Probabilistic Graphical Models

Variable elimination

Siamak Ravanbakhsh

Winter 2021

Learning objective

- an intuition for inference in graphical models
- why is it difficult?
- exact inference by variable elimination

Probability query

marginalization

$$P(X_1) = \sum_{x_2,...,x_n} P(X_1, X_2 = x_2, \ldots, X_n = x_n)$$

Introducing evidence leads to *a similar* problem

$$P(X_1 = x_1 \mid X_m = x_m) = \frac{P(X_1=x_1, X_m=x_m)}{P(X_m=x_m)}$$

Probability query

marginalization

$$P(X_1) = \sum_{x_2,...,x_n} P(X_1, X_2 = x_2, \ldots, X_n = x_n)$$

Introducing evidence leads to *a similar* problem

$$P(X_1 = x_1 \mid X_m = x_m) = \frac{P(X_1 = x_1, X_m = x_m)}{P(X_m = x_m)}$$

MAP inference changes sum to max $\mathbf{x}^* = \arg\max_{\mathbf{x}} P(\mathbf{X} = \mathbf{x})$

maximum a posteriori

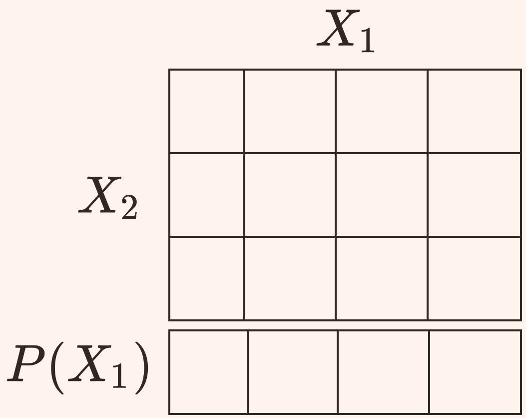
Probability query

marginalization $P(X_1) = \sum_{x_2,...,x_n} P(X_1, X_2 = x_2, \dots, X_n = x_n)$

$n = 2$

representation: $\mathcal{O}(|Val(X_1) \times Val(X_2)|)$

inference: $\mathcal{O}(|Val(X_1) \times Val(X_2)|)$



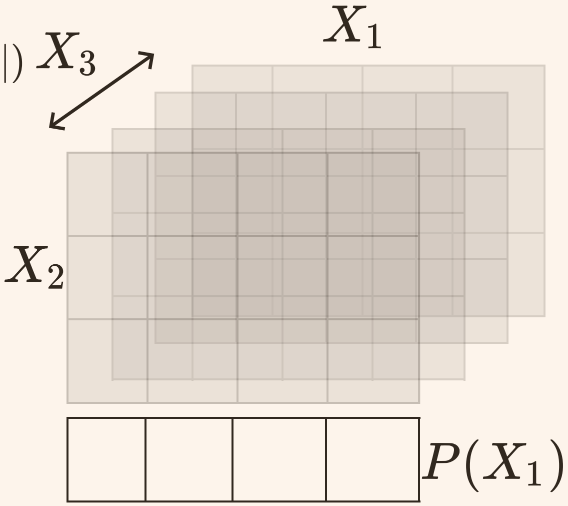
Probability query

marginalization $P(X_1) = \sum_{x_2,...,x_n} P(X_1, X_2 = x_2, \ldots, X_n = x_n)$

$n = 3$

representation: $\mathcal{O}(|Val(X_1) \times Val(X_2) \times Val(X_3)|)$

inference: $\mathcal{O}(|Val(X_1) \times Val(X_2) \times Val(X_3)|)$



Probability query

marginalization $P(X_1) = \sum_{x_2,...,x_n} P(X_1, X_2 = x_2, \ldots, X_n = x_n)$

complexity of representation & inference $\mathcal{O}(\prod_i |Val(X_i)|)$

- binary variables $\mathcal{O}(2^n)$

Probability query

marginalization $P(X_1) = \sum_{x_2,...,x_n} P(X_1, X_2 = x_2, \ldots, X_n = x_n)$

complexity of representation & inference $\mathcal{O}(\prod_i |Val(X_i)|)$

- binary variables $\mathcal{O}(2^n)$

can have a compact representation of P:

- Bayes-net or Markov net
  - *e.g.* $p(x) = \frac{1}{Z} \prod_{i=1}^{n-1} \phi_i(x_i, x_{i+1})$ has an $\mathcal{O}(n)$ representation

Probability query

marginalization $P(X_1) = \sum_{x_2,\ldots,x_n} P(X_1, X_2 = x_2, \ldots, X_n = x_n)$

complexity of representation & inference $\mathcal{O}(\prod_i |Val(X_i)|)$

- binary variables $\mathcal{O}(2^n)$

can have a compact representation of P:

- Bayes-net or Markov net
  - *e.g.* $p(x) = \frac{1}{Z}\prod_{i=1}^{n-1}\phi_i(x_i, x_{i+1})$ has an $\mathcal{O}(n)$ representation

efficient inference ?

Complexity of inference

can we always avoid the exponential cost of inference? No!

can we at least guarantee a good approximation? No!

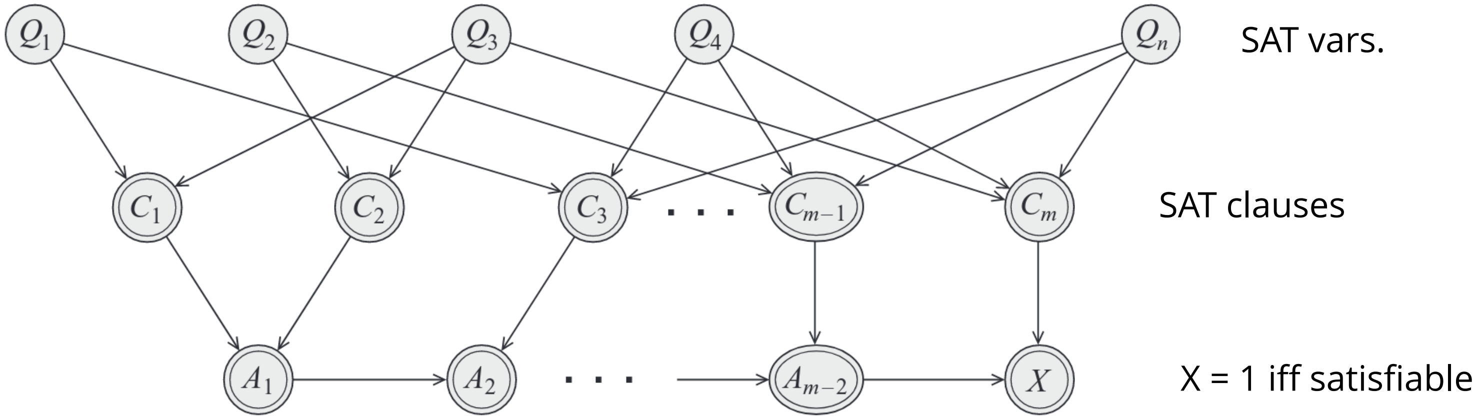
proof idea:

- reduce 3-SAT to inference in a graphical model
  - despite this, graphical models are used for combinatorial optimization (why?)

Complexity of inference: proof

given a BN, decide whether $P(X = x) > 0$ is NP-complete

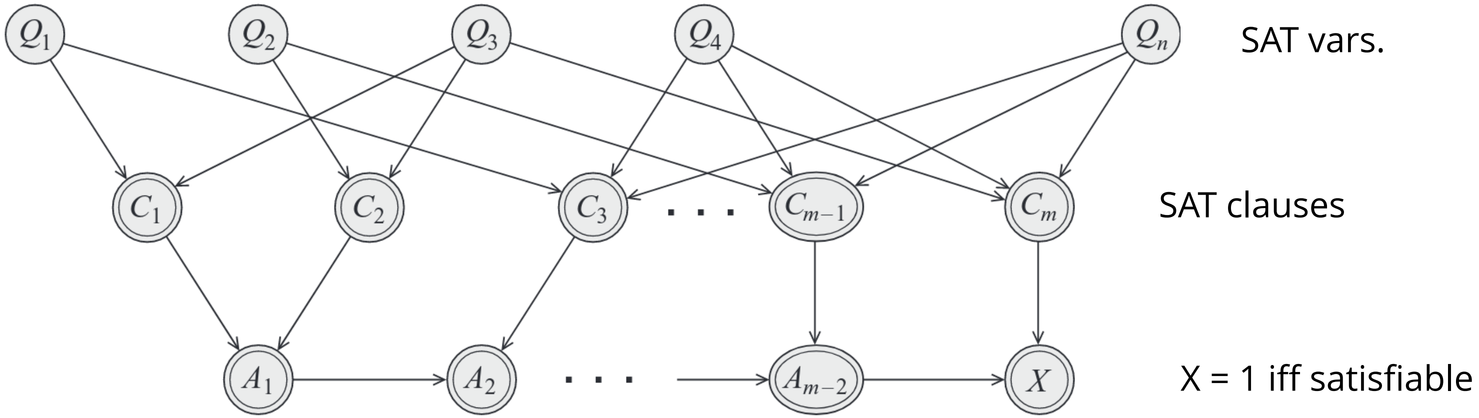
- belongs to NP
- NP-hardness: *answering this query >> solving 3-SAT*



