CS 760: Machine Learning
Graphical Models - II

Josiah Hanna

University of Wisconsin-Madison

November 9, 2023

Announcements

- **Homework 5 due today; homework 6 due Nov 21.**
- **No class on Tuesday, Nov 21.**

Outline

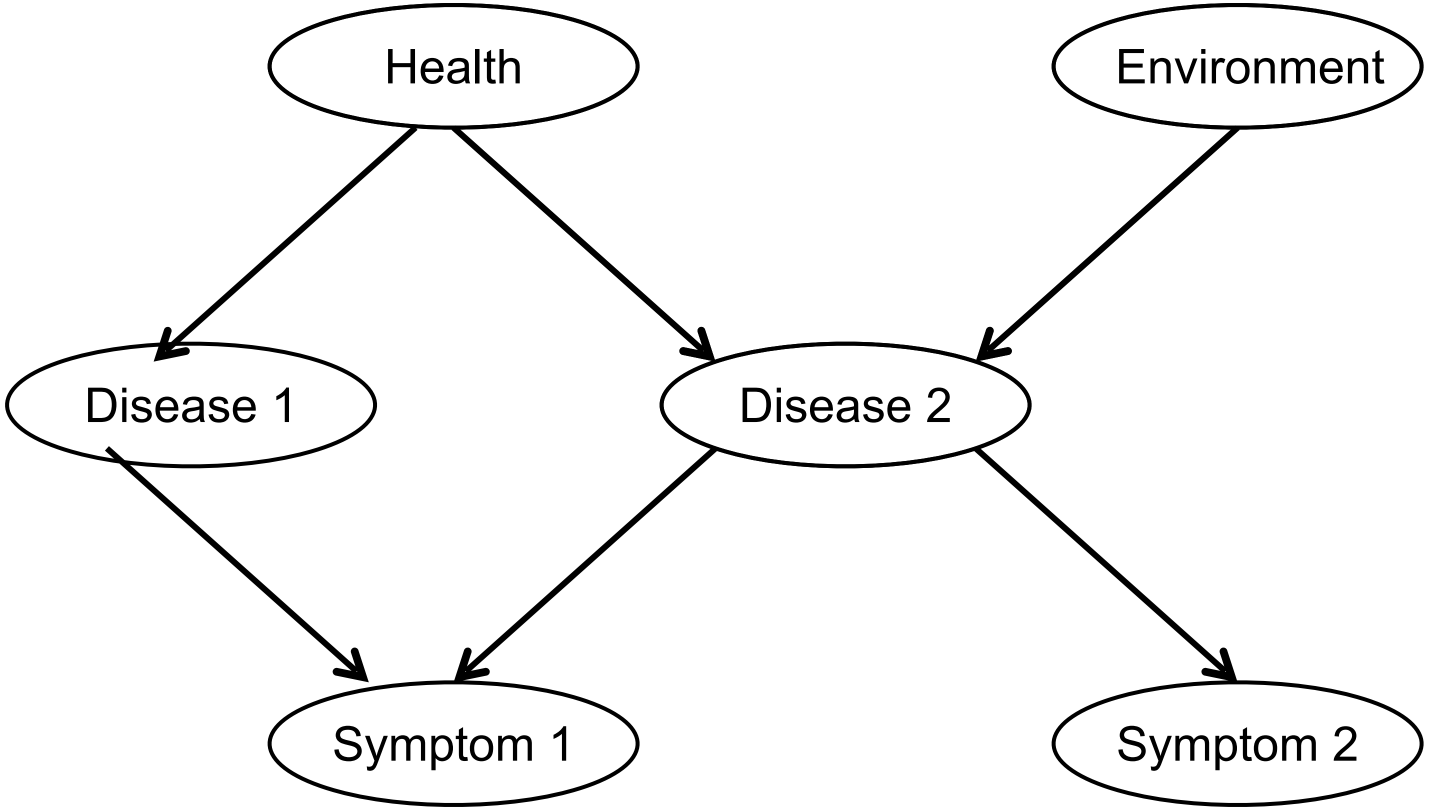
- **Bayesian Networks Review**
  - Definition, examples, inference, learning
- **Structure learning**
  - Chow-Liu Algorithm
- **D-separation**

Outline

- **Bayesian Networks Review**
  - Definition, examples, inference, learning
- **Structure learning**
  - Chow-Liu Algorithm
- **D-separation**

Bayesian Networks Example

- Set up a network that shows how random variables influence others:



Bayesian Networks Example

- Let’s construct a Bayes Network to help us understand a pandemic.

Bayesian Networks Example

- Consider the following 5 binary random variables:

  B = a burglary occurs at the house

  E = an earthquake occurs at the house

  A = the alarm goes off

  J = John calls to report the alarm

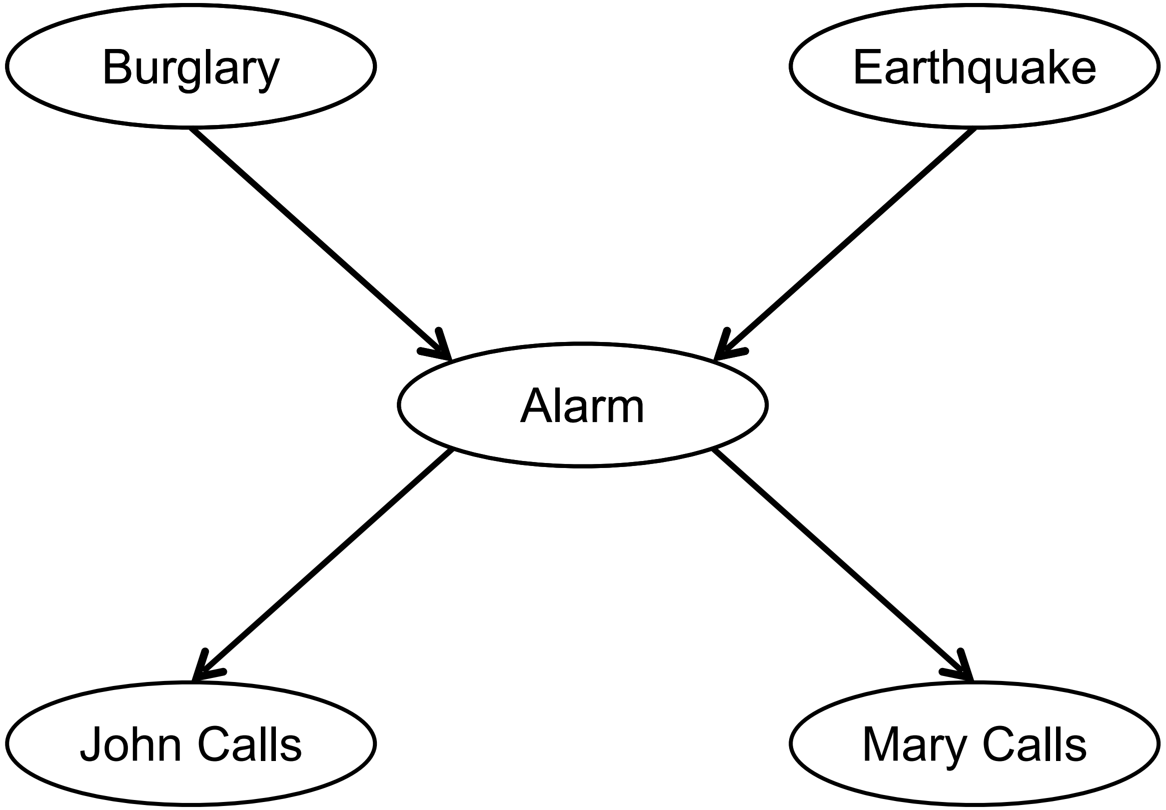
  M = Mary calls to report the alarm

- Suppose the Burglary or Earthquake can trigger Alarm, and Alarm can trigger John's call or Mary's call

- Now we want to answer queries like what is $P(B \mid M, J)$?

