Variable Elimination: Basic Ideas

Sargur Srihari
srihari@cedar.buffalo.edu

Topics

1. Types of Inference Algorithms
2. Variable Elimination: the Basic ideas
3. Variable Elimination
   - Sum-Product VE Algorithm
   - Sum-Product VE for Conditional Probabilities
4. Variable Ordering for VE

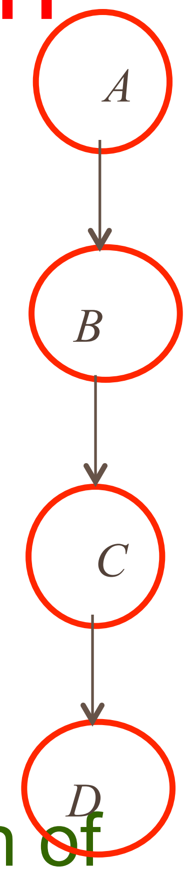
Inference Algorithms

- Types of inference algorithms
  1. Exact
     1. Variable Elimination
     2. Clique trees (Belief Propagation)
  2. Approximate
     1. Optimization
        1. Propagation with approximate messages
        2. Variational (analytical approximations)
     2. Particle-based (sampling)

Variable Elimination: Basic Ideas

- We begin with principles underlying exact inference in PGMs
- As we show, the same BN structure that allows compaction of complex distributions also helps support inference
- In particular we can use dynamic programming techniques to perform inference even for large and complex networks in reasonable time

Intuition for Variable Elimination



- Consider inference is a very simple BN

  $A \rightarrow B \rightarrow C \rightarrow D$

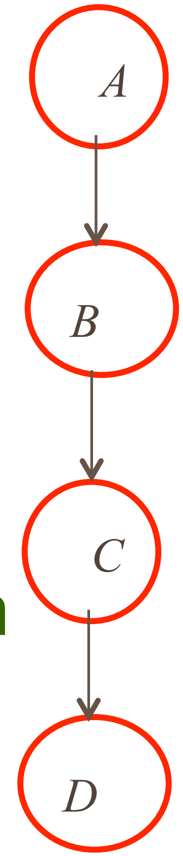
  – E.g., sequence of words
    - CPDs are first order word probabilities
- We consider phased computation
    - Probabilities of four words: *The, quick, brown, fox*
  – Use results of a previous phase in computation of next phase
  – Then reformulate this process in terms of a global computation on the joint distribution

Exact Inference: Variable Elimination

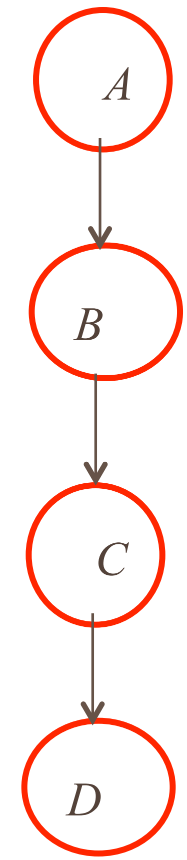
- To compute $P(B)$,
  - i.e., distribution of values b of B, we have

$$P(B) = \sum_a P(A,B) = \sum_a P(a)P(B \mid a)$$

  - required $P(a)$, $P(b|a)$ available in BN
- If A has k values and B has m values
  - For each b: k multiplications and k-1 addition
  - Since there are m values of B, process is repeated for each value of b:
    - this computation is $O(k \text{ x } m)$



Moving Down BN



- Assume we want to compute $P(C)$
- Using same analysis

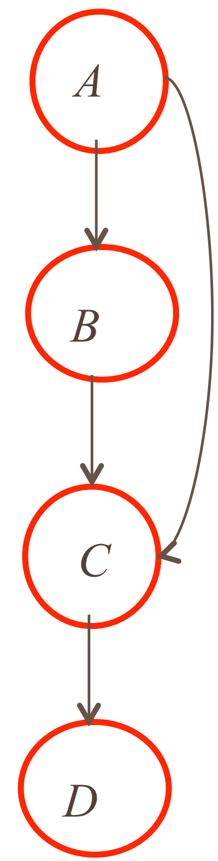
$$P(C) = \sum_b P(B,C) = \sum_b P(b)P(C \mid b)$$

