Lecture 5: Exact Inference via Variable Elimination

Liam Solus

KTH Royal Institute of Technology

solus@kth.se

15 September 2023
Graphical Models PhD Course
WASP

$\mathbf{X} = [X_1, \ldots, X_m]^T$ with pmf $f_{\mathbf{X}}(\mathbf{x}) = P(\mathbf{X} = \mathbf{x})$.

Assume X_i discrete with possible outcomes $\mathrm{Val}[X_i] < \infty$.

Goal (Exact Inference).

1. Given data $\mathbf{x}$, compute $f_{\mathbf{X}_A}(\mathbf{x}_A)$ for $A \subseteq [m]$. (marginals)
2. For $A, B \subseteq [m]$ disjoint and data $\mathbf{x}_B$, compute $f_{\mathbf{X}_A|\mathbf{X}_B}(\mathbf{x}_A|\mathbf{x}_B)$. (posteriors)

Example 1. $\mathbf{X} = [X_1, X_2, X_3]^T$ is Markov to $\mathcal{G} = 1 \to 2 \to 3$.

$\mathrm{Val}(X_i) = \{0, 1\}$ for all $i = 1, 2, 3 \implies \mathrm{Val}(\mathbf{X}) = \{0, 1\}^3$.

$$\begin{array}{lll} X_1 \sim \mathrm{Ber}(\theta_1) & X_2|X_1 = 0 \sim \mathrm{Ber}(\theta_2) & X_3|X_2 = 0 \sim \mathrm{Ber}(\theta_4) \\ & X_2|X_1 = 1 \sim \mathrm{Ber}(\theta_3) & X_3|X_2 = 1 \sim \mathrm{Ber}(\theta_5) \end{array}$$

(marginal computations.) Given n iid oberservations $\mathbf{x}_1, \ldots, \mathbf{x}_n$ from $\mathbf{X}$ we can compute the MLE of the parameters $\theta_1, \ldots, \theta_5$: $\hat{\theta}_1, \ldots, \hat{\theta}_5$. How can we efficiently compute the marginal distributions for

$$X_1, X_2, X_3, \mathbf{X}_{1,2}, \mathbf{X}_{2,3}, \mathbf{X}_{1,3}?$$

(less interesting but still useful)

Example 1 **(continued).**
(posterior computations.) Given an observation from the marginal distribution $\mathbf{X}_{\{1,3\}} = \mathbf{x}_{\{1,3\}}$, how can we efficiently compute $f_{X_2|\mathbf{X}_{\{1,3\}}}(x_2|\mathbf{x}_{\{1,3\}})$? (more useful)

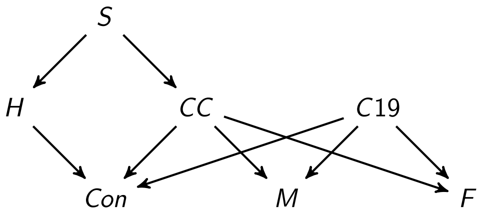
Example 2 **(Medical Diagnosis).** When treating a patient a doctor, considers a variety of possible diseases while measuring symptoms and environmental factors:

- **Diseases:**
  - $CC = 1$ if Common Cold — $CC = 0$ otherwise.
  - $C19 = 1$ if Covid-19 — $C19 = 0$ otherwise.
  - $H = 1$ if Hayfever — $H = 0$ otherwise.
- **Env. Factor:** $S =$ Season

$$\text{Val}(S) = \{\textit{Fall}\,(0), \textit{Winter}\,(1), \textit{Spring}\,(2), \textit{Summer}\,(3)\}$$

- **Symptoms:**
  - $F = 1$ if Fever — $F = 0$ otherwise.
  - $M = 1$ if Muscle Pain — $M = 0$ otherwise.
  - $Con = 1$ if Congestion — $Con = 0$ otherwise.

Example 2 **(continued).** The doctor constructs the following DAG model to represent the distribution of these 7 variables:



Can observe data $\mathbf{y}$ from the marginal distribution $\mathbf{Y} = [S, F, M, Con]^T$ and would like to compute the posterior distributions

$$P(CC|\mathbf{Y} = \mathbf{y}), \qquad P(C19|\mathbf{Y} = \mathbf{y}), \qquad P(H|\mathbf{Y} = \mathbf{y}).$$

Can we use the structure of the graph to help the doctor make this computation efficiently?

Exact Inference for Marginal Computations.

