Informatics 2D: Reasoning and Agents

Alex Lascarides

School of informatics
THE UNIVERSITY OF EDINBURGH

Lecture 24: Exact Inference in Bayesian Networks

Where are we?

Last time ...

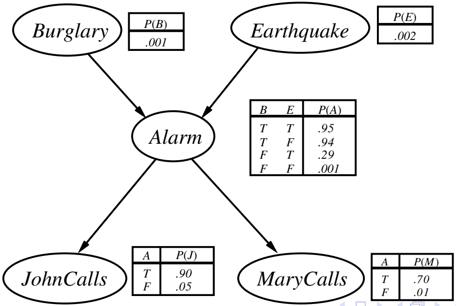
- Introduced Bayesian networks
- Allow for compact representation of JPDs
- Methods for efficient representations of CPTs
- But how hard is inference in BNs?

Today ...

- **Inference in Bayesian networks**

Example BN

- New burglar alarm has been fitted, fairly reliable but sometimes reacts to earthquakes
- Neighbours John and Mary promise to call when they hear alarm
- John sometimes mistakes phone for alarm, and Mary listens to loud music and sometimes doesn't hear alarm



Inference in BNs

- Basic task: compute posterior distribution for set of **query variables** given some observed **event** (i.e. assignment of values to **evidence variables**)
- Formally: determine $\mathrm{P}(X|\mathrm{e})$ given query variables X, evidence variables E (and non-evidence or **hidden** variables Y)
- Example: $\mathrm{P}(Burglary|JohnCalls = true, MaryCalls = true) = \langle 0.284, 0.716 \rangle$
- First we will discuss exact algorithms for computing posterior probabilities then approximate methods later

Inference by enumeration

- We have seen that any conditional probability can be computed from a full JPD by summing terms
- $\mathrm{P}(X|\mathrm{e}) = \alpha \mathrm{P}(X,\mathrm{e}) = \alpha \sum_{\mathrm{y}} \mathrm{P}(X,\mathrm{e},\mathrm{y})$
- Since BN gives complete representation of full JPD, we must be able to answer a query by computing sums of products of conditional probabilities from the BN
- Consider query $\mathrm{P}(Burglary|JohnCalls = true, MaryCalls = true) = \mathrm{P}(B|j,m)$
- $\mathrm{P}(B|j,m) = \alpha \mathrm{P}(B,j,m) = \alpha \sum_{e} \sum_{a} \mathrm{P}(B,e,a,j,m)$

Inference by enumeration

- Recall $P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i | parents(X_i))$
- We can use CPTs to simplify this exploiting BN structure
- For *Burglary* = *true*:

$$P(b|j,m) = \alpha \sum_e \sum_a P(b)P(e)P(a|b,e)P(j|a)P(m|a)$$

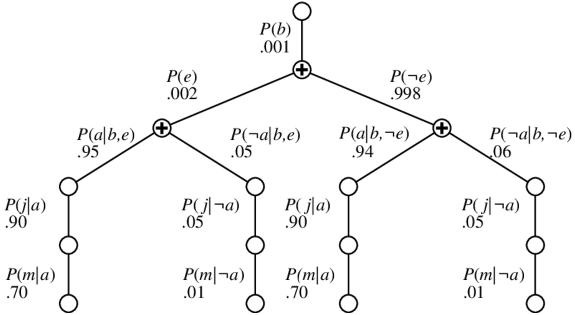
- But we can improve efficiency of this by moving terms outside that don't depend on sums

$$P(b|j,m) = \alpha P(b) \sum_e P(e) \sum_a P(a|b,e)P(j|a)P(m|a)$$

- To compute this, we need to loop through variables in order and multiply CPT entries; for each summation we need to loop over variable's possible values

The variable enumeration algorithm

- Evaluation of expression shown in the following tree:



- But this enumeration method makes you compute the same thing several times;
  e.g. $P(j|a)P(m|a)$ and $P(j|\neg a)P(m|\neg a)$ for each value of e

The variable elimination algorithm

- Idea of **variable elimination**: avoid repeated calculations
- Basic idea: store results after doing calculation once
- Works bottom-up by evaluating subexpressions
- Assume we want to evaluate

$$\mathrm{P}(B|j,m) = \alpha \underbrace{\mathrm{P}(B)}_{\mathrm{f}_1(B)} \sum_e \underbrace{\mathrm{P}(e)}_{\mathrm{f}_2(E)} \sum_a \underbrace{\mathrm{P}(a|B,e)}_{\mathrm{f}_3(A,B,E)} \underbrace{P(j|a)}_{\mathrm{f}_4(A)} \underbrace{P(m|a)}_{\mathrm{f}_5(A)}$$

- We've annotated each part with a **factor**.
- A factor is a **matrix**, indexed with its argument variables. E.g:
  - Factor $\mathrm{f}_5(A)$ corresponds to $P(m|a)$ and depends just on A because m is fixed (it's a 2×1 matrix).

