CS 760: Machine Learning
Graphical Models

Josiah Hanna

University of Wisconsin-Madison

November 7, 2023

Announcements

- HW 5 due Thursday at 9:30am
- Midterm evaluation follow-up

Graphical Models Motivation

- Still considering generative modeling (unsupervised learning).
- **Given** a collection of variables: $X_1, X_2, \ldots, X_k$.
- **Want:** model of $P(X_1 = x_1, X_2 = x_2, \ldots, X_k = x_k)$.
- **Why?** Make flexible predictions by computing posteriors.
- Example: compute $P(X_1 = x_1 \mid X_2 = x_2)$.

Outline

- **Probability Review**
  - Basics, joint probability, conditional probabilities, etc
- **Bayesian Networks**
  - Definition, examples, inference, learning
- **Undirected Graphical Models**
  - Definitions, MRFs, exponential families
- **Structure learning**
  - Chow-Liu Algorithm

Outline

- **Probability Review**
  - Basics, joint probability, conditional probabilities, etc
- **Bayesian Networks**
  - Definition, examples, inference, learning
- **Undirected Graphical Models**
  - Definitions, MRFs, exponential families
- **Structure learning**
  - Chow-Liu Algorithm

Basics: **Joint Distributions**

- Joint distribution of 2 random variables X and Y

$$P(X = a, Y = b)$$

- Or more variables.

$$P(X_1 = x_1, X_2 = x_2, \ldots, X_k = x_k)$$

- If each of the k variables can take on m values then the joint distribution requires storing m^k values.

Basics: **Marginal** Probability

- Given a joint distribution

$$P(X = a, Y = b)$$

  - Compute the distribution of just one variable:

$$P(X = a) = \sum_b P(X = a, Y = b)$$

  - This is the "marginal" distribution.

Basics: **Marginal** Probability

$$P(X = a) = \sum_b P(X = a, Y = b)$$

	Sunny	Cloudy	Rainy
hot	150/365	40/365	5/365
cold	50/365	60/365	60/365

$$[P(\text{hot}), P(\text{cold})] = [\tfrac{195}{365}, \tfrac{170}{365}]$$

Independence

- Independence for a set of events $A_1, \ldots, A_k$

$$P(A_{i_1} A_{i_2} \cdots A_{i_j}) = P(A_{i_1}) P(A_{i_2}) \cdots P(A_{i_j})$$

for all possible choices of $i_1, \ldots, i_j$ with
$1 \leq i_1 < i_2 < \ldots < i_j \leq k$.

- **Why useful?** Dramatically reduces the complexity
- Collapses joint into **product** of marginals
    - Note sometimes we have only pair-wise independence.

Uncorrelatedness

- For random variables, uncorrelated means

$$E[XY] = E[X]E[Y]$$

Note: weaker condition than independence.

- Independence implies uncorrelated (easy to see)

$$E[XY] = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} \Pr(X = x, Y = y) xy$$

$$= \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} \Pr(X = x) \Pr(Y = y) xy$$

$$= \sum_{x \in \mathcal{X}} \Pr(X = x) x \sum_{y \in \mathcal{Y}} \Pr(Y = y) y = E[X]E[Y]$$

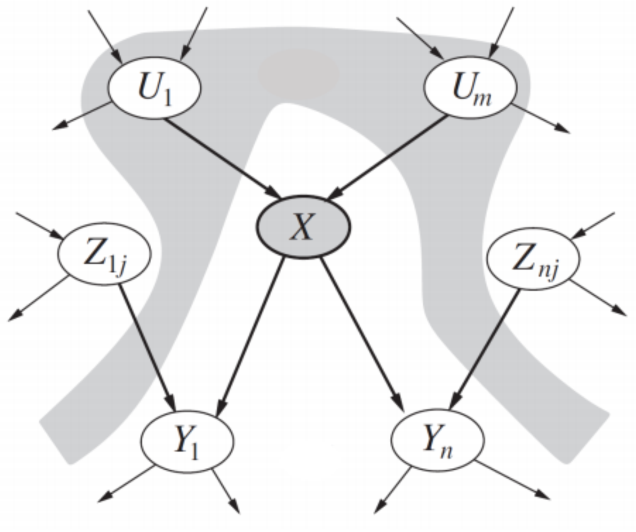
Conditional Probability

- When we know something,

$$P(X = a|Y = b) = \frac{P(X = a, Y = b)}{P(Y = b)}$$

- **Conditional independence**

$$P(X, Y|Z) = P(X|Z)P(Y|Z)$$



Credit: Devin Soni

Chain Rule (of Probability)

- Apply repeatedly,

$$P(A_1, A_2, \ldots, A_n)$$
$$= P(A_1)P(A_2|A_1)P(A_3|A_2, A_1) \ldots P(A_n|A_{n-1}, \ldots, A_1)$$

- Holds for any probability distribution over $A_1, A_2, \ldots, A_n$
- Note: probability table is still big!
  - If some **conditional independence**, can factor!
  - Leads to **probabilistic graphical models (this lecture)**

Law of Total Probability

- Partition the sample space into disjoint B_1, …, B_k
- Then,

$$P(A) = \sum_i P(A|B_i)P(B_i)$$

Bayesian Inference

- Bayes rule:

$$P(H|E_1, E_2, \ldots, E_n) = \frac{P(E_1, \ldots, E_n|H)P(H)}{P(E_1, E_2, \ldots, E_n)}$$

- Under **conditional independence**

$$P(H|E_1, E_2, \ldots, E_n) = \frac{P(E_1|H)P(E_2|H)\cdots, P(E_n|H)P(H)}{P(E_1, E_2, \ldots, E_n)}$$

Random Vectors & Covariance

- Recall variance: $$\mathbb{E}[(X - E[X])^2]$$
- For a **random vector**
  - Note: size *d x d*. All variables are centered

$$\Sigma = \begin{bmatrix} \mathbb{E}[(X_1 - \mathbb{E}[X_1])^2] & \dots & [(X_1 - \mathbb{E}[X_1])((X_n - \mathbb{E}[X_n])] \\ \vdots & \vdots & \vdots \\ [(X_n - \mathbb{E}[X_n])((X_1 - \mathbb{E}[X_1])] & \dots & \mathbb{E}[(X_n - \mathbb{E}[X_n])^2] \end{bmatrix}$$

Covariance

Diagonals: Variance

Break & Quiz

Break & Quiz

50% of emails are spam. Software has been applied to filter spam. A certain brand of software claims that it can detect 99% of spam emails, and the probability for a false positive (a non-spam email detected as spam) is 5%. Now if an email is detected as spam, then what is the probability that it is in fact a nonspam email?

A. 5/104
B. 95/100
C. 1/100
D. 1/2

Break & Quiz

50% of emails are spam. Software has been applied to filter spam. A certain brand of software claims that it can detect 99% of spam emails, and the probability for a false positive (a non-spam email detected as spam) is 5%. Now if an email is detected as spam, then what is the probability that it is in fact a nonspam email?