$\mathbb{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ an iid sample from $\mathbf{X}$.

Naive approach: Estimate

$$f_{\mathbf{X}}(\mathbf{x}) = P(x_1, \ldots, x_m) \approx \frac{\#\{[x_1, \ldots, x_m]^T \in \mathbb{D}\}}{\#\mathbb{D}} \qquad \text{for all } \mathbf{x} \in \mathrm{Val}(\mathbf{X})$$

Then estimate $f_{\mathbf{X}_A}(\mathbf{x}_A)$ by computing the sum

$$f_{\mathbf{X}_A}(\mathbf{x}_A) = P(\mathbf{X}_A = \mathbf{x}_A) = \sum_{\mathbf{x}_{[m]\setminus A} \in \mathrm{Val}(\mathbf{X}_{[m]\setminus A})} f_{\mathbf{X}}(\mathbf{x}_A, \mathbf{x}_{[m]\setminus A}).$$

Computationally expensive... the graph structure can tell us when the complexity is feasible.

1. Compute $f_{\mathbf{X}_A}(\mathbf{x}_A)$ by changing the order of summation in smart ways.
2. Use **dynamic programming**: Store some computed sums that we will use multiple times.

Example 3 **(Markov Chain).** $\mathbf{X} = [X_1, X_2, X_3, X_4]^T$ markov to

$$\mathcal{G} = 1 \to 2 \to 3 \to 4$$

where $\mathrm{Val}(X_i) = \{0, 1\}$ for all $i = 1, 2, 3, 4$.

Suppose we know $f_{X_i|\mathbf{X}_{\mathrm{pa}_{\mathcal{G}}(i)}}(x_i|\mathbf{x}_{\mathrm{pa}_{\mathcal{G}}(i)})$ for all $\mathbf{x}_{\mathrm{pa}_{\mathcal{G}}(i)}$, for all i.

Goal: Compute $f_{X_4}(x_4)$.

Naive approach:

$$\begin{aligned} f_{X_4}(x_4) &= \sum_{[x_1,x_2,x_3]^T \{0,1\}^3} f_{\mathbf{X}}(x_1, x_2, x_3, x_4), \\ &= \sum_{[x_1,x_2,x_3]^T \in \{0,1\}^3} f_{X_1}(x_1) f_{X_2|X_1}(x_2|x_1) f_{X_3|X_2}(x_3|x_2) f_{X_4|X_3}(x_4|x_3) \end{aligned}$$

Since $|\mathrm{Val}(X_i)| = 2$, need

- $(3)(8) = 24$ multiplications: 3 for each $[x_1, x_2, x_3]^T$ with $x_4 = 0$
- 7 summations for $x_4 = 0$
- 1 difference to get $f_{X_4}(1) = 1 - f_{X_4}(0)$.

36 operations

Less Naive approach:

Change the order of summation:

$$
\begin{aligned}
f_{X_4}(x_4) &= \sum_{[x_1,x_2,x_3]^T \{0,1\}^3} f_{\mathbf{X}}(x_1, x_2, x_3, x_4), \\
&= \sum_{[x_1,x_2,x_3]^T \in \{0,1\}^3} f_{X_1}(x_1) f_{X_2|X_1}(x_2|x_1) f_{X_3|X_2}(x_3|x_2) f_{X_4|X_3}(x_4|x_3), \\
&= \sum_{x_3 \in \{0,1\}} f_{X_4|X_3}(x_4|x_3) \sum_{x_2 \in \{0,1\}} f_{X_3|X_2}(x_3|x_2) \sum_{x_1 \in \{0,1\}} f_{X_2|X_1}(x_2|x_1) f_{X_1}(x_1),
\end{aligned}
$$

1. $\sum_{x_1 \in \{0,1\}} f_{X_2|X_1}(x_2|x_1) f_{X_1}(x_1) = f_{X_2}(x_2)$ — 4 computations to get $f_{X_2}(x_2)$
2. $\sum_{x_2 \in \{0,1\}} f_{X_3|X_2}(x_3|x_2) f_{X_2}(x_2) = f_{X_3}(x_3)$ — 4 computations to get $f_{X_3}(x_3)$
3. $\sum_{x_3 \in \{0,1\}} f_{X_4|X_3}(x_4|x_3) f_{X_3}(x_3) = f_{X_4}(x_4)$ — 4 computations to get $f_{X_4}(x_4)$

(each step $1, 2, 3$ does 2 multiplications, 1 summation and 1 difference)

12 operations

Variable Elimination (VE).

