CS 583: Probabilistic Graphical Models

Topic: Variable Elimination

Mustafa Bilgic

http://www.cs.iit.edu/~mbilgic

https://twitter.com/bilgicm

Task

- Given a graphical model over $\mathcal{X}$ (structure and parameters)
- Compute P($\boldsymbol{Y}$ | $\boldsymbol{e}$), where $\boldsymbol{Y} \subseteq \mathcal{X}$ and $\boldsymbol{E} \subseteq \mathcal{X}$
- There are several approaches
  - Exact inference
    - Variable elimination
    - Belief propagation
  - Approximate inference
    - Sampling
- Today, we'll cover variable elimination

Variable elimination

- $P(\boldsymbol{Y} \mid \boldsymbol{e}) = P(\boldsymbol{Y}, \boldsymbol{e}) / P(\boldsymbol{e})$
- $\boldsymbol{W} = \mathcal{X} - \boldsymbol{Y} - \boldsymbol{E}$
- $P(\boldsymbol{y}, \boldsymbol{e}) = \Sigma_{\boldsymbol{w}} P(\boldsymbol{y}, \boldsymbol{e}, \boldsymbol{w})$
- $P(\boldsymbol{e}) = \Sigma_{\boldsymbol{y},\boldsymbol{w}} P(\boldsymbol{y}, \boldsymbol{e}, \boldsymbol{w})$
- Or, better yet: $P(\boldsymbol{e}) = \Sigma_{\boldsymbol{y}} P(\boldsymbol{y}, \boldsymbol{e})$

$P(\boldsymbol{Y}, \boldsymbol{E}) = \Sigma_{\boldsymbol{W}} P(\boldsymbol{Y}, \boldsymbol{E}, \boldsymbol{W})$

- $P(\boldsymbol{Y}, \boldsymbol{E}, \boldsymbol{W})$ can be represented as
  - $\prod P(X_i \mid Pa(X_i))$
  - $1/Z \prod \phi(\boldsymbol{D}_i)$
- The problem with $P(\boldsymbol{y}, \boldsymbol{e}) = \Sigma_{\boldsymbol{w}} P(\boldsymbol{y}, \boldsymbol{e}, \boldsymbol{w})$ is that the joint representation is exponential
  - The very first problem we were trying to avoid

COMPLEXITY

- Unfortunately, exact inference is $\mathcal{NP}$-hard in worst case
  - Proof: pages 288 and 289. Reduction from 3-SAT
- Approximate inference is also $\mathcal{NP}$-hard in worst case
  - Proof: pages 291 and 292.
- Good news:
  - In general, we care about the cases we encounter in practice, not the worst-case scenario

KEY IDEA

- Summation can be moved inside
- $\Sigma_x \Sigma_y\, x*y = \Sigma_x x*(\Sigma_y\, y)$
- If x has n and y has m possible values, how many operations are needed, if we use
  - $\Sigma_x \Sigma_y\, x*y$?
  - $\Sigma_x x*(\Sigma_y\, y)$?

Outline

- First, focus on Bayesian networks
  - Simple linear chains
  - More complex structures
- Two cases
  - Marginal queries: $\boldsymbol{E} = \varnothing$
  - Conditional queries: $\boldsymbol{E} \neq \varnothing$