A. **5/104**
B. 95/100
C. 1/100
D. 1/2

Outline

- **Probability Review**
  - Basics, joint probability, conditional probabilities, etc
- **Bayesian Networks**
  - Definition, examples, inference, learning
- **Undirected Graphical Models**
  - Definitions, MRFs, exponential families
- **Structure learning**
  - Chow-Liu Algorithm

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

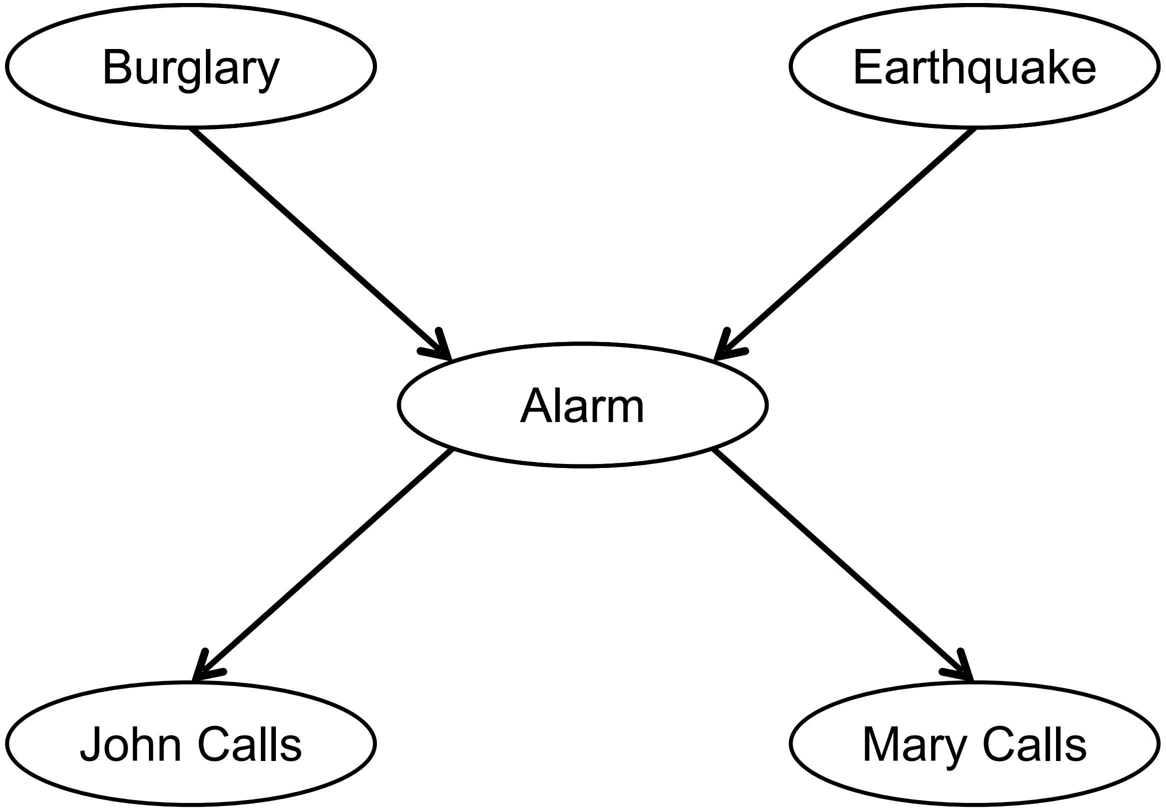
  M = Mary calls to report the alarm

- 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:

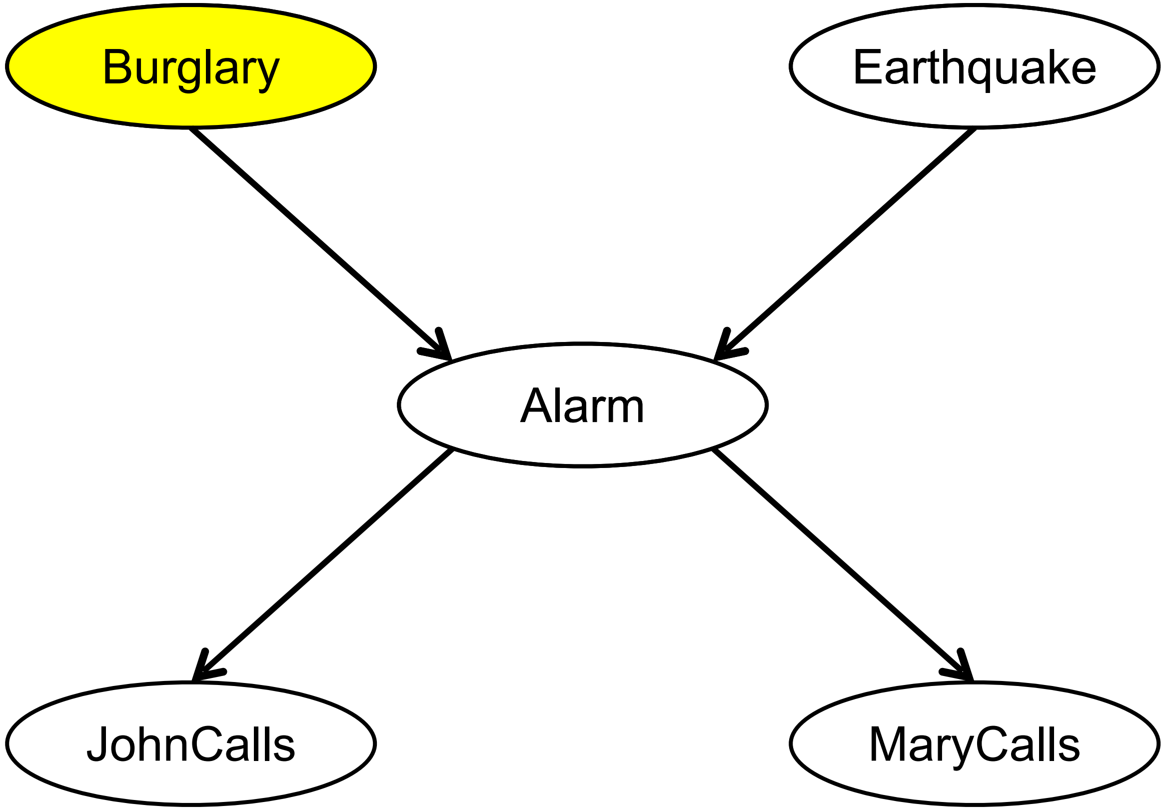


Bayesian Networks Example

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

P (*B*)

