Homework 6 solutions

Intro to AI

Vibhav Gogate

Exercise

13.13

Let V be the statement that the patient has the virus, and A and B the statements that the medical tests A and B returned positive, respectively. The problem statement gives:

$$\begin{aligned} P(V) &= 0.01 \\ P(A|V) &= 0.95 \\ P(A|\neg V) &= 0.10 \\ P(B|V) &= 0.90 \\ P(B|\neg V) &= 0.05 \end{aligned}$$

The test whose positive result is more indicative of the virus being present is the one whose posterior probability, $P(V|A)$ or $P(V|B)$ is largest. One can compute these probabilities directly from the information given, finding that $P(V|A) = 0.0876$ and $P(V|B) = 0.1538$, so B is more indicative.

Equivalently, the questions is asking which test has the highest posterior odds ratio $P(V|A)/P(\neg V|A)$. From the odd form of Bayes theorem:

$$\frac{P(V|A)}{P(\neg V|A)} = \frac{P(A|V)}{P(A|\neg V)} \frac{P(V)}{P(\neg V)}$$

we see that the ordering is independent of the probability of V, and that we just need to compare the likelihood ratios $P(A|V)/P(A|\neg V) = 9.5$ and $P(B|V)/P(V|\neg V) = 18$ to find the answer.

Exercise

13.16 The basic axiom to use here is the definition of conditional probability:

a. We have

$$\mathbf{P}(A, B|E) = \frac{\mathbf{P}(A, B, E)}{\mathbf{P}(E)}$$

and

$$\mathbf{P}(A|B, E)\mathbf{P}(B|E) = \frac{\mathbf{P}(A, B, E)}{\mathbf{P}(B, E)}\frac{\mathbf{P}(B, E)}{\mathbf{P}(E)} = \frac{\mathbf{P}(A, B, E)}{\mathbf{P}(E)}$$

hence

$$\mathbf{P}(A, B|E) = \mathbf{P}(A|B, E)\mathbf{P}(B|E)$$

b. The derivation here is the same as the derivation of the simple version of Bayes' Rule on page 426. First we write down the dual form of the conditionalized product rule, simply by switching A and B in the above derivation:

$$\mathbf{P}(A, B|E) = \mathbf{P}(B|A, E)\mathbf{P}(A|E)$$

Therefore the two right-hand sides are equal:

$$\mathbf{P}(B|A, E)\mathbf{P}(A|E) = \mathbf{P}(A|B, E)\mathbf{P}(B|E)$$

Dividing through by $\mathbf{P}(B|E)$ we get

$$\mathbf{P}(A|B, E) = \frac{\mathbf{P}(B|A, E)\mathbf{P}(A|E)}{\mathbf{P}(B|E)}$$

14.15 This question definitely helps students get a solid feel for variable elimination. Students may need some help with the last part if they are to do it properly.

a.

$$
\begin{aligned}
& P(B|j,m) \\
&= \alpha P(B) \sum_e P(e) \sum_a P(a|b,e) P(j|a) P(m|a) \\
&= \alpha P(B) \sum_e P(e) \left[.9 \times .7 \times \begin{pmatrix} .95 & .29 \\ .94 & .001 \end{pmatrix} + .05 \times .01 \times \begin{pmatrix} .05 & .71 \\ .06 & .999 \end{pmatrix} \right] \\
&= \alpha P(B) \sum_e P(e) \begin{pmatrix} .598525 & .183055 \\ .59223 & .0011295 \end{pmatrix} \\
&= \alpha P(B) \left[.002 \times \begin{pmatrix} .598525 \\ .183055 \end{pmatrix} + .998 \times \begin{pmatrix} .59223 \\ .0011295 \end{pmatrix} \right] \\
&= \alpha \begin{pmatrix} .001 \\ .999 \end{pmatrix} \times \begin{pmatrix} .59224259 \\ .001493351 \end{pmatrix} \\
&= \alpha \begin{pmatrix} .00059224259 \\ .0014918576 \end{pmatrix} \\
&\approx \langle .284, .716 \rangle
\end{aligned}
$$

b. Including the normalization step, there are 7 additions, 16 multiplications, and 2 divisions. The enumeration algorithm has two extra multiplications.