Bayesian Networks Example

- Set up a network that shows how random variables influence others:

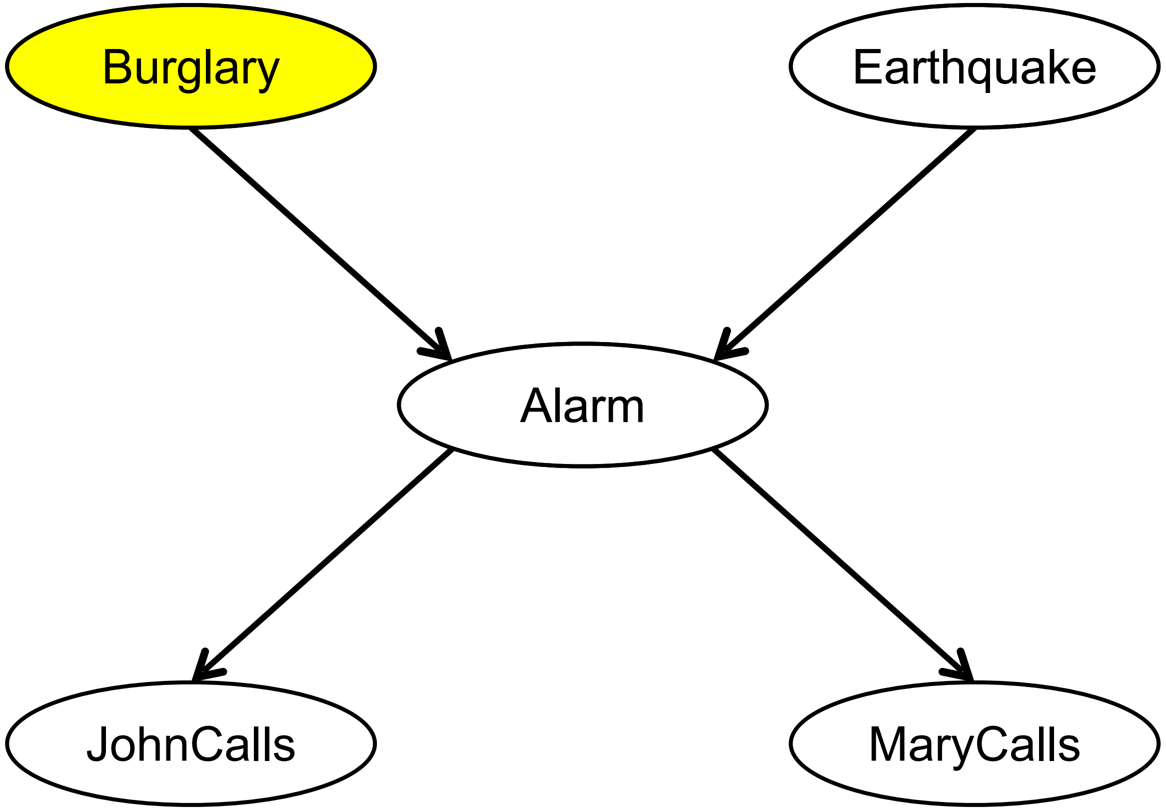


Bayesian Networks Example

- Set up a network that shows how random variables influence others:

P (*B*)

t	f
0.001	0.999

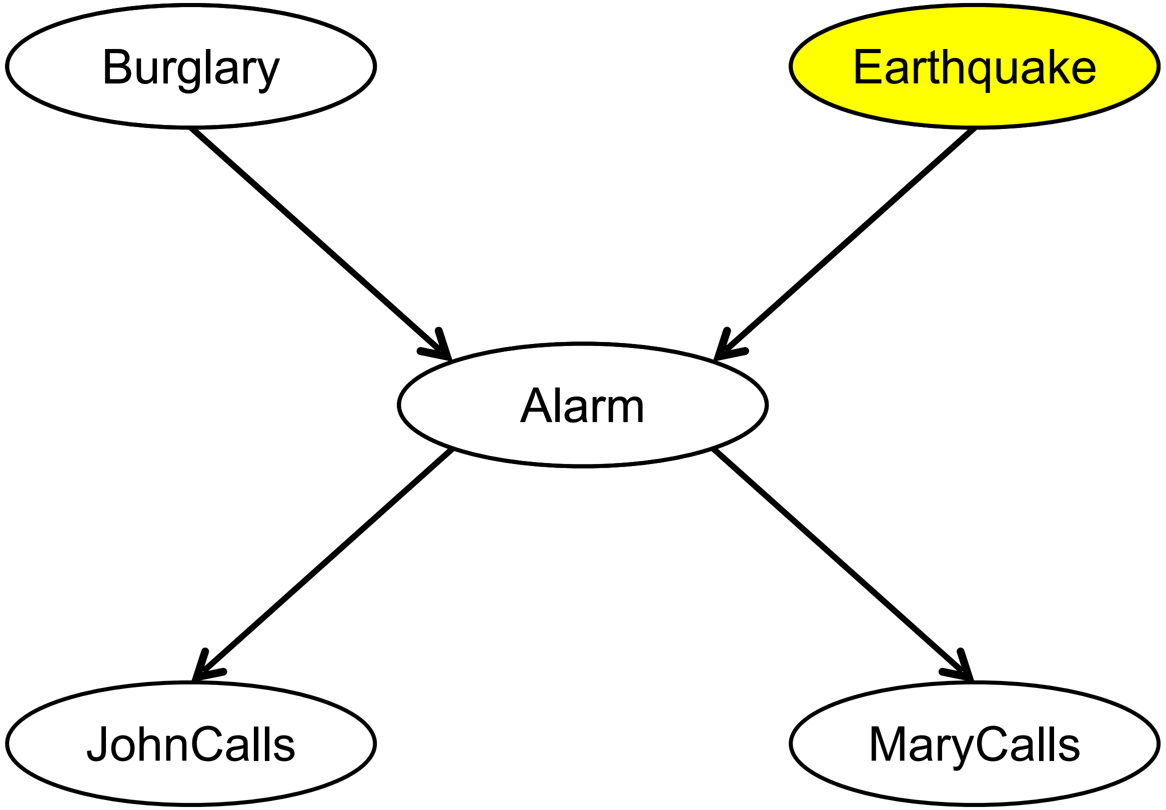


Bayesian Networks Example

- Set up a network that shows how random variables influence others:

P (*B*)

