10-301/601: Introduction to Machine Learning Lecture 21 – Bayesian Networks

Henry Chai

7/17/23

Front Matter

- Announcements

  - PA5 released 7/13, due 7/20 at 11:59 PM

  - Quiz 7: Unsupervised Learning & Naïve Bayes on 7/18 (tomorrow!)

- Recommended Readings

  - Murphy, Chapters 10.1 - 10.5

Recall:
How hard is modelling $P(X|Y)$?

X_1 ("hat")	X_2 ("cat")	X_3 ("dog")	X_4 ("fish")	X_5 ("mom")	X_6 ("dad")	$P(X|Y=1)$	$P(X|Y=0)$
0	0	0	0	0	0	θ_1	θ_{64}
1	0	0	0	0	0	θ_2	θ_{65}
1	1	0	0	0	0	θ_3	θ_{66}
1	0	1	0	0	0	θ_4	θ_{67}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
1	1	1	1	1	1	$1 - \sum_{i=1}^{63} \theta_i$	$1 - \sum_{i=64}^{126} \theta_i$

Recall: Naïve Bayes Assumption

- *Assume* features are conditionally independent given the label:

$$P(X|Y) = \prod_{d=1}^{D} P(X_d|Y)$$

- Pros:

  - <u>Significantly</u> reduces computational complexity

  - Also reduces model complexity, combats overfitting

- Cons:

  - Is a strong, often illogical assumption

    - We'll see a relaxed version of this ~~later in the semester~~ today when we discuss Bayesian networks

Motivating Example

## Hacking Attack Woke Up Dallas With Emergency Sirens, Officials Say

Warning sirens in Dallas, meant to alert the public to emergencies like severe weather, started sounding around 11:40 p.m. Friday, and were not shut off until 1:20 a.m. Rex C. Curry for The New York Times

- "the city's warning system was hacked late on Friday [4/7/2017]"

- "The alarms, which started going off around 11:40 p.m. Friday and lasted until 1:20 a.m. Saturday, … jarring residents awake and flooding 911 with thousands of calls…"

- "…the sirens, which are meant to alert the public to severe weather or other emergencies, …"

- "Social media was flooded with complaints."

Constructing a Bayesian Network
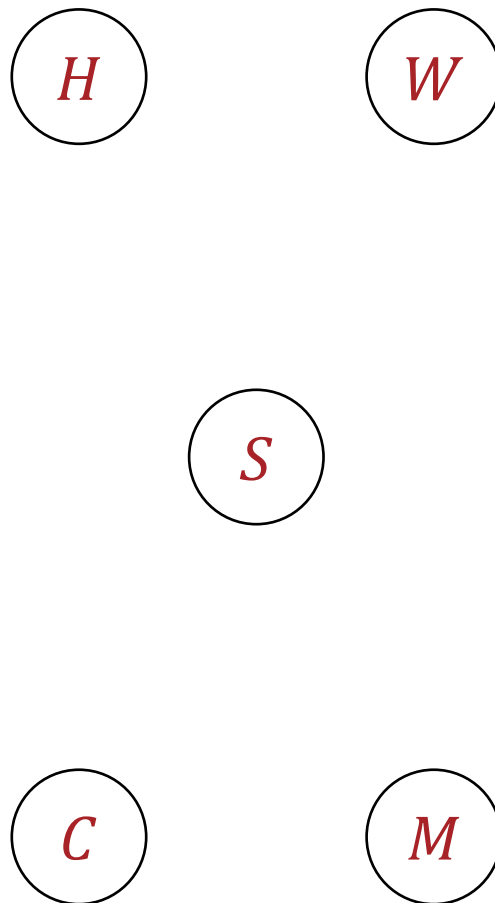
- H = sirens are **h**acked

- W = extreme **w**eather event occurred

- S = **s**irens go off overnight

- C = 911 flooded with phone **c**alls

- M = social **m**edia flooded with posts

- All variables are binary

Constructing a Bayesian Network

H

W

S

C

M
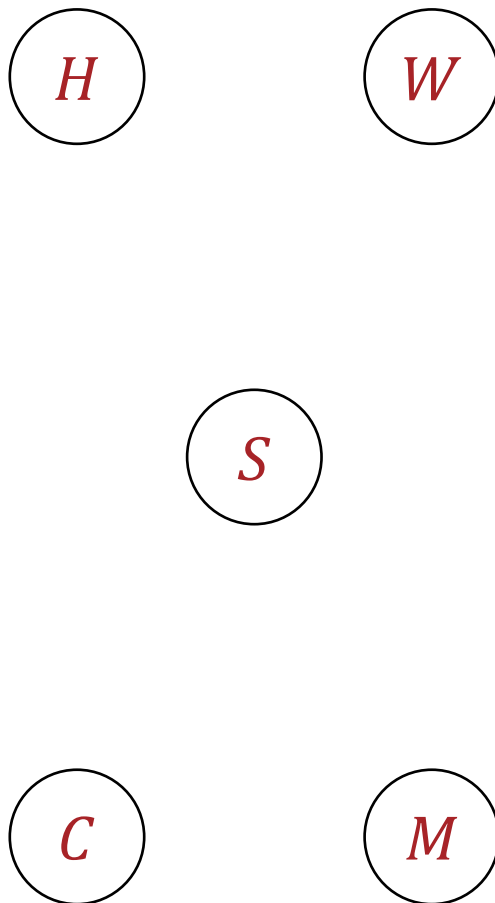
- By the chain rule of probability, the full joint distribution is

