

CSC 412/2506:  
Probabilistic Learning and Reasoning  
Week 3 - 2/2: Exact Inference

Murat A. Erdogdu

University of Toronto

# Overview

- Exact inference on graphical models
- Variable elimination

# Inference as Conditional Distribution

- We explore inference in probabilistic graphical models (PGMs).
  - $x_E$  = The observed evidence
  - $x_F$  = The unobserved variable we want to infer
  - $x_R = x - \{x_F, x_E\}$  = Remaining variables, extraneous to query.
- For inference, we focus on computing the conditional probability distribution

$$p(x_F|x_E) = \frac{p(x_F, x_E)}{p(x_E)} = \frac{p(x_F, x_E)}{\sum_{x_F} p(x_F, x_E)}$$

- for which, we marginalize out these extraneous variables, focussing on the joint distribution over evidence and subject of inference:

$$p(x_F, x_E) = \sum_{x_R} p(x_F, x_E, x_R)$$

# Variable elimination

Order which variables are marginalized affects the computational cost!

Our main tool is **variable elimination**:

- A simple and general **exact inference** algorithm in any probabilistic graphical model (DAGMs or MRFs).
- Has computational complexity that depends on the graph structure of the model. We saw last lecture that the graph structure induces independence assumptions.
- Can use dynamic programming to avoid enumerating all variable assignments.

## Example: Simple chain

- Lets start with the example of a simple chain

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

where we want to compute  $p(D)$ , with no evidence variables.

- We have

$$x_F = \{D\}, \quad x_E = \{\}, \quad x_R = \{A, B, C\}$$

- We saw last lecture that this graphical model describes the factorization of the joint distribution as:

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

- Assume each variable can take on  $k$  different values.

## Example: Simple chain

- The goal is to compute the marginal  $p(D)$  with no observed variables:

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

- However, if we do this sum naively, cost is exponential  $O(k^{n=4})$  :

$$\begin{aligned} p(D) &= \sum_{A,B,C} p(A, B, C, D) \\ &= \sum_C \sum_B \sum_A p(A)p(B|A)p(C|B)p(D|C) \end{aligned}$$

since we are summing over  $A$  for every term in the sum over  $B$ .

- Instead, choose an **elimination ordering**:

$$\begin{aligned} p(D) &= \sum_{C,B,A} p(A, B, C, D) \\ &= \sum_C p(D|C) \sum_B p(C|B) \sum_A p(A)p(B|A). \end{aligned}$$

## Example: Simple chain

- This reduces the complexity by first computing terms that appear across the other sums.

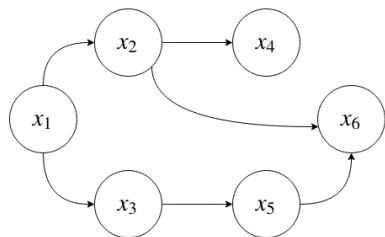
$$p(D) = \sum_C p(D|C) \sum_B p(C|B) \sum_A p(A)p(B|A)$$

•

$$\begin{aligned} p(D) &= \sum_C p(D|C) \sum_B p(C|B) \sum_A p(A)p(B|A) \\ &= \sum_C p(D|C) \sum_B p(C|B)p(B) \\ &= \sum_C p(D|C)p(C) \end{aligned}$$

- The cost of performing inference on the chain in this manner is  $\mathcal{O}(nk^2)$ . In comparison, generating the full joint distribution and marginalizing over it has complexity  $\mathcal{O}(k^n)$ !

## Simple Example: DAGM



- Observe the variable  $X_6 = \bar{x}_6$ .  
What is  $p(X_1|\bar{x}_6)$ ?
- DAGM implies the factorization:

$$p(x_1, \dots, x_6) = p(x_1)p(x_2|x_1)p(x_3|x_1) \\ \times p(x_4|x_2)p(x_5|x_3)p(x_6|x_2, x_5)$$

- We have

$$x_F = \{x_1\}, x_E = \{x_6\}, x_R = \{x_2, x_3, x_4, x_5\}$$

$$p(x_F|x_E) = \frac{\sum_{x_R} p(x_F, x_E, x_R)}{\sum_{x_F, x_R} p(x_F, x_E, x_R)}$$
$$\Rightarrow p(x_1|\bar{x}_6) = \frac{p(x_1, \bar{x}_6)}{p(\bar{x}_6)} = \frac{p(x_1, \bar{x}_6)}{\sum_{x \in x_F, x_R} p(x, \bar{x}_6)}$$



