Probabilistic Foundations of Artificial Intelligence

Problem Set 3

Oct 21, 2016

1. Variable elimination

In this exercise you will use variable elimination to perform inference on a bayesian network. Consider the network in figure 1 and its corresponding conditional probability tables (CPTs).

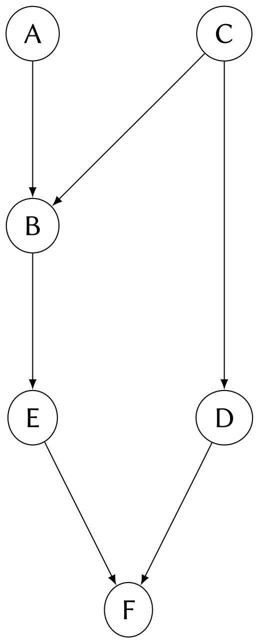


Figure 1: Bayesian network for problem 1.

$$P(A = t) = 0.3 \tag{1}$$
$$P(C = t) = 0.6 \tag{2}$$

Table 1: CPTs for problem 1.

(a) $P(B|A, C)$

A	C	$P(B = t)$
f	f	0.2
f	t	0.8
t	f	0.3
t	t	0.5

(b) $P(D|C)$

C	$P(D = t)$
f	0.9
t	0.75

(c) $P(E|B)$

B	$P(E = t)$
f	0.2
t	0.4

(d) $P(F|D, E)$

D	E	$P(F = t)$
f	f	0.95
f	t	1
t	f	0
t	t	0.25

Assuming a query on A with evidence for B and D, i.e. $P(A|B, D)$, use the variable elimination algorithm to answer the following queries. Make explicit the selected ordering for the variables and compute the probability tables of the intermediate factors.

1. $P(A = t|B = t, D = f)$
2. $P(A = f|B = f, D = f)$
3. $P(A = t|B = t, D = t)$

Consider now the ordering, C, E, F, D, B, A, use again the variable elimination algorithm and write down the intermediate factors, this time without computing their probability tables. Is this ordering better or worse than the one you used before? Why?

2. Belief propagation

In this exercise, you will implement the belief propagation algorithm for performing inference in Bayesian networks. As you have seen in the class lectures, the algorithm is based on converting the Bayesian network to a factor graph and then passing messages between variable and factor nodes of that graph until convergence.

You are provided some skeleton Python code in the `.zip` file accompanying this document. Take the following steps for this exercise.

1. Install the Python dependencies listed in `README.txt`, if your system does not already satisfy them. After that, you should be able to run `demo.py` and produce some plots, albeit wrong ones for now.
2. Implement the missing code in `bprop.py` marked with `TODO`. In particular, you have to fill in parts of the two functions that are responsible for sending messages from variable to factor nodes and vice versa, as well as parts of the function that returns the resulting marginal distribution of a variable node after message passing has terminated.
3. If your implementation is correct, you should get correct results for the naive Bayes model of the demo file that represents the coin flipping network of exercise 3 in Problem Set 2.
4. Now, set up the full-fledged earthquake network, whose structure was introduced in Problem Set 3 and is shown again in Figure 2. Here is the story behind this network:

   While Fred is commuting to work, he receives a phone call from his neighbor saying that the burglar alarm in Fred's house is ringing. Upon hearing this, Fred immediately turns around to get back and check his home. A few minutes on his way back, however, he hears on the radio that there was an earthquake near his home earlier that day. Relieved by the news, he turns around again and continues his way to work.

   To build up the conditional probability tables (CPTs) for the network of Figure 2 you may make the following assumptions about the variables involved:

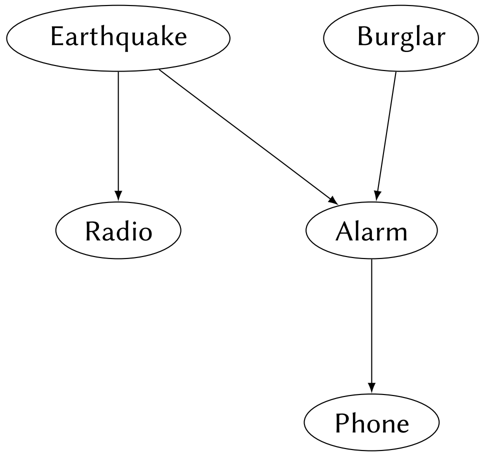


Figure 2: The earthquake network to be implemented.