  - $P(c|b)$ is given in CPD
  - But $P(B)$ is not given as network parameters
  - It can be computed using

$$P(B) = \sum_a P(A,B) = \sum_a P(a)P(B \mid a)$$

  - If B and C have k values each, complexity is $O(k^2)$

Computation depends on Structure



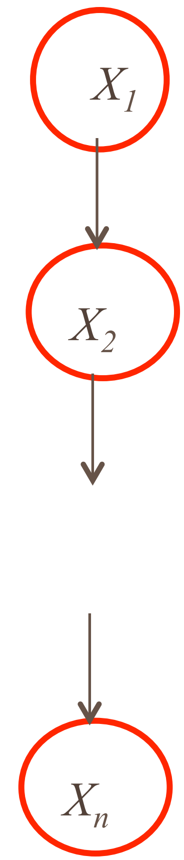
1. Structure of BN is critical for computation
   - If A had been a parent of C

$$P(C) = \sum_b P(b)P(C \mid b)$$

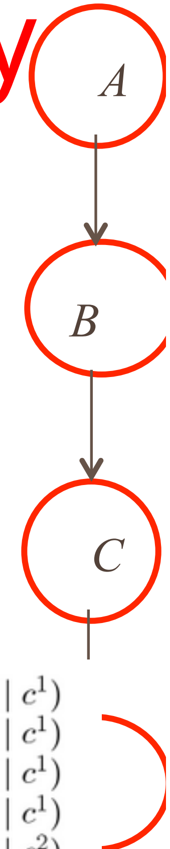
   - would not have sufficed
2. Algorithm does not compute single values but sets of values at a time
   - $P(B)$ over all possible values of B are used to compute $P(C)$

Complexity of General Chain

- In general, if we have $X_1 \rightarrow X_2 \rightarrow \ldots\ldots \rightarrow X_n$
- and there are k values of X_i, total cost is $O(nk^2)$
- Naïve evaluation
- Generate entire joint and summing it out
- Would generate k^n probabilities for the events $x_1, \ldots x_n$
- In this example, despite exponential size of joint distribution **we can do inference in linear time**



Insight that avoids exponentiality



- The joint probability decomposes as

$$P(A,B,C,D)=P(A)P(B|A)P(C|B)P(D|C)$$

  - To compute $P(D)$ we need to sum together all entries where $D=d^1$
    - And separately entries where $D=d^2$
  - Exact computation for $P(D)$ is

$$\begin{array}{lllll} & P(a^1) & P(b^1 \mid a^1) & P(c^1 \mid b^1) & P(d^1 \mid c^1) \\ + & P(a^2) & P(b^1 \mid a^2) & P(c^1 \mid b^1) & P(d^1 \mid c^1) \\ + & P(a^1) & P(b^2 \mid a^1) & P(c^1 \mid b^2) & P(d^1 \mid c^1) \\ + & P(a^2) & P(b^2 \mid a^2) & P(c^1 \mid b^2) & P(d^1 \mid c^1) \\ + & P(a^1) & P(b^1 \mid a^1) & P(c^2 \mid b^1) & P(d^1 \mid c^2) \\ + & P(a^2) & P(b^1 \mid a^2) & P(c^2 \mid b^1) & P(d^1 \mid c^2) \\ + & P(a^1) & P(b^2 \mid a^1) & P(c^2 \mid b^2) & P(d^1 \mid c^2) \\ + & P(a^2) & P(b^2 \mid a^2) & P(c^2 \mid b^2) & P(d^1 \mid c^2) \end{array}$$

