter 3 in Korb and Nicholson [2010].

**Example 2**

Suppose we are given the graph in figure 2 with the respective conditional probabilities.

![Bayes Net for example 2](image)

Figure 2: Bayes Net for example 2

We can compute $P(C)$ and $P(D)$ using (2). This gives $P(C) = .3061$ and $P(D) = .0194$. What if now we learned that C is the case? This will change the probability of the other variables. We can use Bayes rule to calculate the probabilities.

$$P(A|C) = \frac{P(C|A)P(A)}{P(C)} = \frac{(P(C|A,B)P(B)+P(C|A,\neg B)P(\neg B))P(A)}{P(C)}$$

Thus, we see that the probability of A increases. The same holds for B: $P(B|C) = .0294$. Given that B becomes more likely, the same holds for D: $P(D|C) = P(B|C)P(D|B) + (1 - P(B|C))P(D|\neg B) = .0376$.

Suppose that we additionally learn that D is the case. This will increase the probability of B even more: $P(B|C, D) = .7422$. D is pretty good evidence for B, but C also helps: $P(B|D) = .4896$. But what does the evidence for D do with A? It decreases: $P(A|C, D) =$
Evidence for $B$ makes it more likely that $B$ explains $C$. Thereby $A$ becomes a less likely explanation. This effect is known as explaining away. In the initial Bayesian Net the variables $A$ and $B$ where probabilistically independent. But conditional on $C$ they become dependent.

This becomes very intuitive if we give the graph a causal interpretation (which you probably already did). Assume that $A$ stands for having brain cancer, $C$ for having been in a fight, $C$ for having a headache and $D$ for being bruised up. Both, cancer and having been in a fight are potential causes of a headache. A fight will additionally cause you to be bruised up. Assume that you wake up with a headache. This makes it more probable that you have brain cancer and also that you have been in a fight, as we have seen. You walk into your bathroom and notice that you have bruises all over your face. You will think that you have been in a fight (that you forgot all about because of all the alcohol you consumed) and that your headache is due to the fight, not due to brain cancer. The second potential cause (brain cancer) is explained away by evidence for the first (fighting).

### Causal Bayesian Networks

In the last section we discussed a way to compute a graph $G$ that a probability distribution $P$ is Markov compatible with. But the graph is not unique given $P$. Which graph you get depends on the order of variables you start with. We are here interested in one particular graph: the one where the arrow represent direct causal dependencies between the variables. The causal interpretation of Bayes Nets is the dominant way these representations are used nowadays. Sloman describes this interpretation as follows:

Given a probability distribution the causal graph in a number of senses special among all possible graphs a probability function is Markov compatible with. First of all, there is an intuitive sense in which this is the right way to represent the dependencies between the involved variables. The following quote from Sloman [2005] illustrates this nicely: the causal dependency between variables is perceived as the relevant or underlying dependency between variables; the dependency that generates the stochastic dependency. But that’s just saying that the causal reading of the graph might be an adequate description of how we conceptualise the relation between stochastic and causal dependence. But it says nothing about the practical relevance of this particular reading of the graph, or why we should go for the graph matching the causal dependencies between the variables.

One of the central ideas of the causal modeling framework is that stable probabilistic relations between the observed variables of a system are generated by an underlying causal structure. In other words, the world we see around us with all its uncertainty can be attributed to the operation of a big, complicated network of causal mechanisms. On a smaller scale, particular kinds of causal structure will lead to particular patterns of probability in the form of particular patterns of dependence and independence. [Sloman [2005]: 43]

Another way in which the causal graph is special is that it seems to be one of the graphs with the least number of edges that a probability function is Markov compatible with. For practical applications the number of edges is of serious importance; less edges always means better computational properties, in terms of storage as well as inference. From this perspective, even with a purely stochastically interpretation of the nets, using one of the graphs with the

---

2 You can calculate all these values by using the standard formula for conditional probability and then determine the involved quantities by summation over the joints.