1. Switch summations to follow an order specified by the graph to compute different marginals in steps.
2. Store these intermediate values (marginals) to be used in later computations (dynamic programming).

VE is more efficient than the naive approach because the distribution is Markov to the graph $1 \to 2 \to 3 \to 4$.

Graph structure reduces number of variables in each conditional factor allowing us to compute partial sums.

$$f_{X_4}(x_4) = \sum_{[x_1,x_2,x_3]^T \in \{0,1\}^3} f_{X_1}(x_1) f_{X_2|X_1}(x_2|x_1) f_{X_3|\mathbf{X}_{\{1,2\}}}(x_3|\mathbf{x}_{\{1,2\}}) f_{X_4|\mathbf{X}_{\{1,2,3\}}}(x_4|\mathbf{x}_{\{1,2,3\}})$$

requires 36 computations.

(reduce to naive approach when graph is complete.)

Goals:

1. formalize the VE algorithm.
2. deduce complexity bounds according to graph structure.

Formalizing VE.

Definition. Let $\mathbf{X} = [X_1, \ldots, X_m]^T$.

- A **factor** is a function $\phi : \text{Val}(\mathbf{X}) \longrightarrow \mathcal{R}$.
- A factor ϕ is **nonnegative** if $\phi(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in \text{Val}(\mathbf{X})$.
- The **scope** of ϕ is the set of variables that give the input for $\phi(\mathbf{x})$:

$$\text{Scope}[\phi] = \{X_1, \ldots, X_m\}.$$

We can marginalize out variables in factors:

$\mathbf{X} = [X_1, \ldots, X_n]^T$ and $Y \neq X_i$ for all i. Given a factor $\phi(\mathbf{x}, y)$ with $\text{Scope}[\phi] = \{X_1, \ldots, X_m, Y\}$ we get the marginal factor

$$\psi(\mathbf{x}) = \sum_{y \in \text{Val}[Y]} \phi(\mathbf{x}, y).$$

Goal: Given a set of factors Φ whose scopes are contained in $\mathbf{X} = [X_1, \ldots, X_n]^T$ and $\mathbf{Z}$ a subvector of $\mathbf{X}$, compute the factor

1. **marginal computations in DAG models:** $\Phi = \{f_{X_i|\mathbf{x}_{\text{pa}_{\mathcal{G}}(i)}}(x_i|\mathbf{x}_{\text{pa}_{\mathcal{G}}(i)}) : i \in [m]\}$.
2. **marginal computations in UG models:** $\Phi = \{\psi_C(\mathbf{x}_C) : C \in \mathcal{C}(G)\}$.

$\Phi =$ set of factors, $\mathbf{Z} =$ variables to be eliminated, $\prec =$ an ordering on $Z_1, \ldots, Z_k$.

`Eliminate-Var`(Φ, Z_i):

1. $\Phi' := \{\phi \in \Phi : Z_i \in \text{Scope}[\phi]\}$
2. $\Phi'' := \Phi \setminus \Phi'$
3. $\psi_i := \prod_{\phi \in \Phi'} \phi$
4. $\tau_i := \sum_{z_i \in \text{Val}(Z_i)} \psi_i$
5. **return** $\Phi'' \cup \{\tau_i\}$

`VE`$(\Phi, \mathbf{Z}, \prec)$:

1. **for** k **in** $[1, \ldots, k]$:
   - $\Phi :=$ `Eliminate-Var`(Φ, Z_i):
   - $\phi^* := \prod_{\phi \in \Phi} \phi$.
2. **return** ϕ^*

For $\mathbf{X} = [X_1, X_2, X_3, X_4]^T$ markov to

$$\mathcal{G} = 1 \to 2 \to 3 \to 4$$

$$\Phi = \{f_{X_1}(x_1), f_{X_2|X_1}(x_2|x_1), f_{X_3|X_2}(x_3|x_2), \\ f_{X_4|X_3}(x_4|x_3)\}$$

`Eliminate-Var`(Φ, X_i):

- $\Phi' = \{f_{X_1}(x_1), f_{X_2|X_1}(x_2|x_1)\}$
- $\psi_1(x_1, x_2) = f_{X_1}(x_1) f_{X_2|X_1}(x_2|x_1)$
- $\tau_1(x_2) = \sum_{x_1 \in \{0,1\}} \psi_1(x_1, x_2) \\ = f_{X_2}(x_2)$