t	f
0.001	0.999

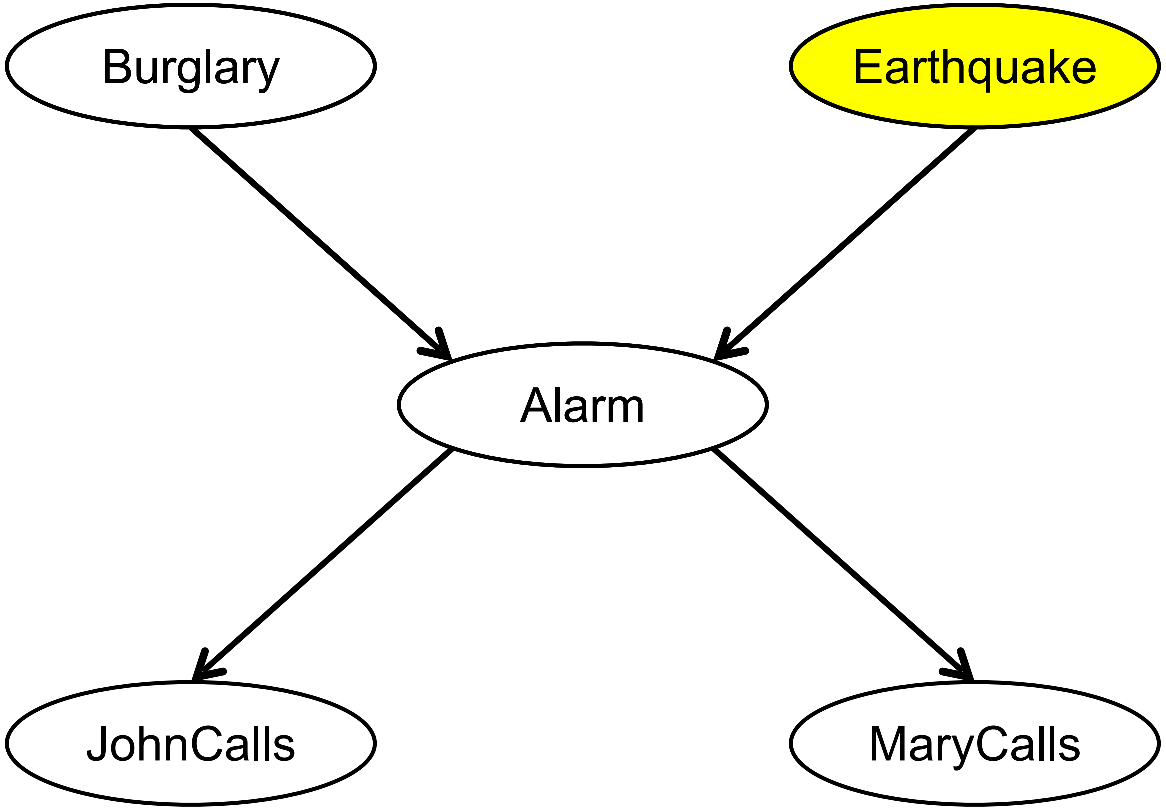


Bayesian Networks Example

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

P (*B*)

t	f
0.001	0.999



P (*E*)

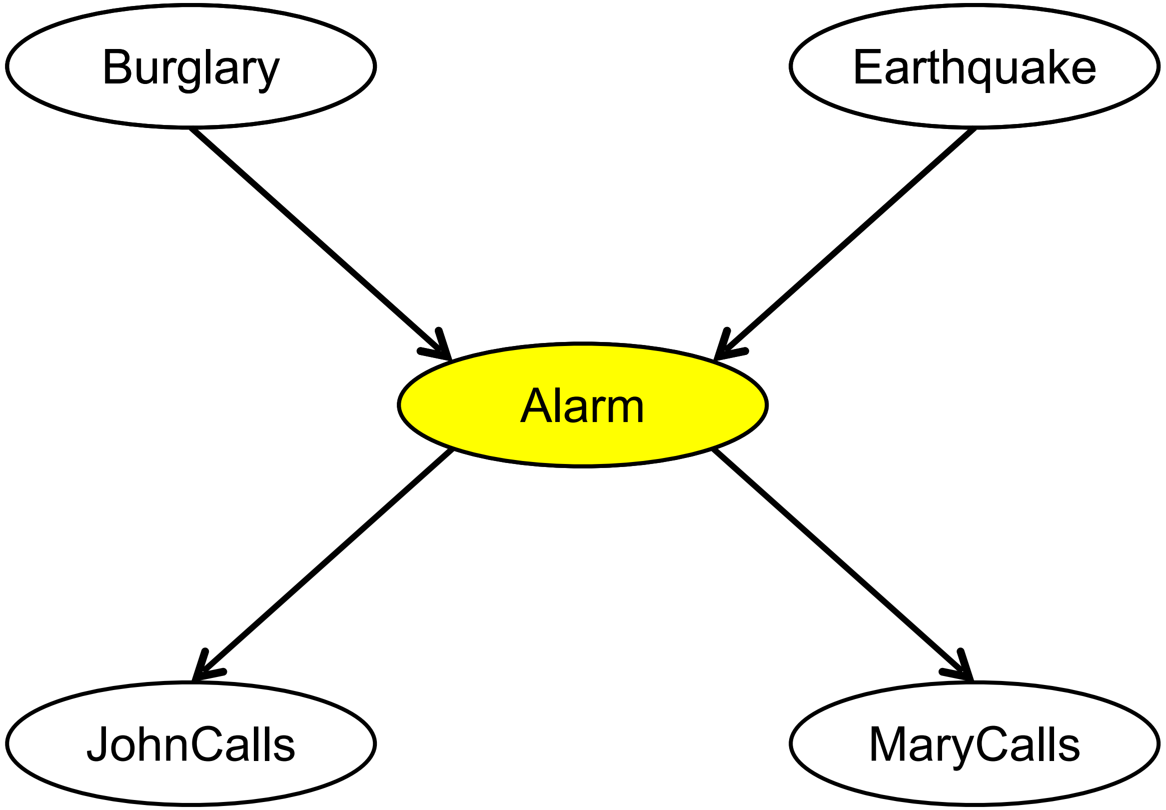
t	f
0.001	0.999

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)

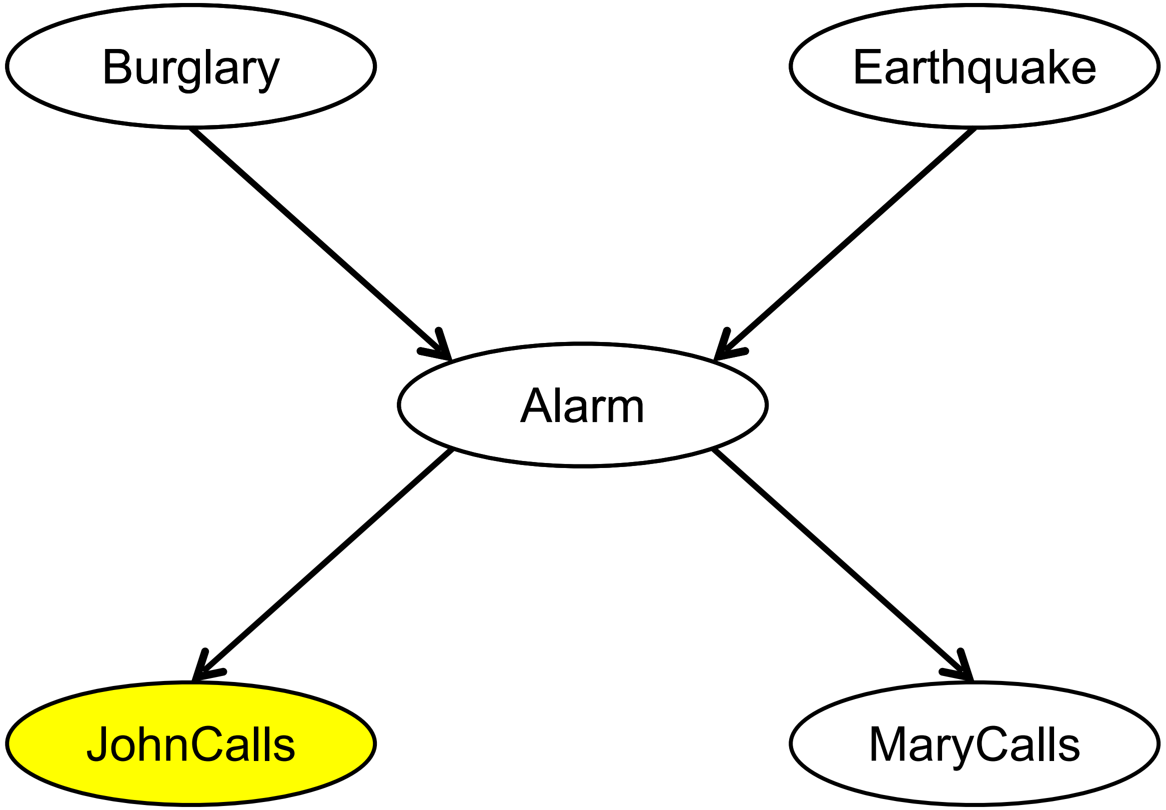
B	*E*	t	f
t	t	0.95	0.05
t	f	0.94	0.06
f	t	0.29	0.71
f	f	0.001	0.999