t	f
0.001	0.999

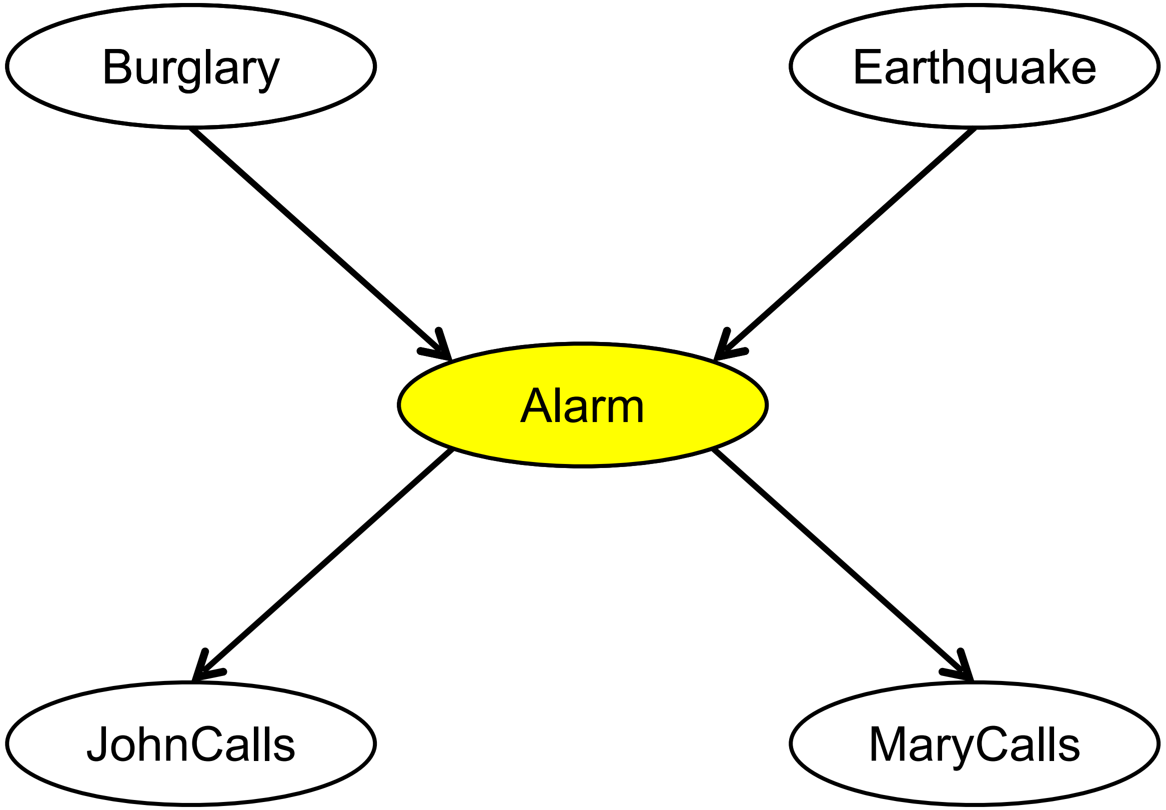


P (*E*)

t	f
0.001	0.999

Bayesian Networks Example

- Set up a network that shows how random variables influence others:



P (B)

t	f
0.001	0.999

P (E)

t	f
0.001	0.999

P (A | B, E)

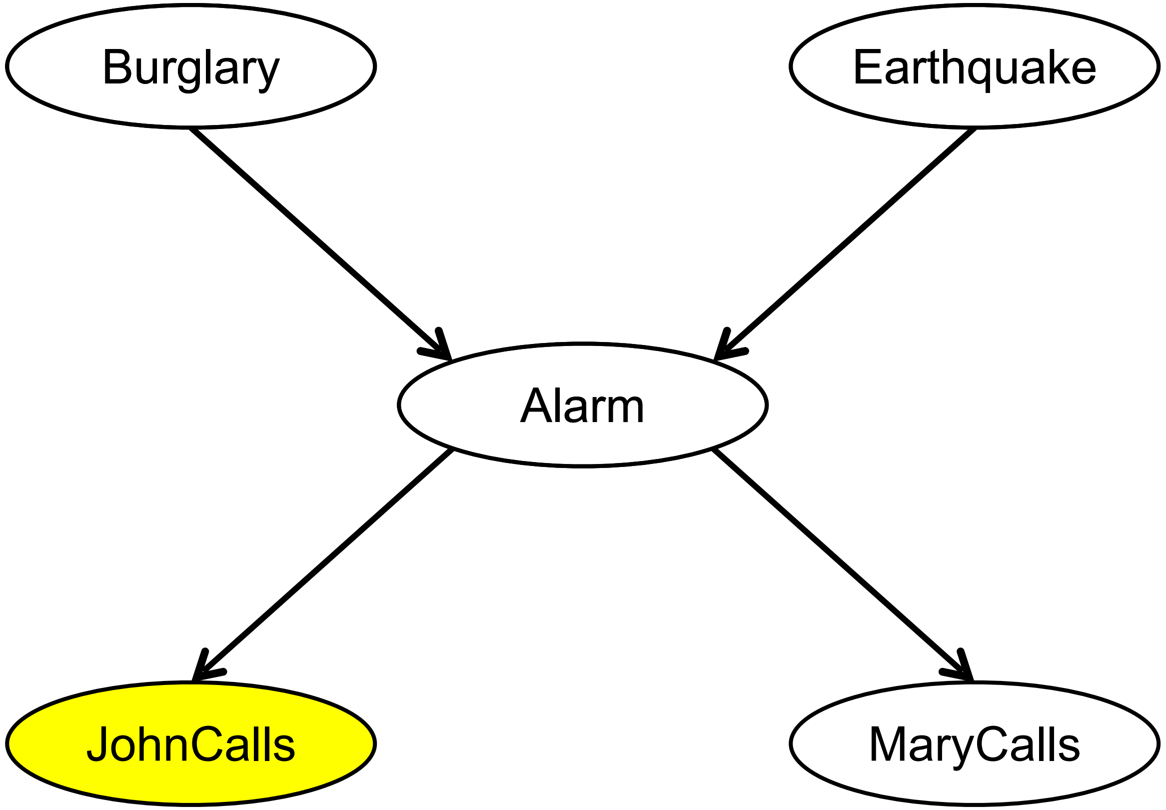
B	*E*	t	f
t	t	0.95	0.05
t	f	0.94	0.06
f	t	0.29	0.71
f	f	0.001	0.999

Bayesian Networks Example

- Set up a network that shows how random variables influence others:

P (B)

t	f
0.001	0.999



P (E)

t	f
0.001	0.999

P (A | B, E)

B	*E*	t	f
t	t	0.95	0.05
t	f	0.94	0.06
f	t	0.29	0.71
f	f	0.001	0.999

P (J | A)

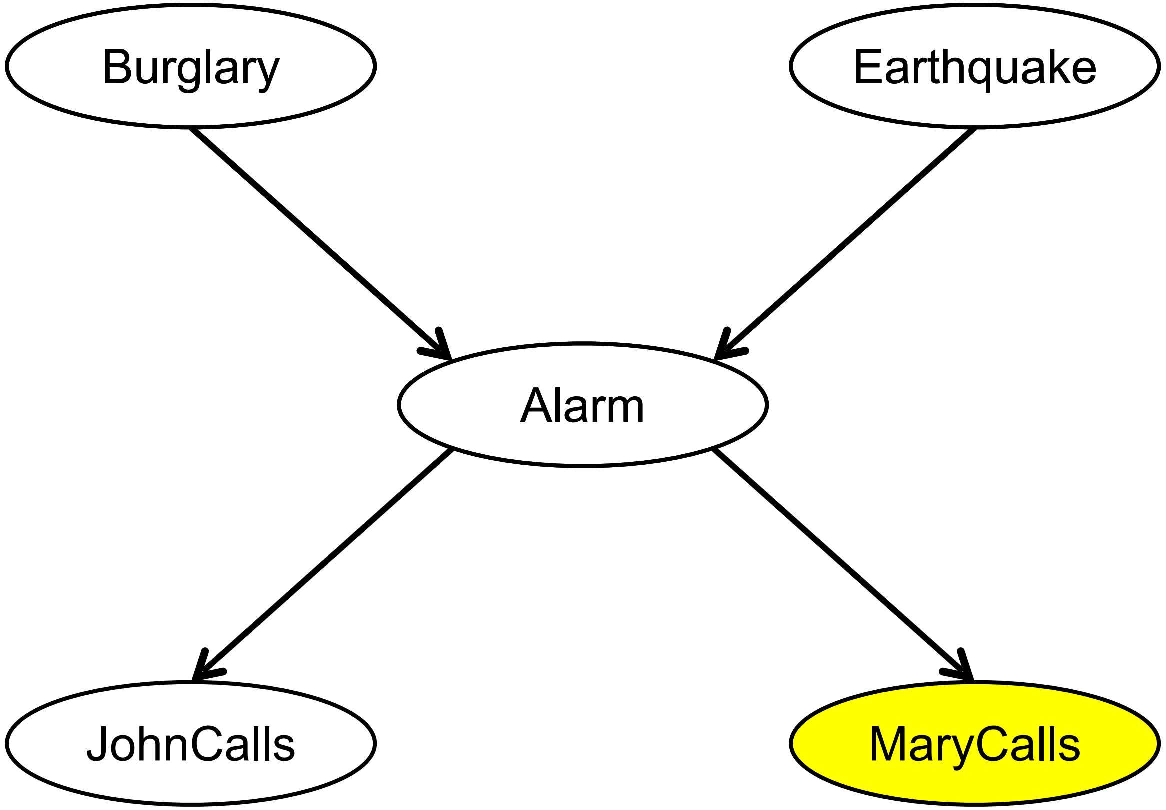
A	t	f
t	0.9	0.1
f	0.05	0.95

Bayesian Networks Example

- Set up a network that shows how random variables influence others:

