Exact Inference: Variable Elimination I

1

Variable Elimination

Recall:

- Let ***X*** be a set of random variables
- A factor ϕ is a function from Val(***X***) $\rightarrow \mathcal{R}$
- The set of variables ***X*** is called the scope of the factor and denoted *Scope[ϕ]*

We will be manipulating factors

2

Variable Elimination

Let ***X*** be a set of variables, and Y $\notin$ ***X*** a variable. Let ϕ(***X***, *Y*) be a factor. We define the factor marginalization of *Y* in ϕ, denoted $\Sigma_Y \phi$ to be a factor ψ over ***X*** such that:

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

This operation is also called summing out of *Y* in ψ

3

Variable Elimination

A	B	C	P(A,B,C)
1	1	1	0.25
1	1	2	0.35
1	2	1	0.08
1	2	2	0.16
2	1	1	0.05
2	1	2	0.07
2	2	1	0
2	2	2	0
3	1	1	0.15
3	1	2	0.21
3	2	1	0.09
3	2	2	0.18

Summing out B

A	C	P(A,C)
1	1	0.25+0.08=0.33
1	2	0.35+0.16=0.51
2	1	0.05+0=0.05
2	2	0.07+0=0.07
3	1	0.15+0.09=0.24
3	2	0.21+0.18=0.39

Note: we only sum up entries in the table where the values of **X** match up.

4

Variable Elimination

In a Bayesian network:

- Summing out all variables results in a factor with value 1

In a Markov network:

- Summing out all variables in the unnormalized distribution $\tilde{P}_\Phi$ defined by the product of factors in the Markov network results in the partition function

Variable Elimination

Recall: Let ***X***, ***Y***, and ***Z*** be three disjoint sets of variables, and let ϕ_1***(X,Y)*** and ϕ_2***(Y,Z)*** be two factors. We define the factor product ϕ_1 *x* ϕ_2 to be a factor ψ: *Val(**X,Y,Z**)* $\rightarrow \mathcal{R}$ as follows:

$$\psi(\mathbf{X},\mathbf{Y},\mathbf{Z}) = \phi_1(\mathbf{X},\mathbf{Y})\ \phi_2(\mathbf{Y},\mathbf{Z})$$

Variable Elimination

Example of a factor product:

A	B	ϕ_1(A,B)
0	0	0.5
0	1	0.8
1	0	0.1
1	1	0
2	0	0.3
2	1	0.9

B	C	ϕ_2(B,C)
0	0	0.5
0	1	0.7
1	0	0.1
1	1	0.2

A	B	C	ψ(A,B,C)
0	0	0	(0.5)(0.5)=0.25
0	0	1	(0.5)(0.7)=0.35
0	1	0	(0.8)(0.1)=0.08
0	1	1	(0.8)(0.2)=0.16
1	0	0	(0.1)(0.5)=0.05
1	0	1	(0.1)(0.7)=0.07
1	1	0	(0)(0.1)=0
1	1	1	(0)(0.2)=0
2	0	0	(0.3)(0.5)=0.15
2	0	1	(0.3)(0.7)=0.21
2	1	0	(0.9)(0.1)=0.09
2	1	1	(0.9)(0.2)=0.18

Variable Elimination

Operations over factors:

- Addition is commutative: $\sum_X \sum_Y \phi = \sum_Y \sum_X \phi$
- Multiplication is commutative: $\phi_1 \cdot \phi_2 = \phi_2 \cdot \phi_1$
- Products are associative: $(\phi_1 \cdot \phi_2) \cdot \phi_3 = \phi_1 \cdot (\phi_2 \cdot \phi_3)$
- Exchanging summations and products:

  If $X \neq Scope[\phi_1]$, $\sum_X (\phi_1 \cdot \phi_2) = \phi_1 \cdot \sum_X \phi_2$

  (X is not in the terms of ϕ_1)

Variable Elimination

Example: $P(D) = \sum_C \sum_B \sum_A P(A,B,C,D)$

$$= \sum_C \sum_B \sum_A \varphi_A \cdot \varphi_B \cdot \varphi_C \cdot \varphi_D$$

$$= \sum_C \sum_B \varphi_C \cdot \varphi_D \cdot \varphi_B \left(\sum_A \varphi_A \right)$$

$$= \varphi_D \sum_C \varphi_C \cdot \left(\sum_B \varphi_B \cdot \left(\sum_A \varphi_A \right) \right)$$