- $P(H, W, S, C, M) =$
$P(M|C, S, H, W)$
$P(C|S, H, W)$
$P(S|H, W)$
$P(H|W)$
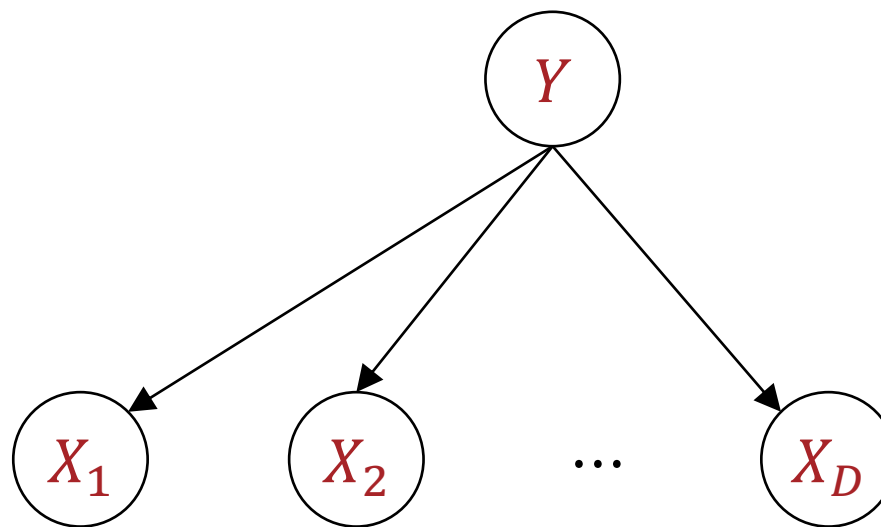$P(W)$

Constructing a Bayesian Network
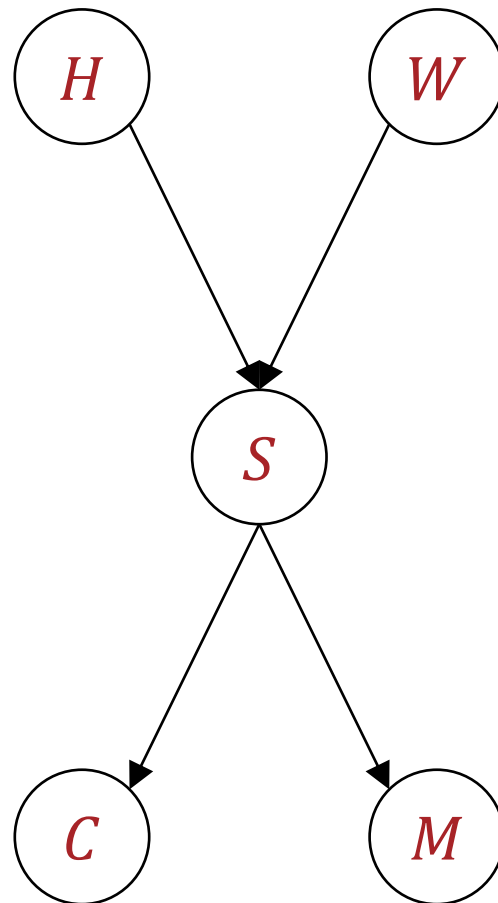
- Directed acyclic graph where edges indicate conditional dependency

- A variable is conditionally independent of all its non-descendants (i.e., upstream variables) given its parents

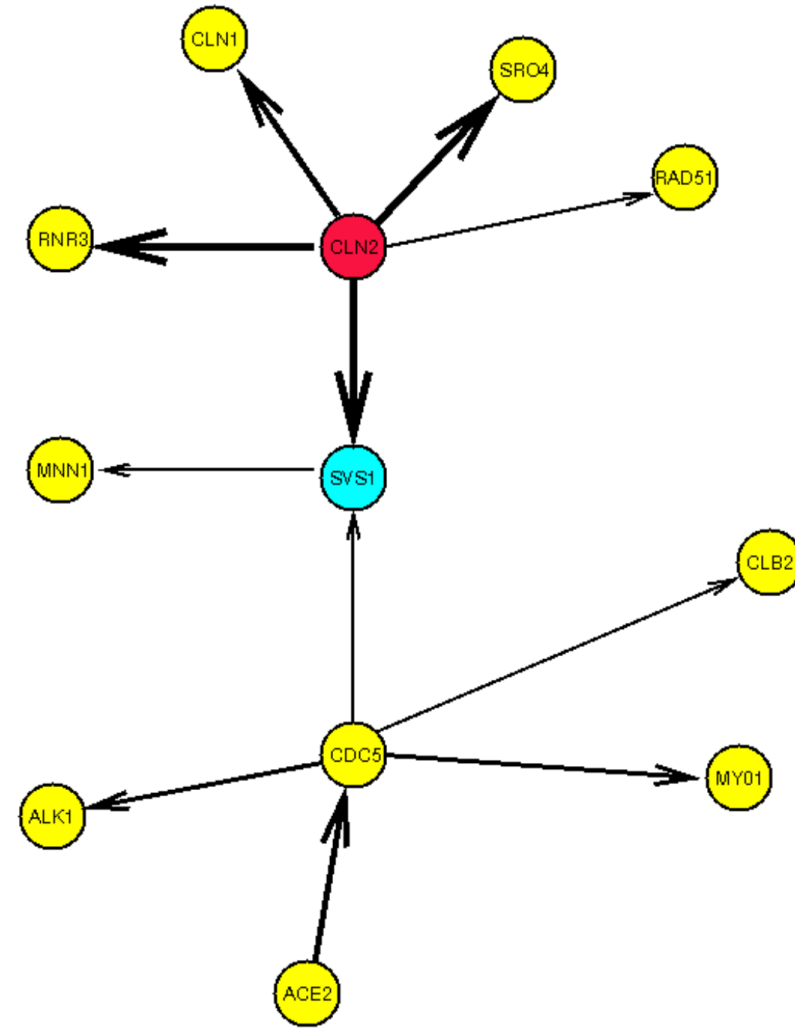- $P(H, W, S, C, M) = P(H)P(W)P(S|H, W) P(C|S)P(M|S)$

Naïve Bayes as a Bayesian Network

- *Assume* features are conditionally independent given the label:

$$P(X, Y) = P(Y)P(X|Y) = P(Y) \prod_{d=1}^{D} P(X_d|Y)$$

Bayesian Network Example: Gene Expression

Figure courtesy of Ziv Bar-Joseph

Bayesian Networks: Outline

- How can we learn a Bayesian network?
    - Learning the graph structure
    - Learning the conditional probabilities

- What inference questions can we answer with a Bayesian network?
    - Computing (or estimating) marginal (conditional) probabilities
    - Implied (conditional) independencies

Learning a Network

1. Specify the random variables

2. Determine the conditional dependencies
   - Prior knowledge
   - Domain expertise
   - Learned from data (model selection)