Variable elimination

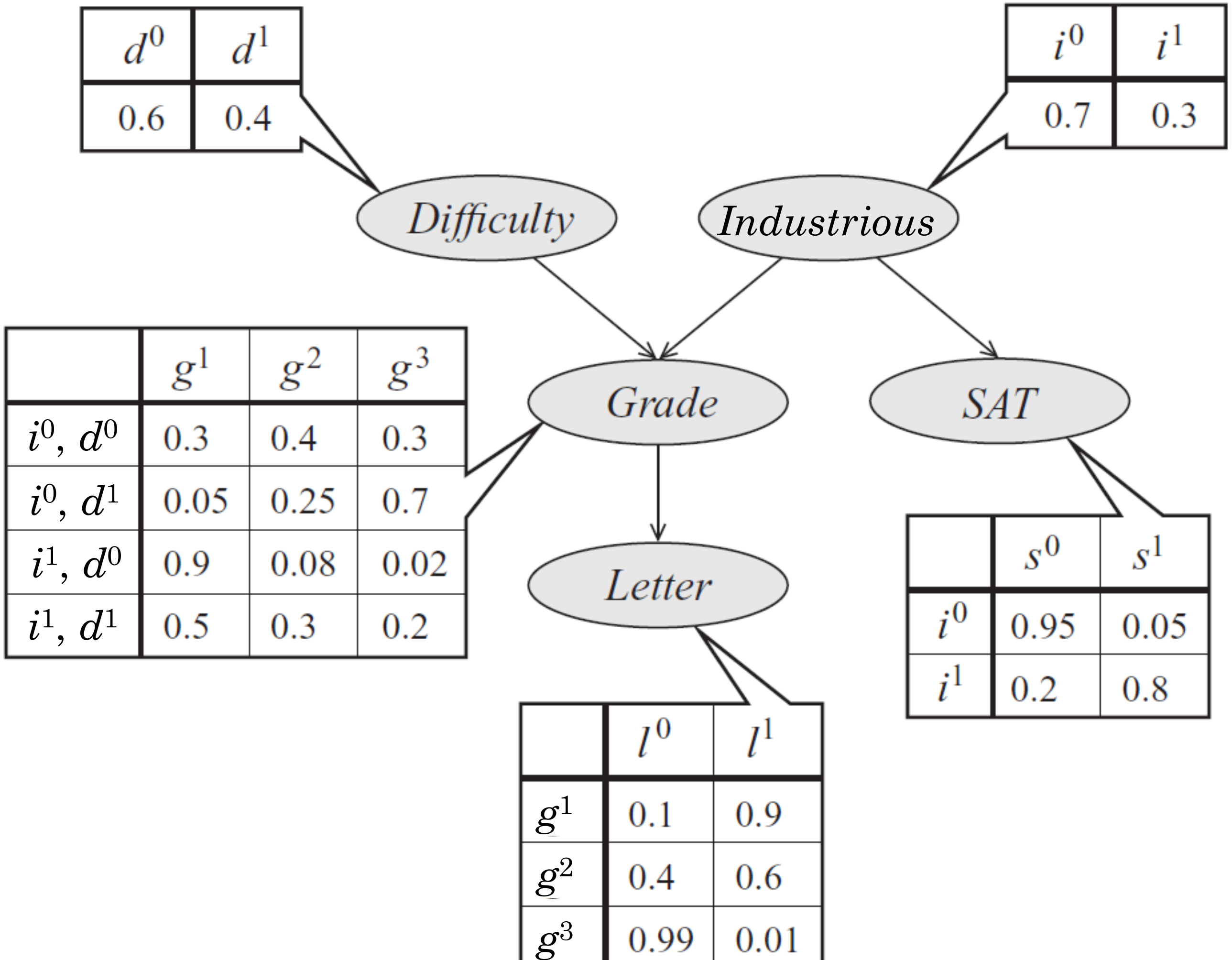
- $\mathcal{X}$: all variables, $\boldsymbol{Y}$: query variables, $\boldsymbol{E}$: evidence variables, $\boldsymbol{W} = \mathcal{X} - \boldsymbol{Y} - \boldsymbol{E}$: remaining variables

1. Write down the joint for P($\mathcal{X}$)
2. Set $X_i \in \boldsymbol{E}$ to their values
3. Pick an order for $X_j \in \boldsymbol{W}$
4. Sum out each X_j from the joint
   a) Multiply the factors $\phi(X_j, Z_1), \ldots, \phi(X_j, Z_k)$ to create $\psi(X_j, Z_1,\ldots, Z_k)$
   b) Sum out X_j from $\psi(X_j, Z_1,\ldots, Z_k)$ to create $\tau(Z_1,\ldots, Z_k)$
5. What remains is $\tau(\boldsymbol{Y}, \boldsymbol{e})$. Normalize it to get $P(\boldsymbol{Y} \mid \boldsymbol{e})$.

EXAMPLES – LINEAR CHAIN BNS

- $A \rightarrow B$
  - P(B) =?
  - P(A) =?
- $X_1 \rightarrow X_2 \rightarrow \ldots \rightarrow X_n$
  - P(X_i) where $1 \leq i \leq n$
- How many operations are needed if we compute the full joint distribution vs. if we use variable elimination?