c. To compute $\mathbf{P}(X_1|X_n = true)$ using enumeration, we have to evaluate two complete binary trees (one for each value of X_1), each of depth $n-2$, so the total work is $O(2^n)$. Using variable elimination, the factors never grow beyond two variables. For example, the first step is

$$
\begin{aligned}
& \mathbf{P}(X_1|X_n = true) \\
&= \alpha \mathbf{P}(X_1) \ldots \sum_{x_{n-2}} P(x_{n-2}|x_{n-3}) \sum_{x_{n-1}} P(x_{n-1}|x_{n-2}) P(X_n = true|x_{n-1}) \\
&= \alpha \mathbf{P}(X_1) \ldots \sum_{x_{n-2}} P(x_{n-2}|x_{n-3}) \sum_{x_{n-1}} \mathbf{f}_{X_{n-1}}(x_{n-1}, x_{n-2}) \mathbf{f}_{X_n}(x_{n-1}) \\
&= \alpha \mathbf{P}(X_1) \ldots \sum_{x_{n-2}} P(x_{n-2}|x_{n-3}) \mathbf{f}_{\overline{X_{n-1} X_n}}(x_{n-2})
\end{aligned}
$$

The last line is isomorphic to the problem with $n-1$ variables instead of n; the work done on the first step is a constant independent of n, hence (by induction on n, if you want to be formal) the total work is $O(n)$.

d. Here we can perform an induction on the number of nodes in the polytree. The base case is trivial. For the inductive hypothesis, assume that any polytree with n nodes can be evaluated in time proportional to the size of the polytree (i.e., the sum of the CPT sizes). Now, consider a polytree with $n+1$ nodes. Any node ordering consistent with the topology will eliminate first some leaf node from this polytree. To eliminate any

leaf node, we have to do work proportional to the size of its CPT. Then, *because the network is a polytree*, we are left with *independent* subproblems, one for each parent. Each subproblem takes total work proportional to the sum of its CPT sizes, so the total work for $n + 1$ nodes is proportional to the sum of CPT sizes.

See Exam solutions for problem 4.

Exercise 15.3

a. The chapter shows that $\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:t})$ can be computed as

$$\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:t}) = \alpha\,\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:k})\mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k) = \alpha\,\mathbf{f}_{1:k}\mathbf{b}_{k+1:t}$$

The forward recursion (Equation 15.3) shows that $\mathbf{f}_{1:k}$ can be computed from $\mathbf{f}_{1:k-1}$ and $\mathbf{e}_k$, which can in turn be computed from $\mathbf{f}_{1:k-2}$ and $\mathbf{e}_{k-1}$, and so on down to $\mathbf{f}_{1:0}$ and $\mathbf{e}_1$. Hence, $\mathbf{f}_{1:k}$ can be computed from $\mathbf{f}_{1:0}$ and $\mathbf{e}_{1:k}$. The backward recursion (Equation 15.7) shows that $\mathbf{b}_{k+1:t}$ can be computed from $\mathbf{b}_{k+2:t}$ and $\mathbf{e}_{k+1}$, which in turn can be computed from $\mathbf{b}_{k+3:t}$ and $\mathbf{e}_{k+2}$, and so on up to $\mathbf{b}_{h+1:t}$ and $\mathbf{e}_h$. Hence, $\mathbf{b}_{k+1:t}$ can be computed from $\mathbf{b}_{h+1:t}$ and $\mathbf{e}_{k+1:h}$. Combining these two, we find that $\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:t})$ can be computed from $\mathbf{f}_{1:0}$, $\mathbf{b}_{h+1:t}$, and $\mathbf{e}_{1:h}$.

b. The reasoning for the second half is essentially identical: for k between h and t, $\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:t})$ can be computed from $\mathbf{f}_{1:h}$, $\mathbf{b}_{t+1:t}$, and $\mathbf{e}_{h+1:t}$.

c. The algorithm takes 3 arguments: an evidence sequence, an initial forward message, and a final backward message. The forward message is propagated to the halfway point and the backward message is propagated backward. The algorithm then calls itself recursively on the two halves of the evidence sequence with the appropriate forward and backward messages. The base case is a sequence of length 1 or 2.

d. At each level of the recursion the algorithm traverses the entire sequence, doing $O(t)$ work. There are $O(\log_2 t)$ levels, so the total time is $O(t \log_2 t)$. The algorithm does a depth-first recursion, so the total space is proportional to the depth of the stack, i.e., $O(\log_2 t)$. With n islands, the recursion depth is $O(\log_n t)$, so the total time is $O(t \log_n t)$ but the space is $O(n \log_n t)$.