$\mathbf{Z} = [X_1, X_2, X_3]^T$ with $X_1 \prec X_2 \prec X_3$:

$$\texttt{VE}(\Phi, \mathbf{Z}, \prec) = f_{X_4}(x_4).$$

Theorem. Let $\mathbf{X} = [X_1, \ldots, X_m]^T$, Φ a set of factors with scopes in $\mathbf{X}$. Suppose $\mathbf{X} = [\mathbf{Y}^T, \mathbf{Z}^T]^T$. For any elimination order $\prec$ on $\mathbf{Z}$, the variable elimination $\mathtt{VE}(\Phi, \mathbf{Z}, \prec)$ returns a factor

$$\phi^* = \sum_{Z \in \mathrm{Val}(\mathbf{Z})} \prod_{\phi \in \Phi} \phi.$$

The proof follows since marginalizing over factors is commutative, associative and fulfills the condition that if $X \notin \mathrm{Scope}[\phi_1]$ then

$$\sum_{x \in \mathrm{Val}[X]} \phi_1 \phi_2 = \phi_1 \sum_{x \in \mathrm{Val}[X]} \phi_2.$$

Hence, VE returns the desired marginals the input factors are the factors in the factorization for the graphical model.

Complexity of VE via Graph Theory.

VE on $X_1, \ldots, X_m$ with k factors in the set Φ:

1. each step creates a factor ψ_i for X_i then sums out X_i to create τ_i.
2. $N_i = \#\,\text{Val}[\text{Scope}[\psi_i]]$
3. at each step $|\Phi| \leq m + k$
4. Each $\phi \in \Phi$ multiplied once to produce ψ_i (at most N_i multiplications)

$$\Longrightarrow \#\text{ multiplications } \leq (m+k)N_i$$

5. $\#$ additions for each ψ_i (to produce τ_i) $= N_i$

$$\Longrightarrow \#\text{ additions } \leq m\left(\max_i N_i\right)$$

Source of possible exponential blow-up are the N_i:

If $\text{Val}[X_i] \leq \eta$ for all i and $\#\,\text{Scope}[\psi_i] = k_i$ then $N_i \leq \eta^{k_i}$.

VE where ψ_i have small scope sizes will be most efficient!

Complexity of VE via Graph Theory.

$\mathrm{Scope}[\psi_i]$ depend on the the graph G and choice of elimination order.

Which graphs admit an elimination order such that $\mathrm{Scope}[\psi_i]$ remain small?

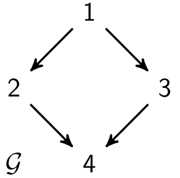
$\mathrm{Scope}[\Phi] := \bigcup_{\phi \in \Phi} \mathrm{Scope}[\phi]$.

$\prec$ an elimination order over all variables in $\mathrm{Scope}[\Phi]$ $(X_1, \ldots, X_m)$

Define the graph $\mathcal{H}_\Phi = (\mathrm{Scope}[\Phi], E_\Phi)$ where

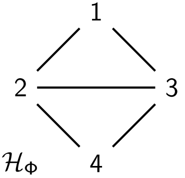
$$X_i - X_k \in E_\Phi \iff \exists \phi \in \Phi : X_i, X_j \in \mathrm{Scope}[\phi].$$

Example.



$\Phi = \{f_{X_1}(x_1), f_{X_2|X_1}(x_2|x_1),$
$f_{X_3|X_2}(x_3|x_2), f_{X_4|X_2,X_3}(x_4|x_2,x_3)\}$

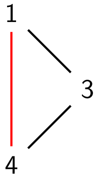
$\mathrm{Scope}[\Phi] = \{X_1, X_2, X_3, X_4\}$



VE on Φ with $\prec = (X_2, X_3, X_1, X_4)$

First step produces

$$\psi_i(x_1, x_2, x_3, x_4) = f_{X_2|X_1}(x_2|x_1) f_{X_4|X_2,X_3}(x_4|x_2, x_3),$$
$$\tau_1(x_1, x_3, x_4) = \sum_{x_2 \in \mathrm{Val}[X_2]} \psi_1(x_1, x_2, x_3, x_4).$$

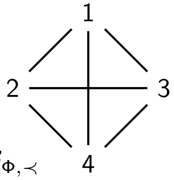


$\mathcal{H}_{\Phi_1}$

$$\Phi_1 = \{f_{X_1}(x_1), f_{X_3|X_2}(x_3|x_2), \tau_1(x_1, x_3, x_4)\}$$

Definition. Edges that appear in $\mathcal{H}_\Phi$ following elimination steps that weren't in the original $\mathcal{H}_\Phi$ are called **fill edges**.