Student network example



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

	g^1	g^2	g^3
i^0, d^0	0.3	0.4	0.3
i^0, d^1	0.05	0.25	0.7
i^1, d^0	0.9	0.08	0.02
i^1, d^1	0.5	0.3	0.2

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

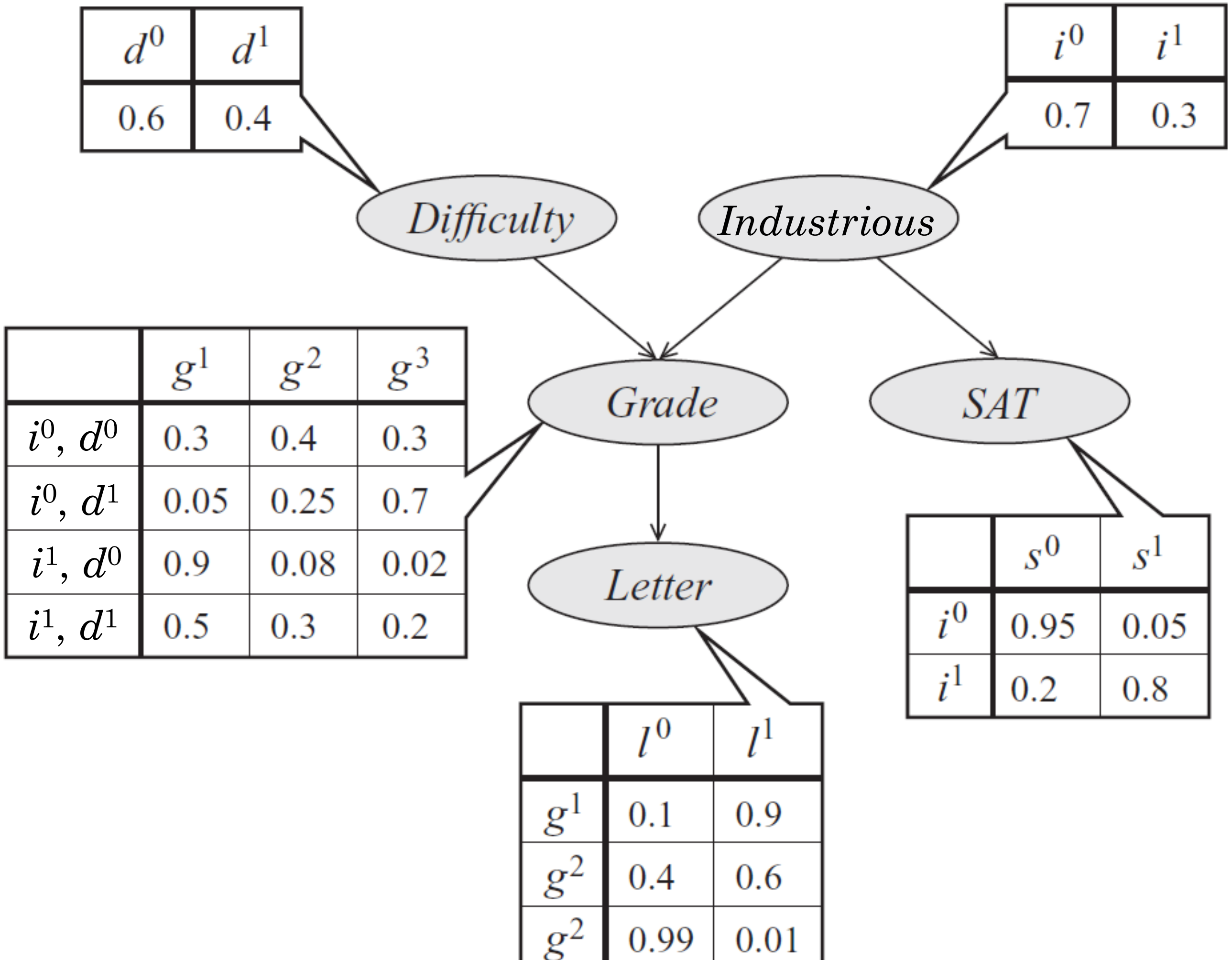
	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^3	0.99	0.01

P(D)
P(I)
P(S)
P(G)
P(L)