P (B)

t	f
0.001	0.999



P (E)

t	f
0.001	0.999

P (A | B, E)

B	*E*	t	f
t	t	0.95	0.05
t	f	0.94	0.06
f	t	0.29	0.71
f	f	0.001	0.999

P (J | A)

A	t	f
t	0.9	0.1
f	0.05	0.95

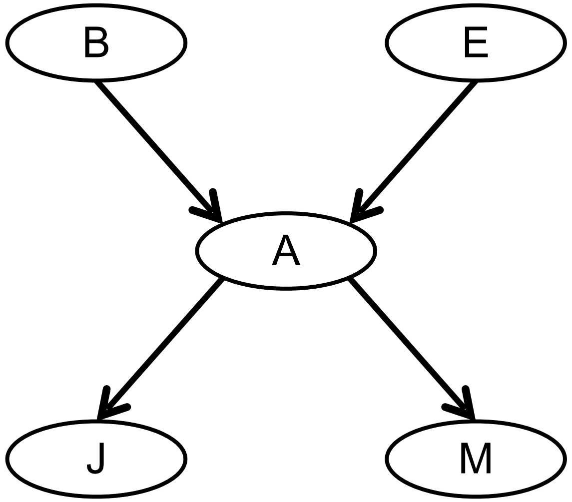
P (M | A)

A	t	f
t	0.7	0.3
f	0.01	0.99

Bayesian Networks: Definition

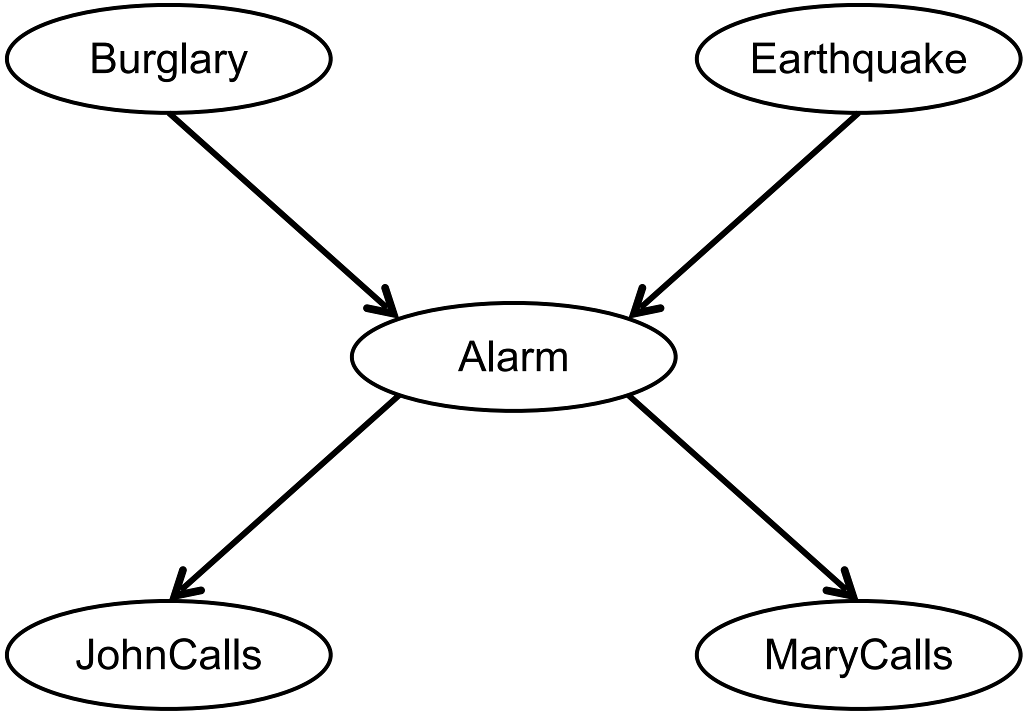
- A BN consists of a **Directed Acyclic Graph (DAG)** and a set of **conditional probability distributions** (CPD)
- The DAG:
  - each node denotes a random variable
  - each edge from X to Y typically represents a causal link from X *to* Y
  - formally: each variable X is independent of its non-descendants given its parents
  - **Each CPD: represents** $P(X \mid Parents(X))$

$$p(x_1, \ldots, x_d) = \prod_{v \in V} p(x_v | x_{\mathrm{pa}(v)})$$



Bayesian Networks: Parameter Counting

- Parameter reduction: standard representation of the joint distribution for Alarm example has $2^5 - 1 = 31$ parameters
- the BN representation of this distribution has 10 parameters



$$
\begin{aligned}
&P(B,E,A,J,M)\\
&= P(B)\\
&\times P(E)\\
&\times P(A \mid B,E)\\
&\times P(J \mid A)\\
&\times P(M \mid A)
\end{aligned}
$$

Inference in Bayesian Networks

Given: values for some variables in the network (*evidence*), and a set of ***query*** variables

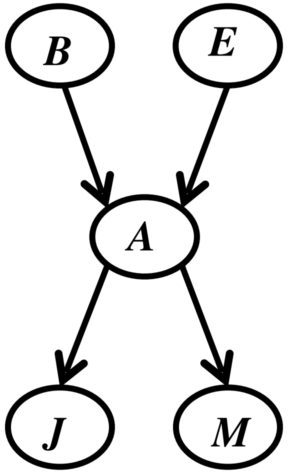
Inference: compute the posterior distribution over the query variables

- Variables that are neither evidence variables nor query variables are ***hidden*** variables
- The BN representation is flexible enough that any set can be the evidence variables and any set can be the query variables

Inference by Enumeration

- Let a denote $\boldsymbol{A}$=true, and $\neg a$ denote $\boldsymbol{A}$=false
- Suppose we're given the query: $P(b \mid j, m)$

  "probability the house is being burglarized given that John and Mary both called"
- From the graph structure, first compute the joint probability:



$$P(b, j, m) = \sum_{e,\neg e} \sum_{a,\neg a} P(b)P(E)P(A \mid b, E)P(j \mid A)P(m \mid A)$$

sum over possible values for E and A variables (e, $\neg e$, a, $\neg a$)

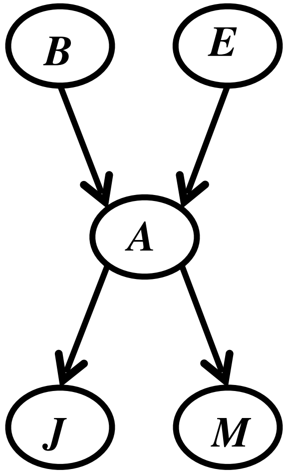
Inference by Enumeration

$$P(b, j, m) = \sum_{e,\neg e} \sum_{a,\neg a} P(b)P(E)P(A \mid b, E)P(j \mid A)P(m \mid A)$$

$$= P(b) \sum_{e,\neg e} \sum_{a,\neg a} P(E)P(A \mid b, E)P(j \mid A)P(m \mid A)$$

P(B)
0.001

P(E)
0.001



B	E	P(A)
t	t	0.95
t	f	0.94
f	t	0.29
f	f	0.001

A	P(J)
t	0.9
f	0.05

A	P(M)
t	0.7
f	0.01

$B \quad E \quad A \quad J \quad M$

$$\begin{aligned} = 0.001 \times (&0.001 \times 0.95 \times 0.9 \times 0.7 + && e, a \\ &0.001 \times 0.05 \times 0.05 \times 0.01 + && e, \neg a \\ &0.999 \times 0.94 \times 0.9 \times 0.7 + && \neg e, a \\ &0.999 \times 0.06 \times 0.05 \times 0.01) && \neg e, \neg a \end{aligned}$$

Inference by Enumeration

- Next do equivalent calculation for $P(\neg b,\ j, m)$

and determine $P(b \mid j, m)$

$$P(b \mid j,m) = \frac{P(b,j,m)}{P(j,m)} = \frac{P(b,j,m)}{P(b,j,m) + P(\neg b,j,m)}$$