To compute  $p(x_1, \bar{x}_6)$ , we use variable elimination in the order 2, 3, 4, 5

$$\begin{aligned} p(x_1, \bar{x}_6) &= p(x_1) \sum_{x_2} \sum_{x_3} \sum_{x_4} \sum_{x_5} p(x_2|x_1)p(x_3|x_1)p(x_4|x_2)p(x_5|x_3)p(\bar{x}_6|x_2, x_5) \\ &= p(x_1) \sum_{x_2} p(x_2|x_1) \sum_{x_3} p(x_3|x_1) \sum_{x_4} p(x_4|x_2) \sum_{x_5} p(x_5|x_3)p(\bar{x}_6|x_2, x_5) \\ &= p(x_1) \sum_{x_2} p(x_2|x_1) \sum_{x_3} p(x_3|x_1) \sum_{x_4} p(x_4|x_2)p(\bar{x}_6|x_2, x_3) \end{aligned}$$

Note that  $p(\bar{x}_6|x_2, x_3)$  does not need to participate in  $\sum_{x_4}$ .

$$\begin{aligned} &= p(x_1) \sum_{x_2} p(x_2|x_1) \sum_{x_3} p(x_3|x_1)p(\bar{x}_6|x_2, x_3) \sum_{x_4} p(x_4|x_2) \\ &= p(x_1) \sum_{x_2} p(x_2|x_1) \sum_{x_3} p(x_3|x_1)p(\bar{x}_6|x_2, x_3) \\ &= p(x_1) \sum_{x_2} p(x_2|x_1)p(\bar{x}_6|x_1, x_2) \\ &= p(x_1)p(\bar{x}_6|x_1) \end{aligned}$$

Finally,

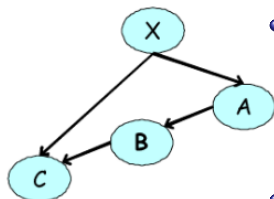
$$p(x_1|\bar{x}_6) = \frac{p(x_1)p(\bar{x}_6|x_1)}{\sum_{x_1} p(x_1)p(\bar{x}_6|x_1)}.$$

# Best Elimination Ordering

- The complexity of variable elimination depends on the elimination ordering!
- Unfortunately, finding the best elimination ordering is NP-hard.

## Intermediate Factors

- In general, eliminating does not produce a valid marginal or conditional distribution of the graphical model.



- Consider the distribution given by

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

- We want to marginalize over  $X$

$$\begin{aligned} p(A, B, C) &= \sum_X p(X)p(A|X)p(B|A)p(C|B, X) \\ &= p(B|A) \sum_X p(X)p(A|X)p(C|B, X) \end{aligned}$$

- The resulting term  $\sum_X p(X)p(A|X)p(C|B, X)$  does not correspond to a valid distribution because it is unnormalized.

## Intermediate Factors

- We introduce **factors**  $\phi$  which are not necessarily normalized distributions, but which describe the local relationship between random variables (just like the ones in MRFs).
- In the above example:

$$\begin{aligned} p(A, B, C) &= \sum_X p(X)p(A|X)p(B|A)p(C|B, X) \\ &= \sum_X \phi_1(X)\phi_2(A, X)\phi_3(A, B)\phi_4(X, B, C) \\ &= \phi_3(A, B) \sum_X \phi_1(X)\phi_2(A, X)\phi_4(X, B, C) \\ &= \phi_3(A, B)\tau(A, B, C) \end{aligned}$$

- Marginalizing over  $X$  we introduce a new factor, denoted by  $\tau$ .

# Sum-Product Inference

- Abstractly, computing  $p(x_F|x_E)$  is given by the **sum-product** algorithm:

$$p(x_F|x_E) \propto \tau(x_F, x_E) = \sum_{x_R} \prod_{\phi_c \in \Phi} \phi_c(x_c)$$

where  $\Phi$  is a set of potentials or factors.

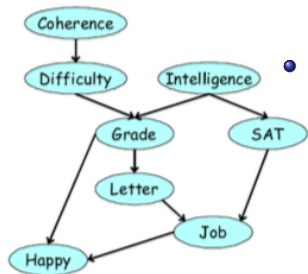
- For DAGMs,  $\Phi$  is given by the conditional probability distributions for all variables

$$\Phi = \{\phi_{x_i}\}_{i=1}^N = \{p(x_i|\text{parents}(x_i))\}_{i=1}^N$$

where the sum is over the set  $x_R = x - x_F - x_E$ .