Complexity of inference: proof

given a BN, decide whether $P(X = x) > 0$ is NP-complete

- belongs to NP
- NP-hardness: *answering this query >> solving 3-SAT*

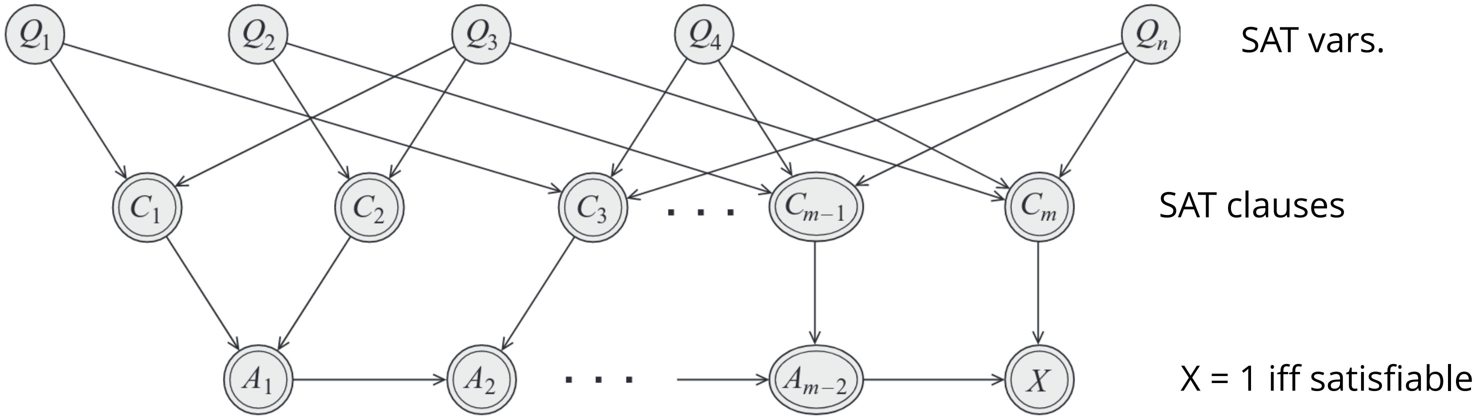


$P(X = x)$

Complexity of inference: proof

given a BN, decide whether $P(X = x) > 0$ is NP-complete

- belongs to NP
- NP-hardness: *answering this query >> solving 3-SAT*



given a BN, calculating $P(X = x)$ is **#P-complete**

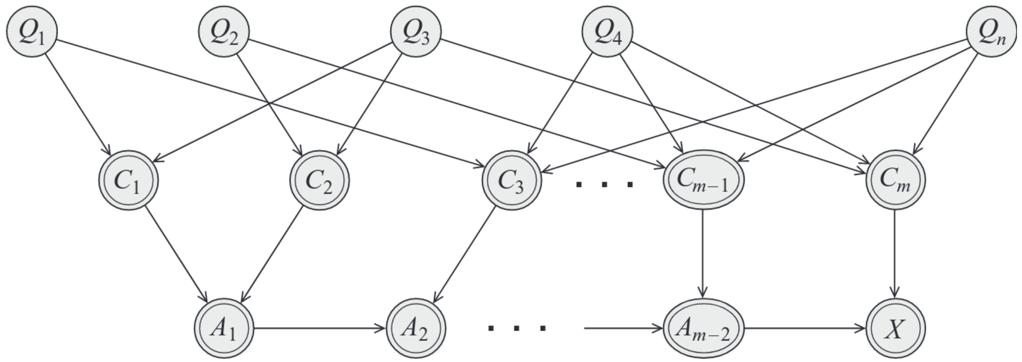
Complexity of approximate inference

given a BN, approximating $P(X = x)$ with a *relative error* ϵ is **NP-hard**

$$\frac{\rho}{1+\epsilon} \leq P(X = x) \leq \rho(1 + \epsilon)$$

Proof: $\rho > 0 \Leftrightarrow P(X = 1) > 0$

our approximation



Complexity of **approximate** inference

given a BN, approximating $P(X = x \mid E = e)$ with an *absolute error* ϵ

for any $0 < \epsilon < \frac{1}{2}$ is **NP-hard**

$$\rho(1 - \epsilon) \leq P(X = x) \leq \rho(1 + \epsilon)$$

Complexity of approximate inference

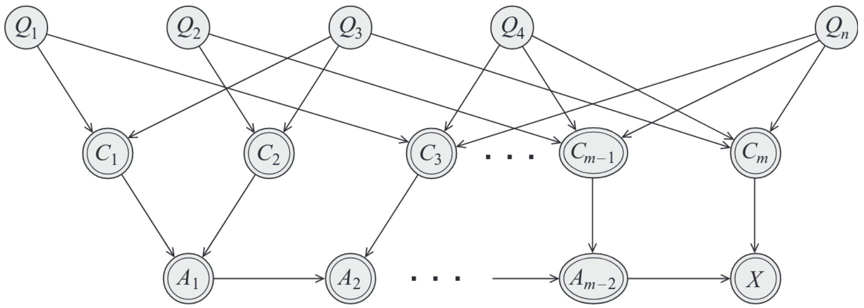
given a BN, approximating $P(X = x \mid E = e)$ with an *absolute error* ϵ

for any $0 < \epsilon < \frac{1}{2}$ is **NP-hard**

$$\rho(1-\epsilon) \leq P(X = x) \leq \rho(1+\epsilon)$$

Proof:

- *sequentially* fix $q_i^* = \arg\max_q P(Q_i = q \mid (Q_1, \ldots, Q_{i-1}) = (q_1^* \ldots q_{i-1}^*), X = 1)$
- either $q_i^0 > \frac{1}{2}$ or $q_i^1 > \frac{1}{2}$
- since $\epsilon < \frac{1}{2}$ this leads to a solution

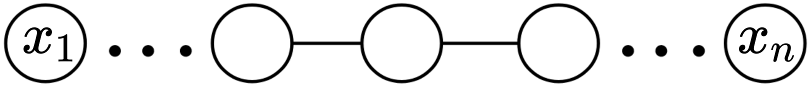


so far...

- reduce the **representation-cost** using a graph structure
- **inference-cost** is in the worst case exponential
- can we reduce it using the graph structure?

Probability query: example

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{i=1}^{n-1} \phi_i(x_i, x_{i+1})$$



$$Val(X_i) = \{1, \ldots, d\} \forall i$$

$p(x_n)$?

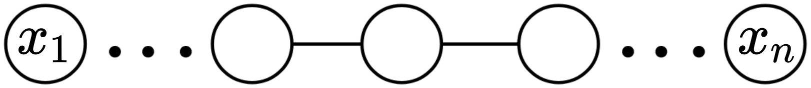
Take 1:

- *calculate n-dim. array* p(x)
- *marginalize it* $p(x_n) = \sum_{-x_n} p(\mathbf{x})$

$\mathcal{O}(d^n)$

Inference: example

$$p(\mathbf{x}) = \frac{1}{Z}\prod_{i=1}^{n-1}\phi_i(x_i, x_{i+1})$$



$p(x_n)$?