$$\begin{array}{lllll} & P(a^1) & P(b^1 \mid a^1) & P(c^1 \mid b^1) & P(d^2 \mid c^1) \\ + & P(a^2) & P(b^1 \mid a^2) & P(c^1 \mid b^1) & P(d^2 \mid c^1) \\ + & P(a^1) & P(b^2 \mid a^1) & P(c^1 \mid b^2) & P(d^2 \mid c^1) \\ + & P(a^2) & P(b^2 \mid a^2) & P(c^1 \mid b^2) & P(d^2 \mid c^1) \\ + & P(a^1) & P(b^1 \mid a^1) & P(c^2 \mid b^1) & P(d^2 \mid c^2) \\ + & P(a^2) & P(b^1 \mid a^2) & P(c^2 \mid b^1) & P(d^2 \mid c^2) \\ + & P(a^1) & P(b^2 \mid a^1) & P(c^2 \mid b^2) & P(d^2 \mid c^2) \\ + & P(a^2) & P(b^2 \mid a^2) & P(c^2 \mid b^2) & P(d^2 \mid c^2) \end{array}$$

  - Examine summation
    - 3rd & 4th terms of first 2 terms:
    - $P(c^1|b^1)P(d^1|c^1)$
    - Modify to first compute
    - $P(a^1)P(b^1|a^1)+P(a^2)P(b^1|a^2)$
    - then multiply by common term

First Transformation of sum

- Same structure is repeated throughout table
- Performing the same transformation we get the summation for $P(D)$ as

$$
\begin{array}{llll}
 & (P(a^1)P(b^1 \mid a^1) + P(a^2)P(b^1 \mid a^2)) & P(c^1 \mid b^1) & P(d^1 \mid c^1) \\
+ & (P(a^1)P(b^2 \mid a^1) + P(a^2)P(b^2 \mid a^2)) & P(c^1 \mid b^2) & P(d^1 \mid c^1) \\
+ & (P(a^1)P(b^1 \mid a^1) + P(a^2)P(b^1 \mid a^2)) & P(c^2 \mid b^1) & P(d^1 \mid c^2) \\
+ & (P(a^1)P(b^2 \mid a^1) + P(a^2)P(b^2 \mid a^2)) & P(c^2 \mid b^2) & P(d^1 \mid c^2)
\end{array}
$$

$$
\begin{array}{llll}
 & (P(a^1)P(b^1 \mid a^1) + P(a^2)P(b^1 \mid a^2)) & P(c^1 \mid b^1) & P(d^2 \mid c^1) \\
+ & (P(a^1)P(b^2 \mid a^1) + P(a^2)P(b^2 \mid a^2)) & P(c^1 \mid b^2) & P(d^2 \mid c^1) \\
+ & (P(a^1)P(b^1 \mid a^1) + P(a^2)P(b^1 \mid a^2)) & P(c^2 \mid b^1) & P(d^2 \mid c^2) \\
+ & (P(a^1)P(b^2 \mid a^1) + P(a^2)P(b^2 \mid a^2)) & P(c^2 \mid b^2) & P(d^2 \mid c^2)
\end{array}
$$

  - Observe certain terms are repeated several times in this expression
  - $P(a^1)P(b^1|a^1)+P(a^2)P(b^1|a^2)$ and
  - $P(a^1)P(b^2|a^1)+P(a^2)P(b^2|a^2)$

    are repeated four times

2^{nd} & 3^{rd} transformation on the sum

- Defining τ_1: $Val(B) \rightarrow R$
  - where $\tau_1(b^1)$ and $\tau_1(b^2)$ are the two expressions, we get

$$
\begin{array}{lll}
 & \tau_1(b^1) & P(c^1 \mid b^1) & P(d^1 \mid c^1) \\
+ & \tau_1(b^2) & P(c^1 \mid b^2) & P(d^1 \mid c^1) \\
+ & \tau_1(b^1) & P(c^2 \mid b^1) & P(d^1 \mid c^2) \\
+ & \tau_1(b^2) & P(c^2 \mid b^2) & P(d^1 \mid c^2)
\end{array}
$$

$$
\begin{array}{lll}
 & \tau_1(b^1) & P(c^1 \mid b^1) & P(d^2 \mid c^1) \\
+ & \tau_1(b^2) & P(c^1 \mid b^2) & P(d^2 \mid c^1) \\
+ & \tau_1(b^1) & P(c^2 \mid b^1) & P(d^2 \mid c^2) \\
+ & \tau_1(b^2) & P(c^2 \mid b^2) & P(d^2 \mid c^2)
\end{array}
$$

