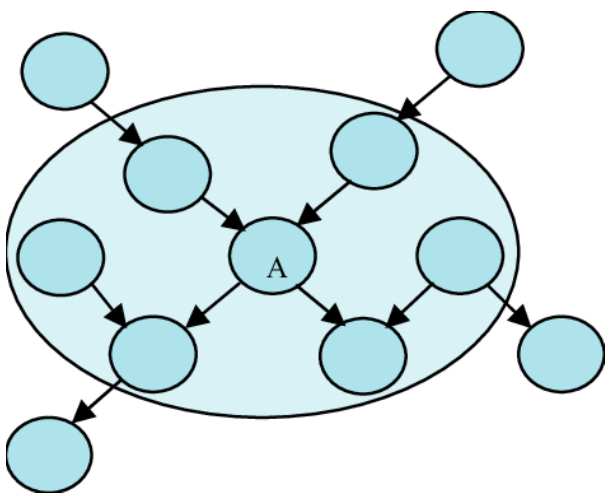
CSE 3521:Bayesian Networks
(DAG Probabilistic Graphical Models)

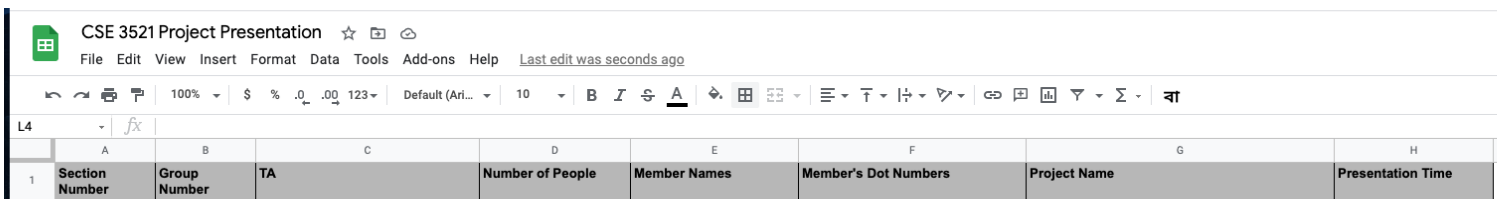


[Many slides are adapted from previous CSE 5521 course at OSU.]

THE OHIO STATE UNIVERSITY

Project Submission

- Presentation
  - Same section group: During class time
  - Multi section group: You can choose which time to present
    - (and update it in the project sheet)



- Report
  - Tentatively 4 pages
  - 1-2 pages the problem description and dataset description
  - 1-2 pages the algorithm description and performance report
  - ½ page comparison among the algorithms and conclude

Today

- Probabilistic graphical models (PGMs)
  - An efficient way to encode conditional independence
  - From PGMs, we can decompose a joint probability much efficiently

- Probabilistic Inference on PGMs

- Independence in PGMs

Problems: dependent feature variables

- Most real-world data have high-dimensional and correlated variables
- Examples:
  - Pixels in an image
  - Words in a document
  - Genes in a microarray

- Sometimes, even data instances are not independent
- Examples:
  - Today’s stock market and yesterday’s stock market

Questions: how to build probability models?

- How to compactly represent $p(X = \boldsymbol{x}|\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ are the parameters?
- How can we use this distribution to infer a set of variables given another?
    - Ex. Given the first 100 pixels, infer the rest
    - Ex. Given today's stock market, predict tomorrow's
- How can we learn the parameters with a reasonable amount of data?

The Chain Rule of Probability

$$p(X[1] = x[1], \dots, X[D] = x[D]) = P(x[1{:}D])$$

$$= p(x[1])p(x[2] | \, x[1]) \dots p(x[D] \mid x[1{:}D-1]) = \prod_d p(x[d] \mid x[1{:}d-1])$$

$$p(X_1 = x_1, \dots, X_N = x_N) = p(x_{1:N}) = p(x_1)p(x_2|x_1) \dots p(x_N|x_{1:N-1}) = \prod_i p(x_i|x_{1:i-1})$$

We will use $x[i]$ or x_i interchangeably sometimes in this lecture!

