Probabilistic Graphical Models

Raquel Urtasun and Tamir Hazan

TTI Chicago

April 18, 2011

Inference: conditional probabilities

- Today we will look into inference in exact inference in graphical models.
- In particular, we will look into variable elimination.
- The factorization of the network is going to be critical in our ability to perform inference.
- We will focus on conditional probability queries

$$p(\mathbf{Y}|\mathbf{E} = \mathbf{e}) = \frac{P(\mathbf{Y}, \mathbf{e})}{P(\mathbf{e})}$$

- Let $\mathbf{W} = \mathcal{X} - \mathbf{Y} - \mathbf{E}$ be the random variables that are neither the query nor the evidence. Each of this joint distributions can be computed by marginalizing the other variables.

$$p(\mathbf{Y}, \mathbf{e}) = \sum_{\mathbf{w}} P(\mathbf{Y}, \mathbf{e}, \mathbf{w})$$

and the probability of the evidence is

$$P(\mathbf{e}) = \sum_{\mathbf{y},\mathbf{w}} P(\mathbf{y}, \mathbf{e}, \mathbf{w})$$

Reuse of computation

- We can reuse the computation as follows

$$P(\mathbf{e}) = \sum_{\mathbf{y},\mathbf{w}} P(\mathbf{y}, \mathbf{e}, \mathbf{w}) = \sum_{y} P(\mathbf{y}, \mathbf{e})$$

- We can now compute the conditional by dividing the probabilities

$$p(\mathbf{Y}|\mathbf{E} = \mathbf{e}) = \frac{P(\mathbf{Y}, \mathbf{e})}{P(\mathbf{e})}$$

- This process is taking the marginal probabilities $p(\mathbf{y}^1, \mathbf{e}), \cdots, p(\mathbf{y}^k, \mathbf{e})$ and renormalizing the entries to sum to 1.

Complexity of inference

- Summing up all the terms has an exponential number of computations.
- Worst case analysis, it is NP-hard.
- Approximate inference in the worst case is also NP-hard.
- It's the same in Bayesian networks and Markov networks.
- In practice there is hope, worst case is not what we care about!

Basic idea of variable elimination

- The structure of the graph helps inference.
- We can use dynamic programming to do efficient inference.
- Let's start with a simple chain $A \rightarrow B \rightarrow C \rightarrow D$.
- Let's assume we want to compute $P(B)$.
- With no assumption:

$$P(B) = \sum_a P(a)P(B|a)$$

- All this information in the Bayesian network: we have the CPD of $P(a)$ and $P(B|a)$.
- The same for $P(C)$

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

and the information in the CPD.

- This algorithm computes sets of values at a time, an entire distribution.

Complexity of a chain

- Example of a chain $X_1 \rightarrow X_2 \rightarrow \cdots \rightarrow X_n$, and each node has k values.
- We can compute at each step

$$P(X_{i+1}) = \sum_{x_i} P(X_{i+1}|x_i)P(x_i)$$

- We need to multiply $P(x_i)$ with each CPD $P(X_{i+1}|X_i)$ for each value of x_i.
- $P(X_i)$ has k values, and the CPD $P(X_{i+1}|X_i)$ has k^2 values.
- k^2 multiplications and $k(k-1)$ additions.
- The cost of the total chain is $\mathcal{O}(nk^2)$.
- By comparison, generating the full joint and summing it up has complexity $\mathcal{O}(k^n)$.
- We have done inference over the joint without generating it explicitly.

Let's be a bit more explicit...

- The joint probability by the chain rule in BN is

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

- In order to compute $P(D)$ we have to sum up all the values

$$P(D) = \sum_{a,b,c} p(A, B, C, D)$$

Let's be a bit more explicit...

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

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

- There is structure on the summation, e.g., repeated $P(c^1|b^1)P(d^1|c^1)$.
- Let's modify the computation to first compute

$$P(a^1)P(b^1|a^1) + P(a^2)P(b^1|a^2)$$

Let's be a bit more explicit...

- Let's modify the computation to first compute

$$P(a^1)P(b^1|a^1) + P(a^2)P(b^1|a^2)$$

- Then we get

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

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

- Certain terms are repeated multiple times

$$P(a^1)P(b^1|a^1) + P(a^2)P(b^1|a^2)$$
$$P(a^1)P(b^2|a^1) + P(a^2)P(b^2|a^2)$$

- We define $\tau_1 : Val(B) \rightarrow \Re, \quad \tau_1(b^i) = P(a^1)P(b^i|a^1) + P(a^2)P(b^i|a^2)$

Let's be a bit more explicit...

- We now have

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

- We can once more reverse the order of the product and the sum and get

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

- We have other repetitive patterns.

Let's be a bit more explicit...

- We define $\tau_2 : Val(C) \to \Re$, with

$$\begin{aligned}
\tau_2(c^1) &= \tau_1(b^1)P(c^1|b^1) + \tau_1(b^2)P(c^1|b^2) \\
\tau_2(c^2) &= \tau_1(b^1)P(c^2|b^1) + \tau_1(b^2)P(c^2|b^2)
\end{aligned}$$

- Thus we can compute the joint $P(A, B, C, D)$ as

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

Even more explicit...