– Can reverse the order of a sum and product

  - sum first, product next

$$
\begin{array}{lll}
 & (\tau_1(b^1)P(c^1 \mid b^1) + \tau_1(b^2)P(c^1 \mid b^2)) & P(d^1 \mid c^1) \\
+ & (\tau_1(b^1)P(c^2 \mid b^1) + \tau_1(b^2)P(c^2 \mid b^2)) & P(d^1 \mid c^2)
\end{array}
$$

$$
\begin{array}{lll}
 & (\tau_1(b^1)P(c^1 \mid b^1) + \tau_1(b^2)P(c^1 \mid b^2)) & P(d^2 \mid c^1) \\
+ & (\tau_1(b^1)P(c^2 \mid b^1) + \tau_1(b^2)P(c^2 \mid b^2)) & P(d^2 \mid c^2)
\end{array}
$$

Fourth Transformation of sum

- Again notice shared expressions that are better computed once and used multiple times
  - We define τ_2: $Val(C) \rightarrow R$

$$\tau_2(c^1)=\tau_1(b^1)P(c^1|b^1)+\tau_1(b^2)P(c^1|b^2)$$

$$\tau_2(c^2)=\tau_1(b^1)P(c^2|b^1)+\tau_1(b^2)P(c^2|b^2)$$

$$\begin{array}{lll} & \tau_2(c^1) & P(d^1 \mid c^1) \\ + & \tau_2(c^2) & P(d^1 \mid c^2) \end{array}$$

$$\begin{array}{lll} & \tau_2(c^1) & P(d^2 \mid c^1) \\ + & \tau_2(c^2) & P(d^2 \mid c^2) \end{array}$$

Summary of computation

- We begin by computing $\tau_1(B)$
- Requires *4* multiplications and *2* additions
- Using it we can compute $\tau_2(C)$ which also requires *4* multis and *2* adds
- Finally we compute $P(D)$ at same cost
- Total no of ops is *18*
- Joint distribution requires *16* x *3=48* mps and *14* adds

Computation Summary

- Transformation we have performed has steps

$$P(D) = \sum_C \sum_B \sum_A P(A)P(B \mid A)P(C \mid B)P(D \mid C)$$

- We push the first summation resulting in

$$P(D) = \sum_C P(D \mid C) \sum_B P(C \mid B) \sum_A P(A)P(B \mid A)$$

- We compute the product $\psi_1(A,B) = P(A)P(B|A)$ and sum out A to obtain the function $\tau_1(B) = \sum_A \psi_1(A,B)$
  - For each value of b, we compute

$$\tau_1(b) = \sum_A \psi_1(A,b) = \sum_A P(A)P(b \mid A)$$

- We then continue

$$\psi_2(B,C) = \tau_1(B)P(C \mid B)$$

$$\tau_2(C) = \sum_B \psi_2(B,C)$$

  - Resulting $\tau_2(C)$ is used to compute $P(D)$

Computation is Dynamic Programming

- Naiive way for $P(D)=\sum_C\sum_B\sum_A P(A)P(B\mid A)P(C\mid B)P(D\mid C)$

  would have us compute every

  $$P(b)=\sum_A P(A)P(b\mid A)$$

  – many times, once for every value of C and D
- For a chain of length n this would be computed exponentially many times
- Dynamic Programming inverts order of computation– performing it inside out rather than outside in
  – First computing once for all values in $\tau_1(B)$, that allows us to compute $\tau_2(C)$ once for all, etc.

Ideas that prevented exponential blowup

- Because of structure of BN, some subexpressions depend only on a small no. of variables
- By computing and caching these results we can avoid generating them exponential no. of times