- Can represent any joint distribution this way
- Using <u>any ordering</u> of the variables…

Problem: this distribution has $O(2^N)$ parameters if each of them is a binary random variable

Conditional Independence

- This is the key to representing large joint distributions
- X and Y are conditionally independent given Z
  - if and only if the conditional joint can be written as a product of the conditional marginals

$$X \perp Y | Z \iff P(X, Y|Z) = P(X|Z)P(Y|Z)$$

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

Markov Models

- "The future is independent of the past given the present"

$$x_{t+1} \perp x_{1:t-1} | x_t$$

$$P(x_1, x_2, x_3, \ldots, x_n)$$

$$= P(x_1)P(x_2|x_1)P(x_3|x_1, x_2) \ldots P(x_n|x_1, x_2, x_3, \ldots, x_{n-1})$$

$$= P(x_1)P(x_2|x_1)P(x_3|x_2) \ldots P(x_n|x_{n-1})$$

- Only O(N) parameters:
  - Fewer parameters, (1) faster learning, (2) fast inference, (3) and fewer training data required!

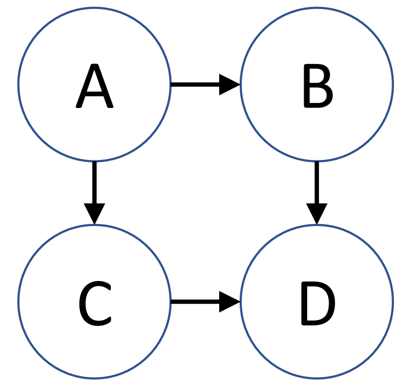
Probabilistic Graphical Models

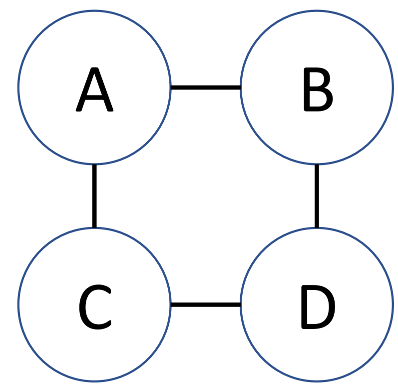
- First order Markov assumption is useful for 1-D sequence data
  - Sequences of words in a sentence or document

- Q: What about 2-D images, 3-D video
  - Or in general arbitrary collections of variables
    - Gene pathways, etc…

Probabilistic Graphical Models

- A way to represent a joint distribution by making conditional independence assumptions
- Nodes represent variables
- Edges: can be directed or undirected
  - directed acyclic graph (DAG): Bayesian networks
- No edges indicate conditional independence assumptions
  - Ex: (top) C and B are conditionally independent given A

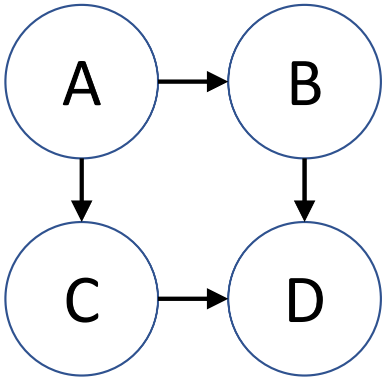




Directed Graphical Models

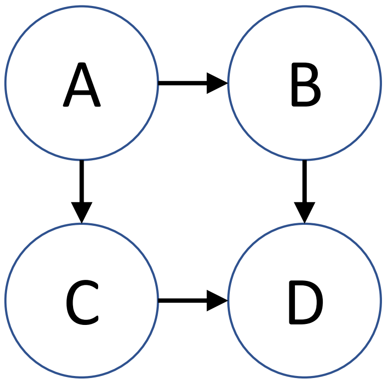
- Graphical Model whose graph is a DAG
  - DAG: Directed acyclic graph (no cycles!)

- A.K.A. Bayesian Networks
  - Also known as Bayes network, belief network
  - Nothing inherently Bayesian about them
    - Just a way of defining conditional independence



Directed Graphical Models

- Key properties: Nodes can be ordered so that parents come before children
  - Topological ordering
  - Can be constructed from any DAG



- Ordered Markov Property:
  - Generalization of first-order Markov Property to general DAGs
  - Node only depends on its parents (not other ancestors)