The **induced graph** $\mathcal{G}_{\Phi,\prec}$ for $(\Phi, \prec)$ is the union over all $\mathcal{H}_\Phi$ for each Φ used/produced in the VE algorithm.



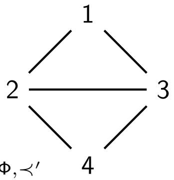
$\mathcal{G}_{\Phi,\prec}$

Cliques in $\mathcal{G}_{\Phi,\prec}$ encode the sizes of scopes of factors used in the VE.

Theorem. $\mathcal{G}_{\Phi,\prec}$ the induced graph for $(\Phi, \prec)$:

1. The scope of every factor produced by VE is a clique in $\mathcal{G}_{\Phi,\prec}$.
2. Every maximal clique in $\mathcal{G}_{\Phi,\prec}$ is the scope of a factor ψ_i.

A different $\prec$ can give smaller cliques: $\prec' = (X_1, X_2, X_3, X_4)$:

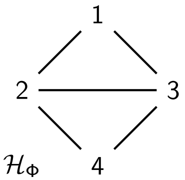


$\mathcal{G}_{\Phi,\prec'}$

Definition. Let Φ be a set of factors and $\prec$ an elimination order on Scope$[\Phi]$.

1. The **width** of $\mathcal{G}_{\Phi,\prec}$ is the size of a maximal clique in $\mathcal{G}_{\Phi,\prec}$ minus 1.
2. If $\mathcal{G}$ is a DAG or UG and $\prec$ an elimination order on its nodes, the **induced width** of $\mathcal{G}$ w.r.t $\prec$ is the width of $\mathcal{G}_{\Phi,\prec}$, and it is denoted $\omega_{\Phi,\prec}$.
3. The **tree-width** of $\mathcal{G}$ is

$$\omega_{\mathcal{G}} := \min_{\prec}(\omega_{\mathcal{G},\prec}).$$



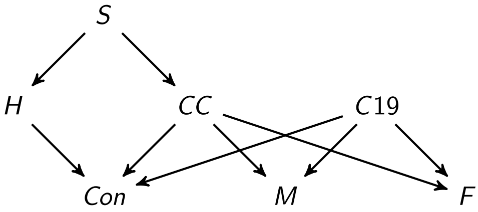
has tree-width 2.

Fact. If $\mathcal{G}$ is a chordal graph then $\omega_{\mathcal{G}}$ equals the size of a maximal clique in $\mathcal{G}$ minus 1.

This is because every chordal graph has a perfect elimination ordering... which are exactly the elimination orderings that do not add fill edges! (Hence why they are called perfect!)

Dealing with Evidence.

Example 2 **(continued).** For a certain patient, the doctor observes the data $[S, Con, M, F]^T = [0, 0, 1, 1]^T$.



They want to compute $f_{C19|S,Con,M,F}(c19|0, 0, 1, 1)$.

$$\begin{aligned}\Phi = \{ & f_S(0) f_{H|S}(h|0),\\ & f_{CC|S}(cc|0),\\ & f_{C19}(c19),\\ & f_{Con|H,CC,C19}(0|h, cc, c19),\\ & f_{M|CC,C19}(1|cc, c19),\\ & f_{F|CC,C19}(1|cc, c19)\}\end{aligned}$$

Consider the elimination orders: $\prec = (H, CC, C19)$, and $\prec' = (H, CC)$

$$\mathtt{VE}(\Phi, [H, CC, C19]^T, \prec) = f_{S,Con,M,F}(0, 0, 1, 1).$$

$$\mathtt{VE}(\Phi, [H, CC]^T, \prec') = f_{C19,S,Con,M,F}(c19, 0, 0, 1, 1).$$

$$\implies f_{C19|S,Con,M,F}(c19|0, 0, 1, 1) = \frac{\mathtt{VE}(\Phi, [H, CC]^T, \prec')}{\mathtt{VE}(\Phi, [H, CC, C19]^T, \prec)}.$$

Applying two runs of VE suffices to compute desired conditional probabilities / posteriors.