Learning the Parameters
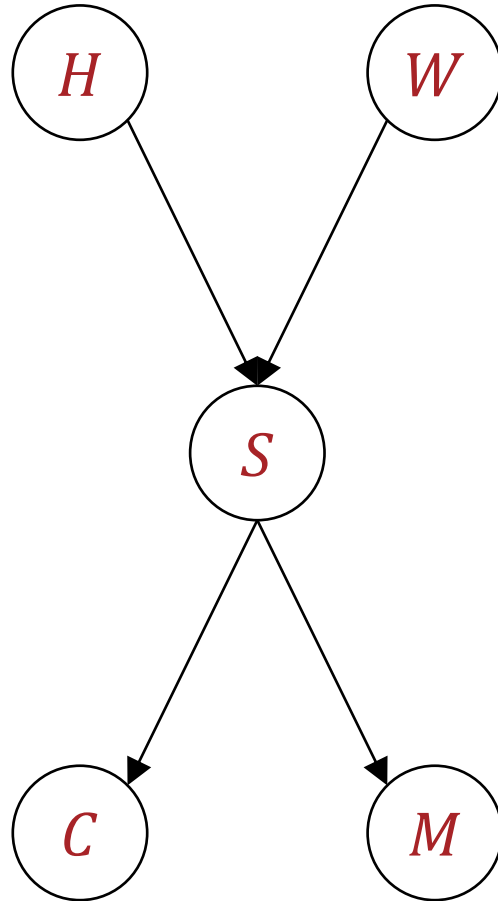
- $P(H, W, S, C, M) = P(H)P(W)P(S|H,W)P(C|S)P(M|S)$

- How many parameters do we need to learn?

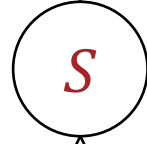# How many parameters do you need to learn in order to fully specify this Bayesian network?

5

9

10

20

31

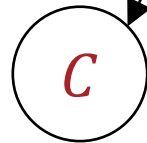# Learning the Parameters

$P(H = 1)$

H

W $P(W = 1)$

S

$P(S = 1 | H = 1, W = 1)$
$P(S = 1 | H = 1, W = 0)$
$P(S = 1 | H = 0, W = 1)$
$P(S = 1 | H = 0, W = 0)$

C

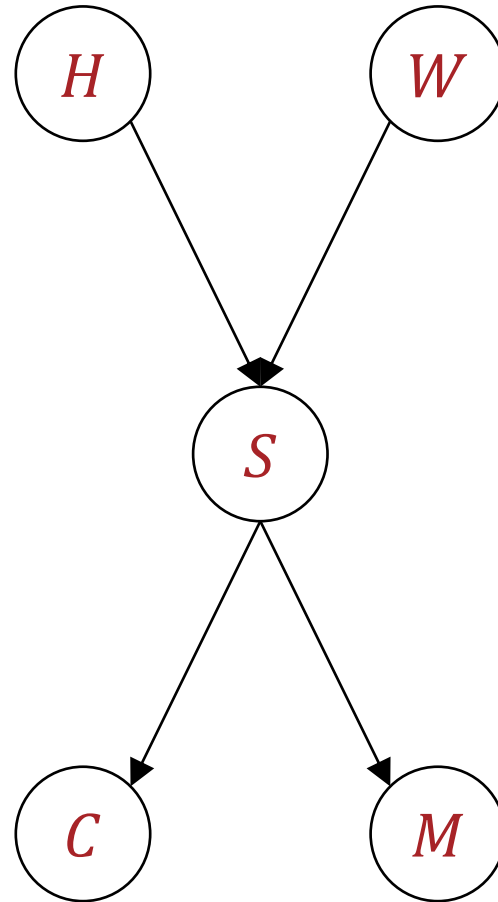M $P(M = 1 | S = 1)$
$P(M = 1 | S = 0)$

$P(C = 1 | S = 1)$
$P(C = 1 | S = 0)$

Learning the Parameters (Fully-observed)

- $\mathcal{D} = \left\{ \left(H^{(n)}, W^{(n)}, S^{(n)}, C^{(n)}, M^{(n)} \right) \right\}_{n=1}^{N}$

- Set parameters via MLE

$$P(H = 1) = \frac{N_{H=1}}{N}$$

$$\vdots$$

$$P(S = 1 | H = 0, W = 1) = \frac{N_{S=1,H=0,W=1}}{N_{H=0,W=1}}$$

$$\vdots$$

Bayesian Networks: Outline

- How can we learn a Bayesian network?

    - Learning the graph structure

    - Learning the conditional probabilities

- What inference questions can we answer with a Bayesian network?

    - Computing (or estimating) marginal (conditional) probabilities

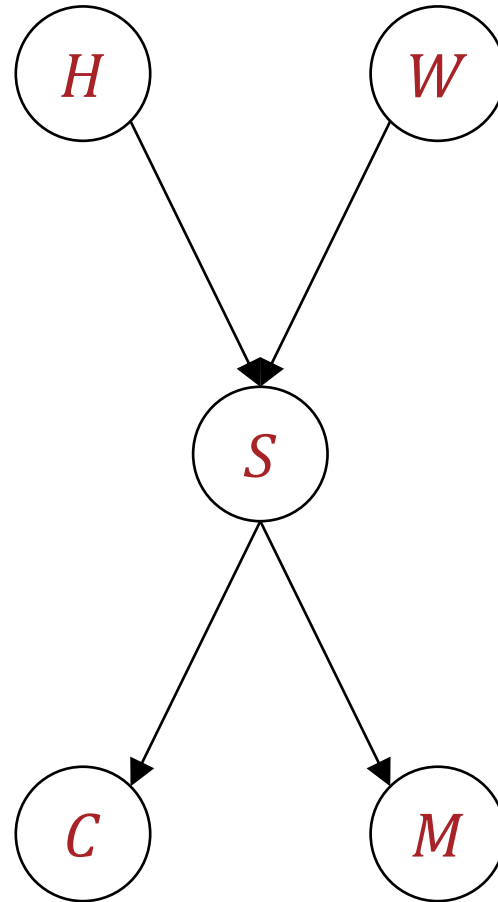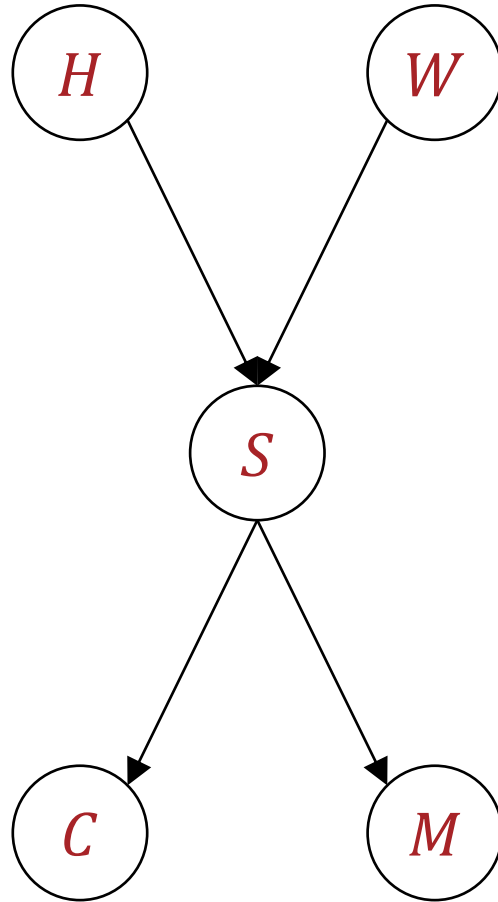   - Implied (conditional) independencies

