Bayesian Networks 2016–2017
Tutorial III – Probabilistic inference

Peter Lucas
LIACS, Leiden University

Pearl's algorithm

Pearl's algorithm is an object-oriented algorithm for probabilistic reasoning, assuming that the question of how to compute the marginal probability distribution $P(V_i)$ for any vertex V_i in a Bayesian network can be answered in terms of the information which should be sent by other vertices in the graph to V_i in order to make it possible for V_i to compute this probability distribution locally. This information is described in terms of two types of *messages*: messages sent to the vertex in the same direction as the arcs in the graph, called *causal parameters*, and messages in the reverse direction, called *diagnostic parameters*. As there may be more than one incoming/outgoing arc for any vertex V_i, those messages must actually be combined by V_i, to obtain what are called a *compound causal* $\pi(V_i)$ and a *compound diagnostic parameter* $\lambda(V_i)$. Those compound parameters are again combined in order to compute the marginal probability $P^*(V_i)$ (known as the *data fusion lemma*):

$$P^*(V_i) = \alpha \cdot \pi(V_i) \cdot \lambda(V_i)$$

where α is a normalisation constant to ensure that $P^*(v_i) + P^*(\neg v_i) = 1$.

A node V_i can compute its compound causal parameter from its own probability function (expressed in terms of conditional probability values) and the causal parameter it receives from its parents:

$$\pi(V_i) = \sum_{\rho(V_i)} P(V_i \mid \rho(V_i)) \cdot \prod_{j=1}^{m} \pi_{V_i}^{V_j}(V_j)$$

with parents $\rho(V_i) = V_1 \wedge \cdots \wedge V_j \wedge \cdots \wedge V_m$ The compound causal parameter for a vertex describes the combined influence on this vertex probabilities of all evidence entered for all its non-descendants.

Further, the compound diagnostic parameter of V_i is computed from the separate diagnostic parameter that it receives from its children (unless it has been observed):

$$\lambda(V_i) = \prod_{j=1}^{m} \lambda_{V_j}^{V_i}(V_i)$$

The compound diagnostic parameter for a vertex describes the combined influence of all evidence that has been entered for its descendants.

A vertex V_i can calculate the causal parameter that it needs to send to a successor V_{i_j} from its compound causal parameter and the diagnostic parameters it receives from its *other* successors (unless it has been observed):

$$\pi^{V_i}_{V_{i_j}}(V_i) = \alpha \cdot \pi_{V_i}(V_i) \cdot \prod_{k=1,\ldots,m,k\neq j} \lambda^{V_i}_{V_{i_k}}(V_i)$$

Finally, a vertex V_i can calculate the diagnostic parameter that it needs to send to a predecessor V_{j_k} from its own probability function and the diagnostic parameter it receives from its successors:

$$\lambda^{V_{j_k}}_{V_i}(V_{j_k}) = \alpha \cdot \sum_{V_i} \lambda_{V_i}(V_i) \cdot [\sum_{\rho(V_i)\setminus\{V_{j_k}\}} P(V_i \mid (\rho(V_i) \setminus \{V_{j_k}\}) \wedge V_{j_k}) \cdot \prod_{l=1,\ldots,n,l\neq k} \pi^{V_{j_l}}_{V_i}(V_{j_l})]$$

The causal parameter is a parameter that a node sends to a descendant to provide this descendant with information regarding its non-descendants, while a diagnostic parameter that a node sends to its predecessor in order to provide the information concerning the vertices in the subtree whose root is the vertex at hand.

Note. The answers below are given only for one probability of the distribution; the complementary probability is computed analogously. At the end, make sure to compute α as well.

Exercise 1

Let $\mathcal{B} = (G, P)$ be a Bayesian network with directed acyclic graph $G = (V(G), A(G))$ and joint probability distribution P, as shown in Figure 1.

