Lecture 7: Exact inference: Variable Elimination

- Given a Bayes net, what kinds of questions can we ask?
- Complexity of inference
- Variable elimination algorithm

Queries

Graphical models (directed or undirected) can answer questions about the underlying probability distribution:

- Conditional or unconditional probability queries:
  - What is the probability of a given value assignment for a subset of variables Y?
  - What is the probability of different value assignments for query variables Y given evidence about variables Z? I.e. compute $P(Y|Z=z)$
- Maximum a posteriori (MAP) queries: given evidence $Z=z$, find the most likely assignment of values to the query variables Y:

$$MAP(Y|Z=z) = \arg\max_y P(Y=y|Z=z)$$

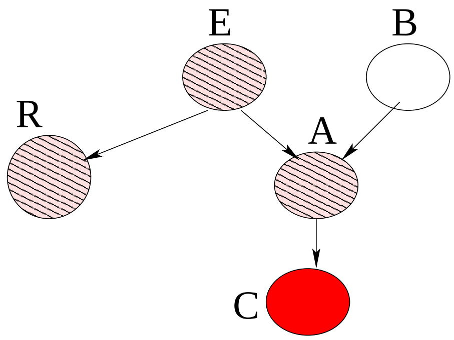
Examples of MAP queries

- In speech recognition, given a speech signal, one can attempt to reconstruct the most likely sequence of words that could have generated the signal.
- In classification, given the training data and a new example, we want to determine the most probable class label of the new example.

Complexity of inference

- Given a Bayesian network and a random variable X, deciding whether $P(X = x) > 0$ is NP-hard.
- This implies that there is no general inference procedure that will work efficiently for all network configurations
- But for particular families of networks, inference can be done efficiently.

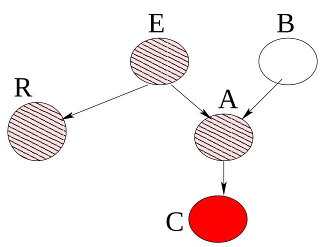
Example



p(B|C=t)=?

$$p(B|C=t) = \frac{p(B, C=t)}{p(C=t)}$$

Naive solution



p(B|C=t)=?

$$\begin{aligned} p(B, C=t) &= \sum_{a,r,e} p(A=a, R=r, E=e, B, C=t) \\ &= \sum_{a,r,e} p(r|e)p(e)p(a|e,B)p(C=t|a) \end{aligned}$$

and same for computing $p(C=t)$

A better solution

- Let's re-arrange the sums slighty:

$$\begin{aligned} p(B, C=t) &= \sum_{a,r,e} p(r|e)p(e)p(a|e,B)p(C=t|a) \\ &= \sum_{a,e} p(e)p(a|e,B)p(C=t|a) \sum_r p(r|e) \end{aligned}$$

- Notice that $\sum_r p(r|e) = 1$! But let's ignore that for the moment. We can call $\sum_r p(r|e) = m_R(e)$ (because it was obtained by summing out over R and only depends on e).
- Now we have:

$$p(B, C=t) = \sum_a \sum_e p(e)p(a|e,B)p(C=t|a)m_R(e)$$

  and we can pick another variable (A or E) to do the same again.
- Instead of $O(2^n)$ factors, we have to sum over $O(n \cdot 2^k)$ factors

Basic idea of variable elimination

- We impose an ordering over the variables, with the query variable coming *last*
- We maintain a list of "factors", which depend on given variables
- We sum over the variables in the order in which they appear in the list
- We **memorize** the result of intermediate computations
- This is a kind of *dynamic programming*

A bit of notation

- Let X_i an evidence variable with observed value $\hat{x_i}$
- Let the **evidence potential** be an indicator function:

$$\delta(x_i, \hat{x_i}) = 1 \text{ iff } X_i = \hat{x_i}$$

This way, we can turn conditionals into sums as well, e.g.

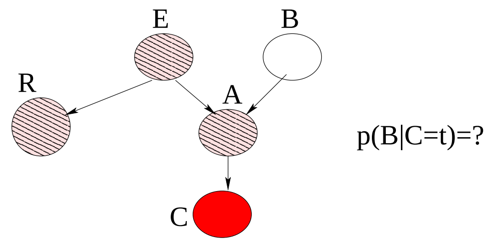
$$p(r|E = t) = \sum_e p(r|e)\delta(e, t)$$

- This is convenient for notation, but in practice we would take "slices" through the probability tables instead.

Variable elimination algorithm

1. Pick a variable ordering with Y at the end of the list
2. Initialize the *active factor list*:
   - with the CPDs in a Bayes net
   - with the potentials in a Markov random field
3. Introduce the *evidence* by adding to the active factor list the evidence potentials $\delta(e, \hat{e})$, for all the variables in E
4. For $i = 1$ to n
   (a) Take the next variable X_i from the ordering.
   (b) Take all the factors that have X_i as an argument off the active factor list, and multiply them, then sum over all values of X_i, creating a new factor m_{X_i}
   (c) Put m_{X_i} on the active factor list

Example



1. Pick a variable ordering: R, E, C, A, B.
2. Initialize the active factor list and introduce the evidence:

   List: $p(R|E), p(E), p(B), p(A|E,B), p(C|A), \delta(C,t)$

3. Eliminate R: take $p(R|E)$ off the list, compute $m_R(e) = \sum_r p(r|e)$.

   List: $p(E), p(B), p(A|E,B), p(C|A), \delta(C,t), m_R(E)$

Example (continued)

4. Eliminate E: $m_E(A,B) = \sum_e p(e)p(a|e,b)m_R(e)$

   List: $p(B), p(C|A), \delta(C,t), m_E(A,B)$

5. Eliminate C: $m_C(a) = \sum_c p(c|a)\delta(C,t)$

   List: $p(B), m_E(A,B), m_C(A)$

6. Eliminate A: $m_A(b) = \sum_a m_E(a,b)m_C(a)$

   List: $p(B), m_A(B)$

7. Eliminate B: $m_B = \sum_b p(b)m_A(b)$

   List: m_B

   This is the answer we needed!

Complexity of variable elimination

- We need at most $O(n)$ multiplications to create one entry in a factor (where n is the total number of variables)
- If k is the maximum number of values that a variable can take, a factor depending on k variables will have $O(k^m)$ entries
- So it is important to have *small factors*!
- But the size of the factors depends on the ordering of the variables!
- Choosing an optimal ordering is NP-complete (more on this later)