Computing Joint Probabilities…

H W

S

C M
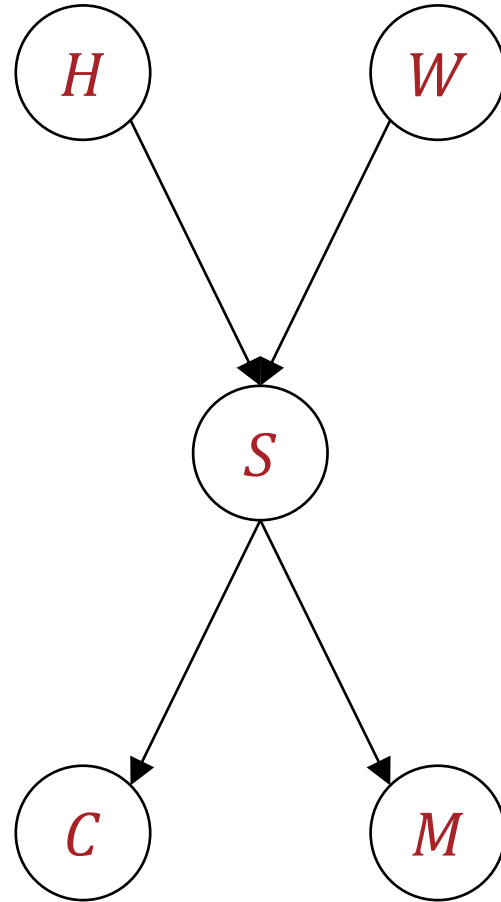
- What is

$P(H = 1, W = 0, S = 1, C = 1, M = 0)$?

Computing Joint Probabilities is easy

$P(H = 1, W = 0, S = 1, C = 1, M = 0)$

$=$

$P(H = 1) *$

$\left(1 - P(W = 1)\right) *$

$P(S = 1 | H = 1, W = 0) *$

$P(C = 1 | S = 1) *$

$\left(1 - P(M = 1 | S = 1)\right)$

Computing Marginal Probabilities...

- What is $P(S = 1)$?
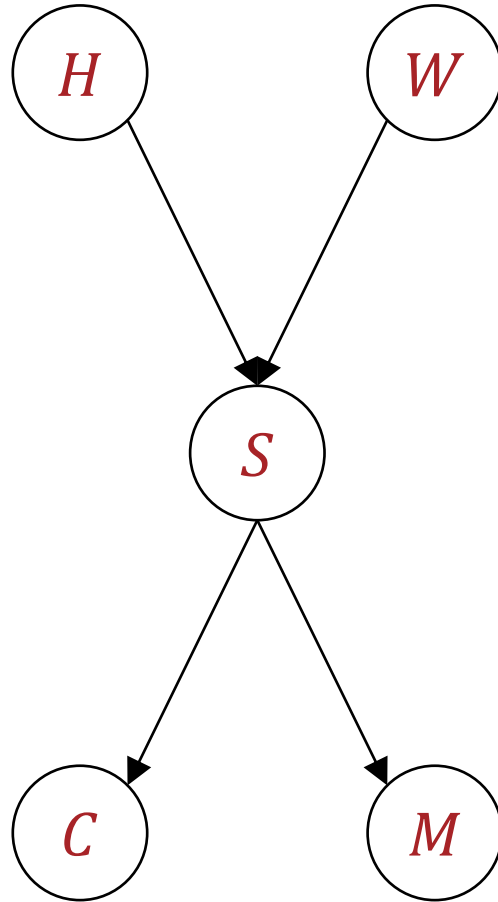
$$P(S = 1)$$

$$= \sum_{h,w,c,m} P(H = h, W = w, S = 1, C = c, M = m)$$

- What is $P(H = 1 | M = 1)$?