9

Variable Elimination

The general problem involves a sum-product inference task:

$$\sum_{\mathbf{Z}} \prod_{\phi \in \Phi} \phi$$

Trick: Push in the summations as far as you can

10

Variable Elimination

Procedure Sum-Product-VE(
  Φ, // A set of factors
  Z, // Set of variables to be eliminated
  < // Ordering on **Z**
)

1. Let $Z_1, \ldots, Z_k$ be an ordering of **Z** such that
2. $Z_i < Z_j$ if and only if i < j
3. **for** i = 1, … k
4. Φ ← Sum-Product-Eliminate-Var(Φ,Z_i)
5. $\phi^* \leftarrow \prod_{\phi \in \Phi} \phi$
6. Return φ*

11

Variable Elimination

Procedure Sum-Product-Eliminate-Var(
  Φ, // A set of factors
  Z, // Variable to be eliminated
)

1. **Φ’** ← {φ∈Φ: Z ∈ Scope[φ]}
2. **Φ’’** ← **Φ** - **Φ’**
3. $\psi \leftarrow \prod_{\phi \in \Phi'} \phi$
4. $\tau \leftarrow \sum_Z \psi$
5. Return **Φ’’** ∪ {τ}

12

Variable Elimination

Let $\boldsymbol{X}$ be some set of variables, and let Φ be a set of factors such that for each $\phi \in \Phi$, $Scope[\Phi] \subseteq \boldsymbol{X}$. Let $\boldsymbol{Y} \subset \boldsymbol{X}$ be a set of query variables, and let $\boldsymbol{Z} = \boldsymbol{X} - \boldsymbol{Y}$. Then for any ordering < over $\boldsymbol{Z}$, *Sum-Product-VE*(Φ,$\boldsymbol{Z}$,<) returns a factor $\phi^*(\boldsymbol{Y})$ such that

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

13

Variable Elimination

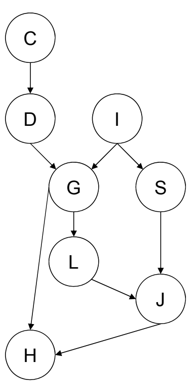
Example: compute $P_{\mathcal{B}}(Y)$ for Bayesian network $\mathcal{B}$. Let:

- $\Phi = \{\phi_{X_i}\}_{i=1}^n$ where $\phi_{X_1} = P(X_i \mid Parents(X_i))$
- $\boldsymbol{Z}=\{Z_1,\ldots,Z_m\} = \mathcal{X} - \boldsymbol{Y}$ (eliminate all non-query variables)

Note: We can do the exact same thing on a Markov network except the final factor $\phi^*(\boldsymbol{Y})$ is unnormalized.

14

Variable Elimination



We will compute *P(J)* using the elimination ordering *C*, *D*, *I*, *H*, *G*, *S*, *L*.

Note that:

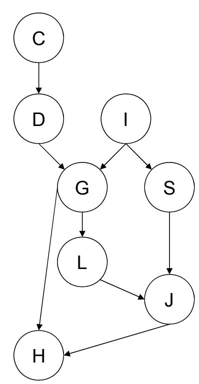
P(C,D,I,G,S,L,J,H)

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

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

15

Variable Elimination



Elimination ordering: *C*, *D*, *I*, *H*, *G*, *S*, *L*

1. Eliminating C:

$$\Psi_1(C,D) = \phi_C(C) \cdot \phi_D(D,C)$$

$$\tau_1(D) = \sum_C \Psi_1(C,D)$$

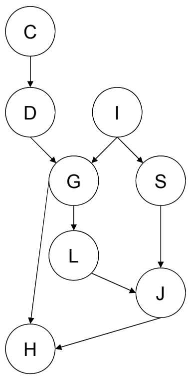
2. Eliminating D:

$$\Psi_2(G,I,D) = \phi_G(G,I,D) \cdot \tau_1(D)$$

$$\tau_2(G,I) = \sum_D \Psi_2(G,I,D)$$

16

Variable Elimination



Elimination ordering: *C*, *D*, *I*, *H*, *G*, *S*, *L*

3. Eliminating I:

$$\Psi_3(G,I,S) = \phi_I(I) \cdot \phi_S(S,I) \cdot \tau_2(G,I)$$

$$\tau_3(G,S) = \sum_I \Psi_3(G,I,S)$$

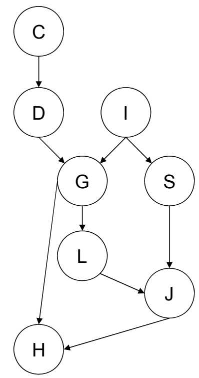
4. Eliminating H:

$$\Psi_4(G,J,H) = \phi_H(H,G,J)$$

$$\tau_4(G,J) = \sum_H \Psi_4(G,J,H)$$

Note: $\tau_4 \equiv 1$ since $\sum_H P(H|G,J)$. However, in this elimination ordering, you do need to generate this factor for the next step.