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)

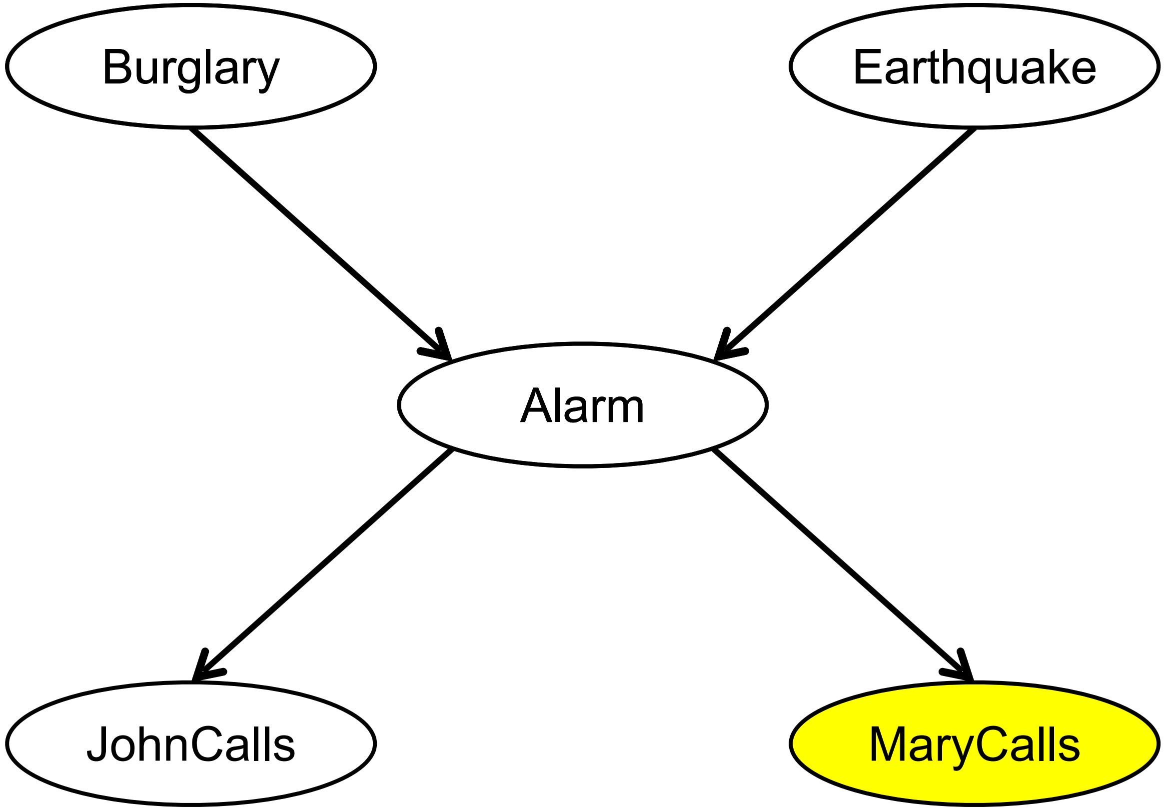
A	t	f
t	0.9	0.1
f	0.05	0.95

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

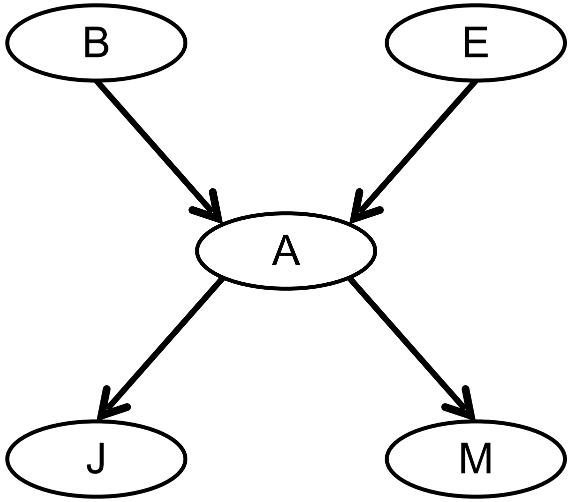
P (M | A)

A	t	f
t	0.7	0.3
f	0.01	0.99

Bayesian Networks: Definition

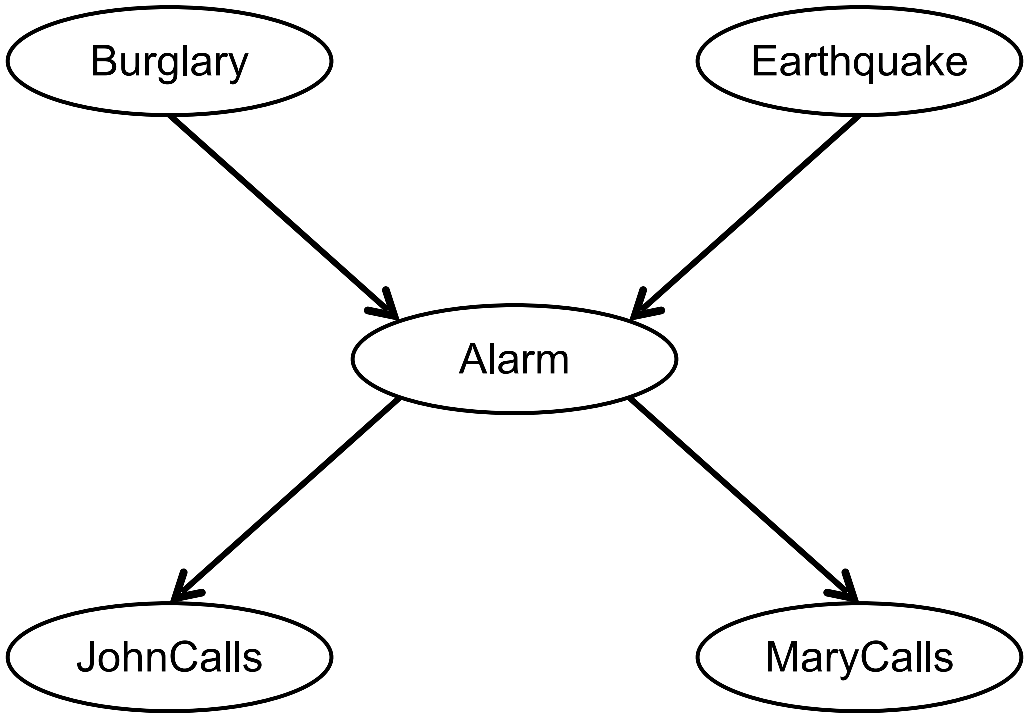
- A BN consists of a **Directed Acyclic Graph (DAG)** and a set of **conditional probability distributions** (CPD)
- The DAG:
  - each node denotes a random variable
  - each edge from X to Y typically represents a causal link from X *to* Y
  - formally: each variable X is independent of its non-descendants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

