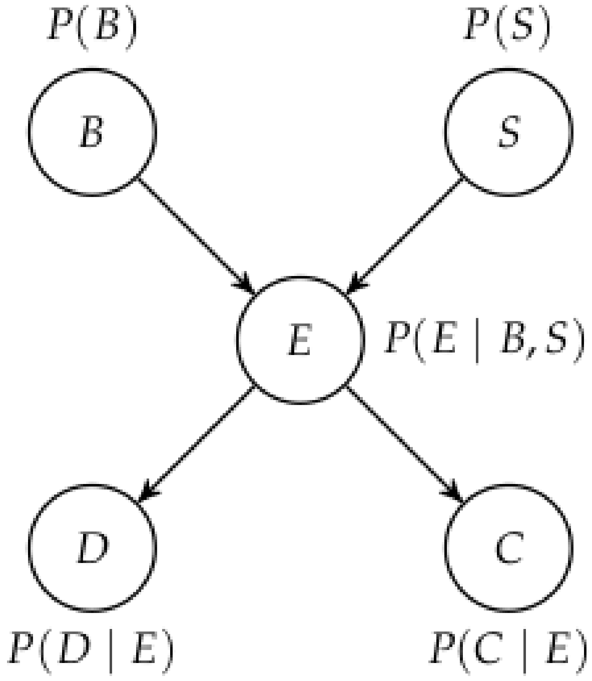
Bayesian Network Learning

- **Last time:**
  - Conditional independence in Bayesian Networks
  - Sampling from Bayesian Networks
- **Today:**
  - Given a **Bayesian Network** and some **values**, how do we calculate the probability of **other values**?
  - Given **data**, how do we **fit** a Bayesian network?

Bayesian Network

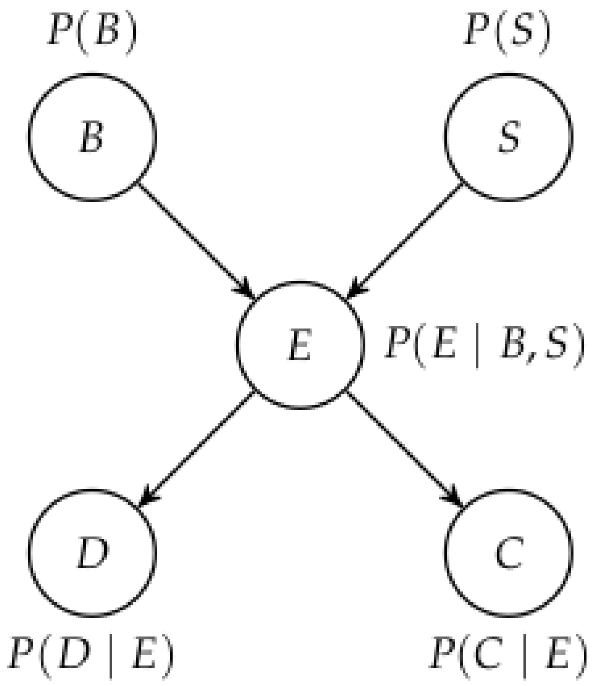


B battery failure
S solar panel failure
E electrical system failure
D trajectory deviation
C communication loss

Structure

Parameters

Inference



B battery failure
S solar panel failure
E electrical system failure
D trajectory deviation
C communication loss

Inputs

- Bayesian network structure
- Bayesian network parameters
- Values of *evidence variables*

Outputs

- Posterior distribution of *query variables*

Given that you have detected a trajectory deviation, and the battery has not failed what is the probability of a solar panel failure?