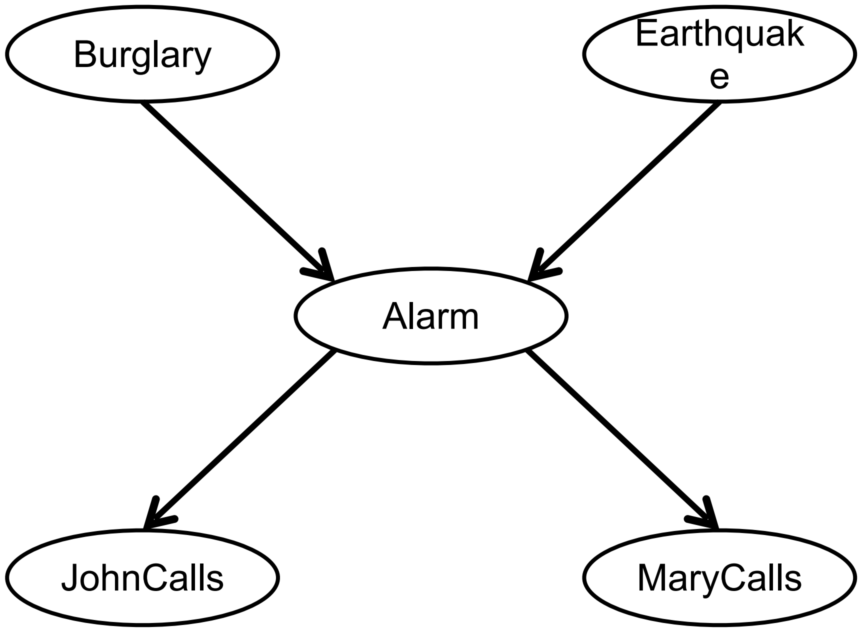
So: exact method, but can be intractably hard.

- Efficient for small BNs
- Approximate inference sometimes available.
  - Example: Markov chain Monte Carlo (MCMC) approaches.

Learning Bayes Nets

- **Problem 1 (parameter learning)**: given a set of training instances and the graph structure of a Bayes Net.

B	E	A	J	M
f	f	f	t	f
f	t	f	f	f
f	f	t	f	t
		…		



- **Goal**: infer the parameters of the CPDs

Learning Bayes Nets

- **Problem 2 (structure learning)**: given a set of training instances

B	E	A	J	M
f	f	f	t	f
f	t	f	f	f
f	f	t	f	t
		…		

- **Goal**: infer the graph structure (and then possibly also the parameters of the CPDs)

Parameter Learning: MLE

- **Goal**: infer the parameters of the CPDs
- As usual, can use maximum likelihood estimation.

Probabilities depend on θ

$$L(\theta; D, G) = P(D; \theta, G) = \prod_{i=1}^{n} P(x_1^{(i)}, x_2^{(i)}, \ldots, x_k^{(i)})$$

Data

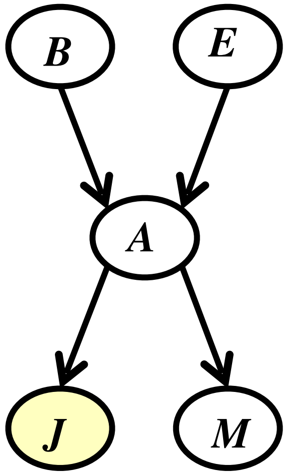
Graph

$$= \prod_{i=1}^{n} \prod_{j=1}^{k} P(x_k^{(i)} \,|\, \texttt{Parents}(x_k^{(i)}))$$

$$= \prod_{j=1}^{k} \underbrace{\prod_{i=1}^{n} \left(P(x_k^{(i)} \,|\, \texttt{Parents}(x_k^{(i)})) \right)}_{\text{independent parameter learning problem for each CPD}}$$

Parameter Learning: MLE Example

- **Goal**: infer the parameters of the CPDs
- Consider estimating the CPD parameters for B and J in the alarm network given the following data set



B	E	A	J	M
f	f	f	t	f
f	t	f	f	f
f	f	f	t	t
t	f	f	f	t
f	f	t	t	f
f	f	t	f	t
f	f	t	t	t
f	f	t	t	t

$$P(b) = \frac{1}{8} = 0.125$$

$$P(\neg b) = \frac{7}{8} = 0.875$$

$$P(j \mid a) = \frac{3}{4} = 0.75$$

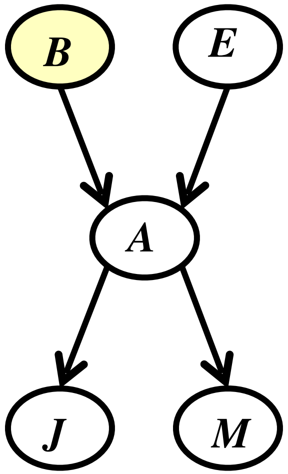
$$P(\neg j \mid a) = \frac{1}{4} = 0.25$$

$$P(j \mid \neg a) = \frac{2}{4} = 0.5$$

$$P(\neg j \mid \neg a) = \frac{2}{4} = 0.5$$

Parameter Learning: MLE Example

- **Goal**: infer the parameters of the CPDs
- Consider estimating the CPD parameters for B and J in the alarm network given the following data set



B	E	A	J	M
f	f	f	t	f
f	t	f	f	f
f	f	f	t	t
f	f	f	f	t
f	f	t	t	f
f	f	t	f	t
f	f	t	t	t
f	f	t	t	t

$$P(b) = \frac{0}{8} = 0$$

$$P(\neg b) = \frac{8}{8} = 1$$

do we really want to set this to 0?

Parameter Learning: Laplace Smoothing

- Instead of estimating parameters strictly from the data, we could start with some <u>prior belief</u> for each
- For example, we could use *Laplace estimates*

$$P(X = x) = \frac{n_x + 1}{\sum_{v \in \text{Values}(X)} (n_v + 1)}$$

pseudocounts

where n_v represents the number of occurrences of value v

- Recall: we did this for Naïve Bayes

Break & Quiz

Quiz

Can the Naïve Bayes' model be represented as a Bayesian network?

If no, explain why. If yes, draw the network.

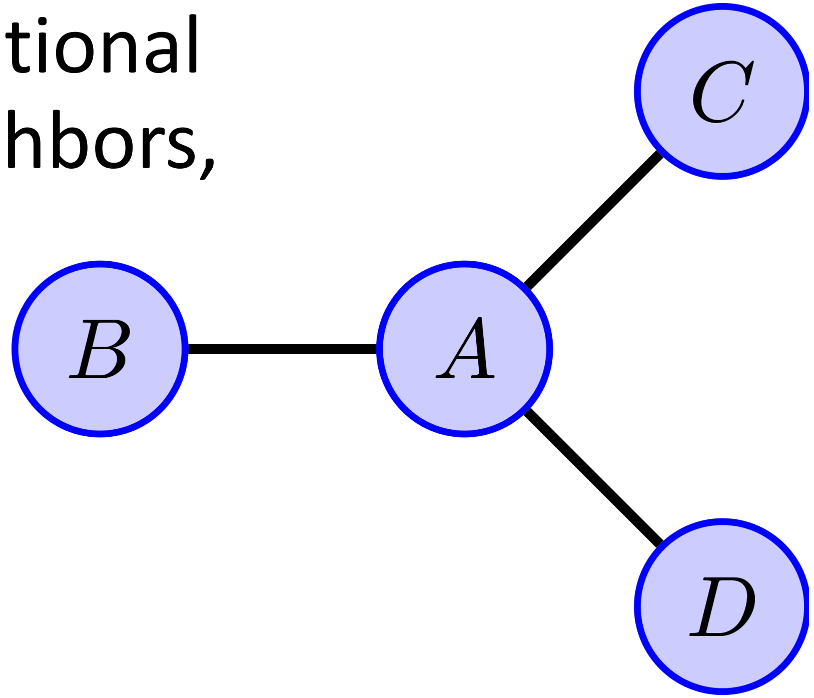
Ans: Yes

Undirected Graphical Models

- Still want to encode conditional independence, but not in a causal way (ie, no parents, direction)
  - **Why**? Allows for modeling other distributions that Bayes nets can't, allows for other algorithms