Take 2:

- calculate $\tilde{p}(x_m) = \sum_{x_1}\ldots\sum_{x_{n-1}}\phi_1(x_1, x_2)\ldots\phi_{n-1}(x_{n-1}, x_n)$
  - without building $p(\mathbf{x})$
- normalize it $p(x_n) = \tilde{p}(x_n)/(\sum_{x_n}\tilde{p}(x_n))$
- **idea**: use the **distributive law:** $ab + ac = a(b + c)$

3 operations 2 operations

Inference and the distributive law

distributive law $ab + ac = a(b + c)$

3 operations 2 operations

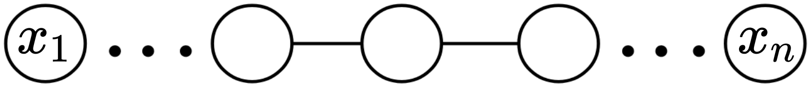
save comutation by **factoring** the operations

in disguise $\sum_{x,y} f(x,y)g(y,z) = \sum_y g(y,z) \sum_x f(x,y)$

- assuming $|Val(X)| = |Val(Y)| = |Val(Z)| = d$
- complexity: from $\mathcal{O}(d^3)$ to $\mathcal{O}(d^2)$

Inference: back to example

$p(x) = \frac{1}{Z} \prod_{i=1}^{n-1} \phi_i(x_i, x_{i+1})$



Take 2:

- objective $\tilde{p}(x_m) = \sum_{x_1} \dots \sum_{x_{n-1}} \phi_1(x_1, x_2) \dots \phi_{n-1}(x_{n-1}, x_n)$
- move in the summations as far as possible

$$\tilde{p}(x_m) = \sum_{x_{n-1}} \phi_{n-1}(x_{n-1}, x_n) \sum_{x_{n-2}} \phi_{n-2}(x_{n-2}, x_{n-1}) \dots \sum_{x_1} \phi_1(x_1, x_2)$$

- complexity is $\mathcal{O}(nd^2)$ instead of $\mathcal{O}(d^n)$

Inference: example 2

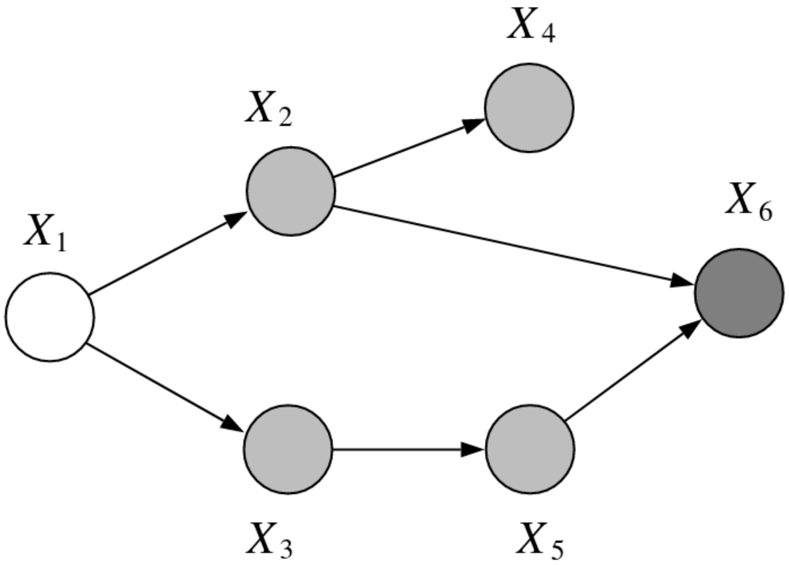
Objective: $p(x_1 \mid \bar{x}_6) = \frac{p(x_1, \bar{x}_6)}{p(\bar{x}_6)}$

$\downarrow$

another way to write (used in Jordan's textbook) $P(X_1 \mid X_6 = \bar{x}_6)$

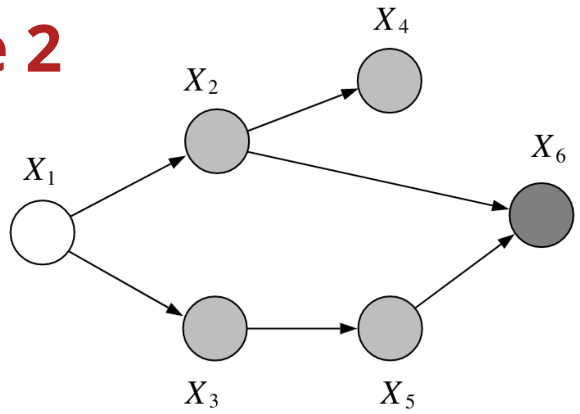
- calculate the numerator
- denominator is then easy

$$p(\bar{x}_6) = \sum_{x_1} p(x_1, \bar{x}_6)$$



source: Michael Jordan's book

Inference: example 2

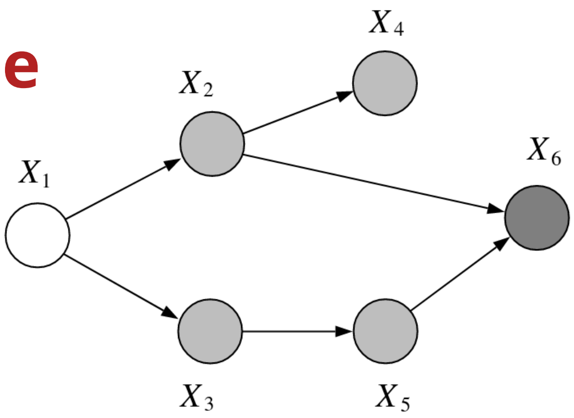


$$
\begin{aligned}
p(x_1, \bar{x}_6) &= \sum_{x_2}\sum_{x_3}\sum_{x_4}\sum_{x_5} p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1)p(x_4 \mid x_2)p(x_5 \mid x_3)p(\bar{x}_6 \mid x_2, x_5) \\
&= p(x_1)\sum_{x_2} p(x_2 \mid x_1)\sum_{x_3} p(x_3 \mid x_1)\sum_{x_4} p(x_4 \mid x_2)\sum_{x_5} p(x_5 \mid x_3)p(\bar{x}_6 \mid x_2, x_5) \\
&= p(x_1)\sum_{x_2} p(x_2 \mid x_1)\sum_{x_3} p(x_3 \mid x_1)\sum_{x_4} p(x_4 \mid x_2)m_5(x_2, x_3)
\end{aligned}
$$

$\mathcal{O}(d^3)$

source: Michael Jordan's book

Inference: **example**



$$p(x_1, \bar{x}_6) = p(x_1) \sum_{x_2} p(x_2 \,|\, x_1) \sum_{x_3} p(x_3 \,|\, x_1) \sum_{x_4} p(x_4 \,|\, x_2) m_5(x_2, x_3)$$

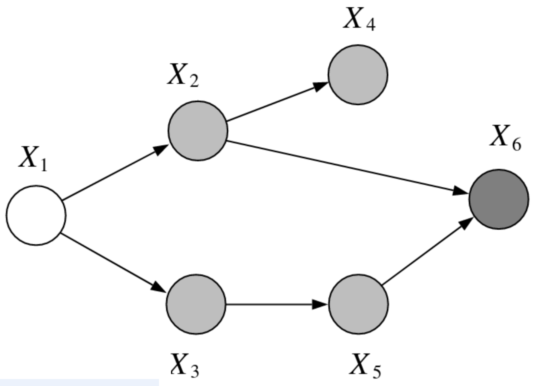
$$= p(x_1) \sum_{x_2} p(x_2 \,|\, x_1) \sum_{x_3} p(x_3 \,|\, x_1) m_5(x_2, x_3) \sum_{x_4} p(x_4 \,|\, x_2)$$

$$= p(x_1) \sum_{x_2} p(x_2 \,|\, x_1) m_4(x_2) \sum_{x_3} p(x_3 \,|\, x_1) m_5(x_2, x_3).$$

$\mathcal{O}(d^2)$

source: Michael Jordan's book

Inference: example



$$p(x_1, \bar{x}_6) = p(x_1) \sum_{x_2} p(x_2 \mid x_1) \sum_{x_3} p(x_3 \mid x_1) \sum_{x_4} p(x_4 \mid x_2) m_5(x_2, x_3)$$

$$= p(x_1) \sum_{x_2} p(x_2 \mid x_1) \sum_{x_3} p(x_3 \mid x_1) m_5(x_2, x_3) \sum_{x_4} p(x_4 \mid x_2)$$

is constant

$$= p(x_1) \sum_{x_2} p(x_2 \mid x_1) m_4(x_2) \sum_{x_3} p(x_3 \mid x_1) m_5(x_2, x_3).$$

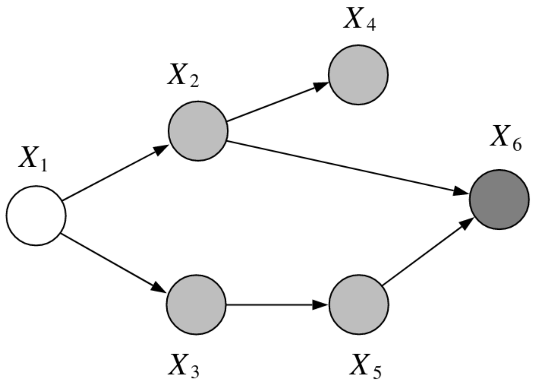
$\mathcal{O}(d^3)$

$$= p(x_1) \sum_{x_2} p(x_2 \mid x_1) m_4(x_2) m_3(x_1, x_2)$$

$\mathcal{O}(d^2)$

$$= p(x_1) m_2(x_1).$$

Inference: example



overall complexity $\mathcal{O}(d^3)$ instead of $\mathcal{O}(d^5)$

if we had built the 5d array of

$$p(x_1, x_2, x_3, x_4, x_5 \mid \bar{x}_6)$$

in the general case $\mathcal{O}(d^n)$

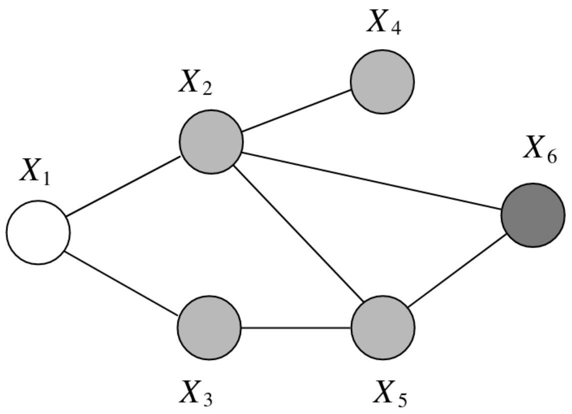
Inference: example *(undirected version)*