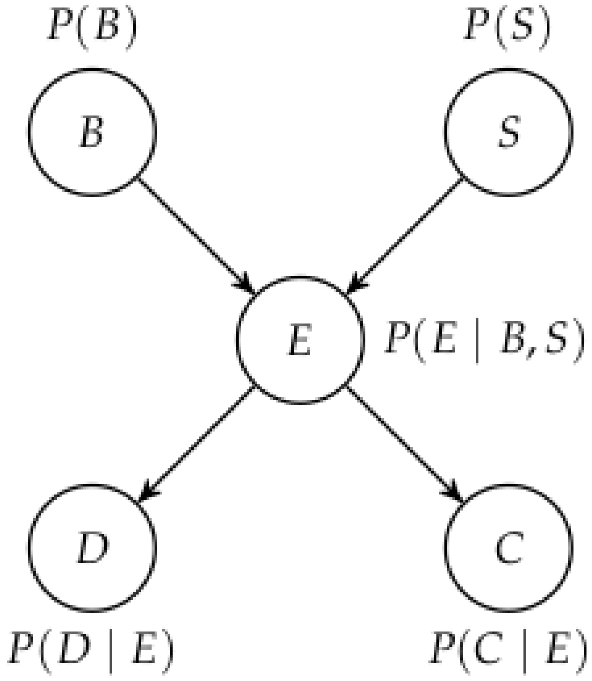
$$P(S = 1 \mid D = 1, B = 0)$$

Exact

Approximate

Exact Inference

Exact Inference



B battery failure
S solar panel failure
E electrical system failure
D trajectory deviation
C communication loss

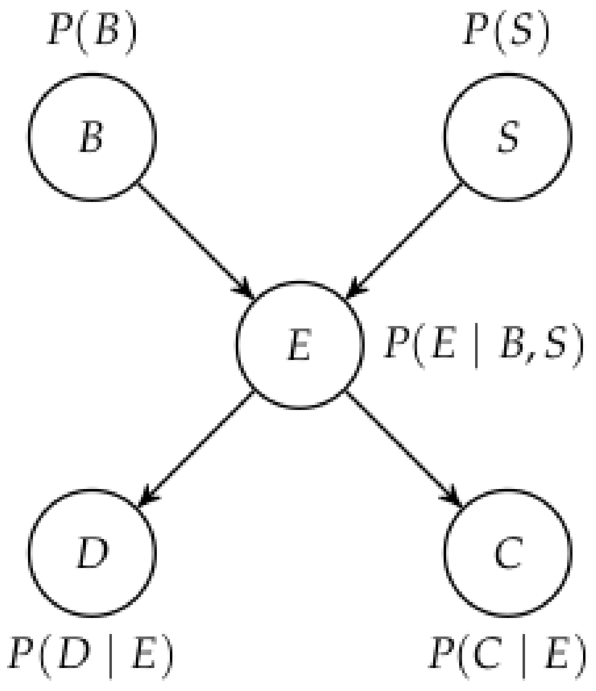
$$P(S{=}1 \mid D{=}1, B{=}0) = \frac{P(S{=}1, D{=}1, B{=}0)}{P(D{=}1, B{=}0)}$$

$$P(S{=}1, D{=}1, B{=}0) = \sum_{e,c} P(B{=}0, S{=}1, E{=}e, D{=}1, C{=}c)$$

$$P(B{=}0, S{=}1, E, D{=}1, C)$$
$$= P(B{=}0)\, P(S{=}1)\, P(E \mid B{=}0, S{=}1)\, P(D{=}1 \mid E)\, P(C{=}1 \mid E)$$

$2^5 = 32$ possible assignments,
but quickly gets too large

Exact Inference



B battery failure
S solar panel failure
E electrical system failure
D trajectory deviation
C communication loss

Product

X	Y	$\phi_1(X,Y)$
0	0	0.3
0	1	0.4
1	0	0.2
1	1	0.1

Y	Z	$\phi_2(Y,Z)$
0	0	0.2
0	1	0.0
1	0	0.3
1	1	0.5

X	Y	Z	$\phi_3(X,Y,Z)$
0	0	0	0.06
0	0	1	0.00
0	1	0	0.12
0	1	1	0.20
1	0	0	0.04
1	0	1	0.00
1	1	0	0.03
1	1	1	0.05

```
struct ExactInference end

function infer(M::ExactInference, bn, query, evidence)
    ϕ = prod(bn.factors)
    ϕ = condition(ϕ, evidence)
    for name in setdiff(variablenames(ϕ), query)
        ϕ = marginalize(ϕ, name)
    end
    return normalize!(ϕ)
end
```