P($\boldsymbol{Y}$ | $\boldsymbol{E}$) EXAMPLES

- $A \rightarrow B$
  - P(B | A=t)
  - P(A=t | B=t)
- $X_1 \rightarrow X_2 \rightarrow \ldots \rightarrow X_n$
  - P(X_i | X_j=x_j)
  - P(X_i | $\boldsymbol{X_j}$=$\boldsymbol{x_j}$)

Student network example



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

	g^1	g^2	g^3
i^0, d^0	0.3	0.4	0.3
i^0, d^1	0.05	0.25	0.7
i^1, d^0	0.9	0.08	0.02
i^1, d^1	0.5	0.3	0.2

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^2	0.99	0.01

$P(S \mid I=i^0)$
$P(D \mid I=i^0)$
$P(I \mid G=g^1)$
$P(D \mid G=g^1)$
$P(D \mid I=i^0, G=g^1)$

P(L) – ORDER: I, S, D, G

Variable	All Factors	Participates	New Factor After *	# *s	New Factor After +	# +s	# Ops
I	P(I), P(D), P(S | I), P(G | D,I), P(L | G)	P(I), P(S | I), P(G | D, I)	ψ_1(G,D,S,I)	2*3*2*2*2= 48	τ_1(G,D,S)	1*3*2*2=12	60
S	P(D), P(L | G), τ_1(G,D,S)	τ_1(G,D,S)	ψ_2(G,D,S)	0	τ_2(G,D)	1*3*2=6	6
D	P(D), P(L | G), τ_2(G,D)	P(D), τ_2(G,D)	ψ_3(G,D)	1*3*2=6	τ_3(G)	1*3	9
G	P(L | G), τ_3(G)	P(L | G), τ_3(G)	ψ_4(L,G)	1*2*3=6	τ_4(L)	2*2=4	10
Normalize	τ_4(L)					1	3 (2 divs)
Total							88

P(L) – ORDER: S, I, D, G

Variable	All Factors	Participates	New Factor After *	# *s	New Factor After +	# +s	# Ops
S	P(I), P(D), P(S | I), P(G | D,I), P(L | G)	P(S | I)	ψ_1(I,S)	0	τ_1(I)	1*2=2	2
I	P(I), P(D), P(G | D,I), P(L | G) τ_1(I)	P(I), P(G | D,I), τ_1(I)	ψ_2(G,D,I)	2*3*2*2=24	τ_2(G,D)	1*3*2=6	30
D	P(D), P(L | G), τ_2(G,D)	P(D), τ_2(G,D)	ψ_3(G,D)	1*3*2=6	τ_3(G)	1*3	9
G	P(L | G), τ_3(G)	P(L | G), τ_3(G)	ψ_4(L,G)	1*2*3=6	τ_4(L)	2*2=4	10
Normalize	τ_4(L)					1	3 (2 divs)
Total							54

Markov network example

A	B	ϕ(A,B)
T	T	5
T	F	1
F	T	1
F	F	3

B	C	ϕ(B,C)
T	T	1
T	F	2
F	T	6
F	F	1

A	B	C	ϕ(A,B)*ϕ(B,C)	P(A,B,C)
T	T	T	5	0.11
T	T	F	10	0.22
T	F	T	6	0.13
T	F	F	1	0.02
F	T	T	1	0.02
F	T	F	2	0.04
F	F	T	18	0.39
F	F	F	3	0.07
		Z	46	1.00