- All variables in the network are binary.
- As can be seen from the network structure, burglaries and earthquakes are assumed to be independent. Furthermore, each of them is assumed to occur with probability 0.1%.
- The alarm is triggered in the following ways: (1) When a burglar enters the house, the alarm will ring 99% of the time; (2) when an earthquake occurs, there will be a false alarm 1% of the time; (3) the alarm might go off due to other causes (wind, rain, etc.) 0.1% of the time. These three types of causes are assumed to be independent of each other.
- The neighbor is assumed to call only when the alarm is ringing, but only does so 70% of the time when it is actually ringing.
- The radio is assumed to never falsely report an earthquake, but it might fail to report an earthquake that actually happened 50% of the time. (This includes the times that Fred fails to listen to the announcement.)

5. After having set up the network and its CPTs, answer the following questions using your belief propagation implementation:

   (a) Before Fred gets the neighbor's call, what is the probability of a burglary having occurred? What is the probability of an earthquake having occurred?

   (b) How do these probabilities change after Fred receives the neighbor's phonecall?

   (c) How do these probabilities change after Fred listens to the news on the radio?

3. Belief propagation on tree factor graphs*

In this exercise we will prove that the belief propagation algorithm converges to the exact marginals after a fixed number of iterations given that the factor graph is a tree, that is, given that the original Bayesian network is a polytree.

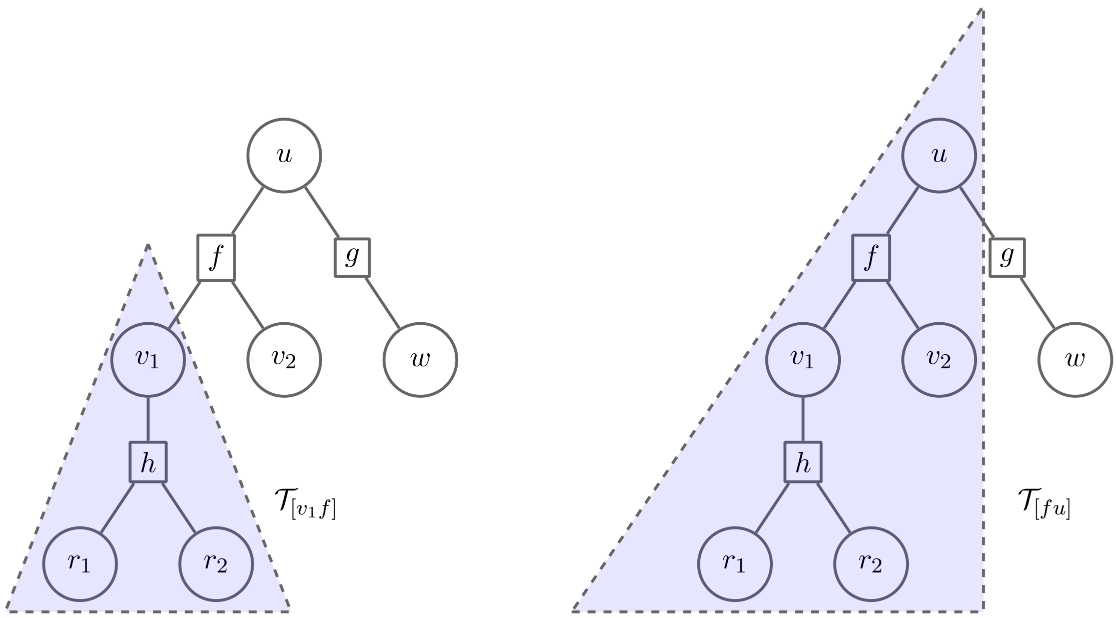


Figure 3: An example factor graph and two of its subtrees.

We will assume that the factor graph contains no single-variable factors. (You have already seen that if those exist, they can easily be incorporated into multi-variable factors without increasing the complexity of the algorithm.) Since the factor graph is a tree, we will designate a variable node, say a, as the root of the tree. We will consider subtrees $\mathcal{T}_{[rt]}$, where r and t are adjacent nodes in the factor graph and t is closer to the root than r, which are of two types:

- if r is a factor node (and t a variable node), then $\mathcal{T}_{[rt]}$ denotes a subtree that has t as its root, contains the whole subtree under r and, additionally, the edge $\{r, t\}$,
- if r is a variable node (and t a factor node), then $\mathcal{T}_{[rt]}$ denotes the whole subtree under r with r as its root.