$$p(x_1, \bar{x}_6) = \frac{1}{Z} \sum_{x_2,...,x_5} \phi(x_1, x_2)\phi(x_1, x_3)\phi(x_2, x_3)\phi(x_3, x_5)\phi(x_2, x_5, x_6)\delta(x_6, \bar{x}_6)$$

using a delta-function for conditioning

$$\delta(x_6, \bar{x}_6) \triangleq \begin{cases} 1, & \text{if } x_6 = \bar{x}_6 \\ 0, & \text{otherwise} \end{cases}$$

add it as a local potential



Inference: example *(undirected version)*

every step remains the same

$$p(x_1, \bar{x}_6) = \frac{1}{Z} \sum_{x_2,...,x_5} \phi(x_1, x_2)\phi(x_1, x_3)\phi(x_2, x_3)\phi(x_3, x_5)\phi(x_2, x_5, x_6)\delta(x_6, \bar{x}_6)$$

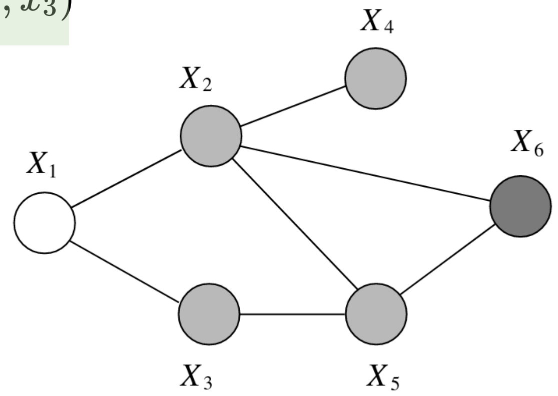
$$= \frac{1}{Z} \sum_{x_2,...,x_5} \phi(x_1, x_2)\phi(x_1, x_3)\phi(x_2, x_3)\phi(x_3, x_5)m_6(x_2, x_5)$$

$$\cdots$$

$$= \frac{1}{Z} \sum_{x_2} \phi(x_1, x_2) \ldots, m_4(x_2) \sum_{x_3} \phi(x_1, x_3)m_5(x_2, x_3)$$

$$= \frac{1}{Z} \sum_{x_2} \phi(x_1, x_2) \ldots, m_4(x_2)m_3(x_1, x_2)$$

$$= \frac{1}{Z} m_2(x_1)$$



except: in Bayes-nets Z=1

- *at this point normalization is easy!*

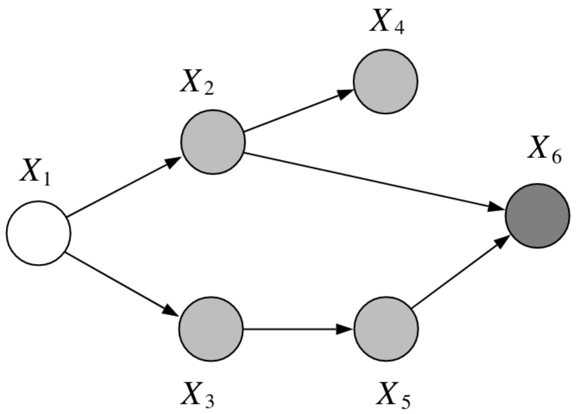
Variable elimination

- input: $\Phi^{t=0} = \{\phi_1, \ldots, \phi_K\}$ a set of factors *(e.g. CPDs)*
- output: $\sum_{x_{i_1},\ldots,x_{i_m}} \prod_k \phi_k(\mathbf{D}_k)$
- go over $x_{i_1}, \ldots, x_{i_m}$ in some order:
  - collect all the relevant factors: $\Psi^t = \{\phi \in \Phi^t \mid x_{i_t} \in Scope[\phi]\}$
  - calculate their product: $\psi_t = \prod_{\phi \in \Psi^t} \phi$
  - marginalize out x_{i_t}: $\psi'_t = \sum_{x_{i_t}} \psi_t$
  - update the set of factors: $\Phi^t = \Phi^{t-1} - \Psi^t + \{\psi'_t\}$
- return the product of factors in $\Phi^{t=m}$

Variable elimination: example

- input: $\Phi^{t=0} = \{\phi_1, \ldots, \phi_K\}$ a set of factors *(e.g. CPDs)*

$$\Phi^0 = \{p(x_2 \mid x_1), p(x_3 \mid x_1), p(\bar{x}_6 \mid x_2, x_5), p(x_4 \mid x_2), p(x_5 \mid x_3)\}$$



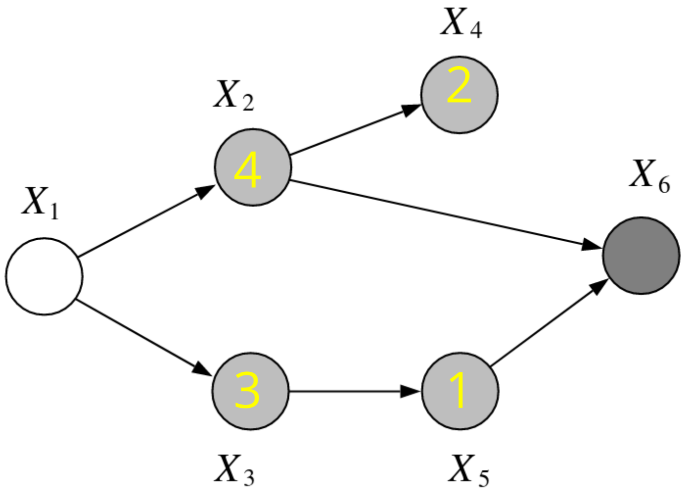
- output: $\sum_{x_{i_1}, \ldots, x_{i_m}} \prod_k \phi_k(\mathbf{D}_k)$

$$p(x_1, \bar{x}_6) = \sum_{x_2} \sum_{x_3} \sum_{x_4} \sum_{x_5} p(x_1) p(x_2 \mid x_1) p(x_3 \mid x_1) p(x_4 \mid x_2) p(x_5 \mid x_3) p(\bar{x}_6 \mid x_2, x_5)$$

Variable elimination: example

- go over $x_{i_1}, \ldots, x_{i_m}$ in some order:

$$x_5, x_4, x_3, x_2$$

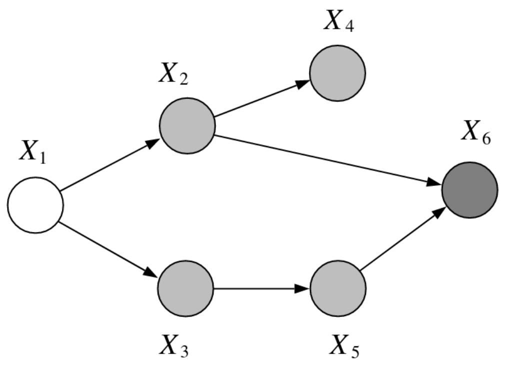


Variable elimination: example

- for x_5 :
  - collect all the relevant factors $\Psi^t = \{\phi \in \Phi^t \mid x_{i_t} \in Scope[\phi]\}$
  - calculate their product $\psi_t = \prod_{\phi \in \Psi^t} \phi$

$$\Psi^0 = \{p(\bar{x}_6 \mid x_2, x_5), p(x_5 \mid x_3)\}$$

$$\psi_t(x_2, x_3, x_5) = p(\bar{x}_6 \mid x_2, x_5) p(x_5 \mid x_3)$$



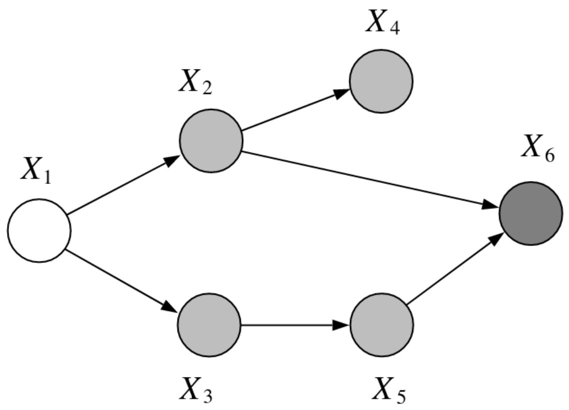
Variable elimination: example

- for x_5 :
  - collect all the relevant factors $\Psi^t = \{\phi \in \Phi^t \mid x_{i_t} \in Scope[\phi]\}$
  - calculate their product $\psi_t = \prod_{\phi \in \Psi^t} \phi$
  - marginalize out x_5