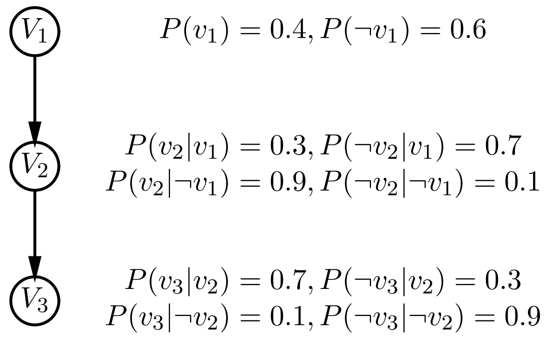
V_1
V_2
V_3
$P(v_1) = 0.4, P(\neg v_1) = 0.6$
$P(v_2|v_1) = 0.3, P(\neg v_2|v_1) = 0.7$
$P(v_2|\neg v_1) = 0.9, P(\neg v_2|\neg v_1) = 0.1$
$P(v_3|v_2) = 0.7, P(\neg v_3|v_2) = 0.3$
$P(v_3|\neg v_2) = 0.1, P(\neg v_3|\neg v_2) = 0.9$

Figure 1: Bayesian network.

a. Which probabilistic information is needed by vertex V_2 in order to locally compute $P(V_2)$. Give your answer in terms of causal parameters and diagnostic parameters. Use the data fusion lemma to compute $P(V_2)$.
[**Answer:** $P^*(v_2) = \alpha \cdot 0.66$]

b. Assume that $V_1 = true$ has been entered into the network. Now, compute the causal and diagnostic parameters for V_2. Also compute the compound causal and diagnostic parameters for V_3. Finally, apply the data fusion lemma to determine $P^*(V_3)$.
[**Answer:** $P^*(v_2) = \alpha \cdot 0.3$, $P^*(v_3) = \alpha \cdot 0.28$]

c. Now assume that in addition to $V_1 = \textit{true}$ the value *false* has been entered for V_3. Compute $P^*(V_i)$, for $i = 1, 2, 3$.

[**Answer:** $P^*(v_2) = \alpha \cdot 0.09$]

Exercise 2

Consider Figure 2 and the following probability distribution:

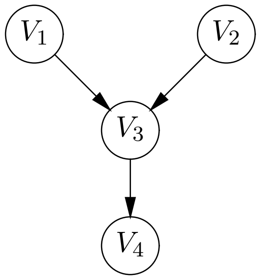

Figure 2: Bayesian network.

$P(v_4 \mid v_3) = 0.3$	$P(\neg v_4 \mid v_3) = 0.7$
$P(v_4 \mid \neg v_3) = 0.5$	$P(\neg v_4 \mid \neg v_3) = 0.5$
$P(v_3 \mid v_1, v_2) = 0.4$	$P(\neg v_3 \mid v_1, v_2) = 0.6$
$P(v_3 \mid \neg v_1, v_2) = 0.2$	$P(\neg v_3 \mid \neg v_1, v_2) = 0.8$
$P(v_3 \mid v_1, \neg v_2) = 0.7$	$P(\neg v_3 \mid v_1, \neg v_2) = 0.3$
$P(v_3 \mid \neg v_1, \neg v_2) = 0.3$	$P(\neg v_3 \mid \neg v_1, \neg v_2) = 0.7$
$P(v_1) = 0.1$	$P(\neg v_1) = 0.9$
$P(v_2) = 0.8$	$P(\neg v_2) = 0.2$

a. Compute the marginal probability distributions $P(V_4)$ and $P(V_1)$ using data fusion.

[**Answer:** $P^*(v_1) = \alpha \cdot 0.1$, $P^*(v_4) = \alpha \cdot 0.4532$]

b. Next, assume $V_2 = \textit{true}$ and $V_3 = \textit{true}$ have been entered into the network. Again compute $P^*(V_4)$ and $P^*(V_1)$ using data fusion.

[**Answer:** $P^*(v_1) = \alpha \cdot 0.04$, $P^*(v_4) = \alpha \cdot 0.3$]

Compare your results with those obtained when using standard probability theory.