- Graph directly encodes a type of conditional independence. If nodes i,j are not neighbors,

$$X_i \perp X_j | X_{V \setminus \{i,j\}}$$



Outline

- **Bayesian Networks Review**
  - Definition, examples, inference, learning
- **Structure learning**
  - Chow-Liu Algorithm
- **D-separation**

Structure Learning

- Generally a hard problem, many approaches.
  - Exponentially (or worse) many structures in # variables
  - Can either use heuristics or restrict to some tractable subset of networks. Ex: **trees**

- Chow-Liu Algorithm
  - Learns a BN with a tree structure that **maximizes the likelihood of the training data**

1. Compute weight $I(X_i, X_j)$ of each possible edge (X_i, X_j)
2. Find maximum weight spanning tree (MST)

Chow-Liu: Computing weights

- Use mutual information to calculate edge weights

$$I(X,Y) = \sum_{x \in \text{values}(X)} \sum_{y \in \text{values}(Y)} P(x,y) \log_2 \frac{P(x,y)}{P(x)P(y)}$$

- The probabilities are calculated empirically using data.
    - Recall decision trees: how much information does knowing Y give us about the value of X.

Chow-Liu: Finding MST

- Many algorithms for calculating MST (e.g Kruskal's, Prim's)

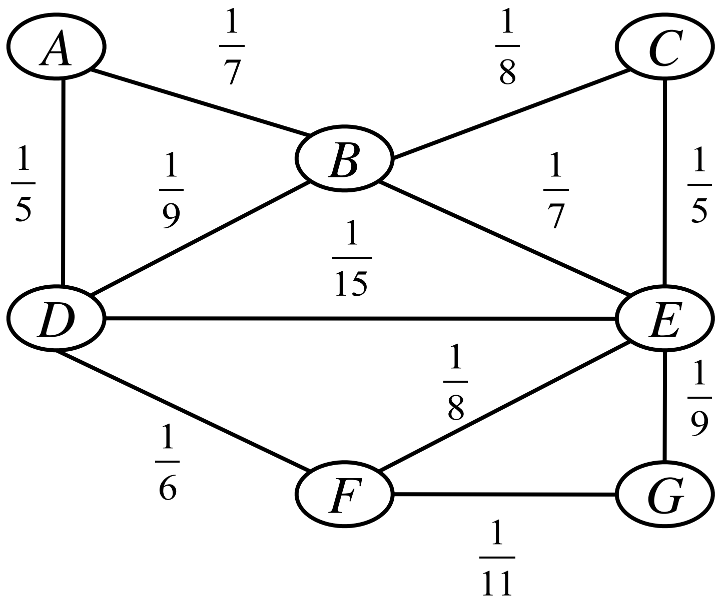
- Kruskal's algorithm

given: graph with vertices V and edges E

$E_{new} \leftarrow \{\ \}$
for each (u, v) in E ordered by weight (from high to low)
$\{$
  remove (u, v) from E
  if adding (u, v) to E_{new} does not create a cycle
    add (u, v) to E_{new}
$\}$
return V and E_{new} which represent an MST

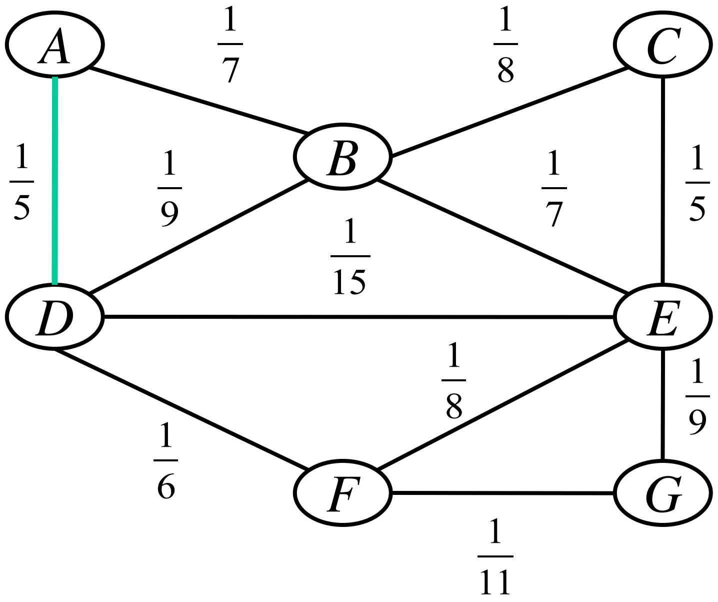
Chow-Liu: Example

- First, calculate empirical mutual information for each pair and calculate edge weights.
    - Graph is usually fully connected (using a non-complete graph for clarity)

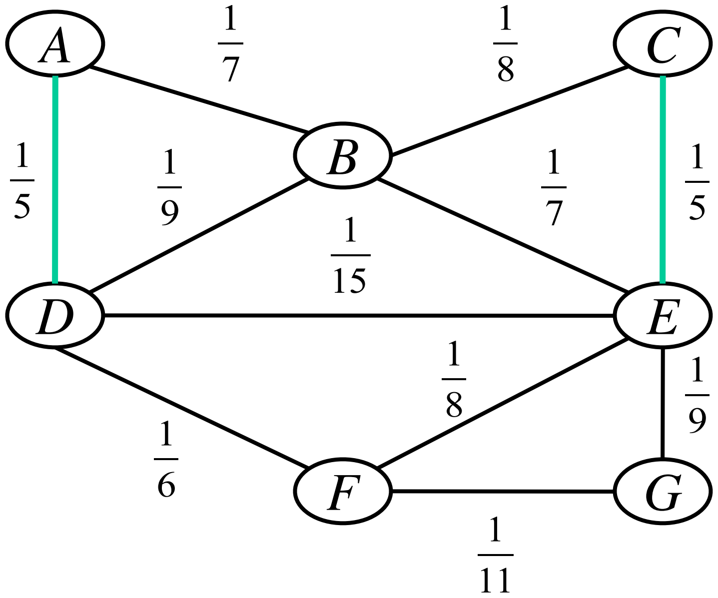


Chow-Liu: Example (cont'd)

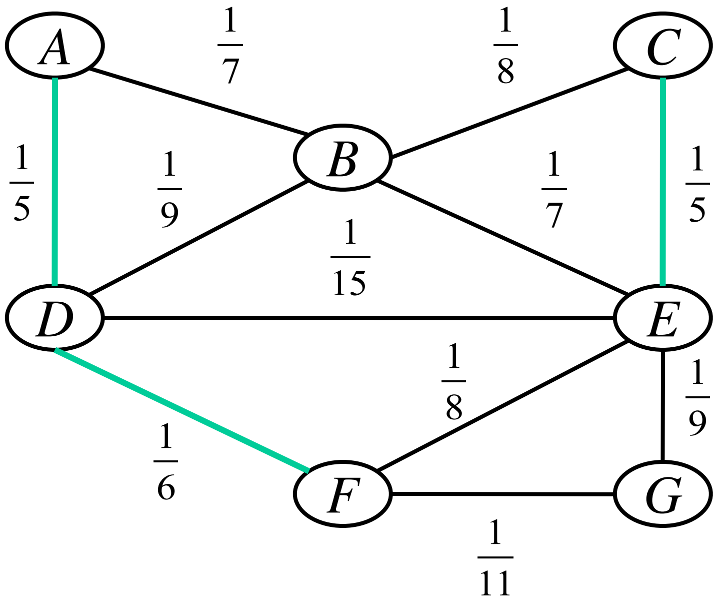
i.