$$\Psi^0 = \{p(\bar{x}_6 \mid x_2, x_5), p(x_5 \mid x_3)\}$$

$$\psi_t(x_2, x_3, x_5) = p(\bar{x}_6 \mid x_2, x_5) p(x_5 \mid x_3)$$

$$\psi'_t(x_2, x_3) = \sum_{x_5} \psi_t(x_2, x_3, x_5)$$

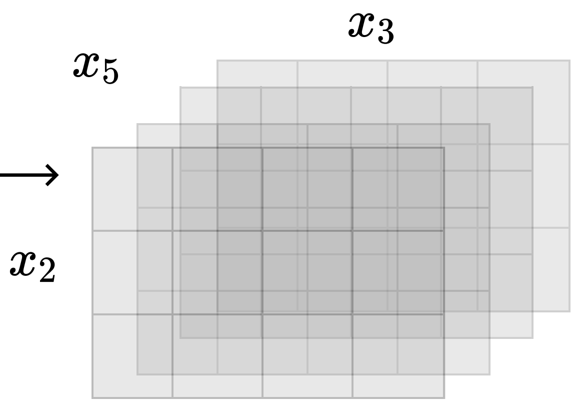


Variable elimination: example

- for x_5 :
  - collect all the relevant factors $\Psi^t = \{\phi \in \Phi^t \mid x_{i_t} \in Scope[\phi]\}$
  - calculate their product $\psi_t = \prod_{\phi \in \Psi^t} \phi$
  - marginalize out x_5

$$\Psi^0 = \{p(\bar{x}_6 \mid x_2, x_5), p(x_5 \mid x_3)\}$$

$$\psi_t(x_2, x_3, x_5) = p(\bar{x}_6 \mid x_2, x_5) p(x_5 \mid x_3) \longrightarrow$$



$$\psi'_t(x_2, x_3) = \sum_{x_5} \psi_t(x_2, x_3, x_5)$$

Variable elimination: example

- for x_5 :
  - collect all the relevant factors $\Psi^t = \{\phi \in \Phi^t \mid x_{i_t} \in Scope[\phi]\}$
  - calculate their product $\psi_t = \prod_{\phi \in \Psi^t} \phi$
  - marginalize out x_5
  - update the set of factors $\Phi^t = \Phi^{t-1} - \Psi^t + \{\psi'_t\}$

$$\psi'_t(x_2, x_3) = \sum_{x_5} \psi_t(x_2, x_3, x_5)$$

$$\Phi^0 = \{p(x_2 \mid x_1), p(x_3 \mid x_1), p(x_4 \mid x_2), p(\bar{x}_6 \mid x_2, x_5), p(x_5 \mid x_3)\}$$

$$\downarrow$$

$$\Phi^1 = \{p(x_2 \mid x_1), p(x_3 \mid x_1), p(x_4 \mid x_2), \psi'_t(x_2, x_3)\}$$

Variable elimination: example

- for x_5 :
  - collect all the relevant factors $\Psi^t = \{\phi \in \Phi^t \mid x_{i_t} \in Scope[\phi]\}$
  - calculate their product $\psi_t = \prod_{\phi \in \Psi^t} \phi$
  - marginalize out x_5
  - update the set of factors $\Phi^t = \Phi^{t-1} - \Psi^t + \{\psi'_t\}$

$$\Phi^1 = \{p(x_2 \mid x_1), p(x_3 \mid x_1), p(x_4 \mid x_2), \psi'_t(2,3)\}$$

repeat for x_4, x_3, x_2

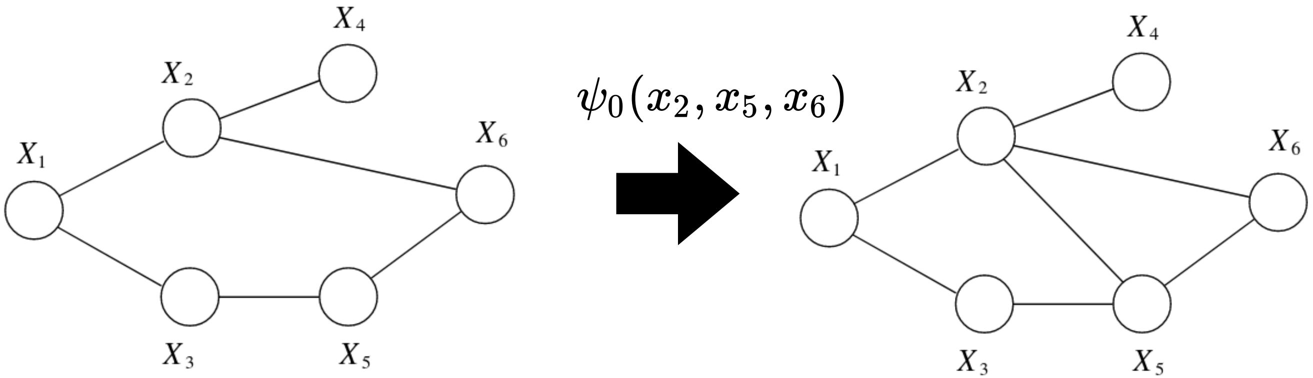
Variable elimination: example

calculating $p(x_1)$: following the graph

using the order x_6, x_5, x_4, x_3, x_2

$\Phi^0 = \{p(x_2 \mid x_1), p(x_3 \mid x_1), p(x_6 \mid x_2, x_5), p(x_4 \mid x_2), p(x_5 \mid x_3)\}$

t=1



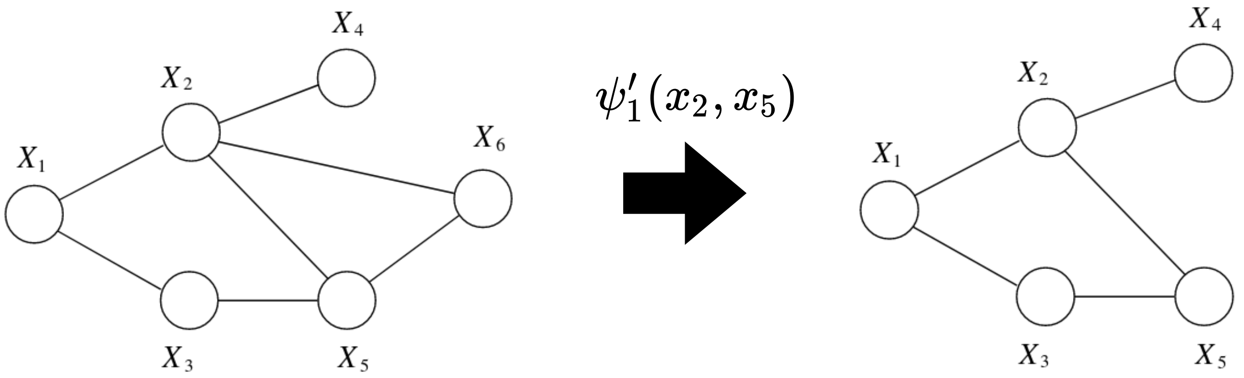
Variable elimination: example

calculating $p(x_1)$

using the order x_6, x_5, x_4, x_3, x_2

$\Phi^1 = \{p(x_2 \mid x_1), p(x_3 \mid x_1), \psi'_1(x_2, x_5), p(x_4 \mid x_2), p(x_5 \mid x_3)\}$

t=1



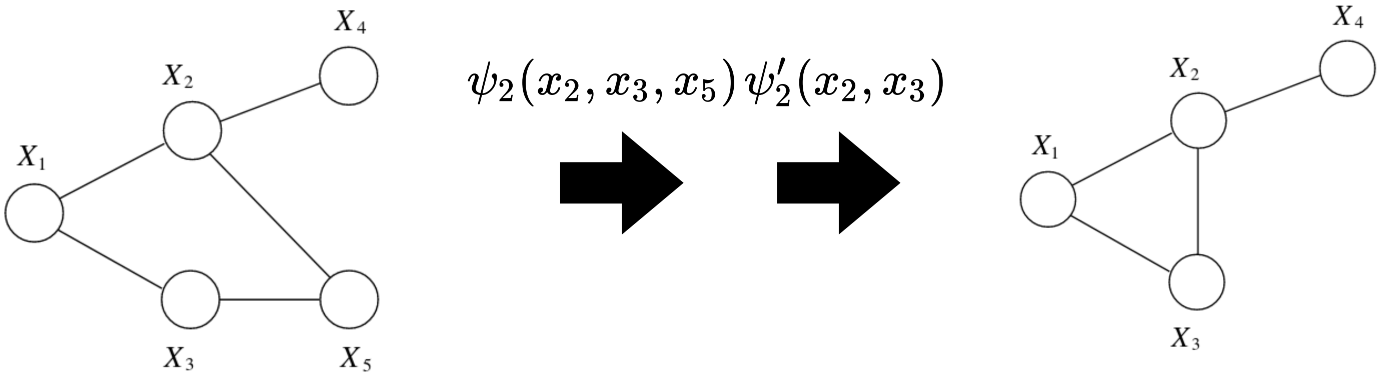
Variable elimination: example

calculating $p(x_1)$

using the order x_6, x_5, x_4, x_3, x_2

$\Phi^1 = \{p(x_2 \mid x_1), p(x_3 \mid x_1), \psi_1'(x_2, x_5), p(x_4 \mid x_2), p(x_5 \mid x_3)\}$