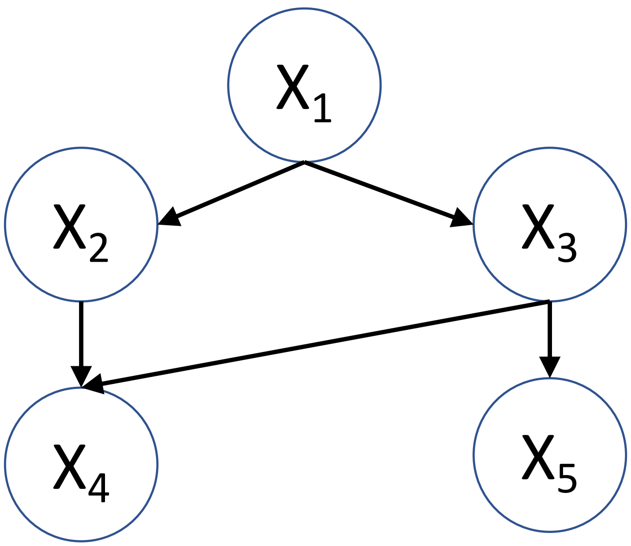
$$X_s \perp X_{\text{any ancestor}(s)-\text{parents}(s)} \mid X_{\text{parents}(s)}$$

    - EX. D ⊥ A | B, C

- Decomposition (nodes are ordered): $$p(x_{1:N}) = \prod_i p(x_i \mid \text{parents}(i))$$

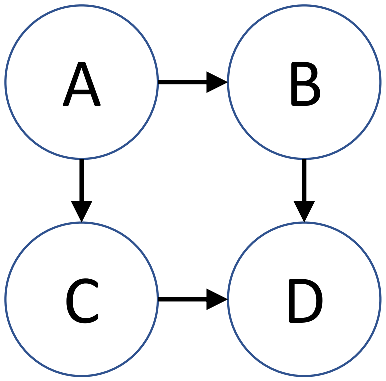
Example

$$P(x_{1:5}) = P(x_1)P(x_2|x_1)P(x_3|x_1,\mathbf{x_2})P(x_4|\mathbf{x_1},x_2,x_3)p(x_5|\mathbf{x_1},\mathbf{x_2},x_3,\mathbf{x_4})$$
$$= P(x_1)P(x_2|x_1)P(x_3|x_1)P(x_4|x_2,x_3)p(x_5|x_3)$$



- Given the decomposition, you can draw the DAG
- Given the DAG, you can derive the decomposition

