Informatics 2D: Reasoning and Agents

Alex Lascarides

School of Informatics
The University of Edinburgh

Lecture 24a: Exact Inference in Bayesian Networks Variable Enumeration

Where are we?

Last time ...

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

Today ...

- **Inference in Bayesian networks**

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 $\mathbf{P}(X|\mathbf{e})$ given query variables $\mathbf{X}$, evidence variables $\mathbf{E}$ (and non-evidence or **hidden** variables $\mathbf{Y}$)
- Example: $\mathbf{P}(Burglary|JohnCalls = true, MaryCalls = true) = \langle 0.284, 0.716 \rangle$
- 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
- $\mathbf{P}(X|\mathbf{e}) = \alpha\mathbf{P}(X,\mathbf{e}) = \alpha\sum_{\mathbf{y}}\mathbf{P}(X,\mathbf{e},\mathbf{y})$
- 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 $\mathbf{P}(Burglary|JohnCalls = true, MaryCalls = true) = \mathbf{P}(B|j,m)$
- $\mathbf{P}(B|j,m) = \alpha\mathbf{P}(B,j,m) = \alpha\sum_e\sum_a\mathbf{P}(B,e,a,j,m)$

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*:

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

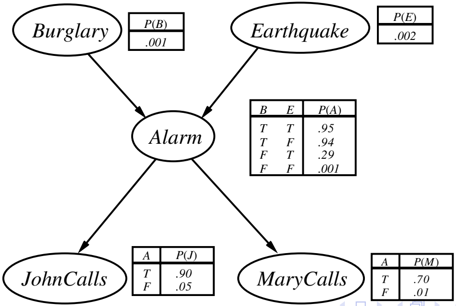
- But we can improve efficiency of this by moving terms outside that don't depend on sums

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

- 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

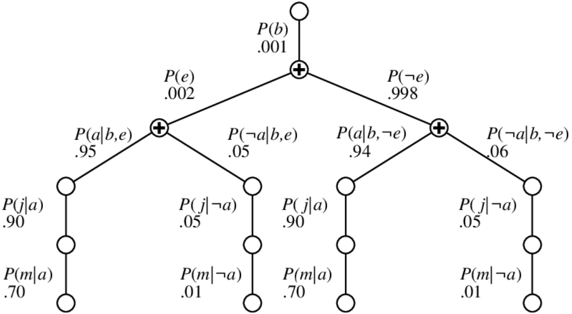
Example

- 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



The variable enumeration algorithm

- Enumeration method is computationally quite hard.
- You often 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
- Evaluation of expression shown in the following tree:



Summary

- Variable enumeration: an exact inference in BNs
- Needless repetition of some calculations
- Next time: **A (slightly) more efficient exact inference method**