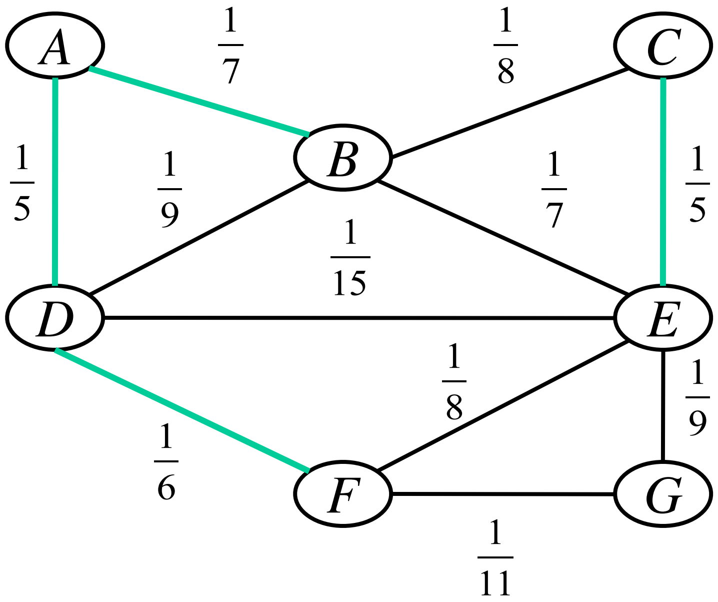
ii.



iii.

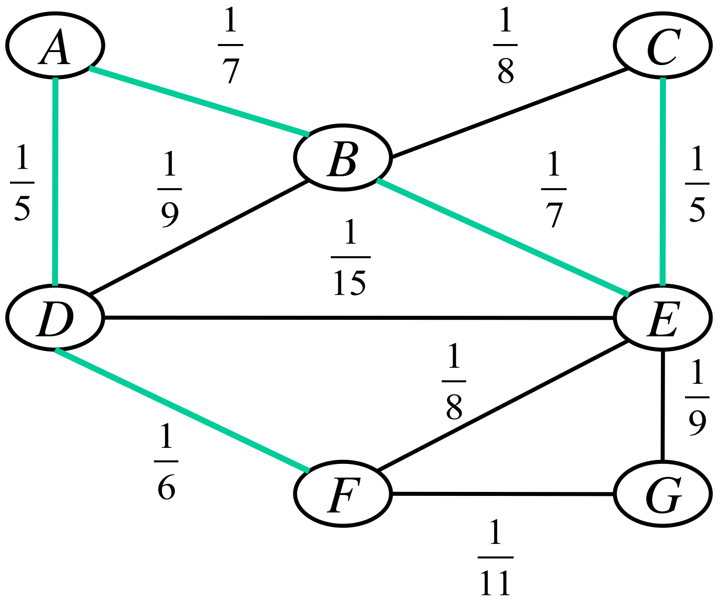


iv.

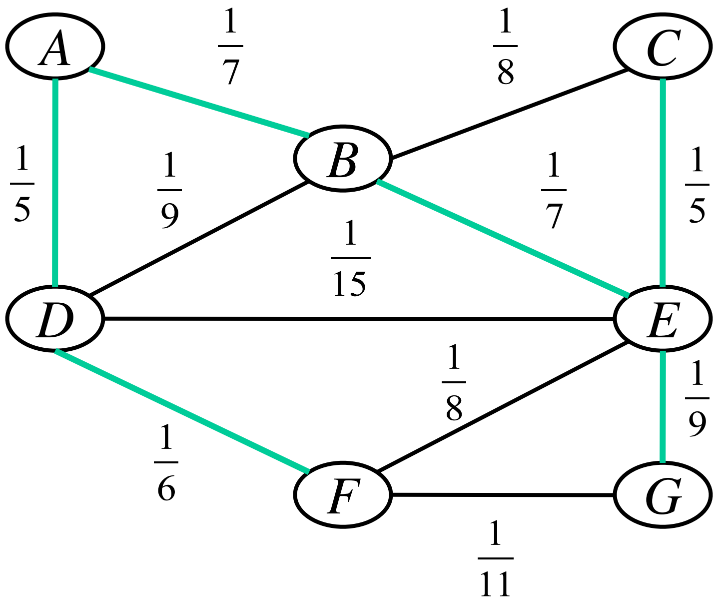


Chow-Liu: Example (cont'd)

v.



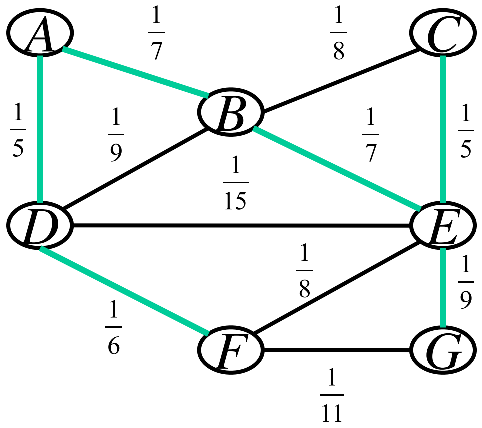
vi.

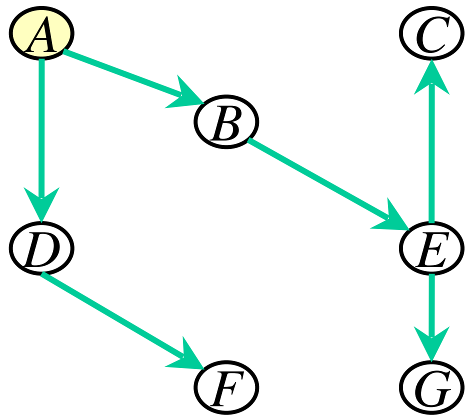


Chow-Liu Algorithm

1. Finding tree structures is a ‘second order’ approximation
   - First order: product of marginals
     $$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i)$$
   - Second order: allow conditioning on one variable
     $$P(X_1, \ldots, X_n) = P(X_1) \prod_{i=2}^{n} P(X_i | X_{i-1})$$

2. To assign directions in a Bayes’ network, pick a root and making everything directed from root (may require domain expertise)

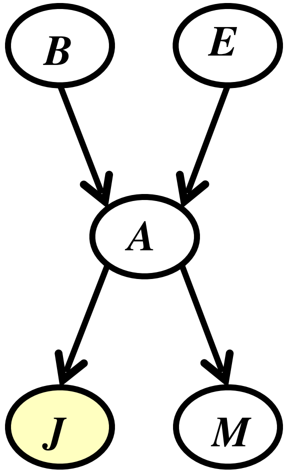




Outline

- **Bayesian Networks Review**
  - Definition, examples, inference, learning
- **Structure learning**
  - Chow-Liu Algorithm
- **D-separation**

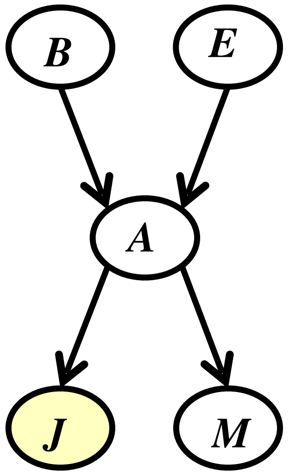
D-separation in Bayesian Networks



- Which of the following are true?
  1. $J \perp\!\!\!\perp M$
  2. $J \perp\!\!\!\perp M \mid A$
  3. $B \perp\!\!\!\perp J$
  4. $B \perp\!\!\!\perp J \mid A$
  5. $B \perp\!\!\!\perp E$
  6. $B \perp\!\!\!\perp E \mid A$

D-separation in Bayesian Networks

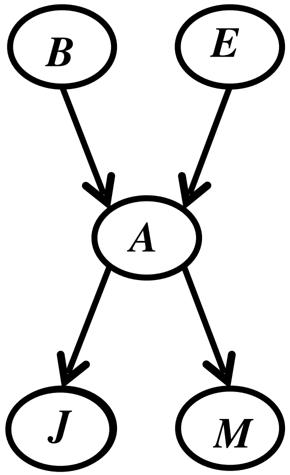
- Still want to encode conditional independence, but not in a,



- Which of the following are true?
  1. $J \perp\!\!\!\perp M$ **(False)**
  2. $J \perp\!\!\!\perp M \mid A$ **(True)**
  3. $B \perp\!\!\!\perp J$ **(False)**
  4. $B \perp\!\!\!\perp J \mid A$ **(True)**
  5. $B \perp\!\!\!\perp E$ **(True)**
  6. $B \perp\!\!\!\perp E \mid A$ **(False)**

D-separation in Bayesian Networks

- D-separation: A formal way to answer questions of conditional independence:
  - E.g. $J \perp\!\!\!\perp M \mid A$, $J \perp\!\!\!\perp E \mid B, M$ etc.