A	B	P(A,B)
T	T	0.33
T	F	0.15
F	T	0.07
F	F	0.46

B	C	P(B,C)
T	T	0.13
T	F	0.26
F	T	0.52
F	F	0.09

A	P(A)
T	0.48
F	0.52

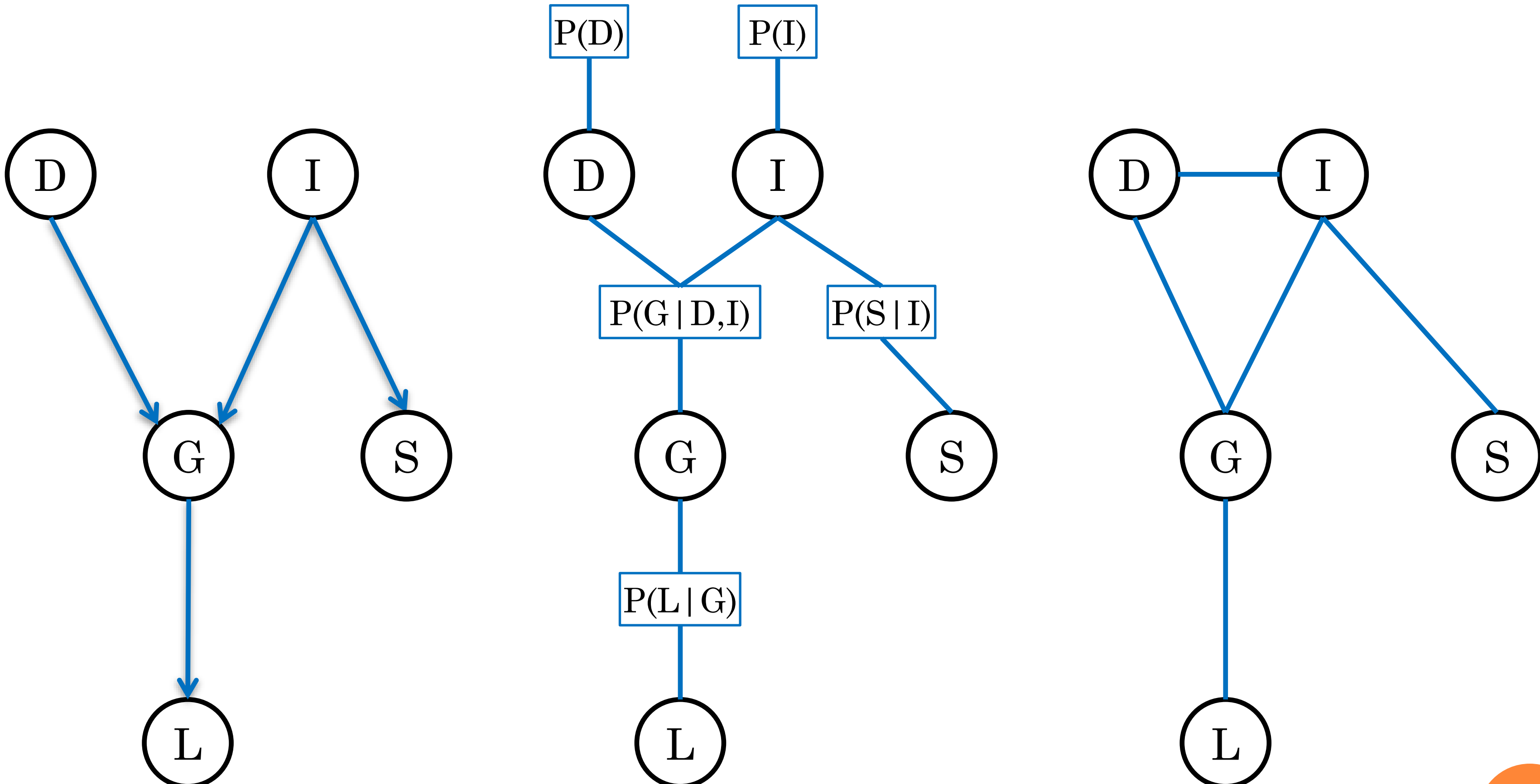
B	P(B)
T	0.39
F	0.61

C	P(C)
T	0.65
F	0.35

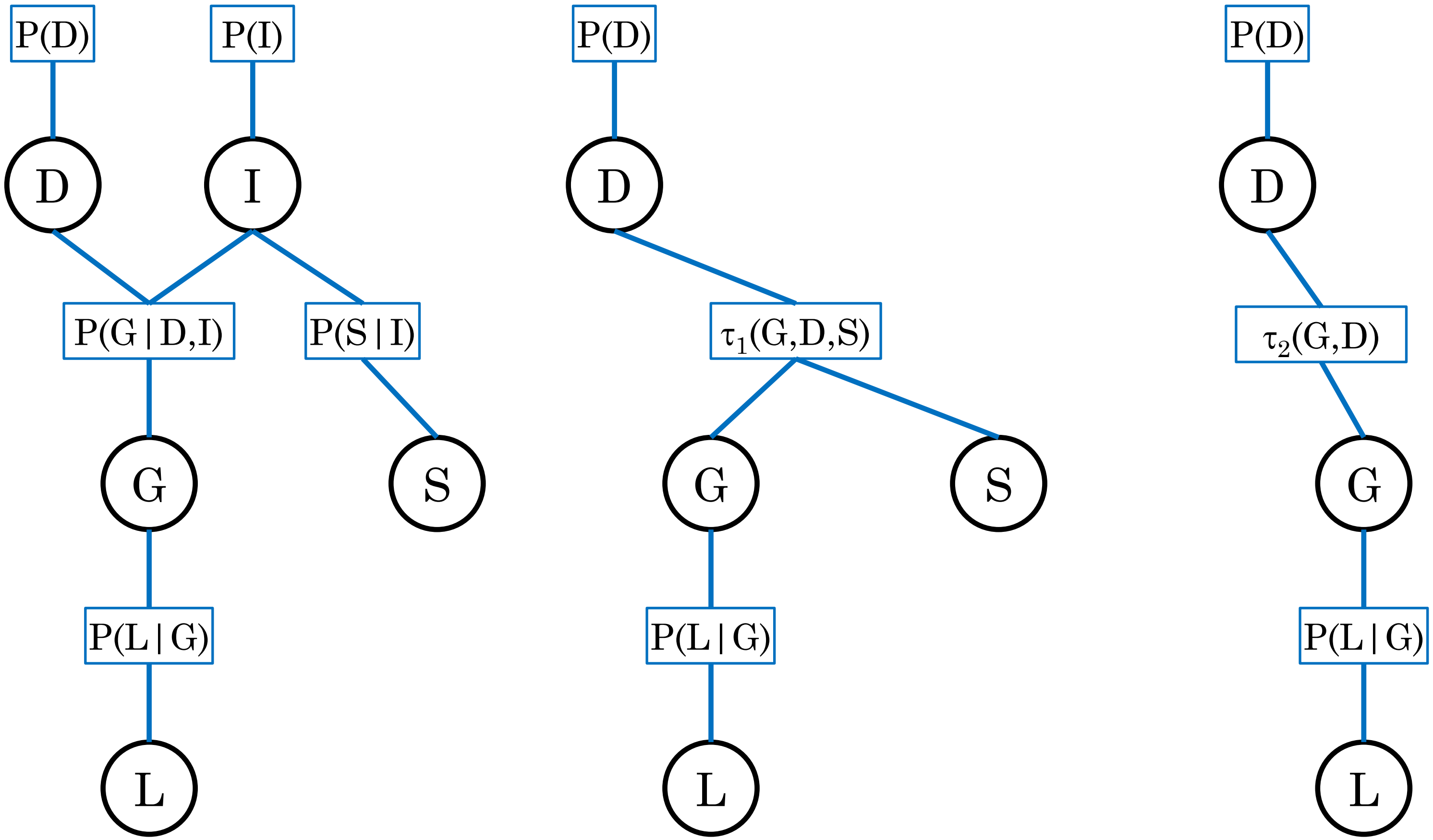
Elimination as graph transformation

- Eliminating X
  - Multiply all the factors X participates in
  - Sum out X
- Graph transformation (need to be moralized first)
  - Connect all of X's neighbors
  - Remove X