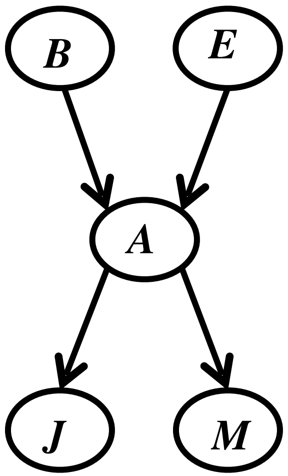
Given: values for some variables in the network (*evidence*), and a set of ***query*** variables

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, $\neg a$)

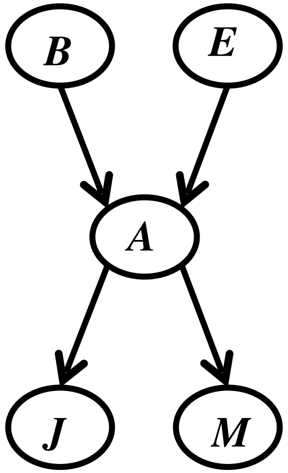
Inference by Enumeration

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

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

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 *E* *A* *J* *M*

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

Inference by Enumeration

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

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

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

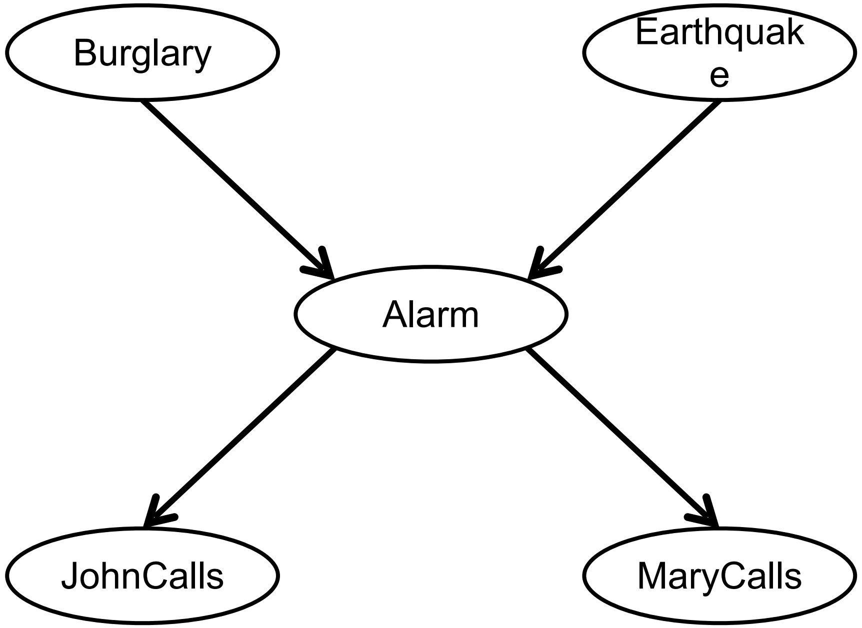
So: exact method, but can be intractably hard.

- Efficient for small BNs
- Approximate inference sometimes available

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 MLE

Probabilities depend on θ

$$L(\theta : D, G) = P(D \mid G, \theta) = \prod_{d \in D} P(x_1^{(d)}, x_2^{(d)}, \ldots, x_n^{(d)})$$

Data Graph

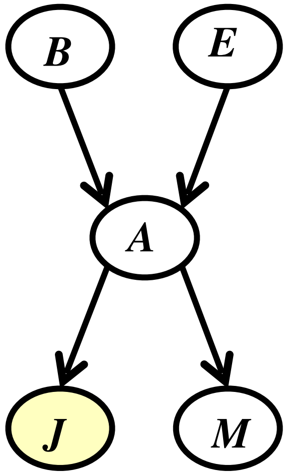
$$= \prod_{d \in D} \prod_i P(x_i^{(d)} \mid Parents(x_i^{(d)}))$$

$$= \prod_i \underbrace{\left(\prod_{d \in D} P(x_i^{(d)} \mid Parents(x_i^{(d)})) \right)}_{\text{independent parameter learning problem for each CPD}}$$

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

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

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

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

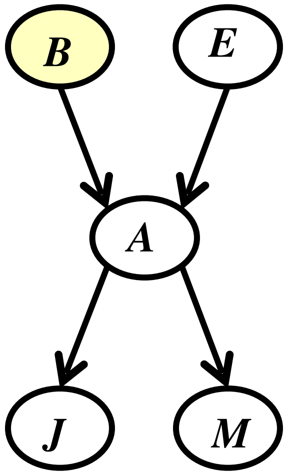
$$P(\neg j \mid a) = \frac{1}{4} = 0.25$$

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

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

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 prior belief 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

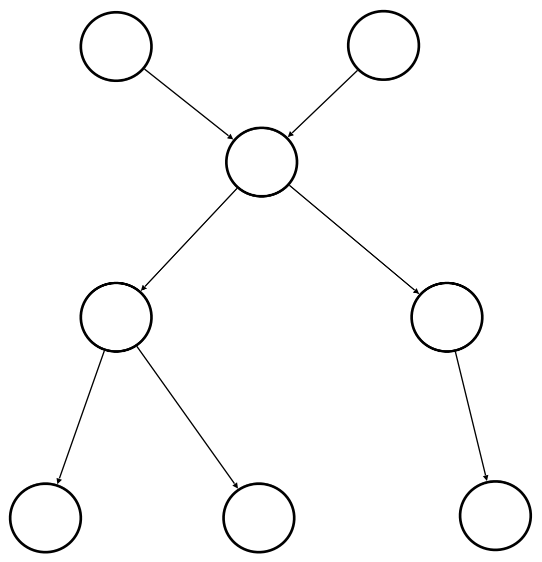
where n_v represents the number of occurrences of value v

- Recall: we did this for Naïve Bayes

Break & Quiz

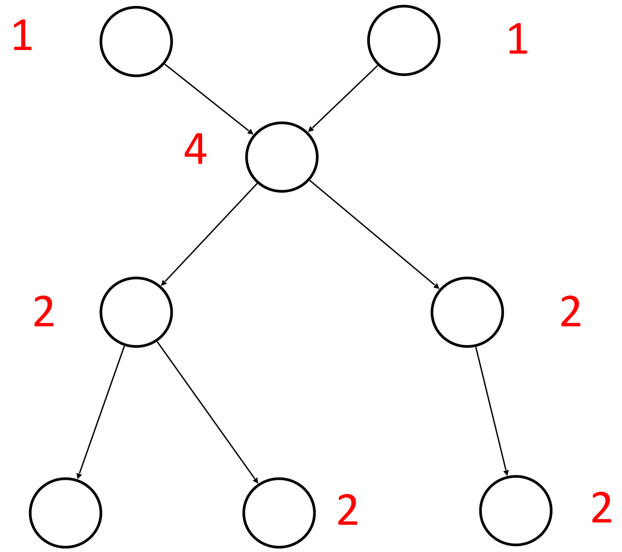
Q2-1: Consider a case with 8 binary random variables, how many parameters does a BN with the following graph structure have?