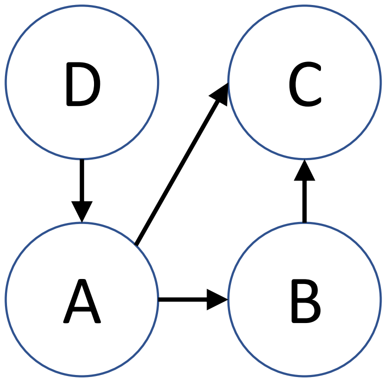
Practice



$$P(A, B, C, D) = P(A)P(B|A)P(C|A)P(D|B, C)$$

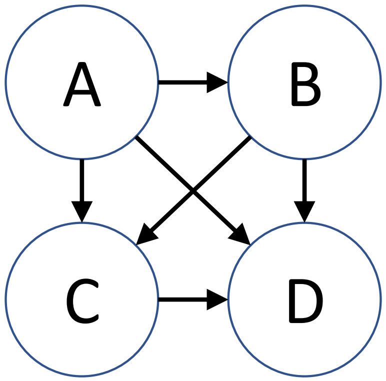
Practice

$$P(A, B, C, D) = P(A|D)P(B|A)P(C|B, A)P(D)$$

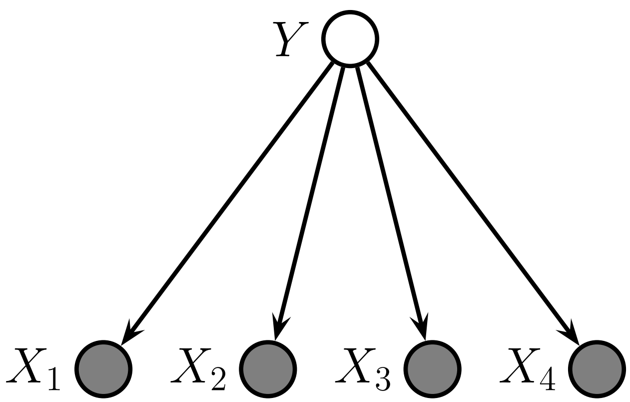


Practice

$$P(A, B, C, D)$$
$$= P(A)P(B|A)P(C|A, B)P(D|A, B, C)$$



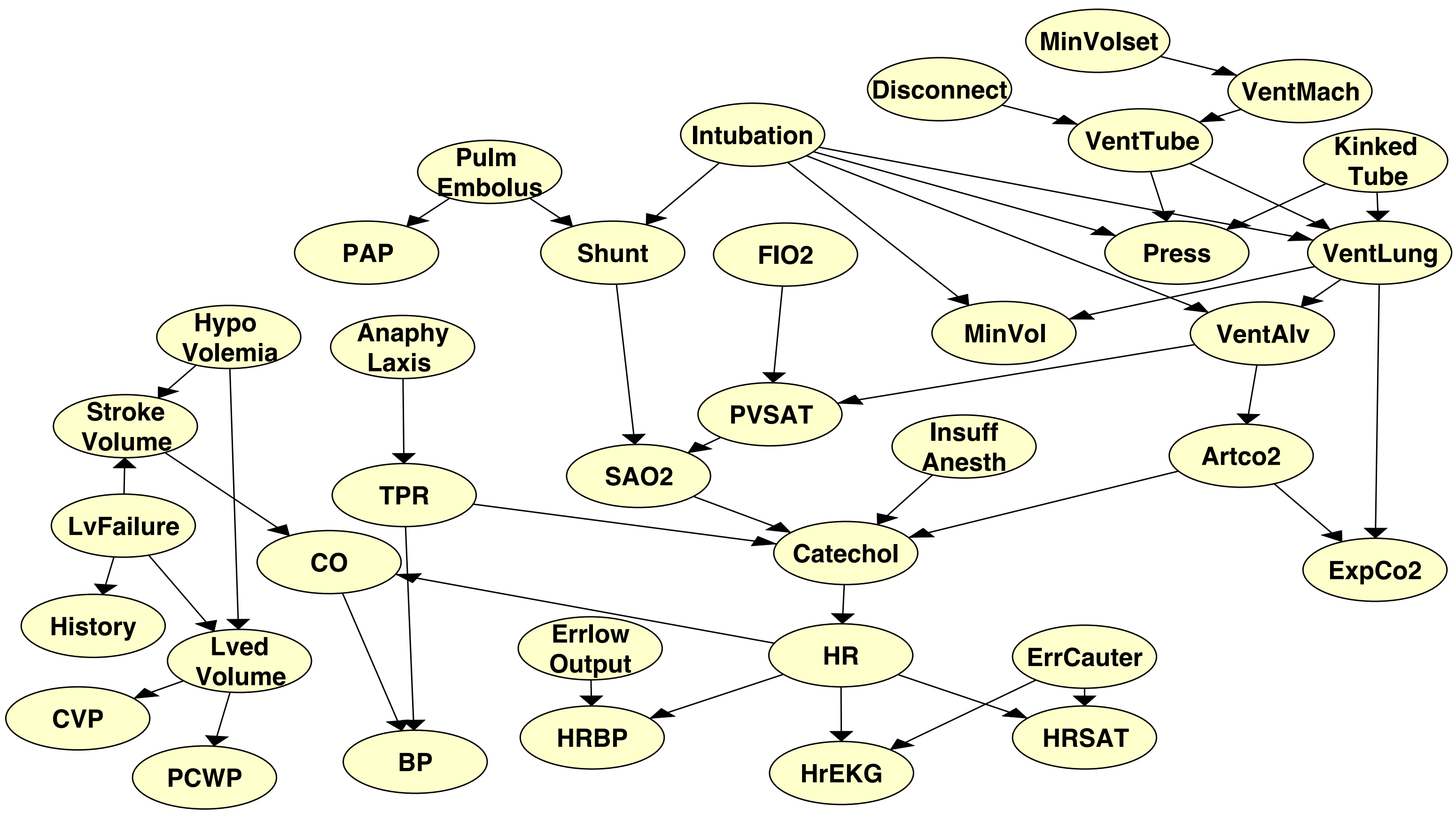
Naïve Bayes



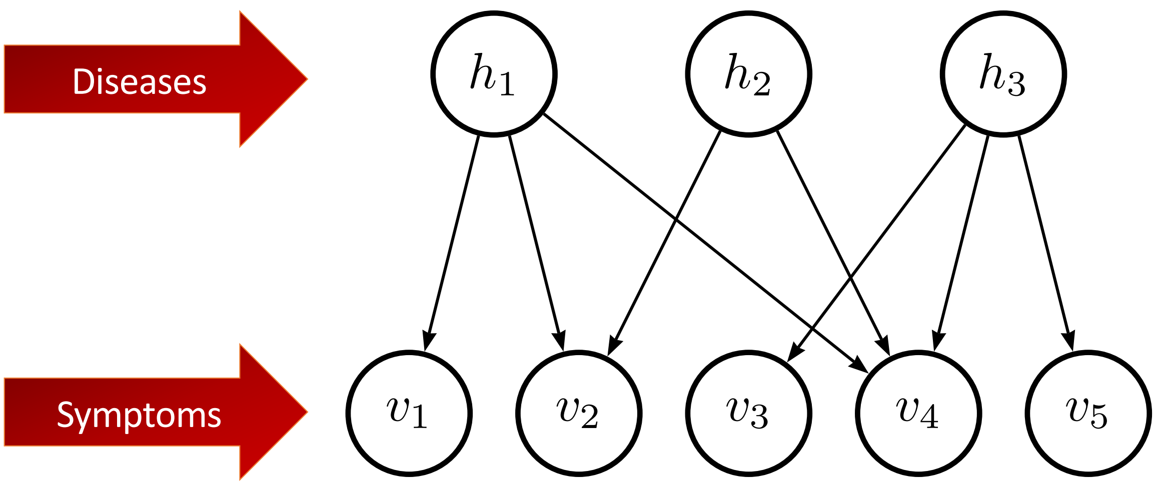
$$P(y, x_{1:D}) = P(y) \prod_{j=1}^{D} P(x_j | y)$$

- White nodes: unknown
- Gray nodes: observed

Example: medical Diagnosis (The Alarm Network)



Another medical diagnosis example: QMR network



Today

- Probabilistic graphical models (PGMs)
  - An efficient way to encode conditional independence
  - From PGMs, we can decompose a joint probability much efficiently

- Probabilistic Inference on PGMs

- Independence in PGMs

Probabilistic Inference

- Graphical Models provide a compact way to represent complex joint distributions

- **Q:** Given a joint distribution, what can we do with it?