$$P(H = 1 | M = 1) = \frac{P(H = 1, M = 1)}{P(M = 1)}$$

$$= \frac{\sum_{w,s,c} P(H = 1, W = w, S = s, C = c, M = 1)}{\sum_{h,w,s,c} P(H = h, W = w, S = s, C = c, M = 1)}$$
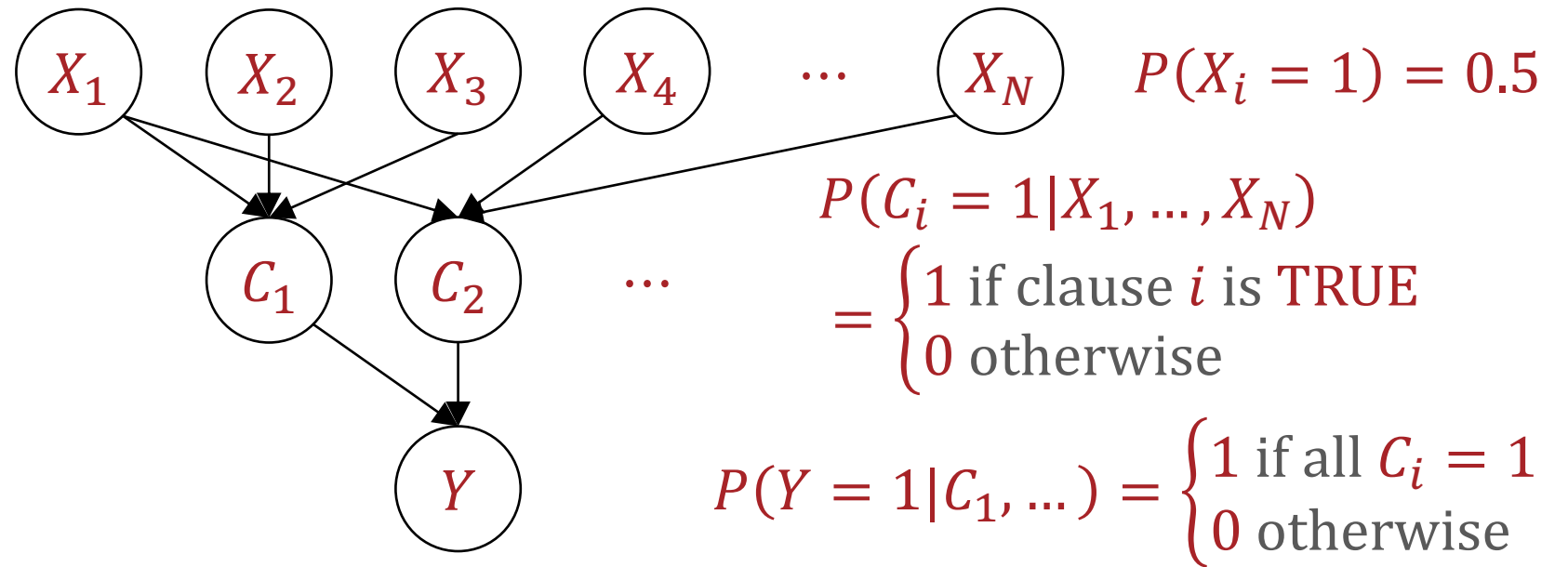
Computing Marginal Probabilities…

- Computing arbitrary marginal (conditional) distributions requires summing over exponentially many possible combinations of the unobserved variables
- Computation can be improved by storing and reusing calculated values (dynamic programming)
 - Still exponential in the worst case

Computing Marginal Probabilities is (NP-)hard!

- Claim: 3-SAT reduces to computing marginal probabilities in a Bayesian network

- Proof (sketch): Given a Boolean equation in 3-CNF, e.g., $(X_1 \vee X_2 \vee X_3) \wedge (\neg X_1 \vee X_4 \vee \neg X_N) \wedge \cdots$, construct the corresponding Bayesian network

$$P(X_i = 1) = 0.5$$

$$P(C_i = 1 | X_1, \ldots, X_N) = \begin{cases} 1 \text{ if clause } i \text{ is TRUE} \\ 0 \text{ otherwise} \end{cases}$$

$$P(Y = 1 | C_1, \ldots) = \begin{cases} 1 \text{ if all } C_i = 1 \\ 0 \text{ otherwise} \end{cases}$$

- $P(Y = 1) > 0$ means the 3-CNF is satisfiable!

Sampling for Bayesian Networks
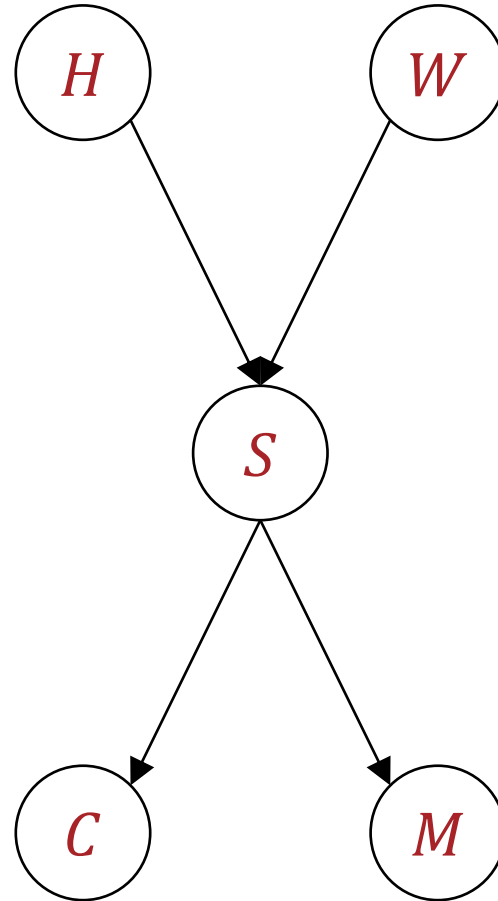
- Sampling from a Bayesian network is easy!