1. 12
2. 14
3. 16
4. 26



Q2-1: Consider a case with 8 binary random variables, how many parameters does a BN with the following graph structure have?

1. 12
2. 14
3. 16
4. 26



So we have 16 parameters in total.

Outline

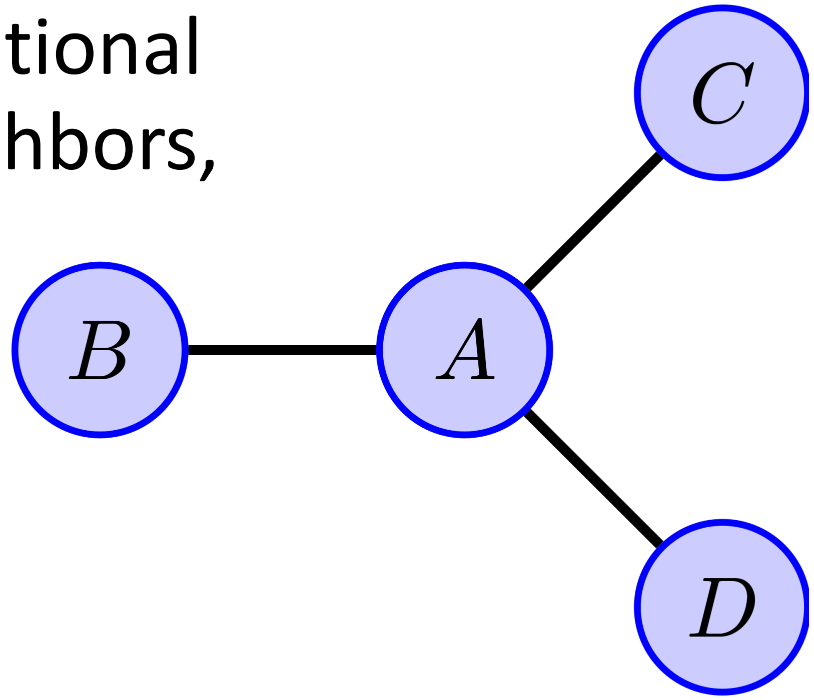
- **Probability Review**
  - Basics, joint probability, conditional probabilities, etc
- **Bayesian Networks**
  - Definition, examples, inference, learning
- **Undirected Graphical Models**
  - Definitions, MRFs, exponential families
- **Structure learning**
  - Chow-Liu Algorithm

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,

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



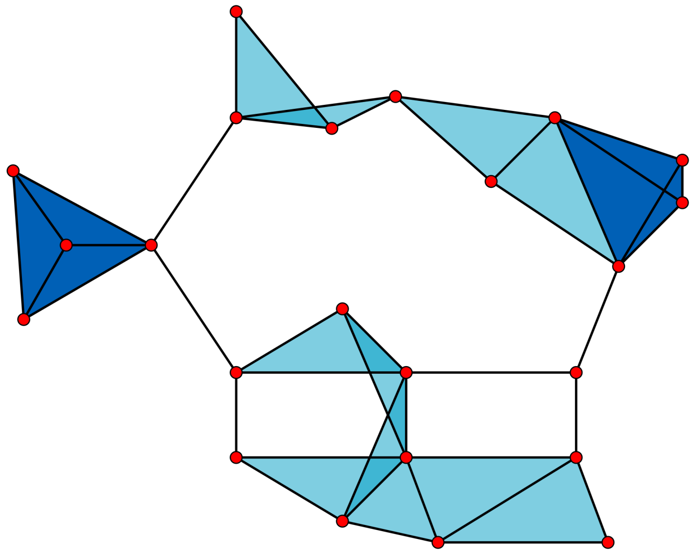
Markov Random Fields

- A particularly popular kind of undirected model. As above, can describe in terms of:
  - 1. Conditional independence: $X_i \perp X_j | X_{V \setminus \{i,j\}}$
  - 2. Factorization. (Clique: maximal fully-connected subgraphs)
    - Bayes nets: factorize over CPTs with **parents**; MRFs: factorize over **cliques**

$$P(X) = \frac{1}{Z} \prod_{C \in \text{cliques}(G)} \phi_C(X_C)$$

Partition function

Potential functions



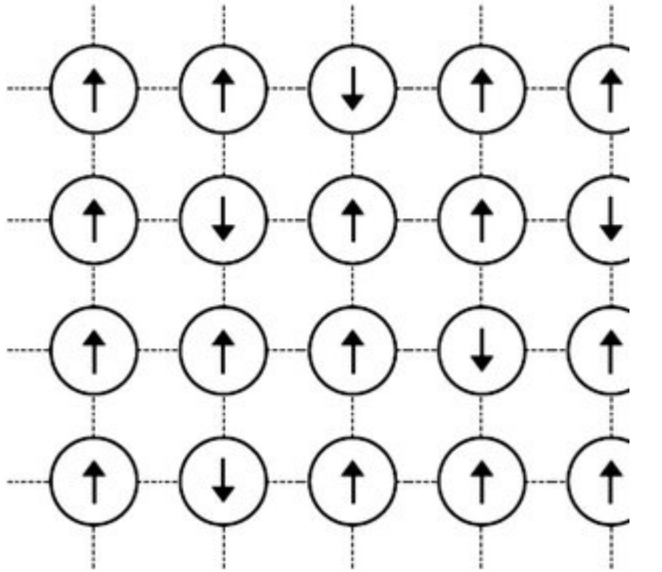
Ising Models

- Ising models: a particular kind of MRF usually written in exponential form
  - Popular in statistical physics
  - **Idea**: pairwise interactions (biggest cliques of size 2)