Condition

X	Y	Z	$\phi(X,Y,Z)$
0	0	0	0.08
0	0	1	0.31
0	1	0	0.09
0	1	1	0.37
1	0	0	0.01
1	0	1	0.05
1	1	0	0.02
1	1	1	0.07

$Y = 1$

X	Z	$\phi(X,Z)$
0	0	0.09
0	1	0.37
1	0	0.02
1	1	0.07

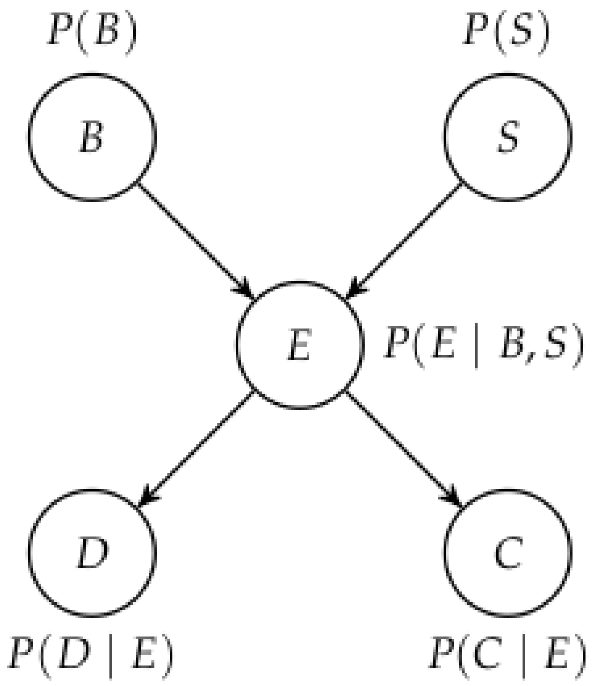
Marginalize

X	Y	Z	$\phi(X,Y,Z)$
0	0	0	0.08
0	0	1	0.31
0	1	0	0.09
0	1	1	0.37
1	0	0	0.01
1	0	1	0.05
1	1	0	0.02
1	1	1	0.07

X	Z	$\phi(X,Z)$
0	0	0.17
0	1	0.68
1	0	0.03
1	1	0.12

$2^5 = 32$ possible assignments,
but quickly gets too large

Exact Inference: Variable Elimination



B battery failure
S solar panel failure
E electrical system failure
D trajectory deviation
C communication loss

$P(B \mid d^1, c^1)$

Start with

$\phi_1(B), \phi_2(S), \phi_3(E, B, S), \phi_4(D, E), \phi_5(C, E)$

Eliminate D and C (evidence) to get $\phi_6(E)$ and $\phi_7(E)$

Eliminate E

$$\phi_8(B, S) = \sum_e \phi_3(e, B, S)\phi_6(e)\phi_7(e)$$

Eliminate S

$$\phi_9(B) = \sum_s \phi_2(s)\phi_8(B, s)$$

$$P(B \mid d^1, c^1) \propto \phi_1(B) \sum_s \left(\phi_2(s) \sum_e \left(\phi_3(e \mid B, s)\phi_4(d^1 \mid e)\phi_5(c^1 \mid e) \right) \right)$$

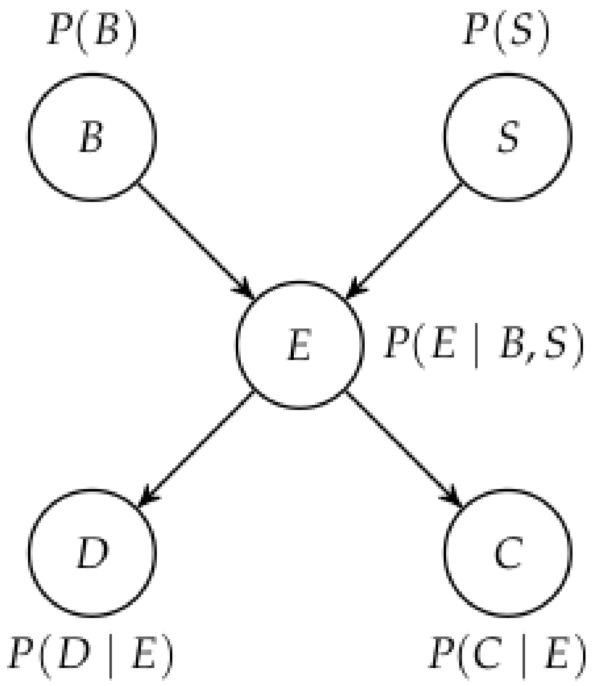
vs

$$P(B \mid d^1, c^1) \propto \sum_s \sum_e \phi_1(B)\phi_2(s)\phi_3(e \mid B, s)\phi_4(d^1 \mid e)\phi_5(c^1 \mid e)$$