3 Remember that our initial motivation for looking at Causal Bayesian Networks was to find a representation of causal dependencies that we can use to describe the meaning of conditional sentences.
smallest number of edges is preferred. However, in general there is no unique minimum. This can be illustrated with a simple example. Suppose there are three variables $A$, $B$ and $C$, and suppose that $A$ and $B$ are dependent and also that $A$ and $C$ are dependent but independent given $B$. Then there are still three possibilities to represent this information in a graph (see figure 3). Purely based on stochastic information it is not possible to distinguish between these graphs. They are Markov-equivalent or observational equivalent. In general, one can prove the following theorem.

![Figure 3: Three possible graphs describing the same observations on independence between the involved variables.](image)

**Theorem 2** Two DAGs are observationally equivalent if and only if they have the same skeletons (the same undirected edges) and the same sets of v-structures, that is, two converging arrows whose tails are not connected by an arrow (Verma and Pearl 1990).

For many application establishing causal dependence is of great importance. Think, for instance, of medicine. This observation that causal dependence is related to a graph that is Markov compatible and has a minimum of edges can be used to detect causal dependence. But as we saw, the minimal graph is not unique. Still, there exist very successful algorithms that compute causal structure dependent on results like the theorem mentioned above (see Pearl [2000], Spirtes et al. [2001]).

Also from a philosophical point of view this observation about minimal graphs is interesting. Why do we have the notion of causation? One might answer this question as follows: we cannot represent full probability distributions, because of how cognitively expensive the representations would be. There is, thus, (evolutionary) pressure to store information about regularities in a more efficient way. Causality could now be explained as a notion that emerges out of going for the most efficient representation of stochastic information. This gives an interesting twist to the Kantian claim that we have this notion because it is a necessary precondition of the possibility of knowledge. But remember that the minimal graph is generally not uniquely defined, but we do think that there is only one correct representation of the underlying causal dependencies. Thus, this idea gets us only so far. Furthermore, we appear to miss a very important aspect of causation in this story: causation facilitates control.

This brings us to a final and maybe the most important feature of Causal Bayesian Networks. In Pearl’s interpretation we have to understand the arrows in these nets as representing autonomous physical mechanisms. Thereby, one can model a possible change in these mechanisms easily by a local change in the network. Think back to our first example, which we also gave a causal graph for. What, for instance, if we find another method to get me drunk. We can easily model this by introducing another cause for $B$ in the graph, without that we would have
to change anything else except the probability distribution of $B$ given its parents. And we can use this augmented network to calculate the probability that I will have a headache after. This would be impossible if one would work with a different graph just based on the probabilistic dependencies between the variables. Furthermore, with the help of the notion of manipulation in networks we can make sense of the statement that causation facilitates control. How exactly we can make reasoning based on manipulation precise will be the topic on the next section.

Interventions

Example 3. If a bulgar tries to enter our house ($A$), this will turn on the alarm ($B$) and all the lights in the house ($C$). That, in turn, will bring our neighbours to the scene to check things out. This can be modelled with the left graph given in figure 4. In this case we take the arrow to represent causal dependency.

![Figure 4: Causal Bayes Net for example 4, on the left side is the initial network, on the right side the network after intervention in $B$](image)

If we hear no alarm, our belief that a burglary is going on will decrease: $P(A|\neg B) = 0.0005$. This is the kind of calculation we did in all the previous examples. But what if somebody disables the alarm? This shouldn’t reduce our belief about the probability of a burglary going on. So, there is a difference between learning that $B$ is off and turning $B$ off: both cases facilitate different inferences.

This observation motivates Pearl to distinguish two different ways to reason with Causal Bayesian Nets, one based on observation or learning new information about the world, and one based on intervention: changing the world. The interventions that Pearl considers are forcing a variable to take one of its values. In the example we forced the variable $B$ to take the value 0. To be able to model this second type of reasoning we need Causal Bayes Nets.