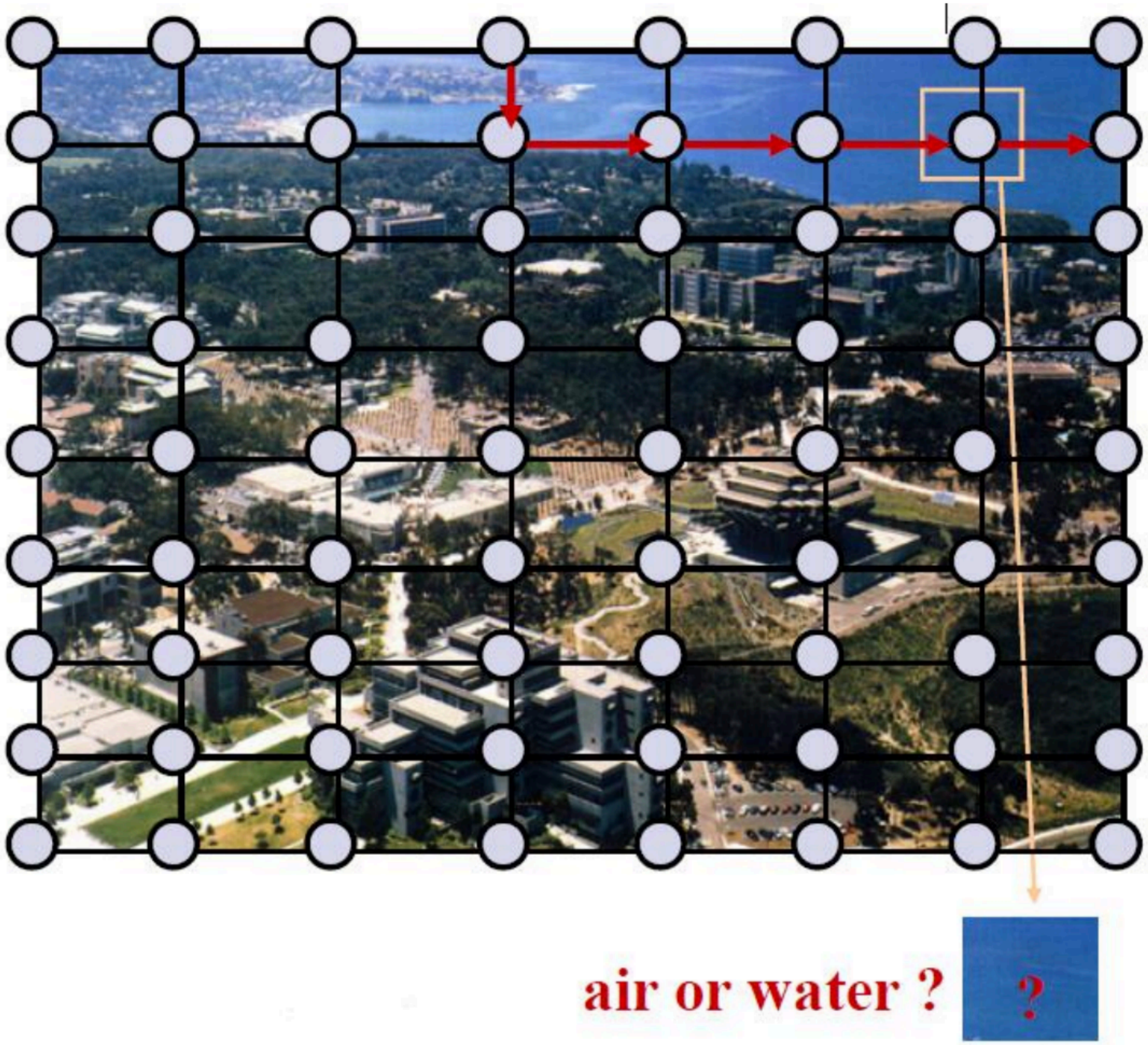
$$P(x_1, \ldots, x_d) = \frac{1}{Z} \exp\Big(\sum_{(i,j) \in E} \theta_{ij} x_i x_j \Big)$$

- Challenges:
  - Compute partition function
  - Perform inference/marginalization

Khudier and Fawaz



Ising Model Example



https://www.cs.cmu.edu/~epxing/Class/10708-17/notes-17/10708-scribe-lecture3.pdf

Outline

- **Probability Review**
  - Basics, joint probability, conditional probabilities, etc
- **Bayesian Networks**
  - Definition, examples, inference, learning
- **Undirected Graphical Models**
  - Definitions, MRFs, exponential families
- **Structure learning**
  - Chow-Liu Algorithm

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 information theory from decision trees!

Chow-Liu: Finding MST

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

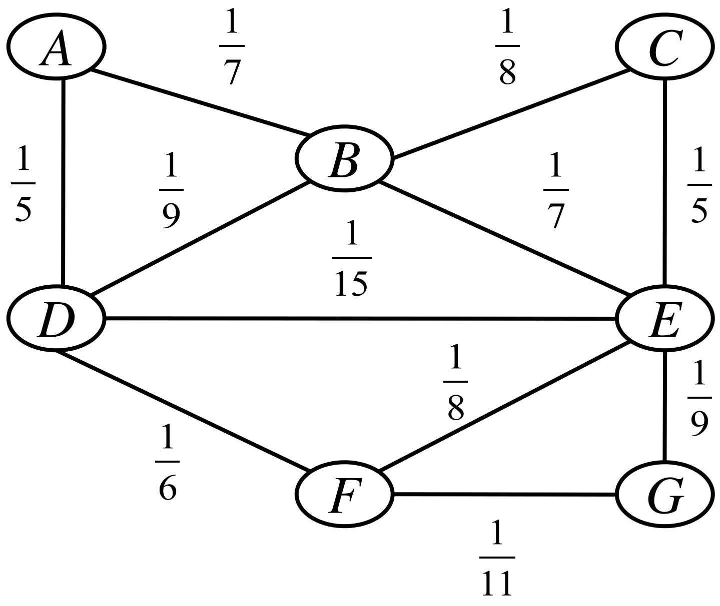
- Kruskal's algorithm

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

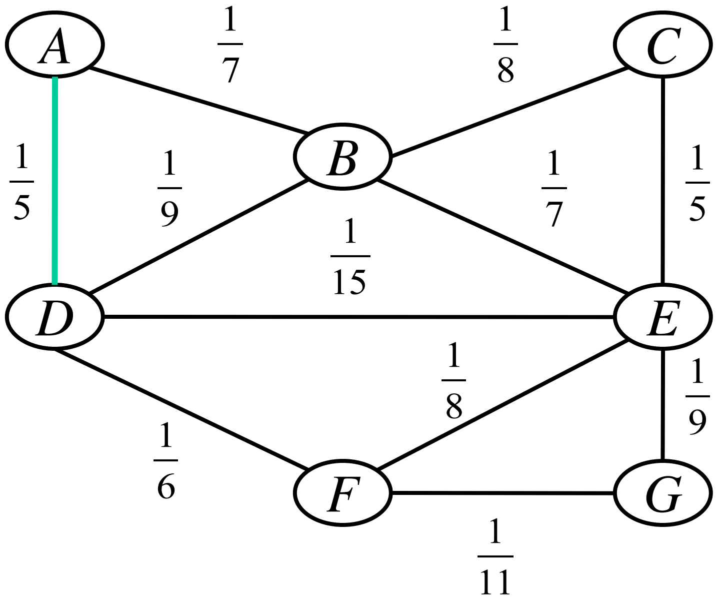
Chow-Liu: Example

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

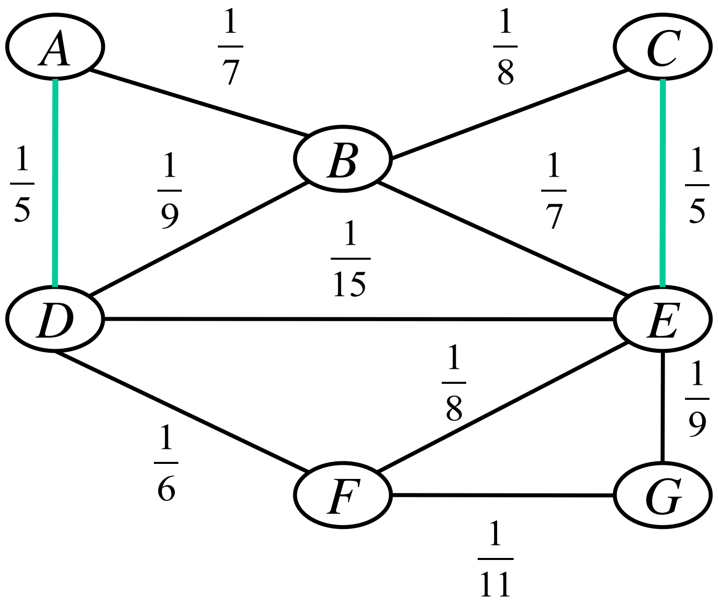


Chow-Liu: Example (cont’d)