- For MRFs,  $\Phi$  is given by the set of unnormalized potentials. Therefore, we must normalize the resulting  $\tau(t)$  by  $\sum_t \tau(y)$ .

## Example



- This describes a factorization:

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

For notational convenience, we write the conditionals as factors.

$$\Phi = \{\phi(C), \phi(C, D), \phi(I), \phi(G, D, I), \phi(L, G), \phi(S, I), \phi(J, S, L), \phi(H, J, G)\}$$

We are interested in the probability of getting a job,  $p(J)$ . We perform exact inference as follows.

# Example

Elimination Ordering  $\prec \{C, D, I, H, G, S, L\}$

$$\begin{aligned} p(J) &= \sum_L \sum_S \phi(J, L, S) \sum_G \phi(L, G) \sum_H \phi(H, G, J) \sum_I \phi(S, I) \phi(I) \sum_D \phi(G, D, I) \underbrace{\sum_C \phi(C) \phi(C, D)}_{\tau(D)} \\ &= \sum_L \sum_S \phi(J, L, S) \sum_G \phi(L, G) \sum_H \phi(H, G, J) \sum_I \phi(S, I) \phi(I) \underbrace{\sum_D \phi(G, D, I) \tau(D)}_{\tau(G, I)} \\ &= \sum_L \sum_S \phi(J, L, S) \sum_G \phi(L, G) \sum_H \phi(H, G, J) \underbrace{\sum_I \phi(S, I) \phi(I) \tau(G, I)}_{\tau(S, G)} \\ &= \sum_L \sum_S \phi(J, L, S) \sum_G \phi(L, G) \tau(S, G) \underbrace{\sum_H \phi(H, G, J)}_{\tau(G, J)} \\ &= \sum_L \sum_S \phi(J, L, S) \underbrace{\sum_G \phi(L, G) \tau(S, G) \tau(G, J)}_{\tau(J, L, S)} \\ &= \sum_L \underbrace{\sum_S \phi(J, L, S) \tau(J, L, S)}_{\tau(J, L)} \\ &= \underbrace{\sum_L \tau(J, L)}_{\tau(J)} \\ &= \tau(J) \end{aligned}$$

Do we need to normalize the final  $\tau$ ?

# Complexity of Variable Elimination Ordering

- We discussed previously that variable elimination ordering determines the computational complexity. This is due to how many variables appear inside each sum.
- Different elimination orderings will involve different number of variables appearing inside each sum.
- The complexity of the VE algorithm is

$$O(mk^{N_{\max}})$$

where

- ▶  $m$  is the number of initial factors.
- ▶  $k$  is the number of states each random variable takes (assumed to be equal here).
- ▶  $N_i$  is the number of random variables inside each sum  $\sum_i$ .
- ▶  $N_{\max} = \max_i N_i$  is the number of variables inside the largest sum.



## Example

Elimination Ordering  $\prec \{C, D, I, H, G, S, L\}$

- Here are all the initial factors:

$$\Phi = \{\phi(C), \phi(C, D), \phi(I), \phi(G, D, I), \phi(L, G), \phi(S, I), \phi(J, S, L), \phi(H, J, G)\}$$

$$\implies m = |\Phi| = 8$$

- Here are the sums, and the number of variables that appear in them

$$\begin{array}{ccc} \underbrace{\sum_C \phi(C)\phi(C, D)}_{N_C=2} & \underbrace{\sum_D \phi(G, D, I)\tau(D)}_{N_D=3} & \underbrace{\sum_I \phi(S, I)\phi(I)\tau(G, I)}_{N_I=3} \\ \underbrace{\sum_H \phi(H, G, J)}_{N_H=3} & \underbrace{\sum_G \phi(L, G)\tau(S, G)\tau(G, J)}_{N_G=4} & \underbrace{\sum_S \phi(J, L, S)\tau(J, L, S)}_{N_S=3} \\ \underbrace{\sum_L \tau(J, L)}_{N_L=2} & \implies \text{the largest sum is } N_G = 4. & \end{array}$$

- For simplicity, assume all variables take on  $k$  states. So the complexity of the variable elimination under this ordering is  $O(8 * k^4)$ .

# Summary

- Variable elimination can be used for exact inference in PGMs.
- The ordering in variable elimination can significantly reduce the computational complexity.
- The overall complexity of the variable elimination algorithm can be computed.