Choosing optimal order is NP-hard

Approximate Inference

Approximate Inference: Direct Sampling



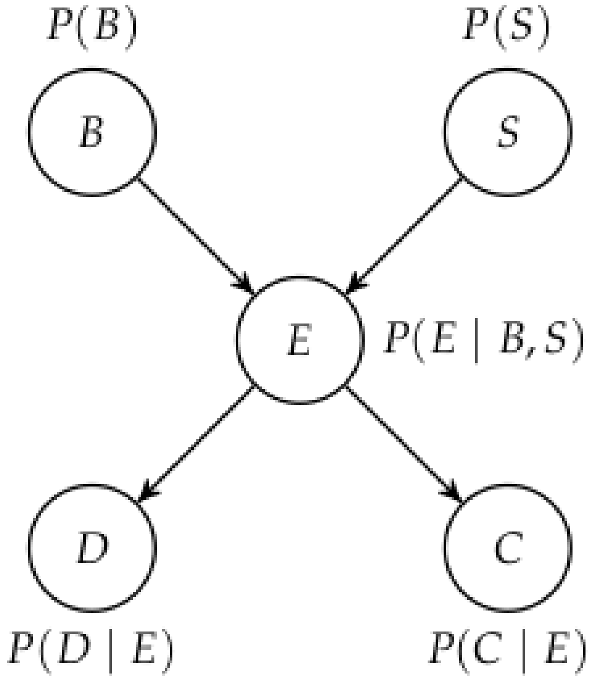
B battery failure
S solar panel failure
E electrical system failure
D trajectory deviation
C communication loss

$$P(b^1 \mid d^1, c^1) \approx \frac{\sum_i (b^{(i)} = 1 \land d^{(i)} = 1 \land c^{(i)} = 1)}{\sum_i (d^{(i)} = 1 \land c^{(i)} = 1)}$$

B	S	E	D	C
0	0	1	1	0
0	0	0	0	0
1	0	1	0	0
1	0	1	1	1
0	0	0	0	0
0	0	0	1	0
0	0	0	0	1
0	1	1	1	1
0	0	0	0	0
0	0	0	1	0

Analogous to **unweighted particle filtering**

Approximate Inference: Weighted Sampling



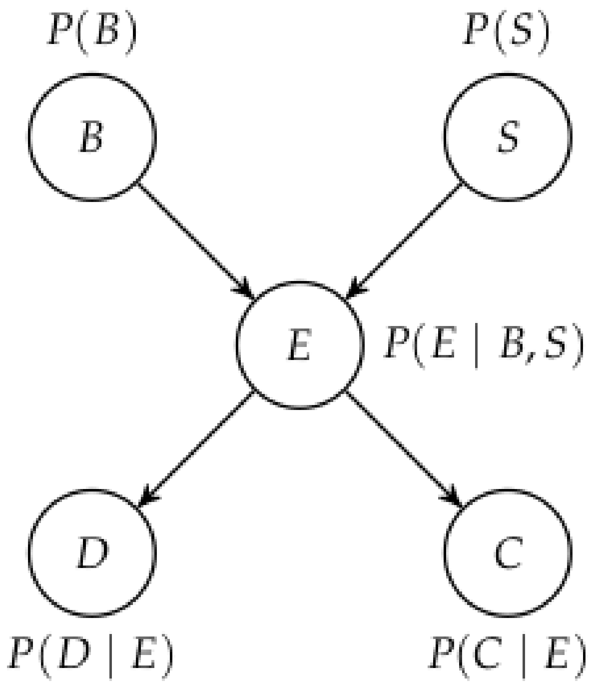
B battery failure
S solar panel failure
E electrical system failure
D trajectory deviation
C communication loss

$$P(b^1 \mid d^1, c^1) \approx \frac{\sum_i w_i(b^{(i)} = 1 \land d^{(i)} = 1 \land c^{(i)} = 1)}{\sum_i w_i(d^{(i)} = 1 \land c^{(i)} = 1)}$$
$$= \frac{\sum_i w_i(b^{(i)} = 1)}{\sum_i w_i}$$