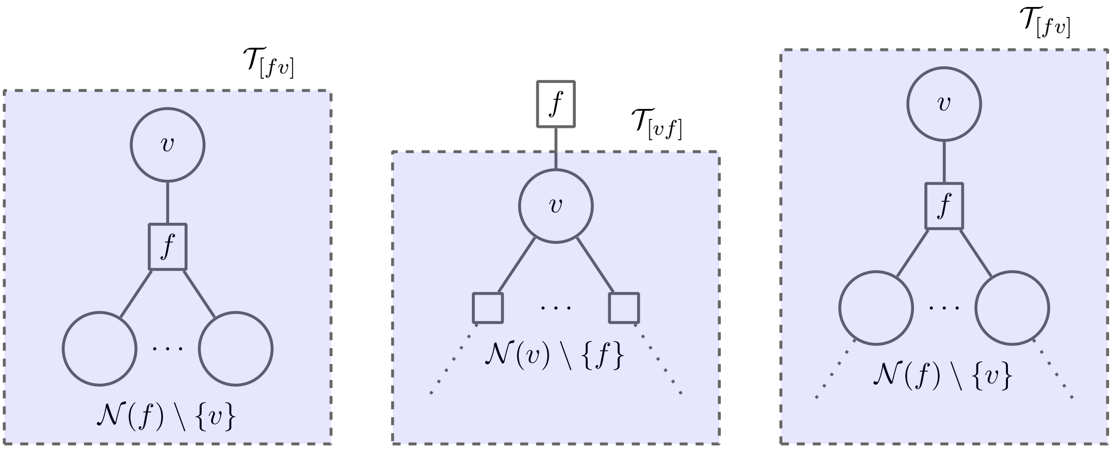
See Figure 3 for two example subtrees, one of each type. The depth of a tree is defined as the maximum distance between the root and any other node. Note that, both types of subtrees $\mathcal{T}_{[rt]}$ defined above have always depths that are even numbers.

We will use the subscript notation $[rt]$ to refer to quantities constrained to the subtree $\mathcal{T}_{[rt]}$. In particular, we denote by $\mathcal{F}_{[rt]}$ the set of factors in the subtree and by $P_{[rt]}(x_v)$ the marginal distribution of v when we only consider the nodes of the subtree. More concretely, if r is a variable node, by the sum rule we get

$$P_{[rt]}(x_r) \simeq \sum_{\boldsymbol{x}_{[rt]\setminus\{r\}}} \prod_{f \in \mathcal{F}_{[rt]}} f(\boldsymbol{x}_f), \tag{1}$$

where $\simeq$ denotes equality up to a normalization constant.

Remember the form of the messages passed between variable and factor nodes at each iteration



(a) Base case. (b) First induction step. (c) Second induction step.

Figure 4: The three cases considered in the proof by induction.

of the algorithm:

$$\mu_{v\to f}^{(t+1)}(x_v) := \prod_{g\in\mathcal{N}(v)\setminus\{f\}} \mu_{g\to v}^{(t)}(x_v) \tag{2}$$

$$\mu_{f\to v}^{(t+1)}(x_v) := \sum_{\boldsymbol{x}_{f\setminus\{v\}}} f(\boldsymbol{x}_f) \prod_{w\in\mathcal{N}(f)\setminus\{v\}} \mu_{w\to f}^{(t)}(x_w). \tag{3}$$

We also define the estimated marginal distribution of variable v at iteration t as

$$\hat{P}^{(t)}(x_v) := \prod_{g\in\mathcal{N}(v)} \mu_{g\to v}^{(t)}(x_v). \tag{4}$$

Our ultimate goal is to show that the estimated marginals are equal to the true marginals for all variables after a number iterations. However, we will first consider the rooted version of the factor graph and show that the previous statement holds for the root node a. More concretely, if we denote variable nodes with v and factor nodes with f, we will show using induction that for all subtrees $\mathcal{T}_{[fv]}$ of depth τ, it holds that, for all $t \geq \tau$,

$$\mu_{f\to v}^{(t)}(x_v) \simeq P_{[fv]}(x_v). \tag{5}$$

(i) Consider the base case of $\mathcal{T}_{[fv]}$ being a subtree of depth $\tau = 2$ (see Figure 4a). Show that (5) holds in this case for all $t \geq 2$.

(ii) Now, assume that (5) holds for all subtrees of depth $\leq \tau$. As a first step, show that for any subtree $\mathcal{T}_{[vf]}$ of depth τ (see Figure 4b) it holds that, for all $t \geq \tau + 1$,

$$\mu_{v\to f}^{(t)}(x_v) \simeq P_{[vf]}(x_v).$$

(iii) Using the result of the previous step, show that, for any subtree $\mathcal{T}_{[fv]}$ of depth $\tau' = \tau + 2$, (5) holds for all $t \geq \tau'$ (see Figure 4c).

(iv) Show that, if the factor graph rooted at a has depth d, then, for all $t \geq d$,

$$\hat{P}^{(t)}(x_a) \simeq P(x_a).$$

(v) To generalize the statement above, assume that the factor graph has diameter $\mathcal{D}$, i.e., the maximum distance between any two nodes in the graph is $\mathcal{D}$. Show that for all $t \geq \mathcal{D}$ the estimated marginal of *any* variable v is exact, that is,

$$\hat{P}^{(t)}(x_v) \simeq P(x_v).$$