Variable Elimination



Elimination ordering: *C*, *D*, *I*, *H*, *G*, *S*, *L*

5. Eliminating G:

$$\Psi_5(G,J,L,S) = \tau_4(G,J) \cdot \tau_3(G,S) \cdot \phi_L(L,G)$$

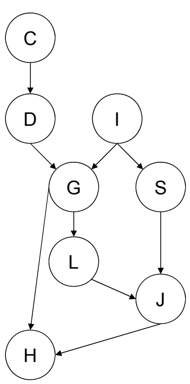
$$\tau_5(J,L,S) = \sum_G \Psi_5(G,J,L,S)$$

6. Eliminating S:

$$\Psi_6(J,L,S) = \tau_5(J,L,S) \cdot \phi_J(J,L,S)$$

$$\tau_6(J,L) = \sum_S \Psi_6(J,L,S)$$

Variable Elimination



Elimination ordering: *C*, *D*, *I*, *H*, *G*, *S*, *L*

7. Eliminating L:

$$\Psi_7(J,L) = \tau_6(J,L)$$

$$\tau_7(J) = \sum_L \Psi_7(J,L)$$

Note: You can use any elimination ordering eg. G, I, S, L, H, C, D. This is a bad ordering because it produces factors with very large scope (see Table 9.2 pg 302)

Variable Elimination

How do we deal with evidence?

eg. P(J = true | I = high, H = false)?

Note that:

$$P(J \mid I = high, H = false) = \frac{P(J, I = high, H = false)}{P(I = high, H = false)}$$

Variable Elimination

Proposition 4.7:

Let $\mathcal{B}$ be a Bayesian network over $\mathcal{X}$ and ***E*** = ***e*** an observation. Let ***W*** = $\mathcal{X}$ – ***E***. Then $P_{\mathcal{B}}$(***W*** |***e***) is a Gibbs distribution defined by the factors where

$$\phi_{X_i} = P_{\mathcal{B}}(X_i \mid Parents(X_i))[\boldsymbol{E} = \boldsymbol{e}]$$

This is a reduced factor, meaning it is the factor with entries inconsistent with E=e removed.

The partition function for this Gibbs distribution is *P*(***e***)

Variable Elimination

- Replace each factor ϕ with ϕ**[E=e]**
- Then do variable elimination like normal
- Remember to normalize with $P(E = e)$

Variable Elimination

Procedure Cond-Prob-VE(
   $\mathcal{K}$,   // A network over $\mathcal{X}$
   Y, // Set of query variables
   E =***e***   // Evidence
)
1. $\Phi \leftarrow$ Factors parameterizing K
2. Replace each $\phi \in \Phi$ by ϕ [***E***=***e***]
3. Select an elimination ordering <
4. ***Z*** $\leftarrow \mathcal{X}$ – ***Y*** – ***E***
5. $\phi^* \leftarrow$Sum-Product-VE(Φ,<,***Z***)
6. $\alpha \leftarrow \sum_{y \in Val(Y)} \phi^*(\boldsymbol{y})$
7. return α, ϕ^*

Note: ϕ^* represents *P*(***Y***,***e***) so divide ϕ^* by α to get *P*(***Y***|***e***)

Variable Elimination

Computing: P(J,I=high,H=false)

Step	Variable Eliminated	Factors Used	Variables Involved	New Factor
1'	C	ϕ_C(C), ϕ_D(D,C)	C, D	τ_1'(D)
2'	D	ϕ_G[I=high](G,D), ϕ_I[I=high](), τ_1'(D)	G, D	τ_2'(G)
5'	G	τ_2'(G), ϕ_L(L,G), ϕ_H[H=false](G.J)	G, L, J	τ_5'(J,L)
6'	S	ϕ_S[I=high](S), ϕ_J(J, L, S)	J, L, S	τ_6'(J,L)
7'	L	τ_6'(J,L), τ_5'(J,L)	J,L	τ_7'(J)