B	S	E	D	C	Weight
1	0	1	1	1	$P(d^1 \mid e^1)P(c^1 \mid e^1)$
0	1	1	1	1	$P(d^1 \mid e^1)P(c^1 \mid e^1)$
0	0	0	1	1	$P(d^1 \mid e^0)P(c^1 \mid e^0)$
0	0	0	1	1	$P(d^1 \mid e^0)P(c^1 \mid e^0)$
0	0	1	1	1	$P(d^1 \mid e^1)P(c^1 \mid e^1)$

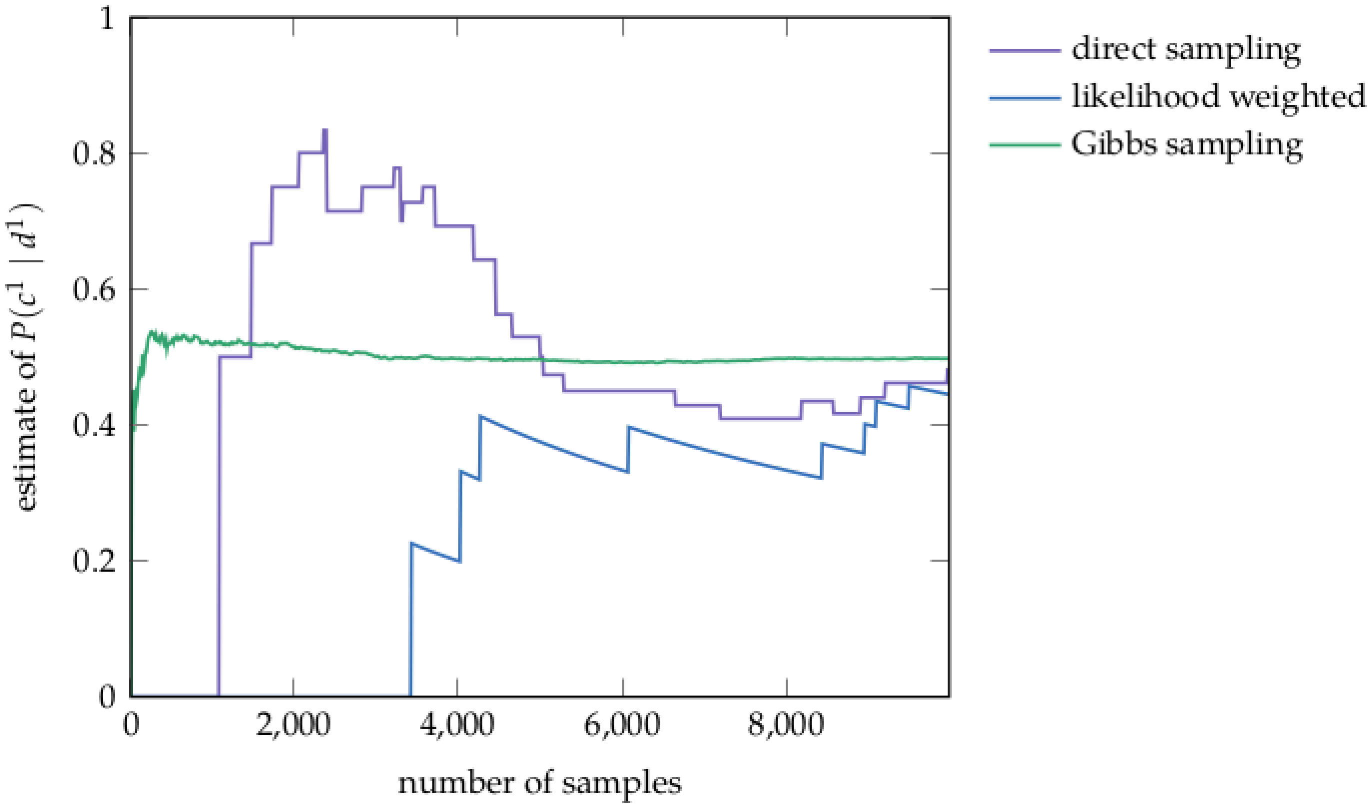
Analogous to **weighted particle filtering**