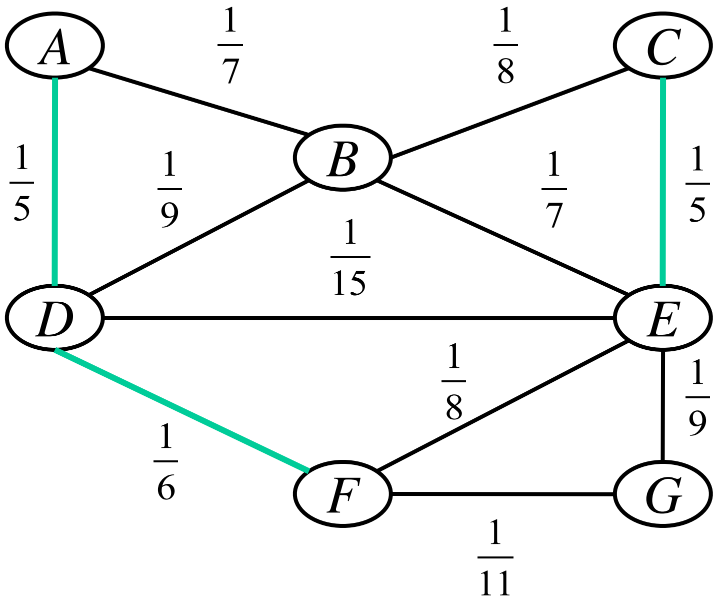
i.



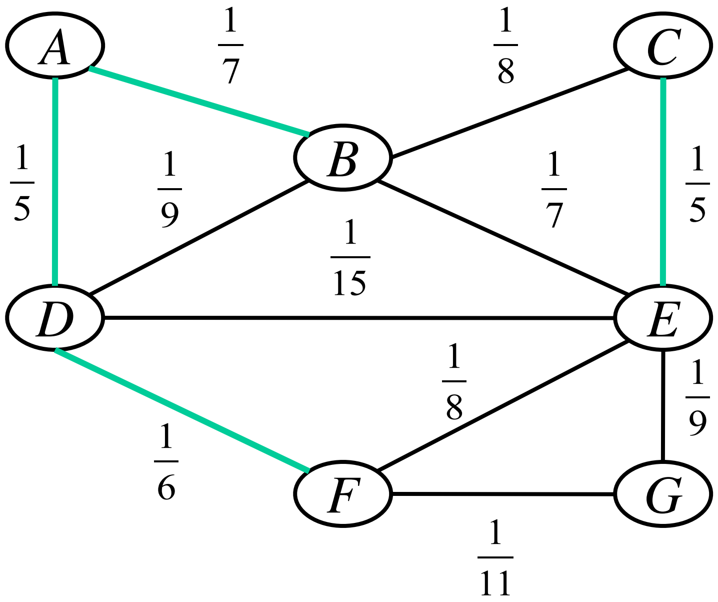
ii.



iii.

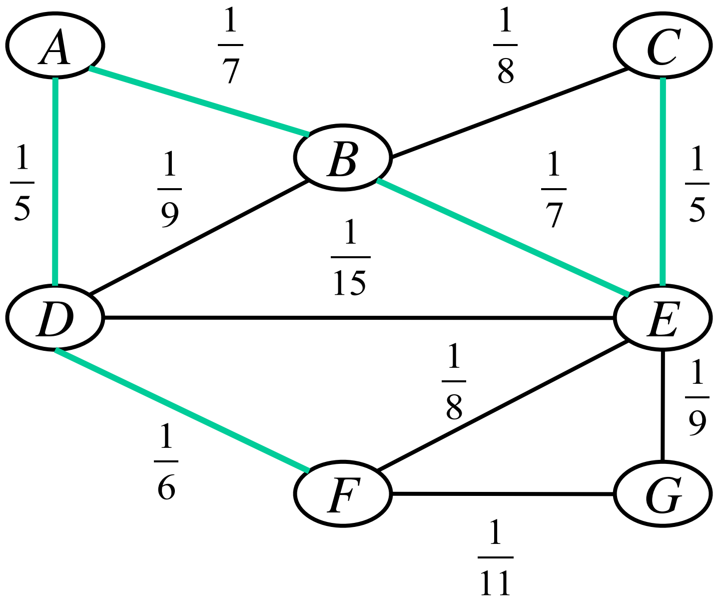


iv.

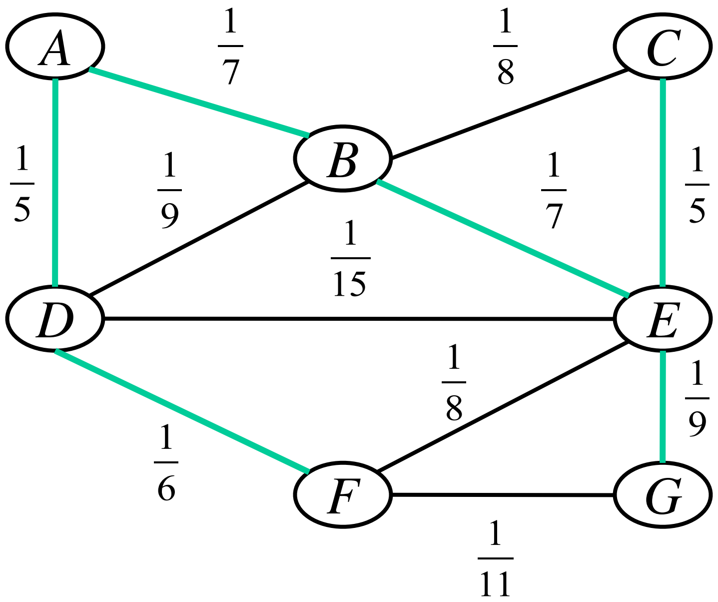


Chow-Liu: Example (cont'd)

v.



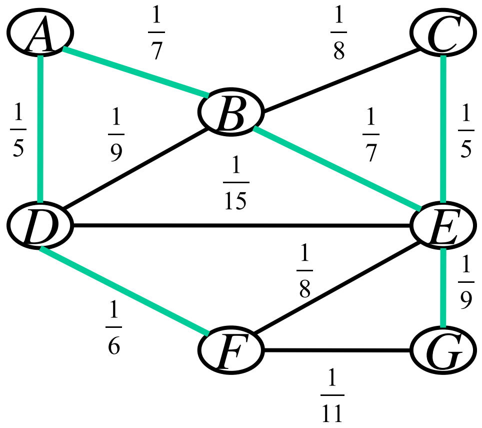
vi.

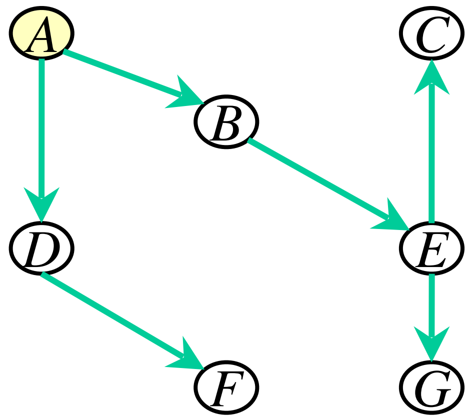


Chow-Liu Algorithm

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)





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