- The joint is

$$P(D) = \sum_{A,B,C} p(A,B,C,D) = \sum_{A,B,C} P(A)P(B|A)P(C|B)P(D|C)$$

- We can push the summation

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

- Let's call $\psi_1(A,B) = P(A)P(B|A)$ and $\tau_1(B) = \sum_A \psi_1(A,B)$.
- We can define $\psi_2(B,C) = \tau_1(B)P(C|B)$ and $\tau_2(C) = \sum_B \psi_1(B,C)$.
- This is $\tau_2(C)$ that we can use in the final computation.
- This procedure is dynamic programming: computation is inside out instead of outside in.

Summary

- Worst case analysis says that computing the joint is NP-hard.
- Even approximating it is NP-hard.
- In practice due to the structure of the Bayesian network some subexpressions in the joint depend only on a subset of variables.
- We can catch up computations that are otherwise computed exponentially many times.

Variable elimination

- We want to go beyond chains!
- We are going to look into Bayesian networks.
- Recall that a factor $\phi : Val(\mathbf{X}) \rightarrow \Re$ with scope $\mathbf{X}$.
- Variable elimination is going to manipulate factors.
- Let $\mathbf{X}$ be a set of variables, and $Y \notin \mathbf{X}$ a variable and $\phi(\mathbf{X}, Y)$ be a factor.
- We define **factor marginalization** to be a factor ψ over $\mathbf{X}$ such that

$$\psi(\mathbf{X}) = \sum_{Y} \phi(\mathbf{X}, Y)$$

- This is called summing out Y in ϕ

Variable elimination

- We only sum up the entries that **X** matches up

a^1	b^1	c^1	0.25
a^1	b^1	c^2	0.35
a^1	b^2	c^1	0.08
a^1	b^2	c^2	0.16
a^2	b^1	c^1	0.05
a^2	b^1	c^2	0.07
a^2	b^2	c^1	0
a^2	b^2	c^2	0
a^3	b^1	c^1	0.15
a^3	b^1	c^2	0.21
a^3	b^2	c^1	0.09
a^3	b^2	c^2	0.18

a^1	c^1	0.33
a^1	c^2	0.51
a^2	c^1	0.05
a^2	c^2	0.07
a^3	c^1	0.24
a^3	c^2	0.39

- Marginalizing a joint distribution $P(\mathbf{X}, \mathbf{Y})$ onto $\mathbf{X}$ in a BN corresponds to summing out the variables $\mathbf{Y}$ in the factor corresponding to P.

Some properties

- If we sum out all the variables in a normalized distribution, what do we get?
- If we sum out all the variables in an unnormalized distribution, what do we get?
- Important property is that sum and product are commutative, and the product is associative $(\phi_1\phi_2)\phi_3 = \phi_1(\phi_2\phi_3)$.
- Therefore, if $X \notin Scope(\phi_1)$ then

$$\sum_X (\phi_1\phi_2) = \phi_1 \sum_X \phi_2$$

Chain example again

- Let's look at the chain again

$$P(A, B, C, D) = \phi_A \phi_B \phi_C \phi_D$$

- The marginal distribution over D

$$\begin{aligned} P(D) &= \sum_{A,B,C} \phi_A \phi_B \phi_C \phi_D \\ &= \sum_C \left(\phi_D \sum_B \left(\phi_C \sum_A (\phi_B \phi_A) \right) \right) \end{aligned}$$

  where we have used the limited scope of the factors.

- Marginalizing involves taking the product of all CPDs and sum over all but the variables in the query.
- We can do this in any order we want; some more efficient than others.
- The **sum product** inference task is

$$\sum_Z \prod_{\phi \in \Phi} \phi$$

Sum-product variable elimination

- Effective as the scope is limited, we push in some of the summations.
- A simple instance of this is the **sum-product variable elimination algorithm**.
- Idea: We sum out variables one at a time.
  - When we do this, we multiply all the factors that have this variable as scope, generating a product factor.
  - We sum out the variable from this product factor, generating a new factor, which enters the set of factors to deal with.

Algorithm 9.1 Sum-Product Variable Elimination algorithm

Procedure Sum-Product-Variable-Elimination (
  Φ, // Set of factors
  $\boldsymbol{Z}$, // Set of variables to be eliminated
  $\prec$ // Ordering on $\boldsymbol{Z}$
)
  Let $Z_1, \ldots, Z_k$ be an ordering of $\boldsymbol{Z}$ such that
    $Z_i \prec Z_j$ iff $i < j$
  for $i = 1, \ldots, k$
    $\Phi \leftarrow$ Sum-Product-Eliminate-Var(Φ, Z_i)
  $\phi^* \leftarrow \prod_{\phi \in \Phi} \phi$
  return ϕ^*

Procedure Sum-Product-Eliminate-Var (
  Φ, // Set of factors
  Z // Variable to be eliminated
)
  $\Phi' \leftarrow \{\phi \in \Phi \ : \ Z \in Scope[\phi]\}$
  $\Phi'' \leftarrow \Phi - \Phi'$
  $\psi \leftarrow \prod_{\phi \in \Phi'} \phi$
  $\tau \leftarrow \sum_Z \psi$
  return $\Phi'' \cup \{\tau\}$