Approximate Inference: Gibbs Sampling



B battery failure
S solar panel failure
E electrical system failure
D trajectory deviation
C communication loss

Markov Chain Monte Carlo (MCMC)



Learning

Bayesian Network Learning

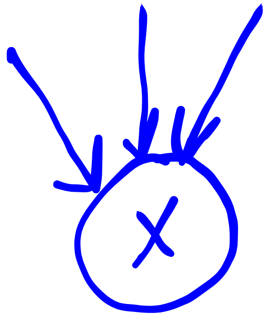
Inputs

- Data, D
- Priors (?)

Outputs

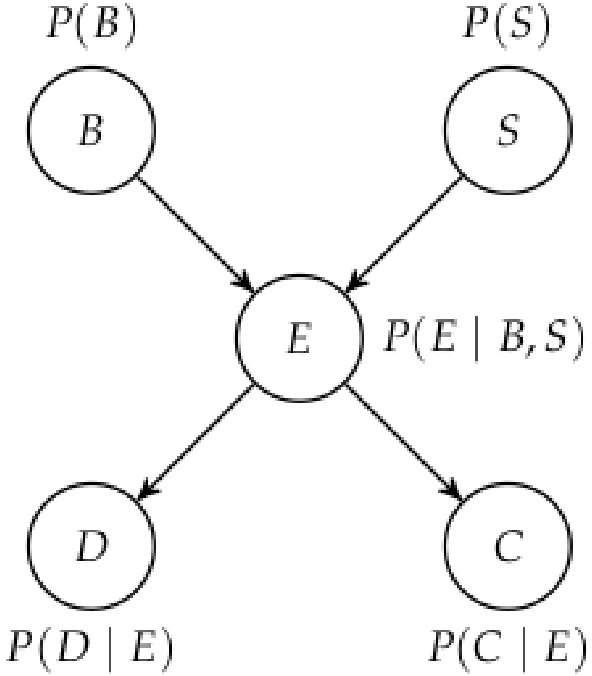
- Bayesian network structure, G
- Bayesian network parameters, θ

Counting Parameters



For discrete R.V.s:

$$\dim(\theta_X) = (|\text{support}(X)| - 1) \prod_{Y \in Pa(X)} |\text{support}(Y)|$$



Structure Learning Example

Parameter Learning

Maximum Likelihood

$$\hat{\theta} = \arg\max_{\theta} P(D \mid \theta)$$

$$P(D \mid \theta) = \prod_i P(o_i \mid \theta)$$

$$\hat{\theta} = \arg\max_{\theta} \sum_i \log P(o_i \mid \theta)$$

Multinomial:

$$\hat{\theta}_i = \frac{n_i}{\sum_{j=1}^{k} n_j}$$

Bayesian

$$\hat{\theta} = \mathbb{E}_{\theta \sim p(\cdot \mid D)}[\theta] = \int \theta p(\theta \mid D)\, \mathrm{d}\theta$$

Multinomial:

$$p(\theta_{1:n} \mid \alpha_{1:n}, m_{1:n}) = \mathrm{Dir}(\theta_{1:n} \mid \alpha_1 + m_1, \ldots, \alpha_n + m_n)$$