  1. Sample all free variables (H and W)

  2. Sample any variable whose parents have already been sampled

  3. Stop once all variables have been sampled

$$P(S = 1) \approx \frac{\text{\# of samples w/ } S = 1}{\text{\# of samples}}$$
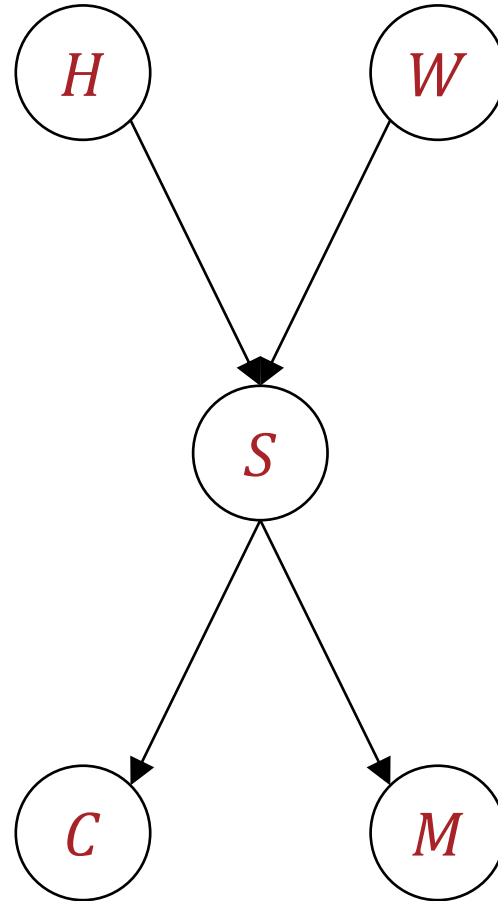
Sampling for Bayesian Networks
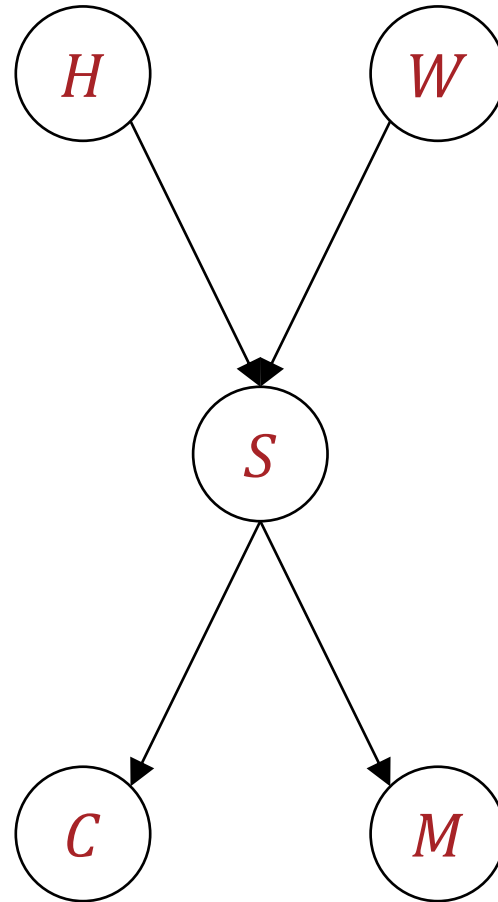

- Sampling from a Bayesian network is easy!

  1. Sample all free variables (H and W)

  2. Sample any variable whose parents have already been sampled

  3. Stop once all variables have been sampled

$P(H = 1 | M = 1)$

$$\approx \frac{\text{\# of samples w/ } H = 1 \text{ and } M = 1}{\text{\# of samples w/ } M = 1}$$

- If the condition is rare, we need lots of samples to get a good estimate

Weighted Sampling for Bayesian Networks

- Initialize $N_{Condition} = N_{Event} = 0$
- Repeatedly
  - Draw a sample from the full joint distribution
  - Set the condition to be true (set $M = 1$)
  - Compute the joint probability of the adjusted sample, w (easy!)
    $$N_{Condition} = N_{Condition} + w$$
  - If the event occurs in the adjusted sample ($H = 1$?), update N_{Event}
    $$N_{Event} = N_{Event} + w$$
- Desired marginal conditional probability is $\approx \dfrac{N_{Event}}{N_{Condition}}$

Conditional Independence
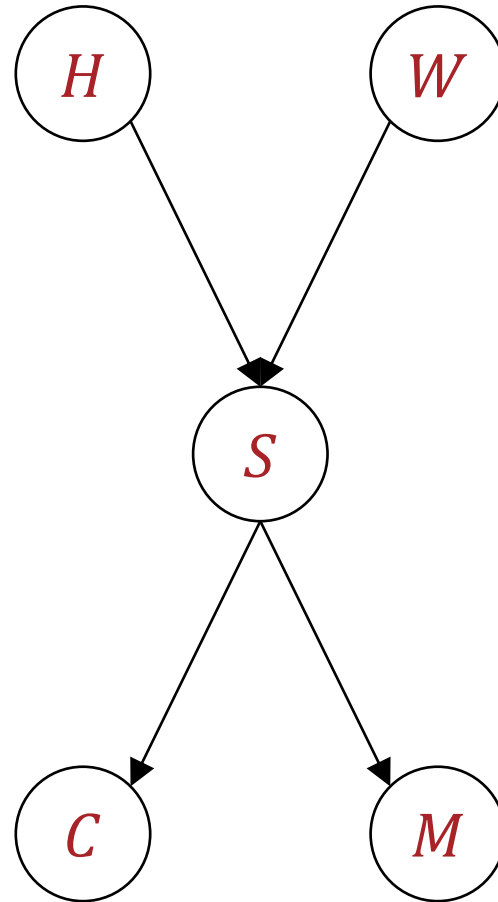
- X and Y are conditionally independent given Z ($X \perp Y \mid Z$) if
$$P(X, Y \mid Z) = P(X \mid Z)P(Y \mid Z)$$

- In a Bayesian network, each variable is conditionally independent of its ***non-descendants*** given its parents
  - H and M are not independent but they are conditionally independent given S

- What other conditional independencies does a Bayesian network imply?
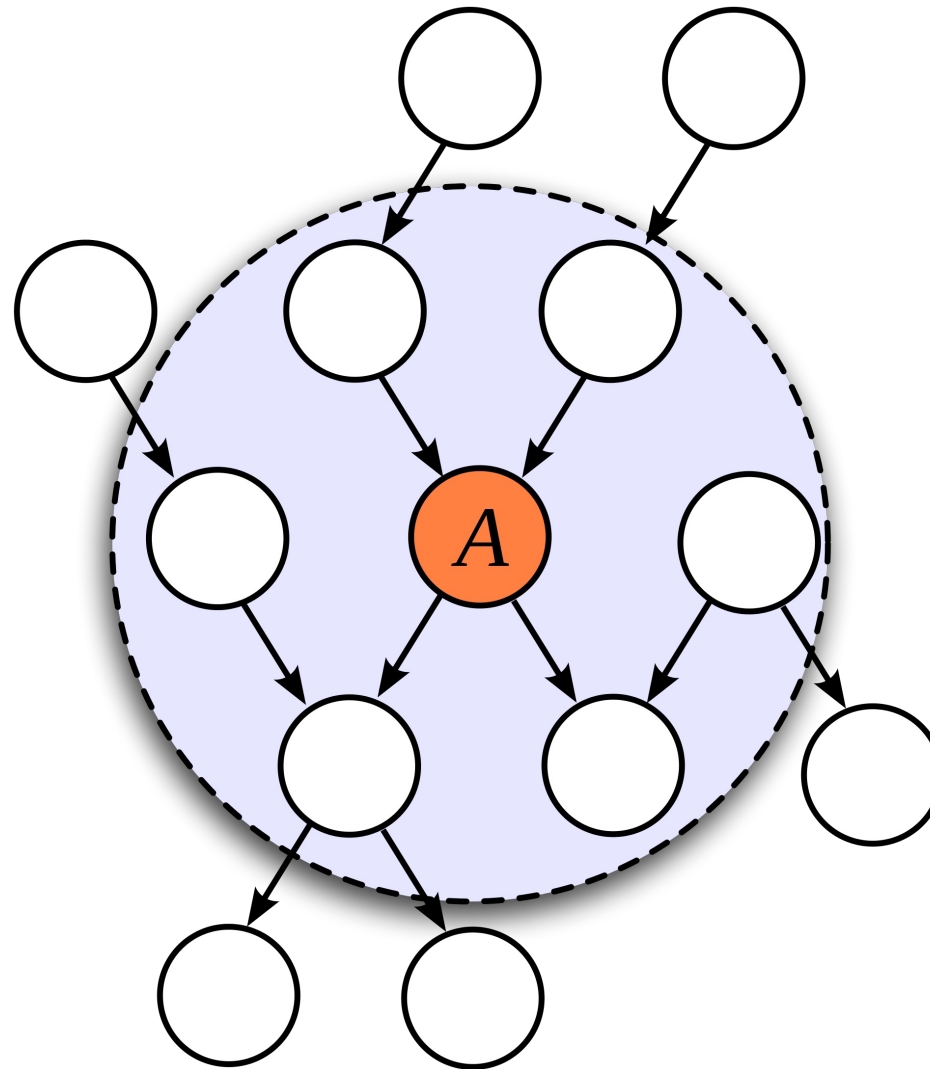
Markov Blanket