$$\mathrm{f}_5(A) = \langle P(m|a), P(m|\neg a) \rangle$$

  - $\mathrm{f}_3(A,B,E)$ is a $2 \times 2 \times 2$ matrix for $\mathrm{P}(a|B,e)$

The variable elimination algorithm

$\mathbf{P}(B|j,m) = \alpha \mathrm{f}_1(B) \times \sum_e \mathrm{f}_2(E) \sum_a \mathrm{f}_3(A,B,E) \times \mathrm{f}_4(A) \times \mathrm{f}_5(A)$

- Summing out A produces a 2×2 matrix (via **pointwise product**):
  $$\begin{aligned} \mathrm{f}_6(B,E) = & \sum_a \mathrm{f}_3(A,B,E) \times \mathrm{f}_4(A) \times \mathrm{f}_5(A) \\ = & (\mathrm{f}_3(a,B,E) \times \mathrm{f}_4(a) \times \mathrm{f}_5(a)) + \\ & (\mathrm{f}_3(\neg a,B,E) \times \mathrm{f}_4(\neg a) \times \mathrm{f}_5(\neg a)) \end{aligned}$$
- So now we have
  $\mathbf{P}(B|j,m) = \alpha \mathrm{f}_1(B) \times \sum_e \mathrm{f}_2(E) \times \mathrm{f}_6(B,E)$
- Sum out E in the same way:
  $\mathrm{f}_7(B) = (\mathrm{f}_2(e) \times \mathrm{f}_6(B,e)) + (\mathrm{f}_2(\neg e) \times \mathrm{f}_6(B,\neg e))$
- Using $\mathrm{f}_1(B) = \mathbf{P}(B)$, we can finally compute
  $$\mathbf{P}(B|j,m) = \alpha \mathrm{f}_1(B) \times \mathrm{f}_7(B)$$
- Remains to define pointwise product and summing out

An example

- Pointwise product yields product for union of variables in its arguments:

$$f(X_1 \ldots X_i, Y_1 \ldots Y_j, Z_1 \ldots Z_k) = f_1(X_1 \ldots X_i, Y_1 \ldots Y_j) f_2(Y_1 \ldots Y_j, Z_1 \ldots Z_k)$$

A	B	$f_1(A,B)$	B	C	$f_2(B,C)$	A	B	C	$f(A,B,C)$
T	T	0.3	T	T	0.2	T	T	T	0.3×0.2
T	F	0.7	T	F	0.8	T	T	F	0.3×0.8
F	T	0.9	F	T	0.6	T	F	T	0.7×0.6
F	F	0.1	F	F	0.4	T	F	F	0.7×0.4
						F	T	T	0.9×0.2
						F	T	F	0.9×0.8
						F	F	T	0.1×0.6
						F	F	F	0.1×0.4

- For example $f(T,T,F) = f_1(T,T) \times f_2(T,F)$

An example

- Summing out is similarly straightforward
- Trick: any factor that does not depend on the variable to be summed out can be moved outside the summation process
- For example

$$\sum_e \mathrm{f}_2(E) \times \mathrm{f}_3(A,B,E) \times \mathrm{f}_4(A) \times \mathrm{f}_5(A)$$
$$= \mathrm{f}_4(A) \times \mathrm{f}_5(A) \times \sum_e \mathrm{f}_2(E) \times \mathrm{f}_3(A,B,E)$$

- Matrices are only multiplied when we need to sum out a variable from the accumulated product

Another Example: $\mathbf{P}(J|b) = \langle P(j|b), P(\neg j|b)\rangle$

$$
\begin{aligned}
\mathbf{P}(J|b) &= \alpha \textstyle\sum_e \sum_a \sum_m \mathbf{P}(J,b,e,a,m) && \text{prod., marg.}\\
&= \alpha \textstyle\sum_e \sum_a \sum_m P(b)P(e)P(a|b,e)\mathbf{P}(J|a)P(m|a) && \text{cond. indep.}\\
&= \alpha' \textstyle\sum_e \underbrace{P(e)}_{f_1(E)} \sum_a \underbrace{P(a|b,e)}_{f_2(A,E)} \underbrace{\mathbf{P}(J|a)}_{f_3(J,A)} \underbrace{\sum_m P(m|a)}_{=1} && \text{move terms}\\
&= \alpha' \textstyle\sum_e \underset{2\times 1}{f_1(E)} \sum_a \underset{2\times 2}{f_2(A,E)} \; \underset{2\times 2}{f_3(J,A)}\\
&= \alpha' \textstyle\sum_e \underset{2\times 1}{f_1(E)} \; \underset{2\times 2}{f_4(J,E)}\\
&= \alpha' f_5(J)
\end{aligned}
$$

Can eliminate all variables that aren't ancestors of query or evidence variables!

Summary

- Inference in Bayesian Networks
- Exact methods: enumeration, variable elimination algorithm
- Computationally intractable in the worst case
- Next time: **Approximate inference in Bayesian Networks**