Loop cutset conditioning

Exercise 3

Consider Figure 3 with associated probability distribution:

$P(v_3 \mid v_1, v_2) = 0.4$	$P(\neg v_3 \mid v_1, v_2) = 0.6$
$P(v_3 \mid \neg v_1, v_2) = 0.2$	$P(\neg v_3 \mid \neg v_1, v_2) = 0.8$
$P(v_3 \mid v_1, \neg v_2) = 0.7$	$P(\neg v_3 \mid v_1, \neg v_2) = 0.3$
$P(v_3 \mid \neg v_1, \neg v_2) = 0.3$	$P(\neg v_3 \mid \neg v_1, \neg v_2) = 0.7$

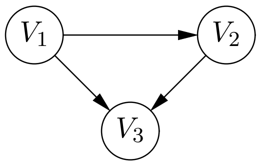

Figure 3: Bayesian network with 'loop' in the underlying undirected graph.

$P(v_2 \mid v_1) = 0.8$	$P(\neg v_2 \mid v_1) = 0.2$
$P(v_2 \mid \neg v_1) = 0.2$	$P(\neg v_2 \mid \neg v_1) = 0.8$
$P(v_1) = 0.1$	$P(\neg v_1) = 0.9$

Suppose we are interested in computing $P(v_3)$.

a. What are the possible loop cutsets for this graph?

b. Suppose you take $\{V_1\}$ as a loop cutset. The idea of loop cutset conditioning is to remove an arc from the loop cutset to a child by 'instantiating' the CPT of that child to possible values of the parent. Call the resulting probability distribution P^*. For example, if $V_1 = \mathit{true}$, then we could remove the edge from V_1 to V_2 so that the CPT of V_2 becomes:

   $P^*(v_2) = 0.8$ $\qquad$ $P^*(\neg v_2) = 0.2$

   What is the CPT of V_2 if we assume that V_1 is false? Which other edges could you remove, given this loop cutset? And is this necessary for applying Pearl's algorithm?

c. Suppose we assume that v_1 ($V_1 = \mathit{true}$) in the CPT of V_2 so that we can delete the arc from V_1 to V_2. Show that $P^*(v_3) \neq P(v_3 \mid v_1)$. Also show that $P^*(v_1, v_3) = P(v_1, v_3)$. What do you now think of the name 'loop cutset *conditioning*'?

d. Compute $P(v_3)$ using loop cutset conditioning.

Junction tree algorithm

The junction tree algorithm is an algorithm for computing probabilistic queries on general graphs. It uses a message passing algorithm on a modified graph called a *junction tree*, which can be obtained from the Bayesian network using moralisation and triangulation (see slides for more detail).

Computation proceeds in the junction tree via the following update equations:

$$\varphi^*_S = \sum_{V \backslash S} \varphi_V$$

$$\varphi^*_W = \frac{\varphi^*_S}{\varphi_S} \varphi_W$$

The message-passing protocol ensures that a clique can only send a message to a neighbouring clique when it has received messages from all of its neighbours. This can be done in different orders, e.g., the 'Hugin algorithm' designates an arbitrary node as root node and propagates messages inward to the root and outward to the leaves.

After message passing, the potentials in the cliques are equal to the marginal probability of the nodes in the clique (given the evidence).

Exercise 4

Again, consider Figure 2 and its corresponding probability distribution.

a. Moralise and triangulate (if necessary) to obtain the junction tree.

b. Give two factorisations of the given Bayesian network in the form:

$$P(V) = \frac{\prod_C \varphi_C}{\prod_S \varphi_S}$$

where C are cliques and S separators in the junction tree. The first factorisation should correspond to the situation before message passing (i.e., $\prod_S \varphi_S = 1$). The second factorisation should correspond to the situation after message passing (i.e., $\varphi_C = P(C)$).

c. Compute $P(V_3)$ by assigning values to the parameters of the junction tree and perform message passing.

d. Suppose v_1 ($V_1 = \textit{true}$). Again compute $P(V_3)$ using the junction tree algorithm.

Exercise 5

Let G^m be a graph resulting from the moralisation of G. Prove that if G is an I-map, then G^m is an I-map.

Exercise 6

The junction tree algorithm uses a forward and backward pass going from and to an arbitrary root of the tree. Proof that after both passes it holds that (*local consistency*):

$$\sum_{V \backslash S} \varphi_V^{**} = \sum_{W \backslash S} \varphi_W^{**}$$

where V and W are neighbouring cliques with separator S.