$$\frac{\alpha_i}{\sum_{j=1}^{n} \alpha_j}$$

Structure Learning

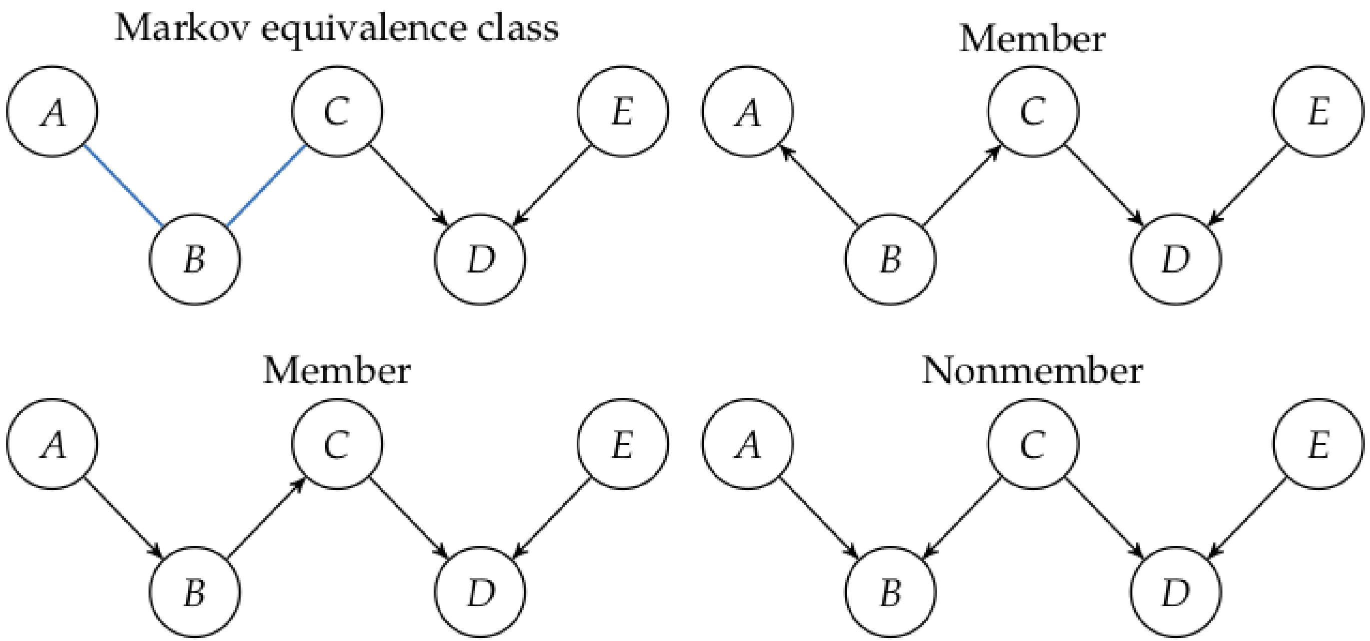
$$P(G \mid D) \propto P(G)P(D \mid G)$$
$$= P(G) \int P(D \mid \theta, G) p(\theta \mid G)\, \mathrm{d}\theta$$

$$P(G \mid D) = P(G) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij0})}{\Gamma(\alpha_{ij0} + m_{ij0})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + m_{ijk})}{\Gamma(\alpha_{ijk})}$$

$$\log P(G \mid D)$$
$$= \log P(G) + \sum_{i=1}^{n} \sum_{j=1}^{q_i} \left(\log\left(\frac{\Gamma(\alpha_{ij0})}{\Gamma(\alpha_{ij0} + m_{ij0})} \right) + \sum_{k=1}^{r_i} \log\left(\frac{\Gamma(\alpha_{ijk} + m_{ijk})}{\Gamma(\alpha_{ijk})} \right) \right)$$

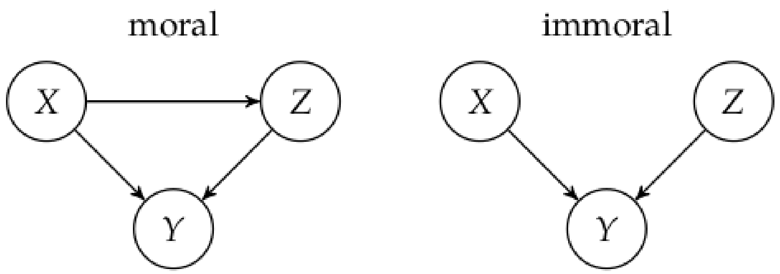
NP-Hard

Markov Equivalence Class



Markov Equivalent iff

1. Same undirected edges
2. Same set of immoral v-structures



Recap

Inference

Learning