Inf2D-Reasoning and Agents
Spring 2017

Lecture (2)4:– Exact Inference with Bayesian Networks

Peggy Seriès, pseries@inf.ed.ac.uk

Based on previous slides by A. Lascarides
Where are we?

- 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
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 $P(X|e)$ given query variables $X$, evidence variables $E$ (and non-evidence or hidden variables $Y$)

• Example:
  $P(\text{Burglary}|\text{JohnCalls} = \text{true}, \text{MaryCalls} = \text{true}) = ?$

First we will discuss exact algorithms for computing posterior probabilities then approximate methods late
Inference by enumeration

- We have seen that any conditional probability can be computed from a full JPD by summing terms:
  \[ P(X|e) = \alpha P(X, e) = \alpha \sum_y P(X, e, y) \]

- 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
  \[ P(\text{Burglary}|\text{JohnCalls} = \text{true}, \text{MaryCalls} = \text{true}) = P(B|j, m) \]

  \[ P(B|j, m) = \alpha P(B, j, m) = \alpha \sum_e \sum_a P(B, e, a, j, m) \]
  \[ = \alpha \sum_e \sum_a P(b)P(e)P(a|b, e)P(j|a)P(m|a) \]
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
Inference by enumeration

- \( P(B|j, m) \)
- \( = \alpha P(B, j, m) \)
- \( = \alpha \sum_e \sum_a P(B, e, a, j, m) \) marginalisation
- \( = \alpha \sum_e \sum_a P(B)P(e)P(a|B, e)P(j|a)P(m|a) \) using BN
- we can improve efficiency of this by moving terms outside that don’t depend on sums
  \[ \begin{align*}
P(b|j, m) &= \alpha P(b) \sum_e P(e) \sum_a P(a|b, e)P(j|a)P(m|a) \\
P(b|j,m) &= \alpha P(b)(P(e)[P(a|b, e)P(j|a)P(m|a)+P(\neg a|b, e)P(j|\neg a)P(m|\neg a)] \\
+P(\neg e)[(P(a|b, \neg e)P(j|a)P(m|a)+P(\neg a|b, \neg e)P(j|\neg a)P(m|\neg a)]
\end{align*} \]
Inference by enumeration

• 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.

• 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 $e$.

• Recursive depth-first enumeration: $O(n)$ space, $O(d^n)$ time.
Inference by enumeration

• Evaluation of expression $P(b|j, m) = \alpha P(b)\left[(P(e)P(a|b, e)P(j|a)P(m|a)+P(\neg a|b, e)P(j|\neg a)P(m|\neg a))+P(\neg e)(P(a|b, \neg e)P(j|a)P(m|a)+P(\neg a|b, \neg e)P(j|\neg a)P(m|\neg a))\right]$ shown in the following tree:
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:

\[
P(B | j, m) = \alpha \sum_e P(e) \sum_a P(a | B, e) P(j | a) P(m | a)\]

\[
f_1(B) f_2(E) f_3(A, B, E) f_4(A) f_5(A)
\]

- We’ve annotated each part with a factor.
The variable elimination algorithm

\[ P(B|j, m) = \alpha \frac{P(B) \sum_e P(e) \sum_a P(a|B, e) P(j|a) P(m|a)}{f_1(B) f_2(E) f_3(A,B,E) f_4(A) f_5(A)} \]

- A factor is a **matrix**, indexed with its argument variables.
- E.g: Factor \( f_5(A) \) corresponds to \( P(m|a) \) and depends just on \( A \) because \( m \) is fixed — it’s a \( 2 \times 1 \) matrix:
  \[ f_5(A) = <P(m|a), P(m|\neg a)> = <0.70, 0.01> \]
- \( f_4(A) = <P(j|a), P(j|\neg a)> = <0.90, 0.05> \)
- \( f_3(A,B, E) \) is a \( 2 \times 2 \times 2 \) matrix for \( P(a|B, e) \)
  its “first” element is \( p(a|b,e)=0.95 \) and “last” \( P(\neg a|\neg b, \neg e)=0.999 \)
The variable elimination algorithm

\[ P(B|j, m) = \alpha f_1(B) \sum_e P(e) \sum_a P(a|B, e) P(j|a) P(m|a) \]

\[ f_6(B, E) = \sum_a f_3(A, B, E) \times f_4(A) \times f_5(A) \]
\[ = (f_3(a, B, E) \times f_4(a) \times f_5(a)) + (f_3(\neg a, B, E) \times f_4(\neg a) \times f_5(\neg a)) \]

with \( f_3(a, B, E) = (P(a|b,e), P(a|\neg b,e); P(a|b,\neg e), P(a|\neg b,\neg e)) \)