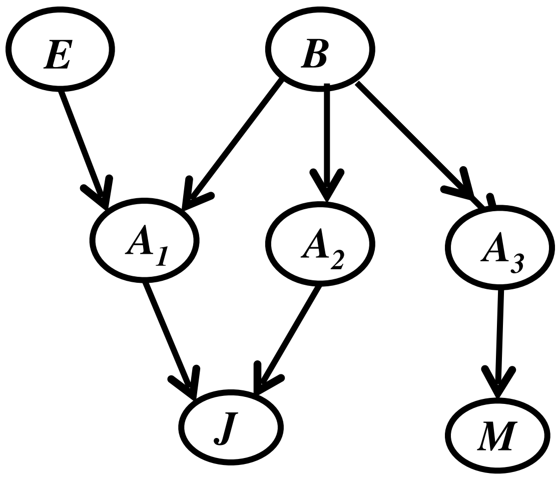
- Triples: Any 3 connected vertices
- We say that a triple is active if
  - (Causal chain): X ➔ Y ➔ Z (Y is unobserved)
  - (Common cause): X 🡐 Y ➔ Z (Y is unobserved)
  - (Common effect): X ➔ Y 🡐 Z (Y or any descendent of Y is observed)

- An (undirected) path is active if all of it’s triples are active.

D-separation in Bayesian Networks

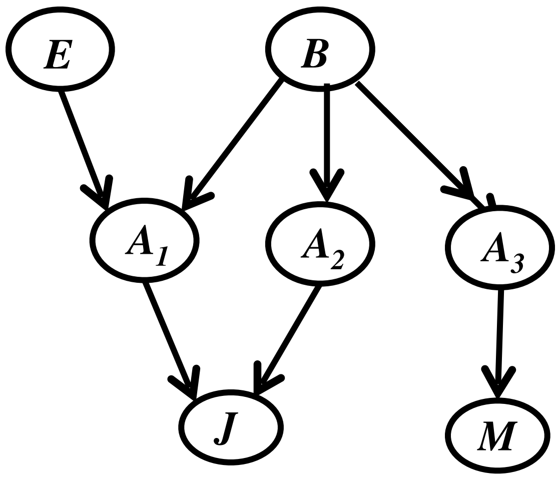
- Goal: Answer queries of the form: $A \perp\!\!\!\perp B \mid \{C, D, \ldots\}$
- D-separation Algorithm:
  - For all (undirected) paths from A to B
    - Check if path is active (i.e all triples are active)
      - Return "$A \perp\!\!\!\perp B \mid \{C, D, \ldots\}$ is **not** guaranteed"
  - If all paths are inactive:
    - Return "$A \perp\!\!\!\perp B \mid \{C, D, \ldots\}$ is true"

D-separation Examples

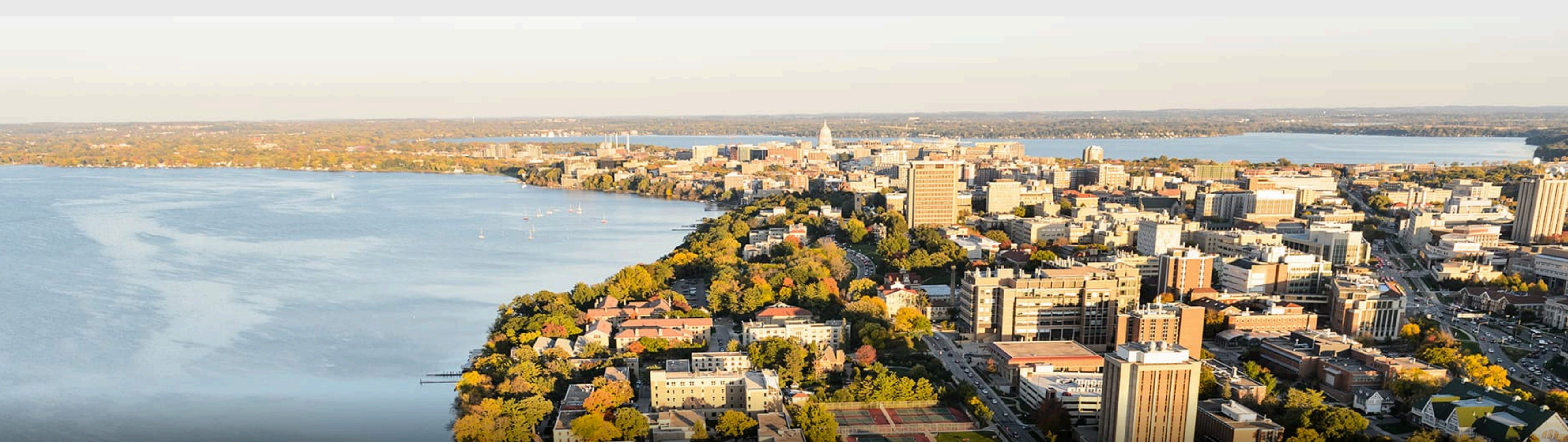


- Are the following conditional independences guaranteed?
  1. $B \perp\!\!\!\perp M$
  2. $B \perp\!\!\!\perp M \mid A_3$
  3. $E \perp\!\!\!\perp B$
  4. $E \perp\!\!\!\perp B \mid A_1$
  5. $E \perp\!\!\!\perp B \mid A_2$
  6. $E \perp\!\!\!\perp B \mid J$
  7. $A_1 \perp\!\!\!\perp A_2$
  8. $A_1 \perp\!\!\!\perp A_2 \mid E$
  9. $A_2 \perp\!\!\!\perp A_3 \mid B$
  10. $J \perp\!\!\!\perp M$
  11. $J \perp\!\!\!\perp M \mid A_3$

D-separation Examples

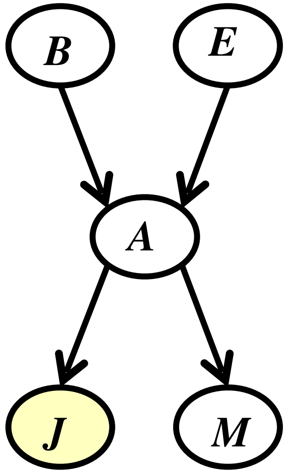


- Are the following conditional independences guaranteed?
  1. $B \perp\!\!\!\perp M$ **(False)**
  2. $B \perp\!\!\!\perp M \mid A_3$ **(True)**
  3. $E \perp\!\!\!\perp B$ **(True)**
  4. $E \perp\!\!\!\perp B \mid A_1$ **(False)**
  5. $E \perp\!\!\!\perp B \mid A_2$ **(True)**
  6. $E \perp\!\!\!\perp B \mid J$ **(False)**
  7. $A_1 \perp\!\!\!\perp A_2$ **(False)**
  8. $A_1 \perp\!\!\!\perp A_2 \mid E$ **(False)**
  9. $A_2 \perp\!\!\!\perp A_3 \mid B$ **(True)**
  10. $J \perp\!\!\!\perp M$ **(False)**
  11. $J \perp\!\!\!\perp M \mid A_3$ **(True)**



Break & Quiz

Quiz



True or False:
Bayesian networks can be used for unsupervised learning only. They cannot be used for supervised learning.

Ans: False

Thanks Everyone!

Some of the slides in these lectures have been adapted/borrowed from materials developed by Mark Craven, David Page, Jude Shavlik, Tom Mitchell, Nina Balcan, Elad Hazan, Tom Dietterich, Pedro Domingos, Jerry Zhu, Yingyu Liang, Volodymyr Kuleshov, Fei-Fei Li, Justin Johnson, Serena Yeung, Pieter Abbeel, Peter Chen, Jonathan Ho, Aravind Srinivas, and Fred Sala