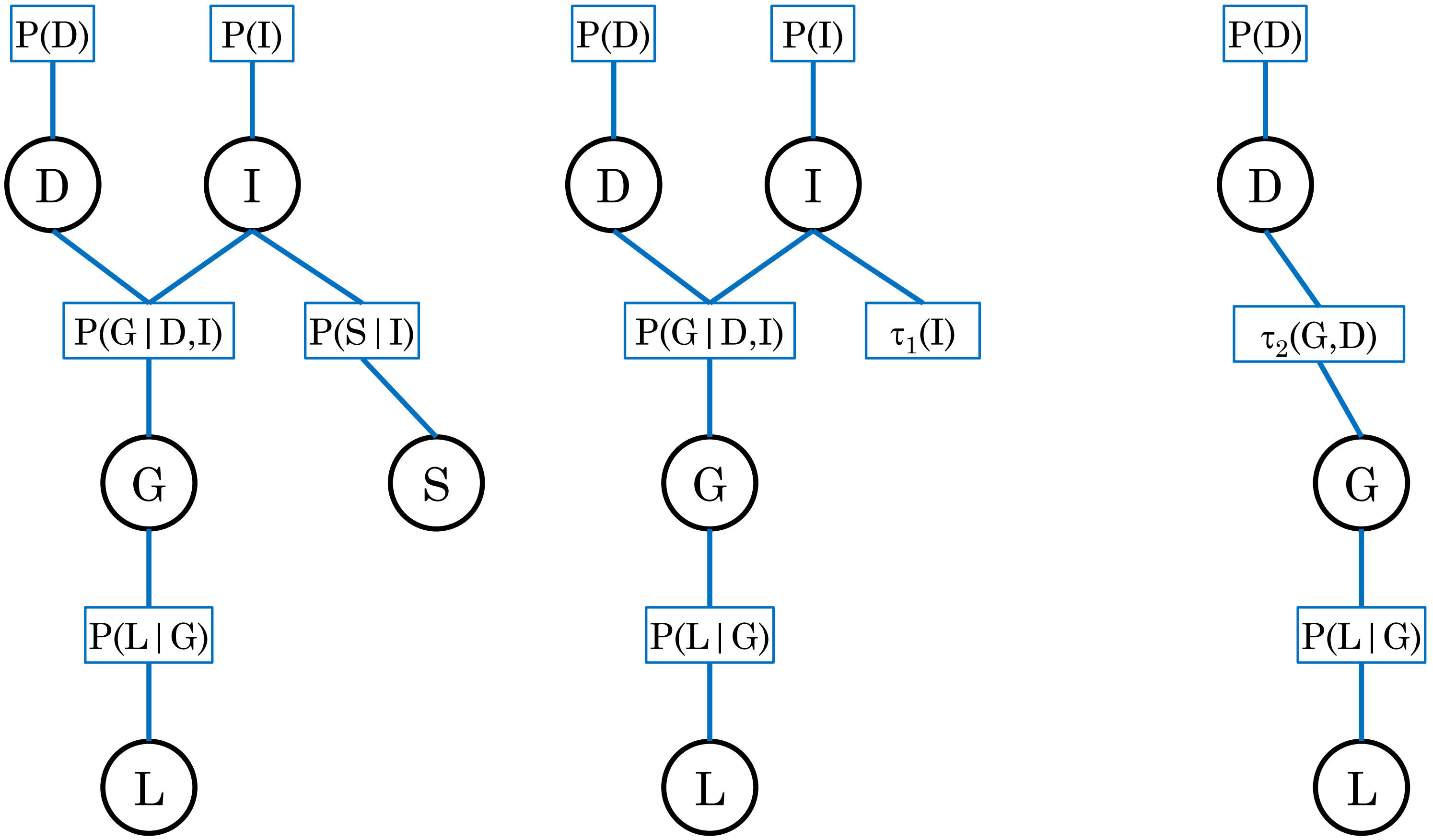
REPRESENTATION



IF WE FIRST ELIMINATE *I* THEN *S*



If we first eliminate *S* then *I*



Finding good elimination orderings

- Finding the best order is NP-hard
  - Best = optimal time and space complexity
- Heuristics
  - Min-neighbors
  - Min-fill
  - Weighted versions of min-neighbors and min-fill

Irrelevant nodes in BNs

- $\mathcal{X}$: all variables, $\boldsymbol{Y}$: query variables, $\boldsymbol{E}$: evidence variables, $\boldsymbol{W} = \mathcal{X} - \boldsymbol{Y} - \boldsymbol{E}$: remaining variables
- A node $X_i \in \boldsymbol{W}$ is irrelevant for the query $P(\boldsymbol{Y} \mid \boldsymbol{e})$ if it can be removed from the network without effecting the value of $P(\boldsymbol{Y} \mid \boldsymbol{e})$
- Obvious:
  - If $\boldsymbol{Z} \subseteq \boldsymbol{W}$ is d-separated from $\boldsymbol{Y}$ given $\boldsymbol{E}$, then $\boldsymbol{Z}$ is irrelevant
- Perhaps less obvious:
  - Let $\boldsymbol{Z}$ be ancestors of $\boldsymbol{Y} \cup \boldsymbol{E}$. Then $\boldsymbol{W} \setminus \boldsymbol{Z}$ is irrelevant