• Eliminate/Summing out \( A \) produces a \( 2 \times 2 \) matrix (via pointwise product):

\[ f_6(B, E) = \sum_a f_3(A, B, E) \times f_4(A) \times f_5(A) \]

• So now we have \( P(B|j, m) = \alpha f_1(B) \times \sum_e f_2(E) \times f_6(B, E) \)

• Eliminating \( E \) in the same way:

\[ f_7(B) = (f_2(e) \times f_6(B, e)) + (f_2(\neg e) \times f_6(B, \neg e)) \]

• Using \( f_1(B) = P(B) \), we can finally compute \( P(B|j, m) = \alpha f_1(B) \times f_7(B) \)

• Remains to define point-wise product and summing out
Pointwise Products

• Pointwise product of 2 factors \( f_1 \) and \( f_2 \) yields a new factor \( f \)
• whose variables are the union of the variables in \( f_1 \) and \( f_2 \)
• and whose elements are given by the product of the corresponding elements in the 2 factors.

\[
\begin{align*}
  f_1(X,Y) \cdot f_2(Y,Z) &= f(X,Y,Z) \\
  f_1(T,T) \cdot f_2(T,F) &= \text{For example: } f(T,T,F) = f_1(T,T) \times f_2(T,F)
\end{align*}
\]
Summing out

- **Summing out** is similarly straightforward
- Summing out is done by adding the sub matrices formed by fixing the variable to each of its values in turn.

- Trick: any factor that does not depend on the variable to be summed out

\[
\sum_{e} f_2(E) \times f_3(A, B, E) \times f_4(A) \times f_5(A)
\]

\[
= f_4(A) \times f_5(A) \times \sum_{e} f_2(E) \times f_3(A, B, E)
\]

- Matrices are only multiplied when we need to sum out a variable from the accumulated product
Another example: \( P(J|b) = <P(j|b), P(\neg j|b)> \)

- \( P(J|b) = \alpha \sum_e \sum_a \sum_m P(J, b, e, a, m) \)
Another example: \( P(J|b) = <P(j|b), P(\neg j|b)> \)

\[
\begin{align*}
\cdot \ P(J|b) & = \alpha \sum_e \sum_a \sum_m P(J, b, e, a, m) \\
& = \alpha \sum_e \sum_a \sum_m P(b)P(e)P(a|b, e)P(J|a)P(m|a)
\end{align*}
\]
Another example: \( P(J|b) = \langle P(j|b), P(\neg j|b) \rangle \)

- \( P(J|b) \)
  \[
  = \alpha \sum_e \sum_a \sum_m P(J, b, e, a, m) \\
  = \alpha \sum_e \sum_a \sum_m P(b)P(e)P(a|b, e)P(J|a)P(m|a) \\
  = \alpha' \sum_e \underbrace{P(e)}_{f_1(E)} \sum_a \underbrace{P(a|b, e)}_{f_2(A, E)} \underbrace{P(J|a)}_{f_3(J, A)} \sum_m P(m|a)
  \]
Another example: \( P(J|b) = \langle P(j|b), P(\neg j|b) \rangle \)

- \( P(J|b) = \alpha \sum_e \sum_a \sum_m P(J, b, e, a, m) \)
  \[= \alpha \sum_e \sum_a \sum_m P(b) P(e) P(a|b, e) P(J|a) P(m|a) \]
  \[= \alpha' \sum_e P(e) \sum_a P(a|b, e) \sum_m P(J|a) \sum_m P(m|a) \]
  \[= 1 \]

- **M was irrelevant** to this query
  The result of this query is unchanged if we remove *MaryCalls* alltogether

- In general, every variable that is not an accents of a query variable or evidence variable is irrelevant to the query
Inference in Bayesian Networks

Exact methods: enumeration, variable elimination algorithm

In general, the complexity of exact inference will depend strongly on the structure of the network:

- Computationally intractable in the worst case
- Linear time in the best cases (tree structures) (as a function of number of CPT entries)

Next time: Approximate inference in Bayesian Networks