The philosophy behind the notion of an intervention is that it is a change in a causal system brought about by a variable not part of the model currently under investigation. This variable can bring about direct change in the value of a target variable, which is part of the model. Furthermore, only this target variable is affected (and its effects) and only through a direct route through this new variable. It should be clear that these are conditions that are difficult to guarantee in practise (see Hausman and Woodward [1999], Woodward [2003] for discussion, and Cartwright [2002] for a critical analysis).
Formally, interventions can be modelled as transformations on the probability function. Setting variable $X$ to value $x$ will change $P$ into $P_{X=x}$ where we change the probability of $X = x$ to 1 but keep for all other variables $Y \neq X$ the old dependency on its parents: $P_{X=x}(y|pa_y) = P(y|pa_y)$. Causal Networks allow an elegant graph transformation corresponding to an intervention. In our example we intervene in $B$. We can models this by cutting $B$ off its parents and setting $P(B) = 0$, while we leave everything else unchanged. The joint distribution changes as given in (6). It is not telling us anything new about causally independent variables.

$$P_{B=0}(a, c, d) = P(d|B = 0, c)P(c|a)P(a) \tag{6}$$

Lets go back to our example. Without any information the probability that we find the neighbours checking the house are rather low: $P(D) = \Sigma_{A,B}P(a)P(b|a)P(c|a)P(D|b,c) = .0625$. If we learn that the alarm is off, that probability decreases even more. No alarm is evidence that no burglary is going on. That lowers the chance that the lights are on, and that, in turn, will keep the neighbours away: $P(D|\neg B) = \Sigma_{A,C}P(a|\neg B)P(c|a)P(D|c, \neg B) = .055$. If the alarm is turned off, things don’t change a lot: $P_{-B}(D) = \Sigma_{A,C}P(a)P(c|a)P(D|c, \neg B) = .0554$. It’s a bit more than $P(D|\neg B)$, because the probability of a burglary doesn’t decrease this time. But the change caused by the intervention will not further propagate through the network. The probability of $C$, for instance, won’t change.

$$P_{-B}(c) = \Sigma_{PA,c}P_{-B}(c|pa_c)P_{-B}(pa_c)$$
$$= \Sigma_A P_{-B}(c|a)P_{-B}(a)$$
$$= \Sigma_A P(c|a)P(a)$$
$$= P(c)$$

If the bulgar breaks into the house after disabling the alarm, he will have decreased the probability that he will be disturbed by noisy neighbours: $P_{-B,A}(D) = 0.0955$ compared with $P_A(D) = .47.4$. So we see, it makes definitely sense to disable the alarm before you break in.

We have now two ways to reason with Causal Bayes Nets:

1. **observational inference**, reasoning from observing that a set of variables $X$ have value $x$, or, in other words, trying to answer the question what one could learn from observing that $X$ has value $x$;
2. **interventional inference**, reasoning from manipulating the value of variables $X$ to a certain setting $x$, or, in other words, trying to answer the question what would happen if one sets the value of $X$ to $x$.

Pearl combines both modes of reasoning into a third.

3. **counterfactual inference**, reasoning of the type: given that $X$ has value $x$, what would have happened had $X$ value $y \neq x$.

We started looking into Causal Bayesian Nets, because they provide an attractive way to model causal dependencies. We wanted such a representation because it appeared to be relevant information for the semantics of conditional sentences. But with this last notion of inference we immediately get a theory for the semantics of counterfactual conditionals. Let us also illustrate this modus of reasoning with our example. Given that the alarm is not on, what would happen if we turned the alarm on. This inference involves a 3-step procedure:

---

4Learning $A$ or intervening in $A$ doesn’t make any difference.
1. We update with the observation that $B$ is false. We just calculate $P(A|\neg B) = .0005$. The conditional dependencies don’t change, thus we can represent the updated probability function as in the graph on the left side in figure 5 (we use $P^*$ to shorten $P(\cdot|\neg B)$).

2. We intervene in the probability function and set $B$ to 1. See the right side in figure 5 for the result. This will set $P^*_B(B) = 1$. The conditional probabilities remain unchanged.

3. We compute inferences. $P^*_B(D) = \Sigma_{A,C} P^*_B(a)P^*_B(c|a)P^*_B(D|B,c) = .4002$, which is slightly less than what we get when intervening in $B$ without learning first that the alarm is off. This is the case because observing that the alarm is off will decrease the probability that a burglary is going on, which, in turn decreases the probability that the lights will be on.

![Figure 5: Counterfactual reasoning for example 4](image-url)

References


Katrin Schulz 8 k.schulz@uva.nl