t=2



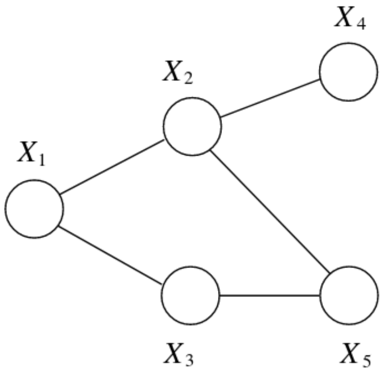
Variable elimination: example

calculating $p(x_1)$

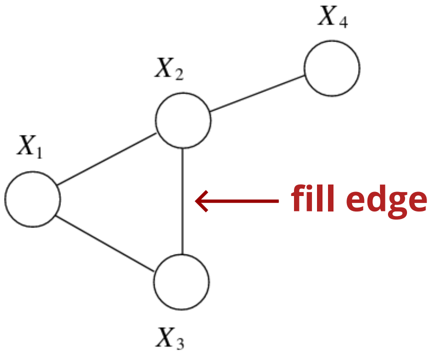
using the order x_6, x_5, x_4, x_3, x_2

$\Phi^2 = \{p(x_2 \mid x_1), p(x_3 \mid x_1), \psi_2'(x_2, x_3), p(x_4 \mid x_2)\}$

t=2



$\psi_2(x_2, x_3, x_5)\,\psi_2'(x_2, x_3)$



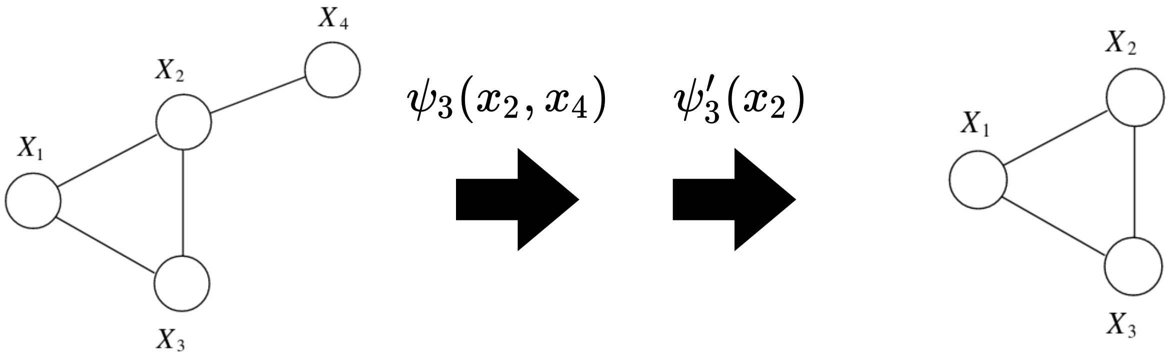
Variable elimination: example

calculating $p(x_1)$

using the order x_6, x_5, x_4, x_3, x_2

$\Phi^2 = \{p(x_2 \mid x_1), p(x_3 \mid x_1), \psi'_2(x_2, x_3), p(x_4 \mid x_2)\}$