- Let $\mathcal{S}$ be the set of all random variables in a Bayesian network

- A *Markov blanket* of $A \in \mathcal{S}$ is any set $B \subseteq \mathcal{S}$ s.t.
$$A \perp \mathcal{S} \backslash B \mid B$$
  - Contains all the useful information about A

- Trivially, $\mathcal{S}$ is always a Markov blanket for any random variable in $\mathcal{S}$

Markov Boundary
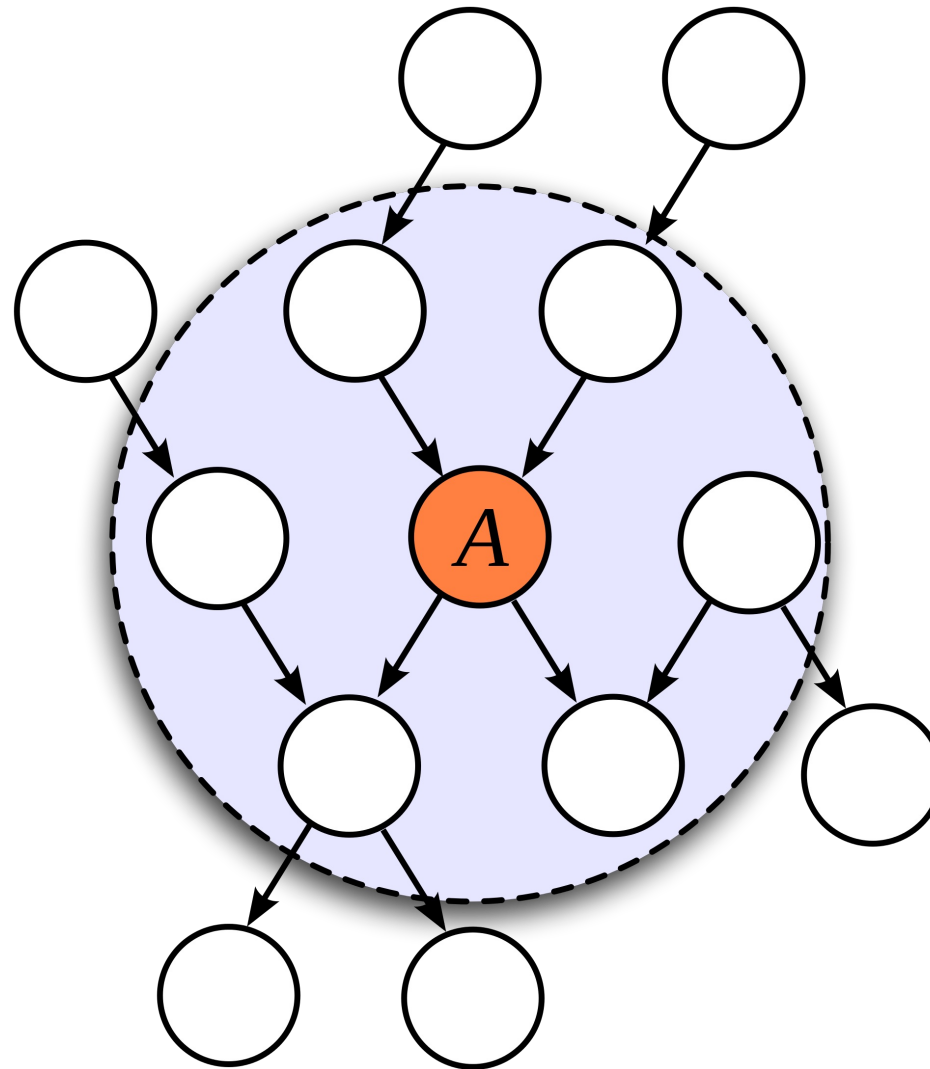
- Let S be the set of all random variables in a Bayesian network

- The *Markov boundary* of A is the smallest possible Markov blanket of A

- The Markov boundary consists of a variable's children, parents and co-parents (the other parents of its children)

But what if I care about the relationship between two variables?