Sum-product variable elimination

- Theorem: Let $\mathbf{X}$ be a set of variables, and let Φ be a set of factors, such that for each $\phi \in \Phi$, $Scope(\phi) \subseteq \mathbf{X}$. Let $Y \subset \mathbf{X}$ be a set of query variables, and let $\mathbf{Z} = \mathbf{X} - \mathbf{Y}$. Then for every ordering $\prec$ over $\mathbf{Z}$, the Sum-Product-Variable-Elimination$(\Phi, \mathbf{Z}, \prec)$ returns a factor $\phi(\mathbf{Y})$ such that

$$\phi(\mathbf{Y}) = \sum_{\mathbf{Z}} \prod_{\phi \in \Phi} \phi$$

- We can apply this to a BN with variables $\mathbf{Y} = \{Y_1, \cdots, Y_k\}$, where Φ is all the CPDs

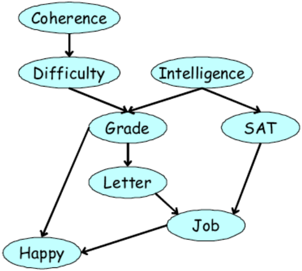
$$\Phi = \{\phi_{X_i}\}_{i=1}^{n} = \{P(X_i | Pa_{X_i})_{i=1}^{n}\}$$

  We apply the elimination algorithm to the set $\{Z_1, \cdots, Z_m\} = \mathcal{X} - \mathbf{Y}$.

- We can apply the same algorithm to a Markov network, where the factors are the clique potentials.

- For Markov networks, the procedure returns an unnormalized distribution. We need to renormalize.

Example of BN



- The joint distribution

$$p(C,D,I,G,S,L,H,J) = p(C)p(D|C)p(I)p(G|D,I)p(L|G)P(S|I)P(J|S,L)p(H|J,G)$$

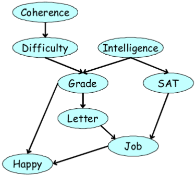
with factors

$$p(C,D,I,G,S,L,H,J) = \phi_c(C)\phi_D(C,D)\phi_I(I)\phi_G(G,D,I)\phi_L(L,G) \\ \phi_S(S,I)\phi_J(J,S,L)\phi_H(H,J,G)$$

- Let's do variable elimination with ordering $\{C,D,I,H,G,S,L\}$ on the board!

Elimination Ordering

- We can pick any order we want, but some orderings introduce factors with much larger scope.



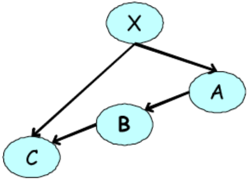
Step	Variable eliminated	Factors used	Variables involved	New factor
1	C	$\phi_C(C)$, $\phi_D(D,C)$	C, D	$\tau_1(D)$
2	D	$\phi_G(G,I,D)$, $\tau_1(D)$	G, I, D	$\tau_2(G,I)$
3	I	$\phi_I(I)$, $\phi_S(S,I)$, $\tau_2(G,I)$	G, S, I	$\tau_3(G,S)$
4	H	$\phi_H(H,G,J)$	H, G, J	$\tau_4(G,J)$
5	G	$\tau_4(G,J)$, $\tau_3(G,S)$, $\phi_L(L,G)$	G, J, L, S	$\tau_5(J,L,S)$
6	S	$\tau_5(J,L,S)$, $\phi_J(J,L,S)$	J, L, S	$\tau_6(J,L)$
7	L	$\tau_6(J,L)$	J, L	$\tau_7(J)$

- Alternative ordering...

Step	Variable eliminated	Factors used	Variables involved	New factor
1	G	$\phi_G(G,I,D)$, $\phi_L(L,G)$, $\phi_H(H,G,J)$	G, I, D, L, J, H	$\tau_1(I,D,L,J,H)$
2	I	$\phi_I(I)$, $\phi_S(S,I)$, $\tau_1(I,D,L,S,J,H)$	S, I, D, L, J, H	$\tau_2(D,L,S,J,H)$
3	S	$\phi_J(J,L,S)$, $\tau_2(D,L,S,J,H)$	D, L, S, J, H	$\tau_3(D,L,J,H)$
4	L	$\tau_3(D,L,J,H)$	D, L, J, H	$\tau_4(D,J,H)$
5	H	$\tau_4(D,J,H)$	D, J, H	$\tau_5(D,J)$
6	C	$\tau_5(D,J)$, $\phi_D(D,C)$	D, J, C	$\tau_6(D,J)$
7	D	$\tau_6(D,J)$	D, J	$\tau_7(J)$

Semantics of Factors

- In the previous example the factors were marginal or conditional probabilities, but this is not true in general.



- The result of eliminating X is not a marginal or conditional probability of the network

$$\tau(A, B, C) = \sum_X P(X)P(A|X)P(C|B, X)$$

B not on the left side as $P(B|A)$ has not been multiplied. It is also not $P(A, C|B)$, why?