t=3



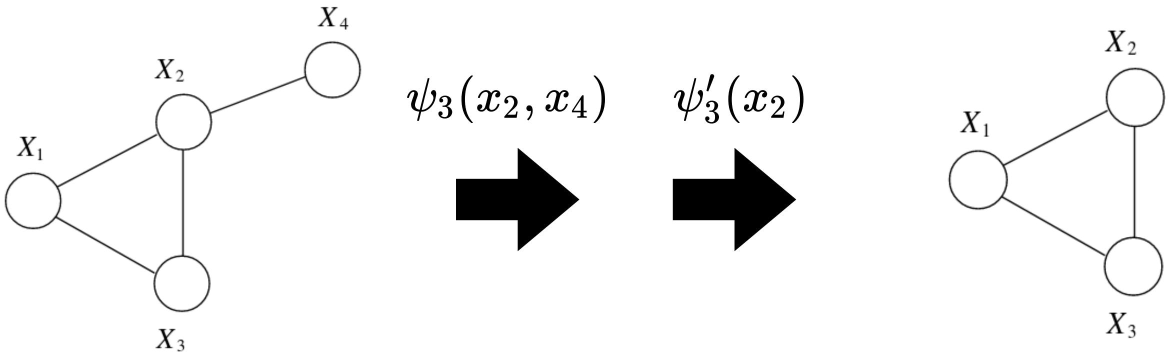
Variable elimination: example

calculating $p(x_1)$

using the order x_6, x_5, x_4, x_3, x_2

$\Phi^3 = \{p(x_2 \mid x_1), p(x_3 \mid x_1), \psi'_2(x_2, x_3), \psi'_3(x_2)\}$

t=3



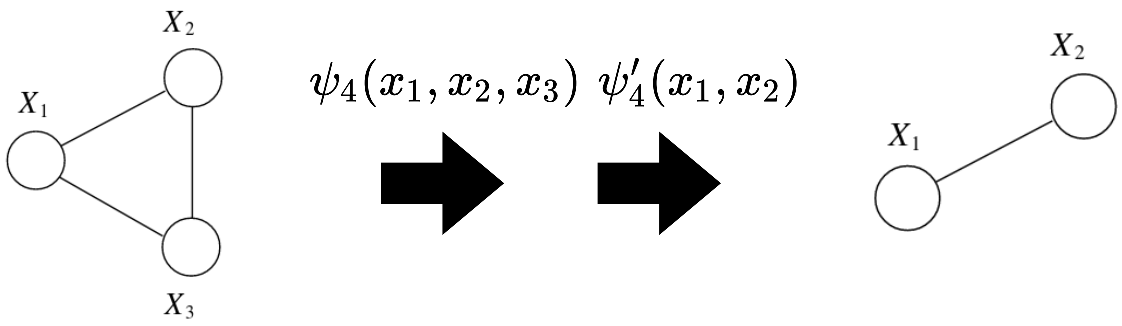
Variable elimination: example

calculating $p(x_1)$

using the order x_6, x_5, x_4, x_3, x_2

$\Phi^3 = \{p(x_2 \mid x_1), p(x_3 \mid x_1), \psi_2'(x_2, x_3), \psi_3'(x_2)\}$

t=4



$\psi_4(x_1, x_2, x_3)$ $\psi_4'(x_1, x_2)$

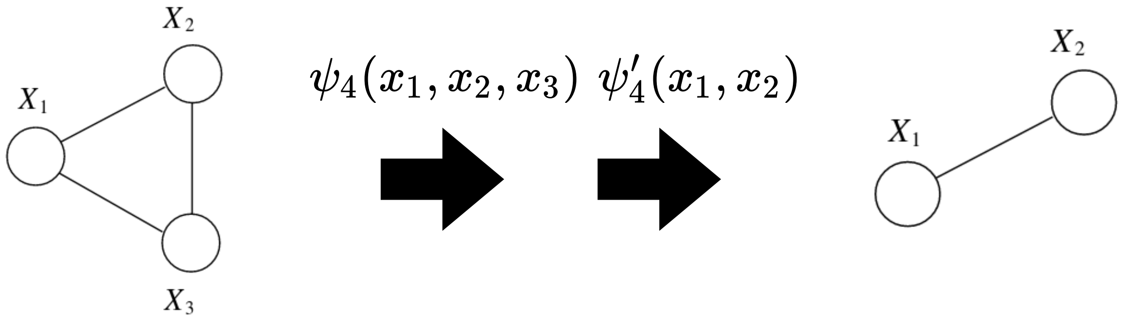
Variable elimination: **example**

calculating $p(x_1)$

using the order x_6, x_5, x_4, x_3, x_2

$\Phi^4 = \{p(x_2 \mid x_1), \psi_3'(x_2), \psi_4'(x_1, x_2)\}$

t=4



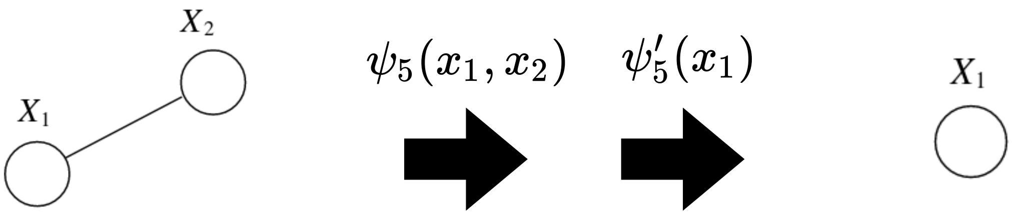
Variable elimination: example

calculating $p(x_1)$

using the order x_6, x_5, x_4, x_3, x_2

$\Phi^4 = \{p(x_2 \mid x_1), \psi'_3(x_2), \psi'_4(x_1, x_2)\}$

t=5



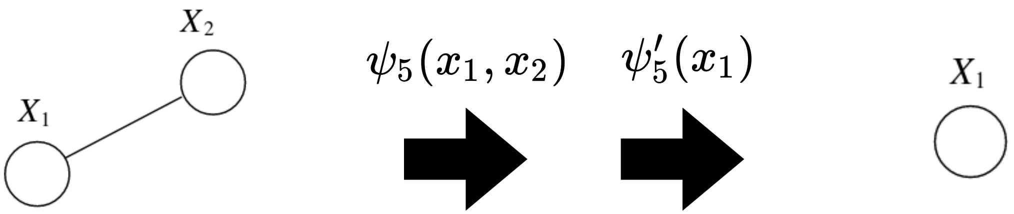
Variable elimination: example

calculating $p(x_1)$

using the order x_6, x_5, x_4, x_3, x_2

$\Phi^5 = \{\psi'_5(x_1)\}$

t=5

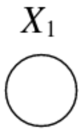


Variable elimination: example

$$p(x_1) = \frac{1}{Z} \sum_{x_2,...,x_6} \phi(x_1, x_2)\phi(x_1, x_3)\phi(x_2, x_3)\phi(x_3, x_5)\phi(x_2, x_5, x_6)$$

at final iteration: $\Phi^5 = \{\psi'_5(x_1)\}$

the **marginal** of interest $p(x_1) = \frac{1}{Z}\psi'_5(x_1)$



One more elimination step: $\Phi^6 = \{\psi'_6(\emptyset) = Z\}$

- gives the **partition function** $Z = \sum_{x_1} \psi'_5(x_1)$

Complexity

- go over $x_{i_1}, \ldots, x_{i_m}$ in some order:
  - collect all the relevant factors: $\Psi^t = \{\phi \in \Phi^t \mid x_{i_t} \in Scope[\phi]\}$
  - calculate their product: $\psi_t = \prod_{\phi \in \Psi^t} \phi$
  - marginalize out x_{i_t} : $\psi'_t = \sum_{x_{i_t}} \psi_t$
  - update the set of factors: $\Phi^t = \Phi^{t-1} - \Psi^t + \{\psi'_t\}$

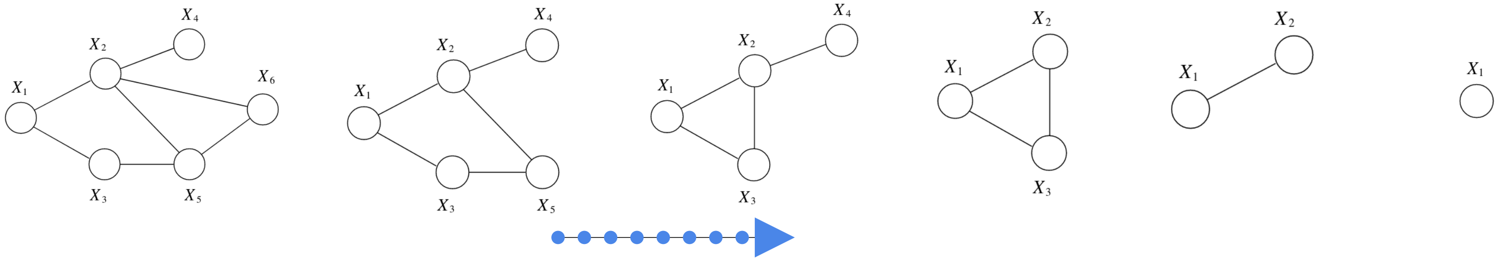
complexity: number of vars in ψ_t: $\mathcal{O}(max_t\, d^{|Scope[\psi_t]|})$

- depends on the ***graph structure***

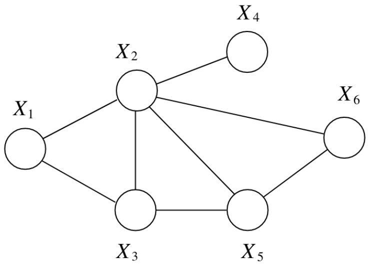
Induced graph

complexity of step t: number of vars in ψ_t $\mathcal{O}(\, d^{|Scope[\psi_t]|})$

- depends on the ***graph structure***



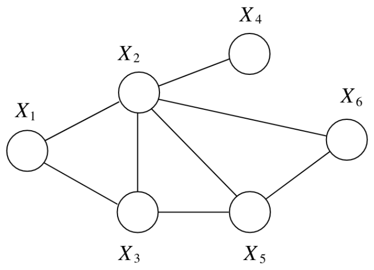
induced graph



- add edges created during the elimination
- maximal cliques correspond to $\psi_t \quad \forall t$

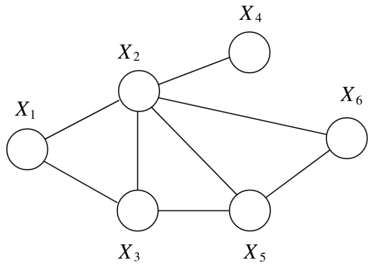
Induced graph

- maximal cliques correspond to some ψ_t **why?**
  - *take one such clique - e.g.,* $\{X_2, X_3, X_5\}$
  - *take the first to be eliminated - e.g.,* X_5
  - *all the edges to* X_5 *exist* ***before*** *its elimination*
  - *therefore, during elimination of* X_5 *we create a factor with* $Scope[\psi_t] = \{X_2, X_3, X_5\}$



Induced graph

- maximal cliques correspond to some ψ_t **why?**
  - *take one such clique - e.g.,* $\{X_2, X_3, X_5\}$
  - *take the first to be eliminated - e.g.,* X_5
  - *all the edges to* X_5 *exist* ***before*** *its elimination*
  - *therefore, during elimination of* X_5 *we create a factor with* $Scope[\psi_t] = \{X_2, X_3, X_5\}$

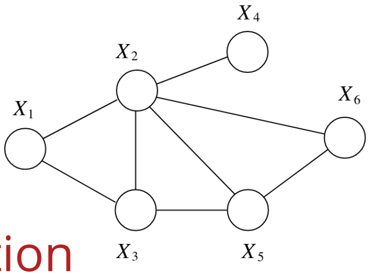


- in fact, the induced graph is **chordal** all the loops > 3 have a *chord*
  - *suppose there is a loop of length > 3:* $X_1 - X_2 - \ldots - X_l - X_1$
  - *suppose X1 is eliminated first*
  - *since no edge is added to X1 after its elimination* $X_l - X_1 - X_2$ *exists before its elimination*
  - *eliminating X1 creates the edge* $X_l - X2$

Tree-width

maximal cliques correspond to ψ_t

cost of marginalizing ψ_t is $\mathcal{O}(d^{|Scope[\psi_t]|})$



largest clique dominates the cost of variable elimination

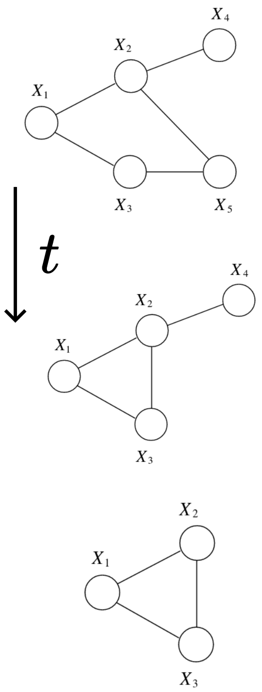
the tree-width $\min_{\text{orderings}} \max_{\psi_t} scope[\psi_t] - 1$

- tree-width of a tree = 1
- NP-hard to calculate the tree-width
- use heuristics to find good orderings

Ordering heuristics

choose the next vertex to eliminate by:

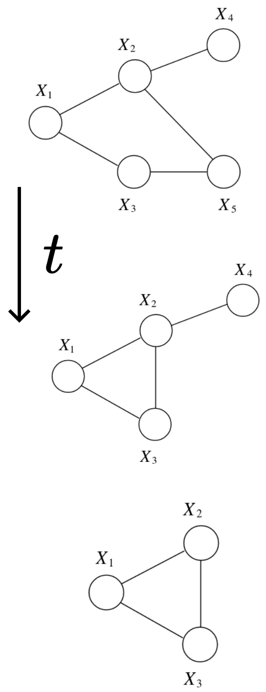
- minimizing the effect of the created clique/factor
  - **min-neighbours:** #neigbours in the current graph
  - **min-weight:** product of cardinality of neighbours