- Let S be the set of all random variables in a Bayesian network

- The *Markov boundary* of A is the smallest possible Markov blanket of A

- The Markov boundary consists of a variable's children, parents and co-parents (the other parents of its children)
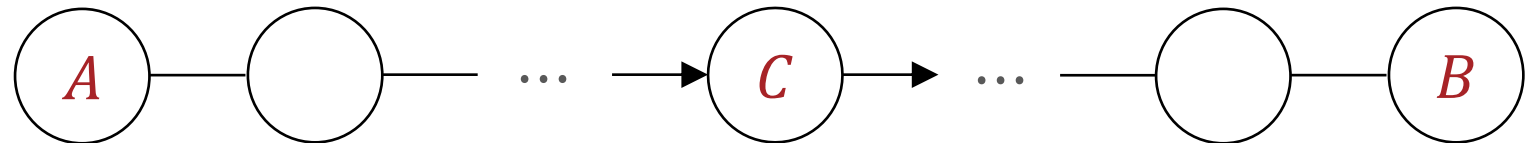
D-separation

- Random variables A and B are *d-separated* given evidence variables Z if $A \perp B \mid Z$

- Definition 1: A and B are d-separated given Z iff every *undirected* path between A and B is *blocked* by Z

- An undirected path between A and B is blocked by Z if

  1. $\exists$ a "common parent" variable C on the path and $C \in Z$

  $$A \quad \bigcirc \quad \cdots \longleftarrow C \longrightarrow \cdots \quad \bigcirc \quad B$$
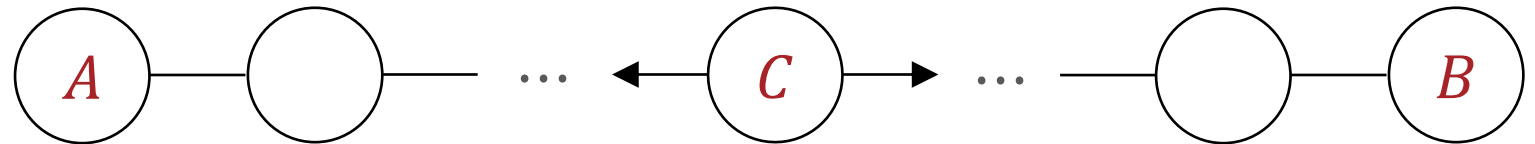
  2. $\exists$ a "cascade" variable C on the path and $C \in Z$

  $$A \quad \bigcirc \quad \cdots \longrightarrow C \longrightarrow \cdots \quad \bigcirc \quad B$$

  3. $\exists$ a "collider" variable C on the path and
  $\{C, \text{descendents}(C)\} \notin Z$
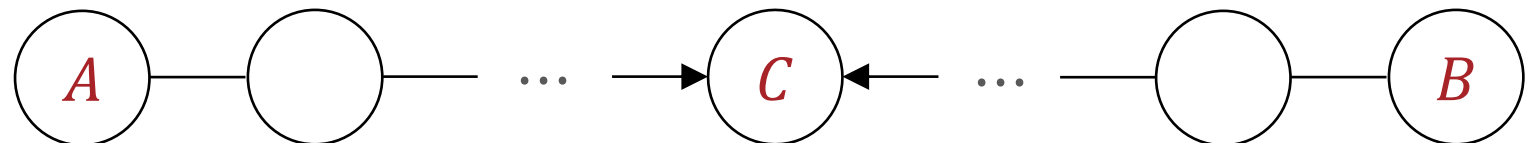
  $$A \quad \bigcirc \quad \cdots \longrightarrow C \longleftarrow \cdots \quad \bigcirc \quad B$$

D-separation

- Random variables A and B are *d-separated* given evidence variables Z if $A \perp B \mid Z$

- Definition 2: A and B are d-separated given Z iff $\nexists$ a path between A and B in the undirected ancestral moral graph with Z removed
  1. Keep only A, B, Z and their ancestors (ancestral graph)
  2. Add edges between all co-parents (moral graph)
  3. Undirected: replace directed edges with undirected ones
  4. Delete Z and check if A and B are connected

- Example: $A \perp B \mid \{D, E\}$?

Original: Ancestral: Moral: Undirected: Givens Removed:

⇒ A and B connected
⇒ not d-separated

Figure courtesy of Matt Gormley
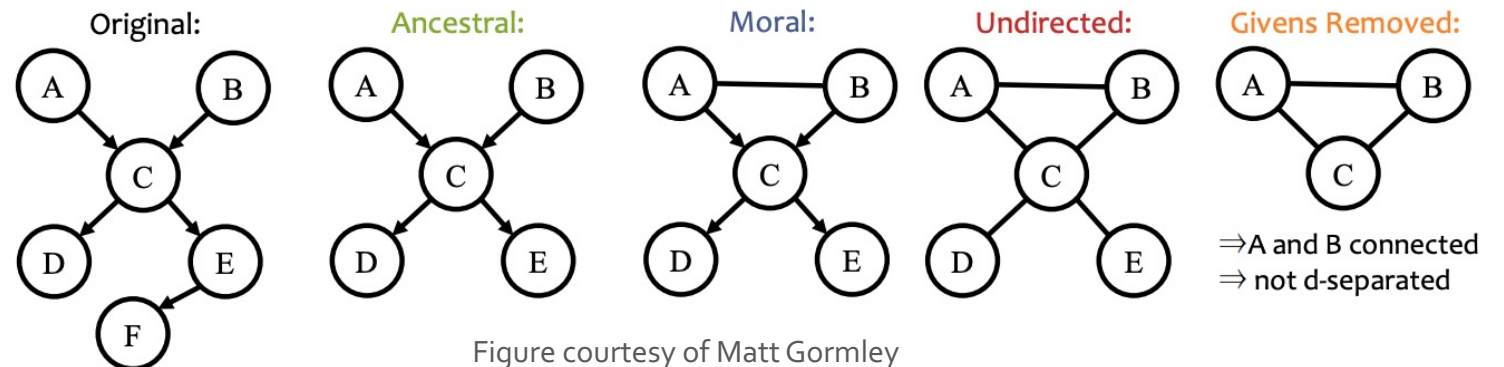
Shortcomings of Bayesian Networks

- Graph structure must be acyclic

- Cannot encode temporal/sequential relationships

- We'll address these (related) problems next with hidden Markov models

Key Takeaways

- Bayesian networks are flexible models for modelling joint probability distributions

  - Trade-off between expressiveness (full joint distributions) and computational tractability (Naïve Bayes)

- Bayesian networks represent conditional dependence though a directed acyclic graph

  - Graph structure usually specified, can be learned

  - Parameters in the fully-observed case learned via MLE

- Computing marginal & conditional distributions is NP-hard

  - Can use sampling for approximate inference

- Markov blanket and d-separation provide notions of conditional independence for single and pairs of variables respectively