Ordering heuristics

choose the next vertex to eliminate by:

- minimizing the effect of the created clique/factor
  - **min-neighbours:** #neigbours in the current graph
  - **min-weight:** product of cardinality of neighbours
- minimizing the effect of fill edges
  - **min-fill:** number of fill-edges after its elimination
  - **weighted min-fill:** edges are weighted by the product of the cardinality of the two vertices

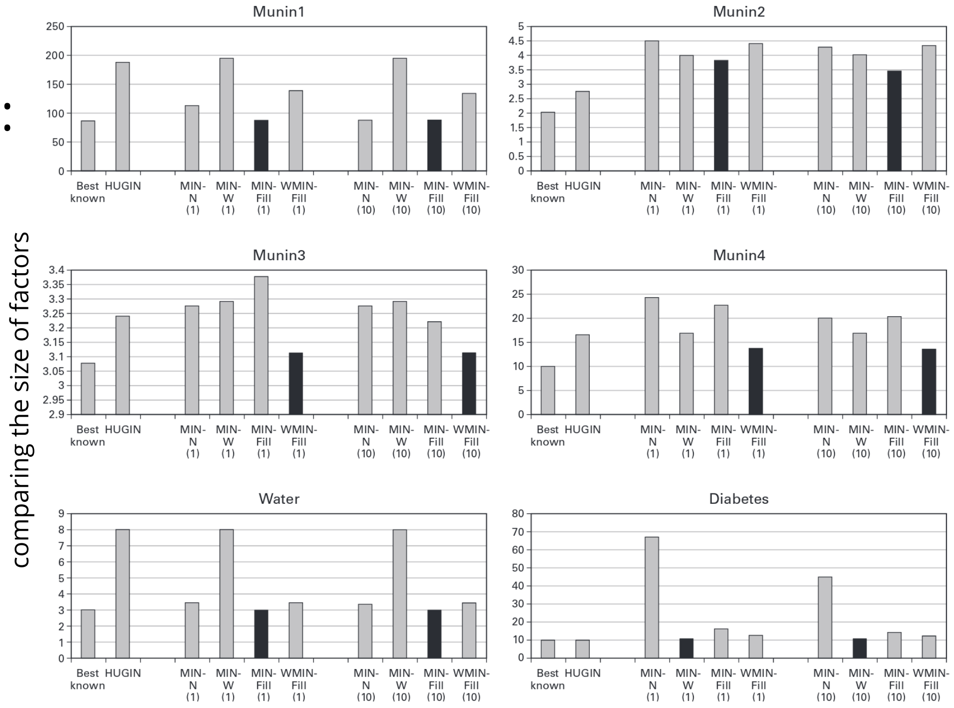


Ordering heuristics

minimizing the #fill edges tends to work better in practice

to minimize the cost one could:

- try different heuristics
- calculate the max-clique size
- pick the best ordering
- apply variable elimination



Answering other queries

we saw variable elimination (VE) for marginalization

$$P(X_1) = \sum_{x_2,...,x_n} P(X_1, X_2 = x_2, \ldots, X_n = x_n)$$

Introducing evidence leads to *a similar* problem

$$P(X_1 \mid X_m = x_m) = \frac{P(X_1, X_m = x_m)}{P(X_m = x_m)}$$

- use VE to get $P(X_1, X_m = x_m)$
- marginalize this to get $P(X_m = x_m)$
- devide!

Answering other queries

we saw variable elimination (VE) for marginalization

$$P(X_1 = x_1) = \sum_{x_2,\ldots,x_n} P(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n)$$

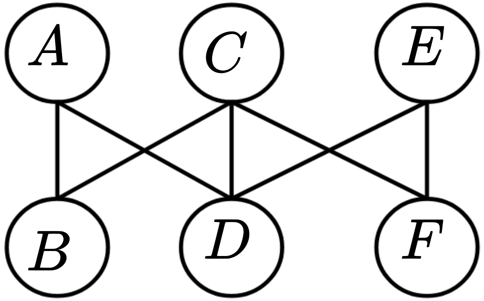
MAP inference: sum $\longrightarrow$ max

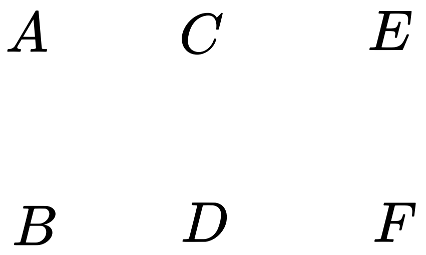
$$Q(X_1 = x_1) = \max_{x_2,\ldots,x_n} P(X_1, X_2 = x_2, \ldots, X_n = x_n)$$

- run VE with maximization instead of summation
- eliminating ALL the variables gives a single value $\max_{\mathbf{x}} P(\mathbf{X} = \mathbf{x})$
- we can also get the maximizing assignment (later!) $\arg\max_{\mathbf{x}} P(\mathbf{X} = \mathbf{x})$

quiz: induced graph

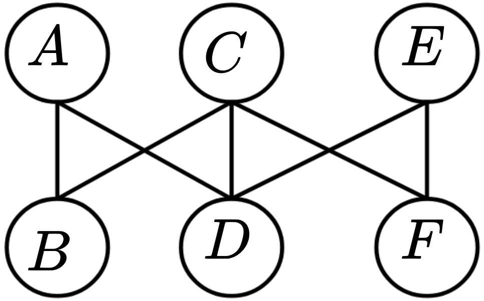
what are the fill-edges corresponding to the following elimination order? A, B, C, D, E, F

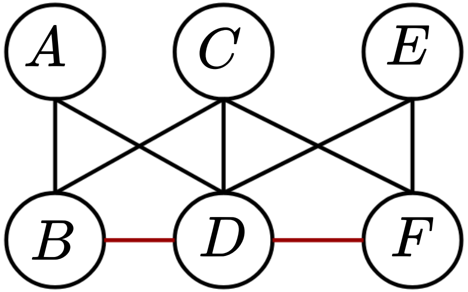




quiz: induced graph

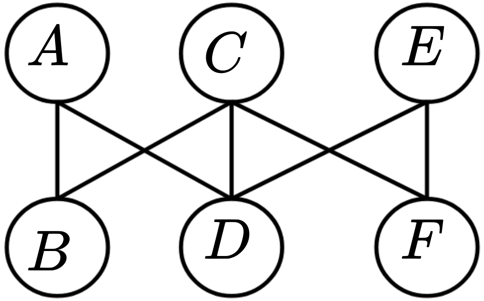
what are the fill-edges corresponding to the following elimination order? A, B, C, D, E, F



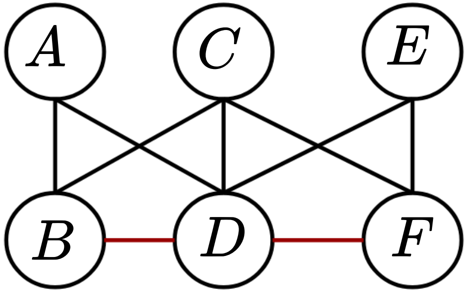


quiz: induced graph

what are the fill-edges corresponding to the following elimination order? A, B, C, D, E, F



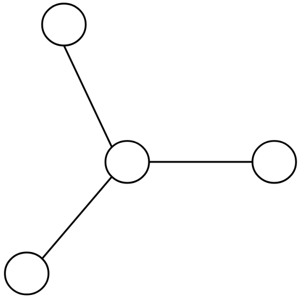
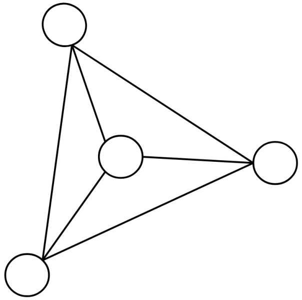
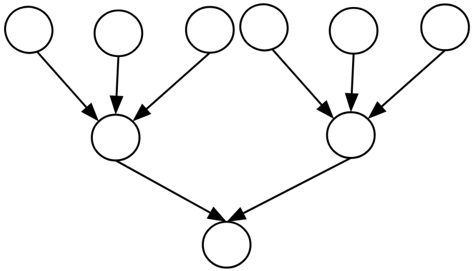
is this graph chordal?



how about this one?

quiz: tree width

what is the tree-width in these graphical models?

Summary

- inference in graphical models is NP-hard
  - even approximating it is NP-hard
- brute-force inference has an exponential cost
- use the graph structure + distributive law:
  - variable elimination algorithm
  - cost grows with the tree-width of the graph
  - NP-hard to calculate the tree-width / optimal ordering
  - use heuristics