- **A:** Main use = probabilistic inference
  - Estimate unknown variables from known ones
  - Ex. Given P(X, Y), predict the most likely assignment of Y given X=x

General Form of Inference

- We have:
  - A correlated set of random variables
  - Joint distribution: $P(x_{1:V}|\theta)$
    - Assumption: parameters are known
- Partition variables into:
  - Visible (with assignments/observations): x_v
  - Hidden: x_h
- Goal: compute unknowns from known ones

$$P(x_h|x_v,\theta) = \frac{P(x_h,x_v|\theta)}{P(x_v|\theta)} = \frac{P(x_h,x_v|\theta)}{\sum_{x'_h} P(x'_h,x_v|\theta)}$$

General Form of Inference

$$P(x_h|x_v,\theta) = \frac{P(x_h,x_v|\theta)}{P(x_v|\theta)} = \frac{P(x_h,x_v|\theta)}{\sum_{x'_h} P(x'_h,x_v|\theta)}$$

- Condition data by clamping visible variables to observed values
- Normalize by probability of evidence

Nuisance Variables

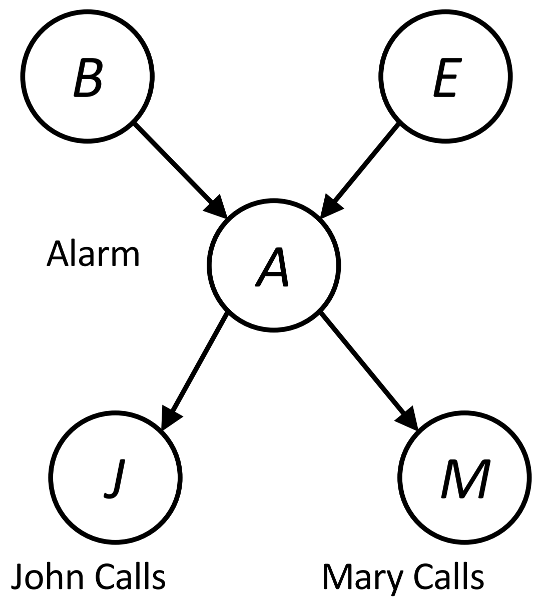
- Partition hidden variables into:
  - Query Variables (interested): x_q
  - Nuisance variables (not interested): x_u

$$P(x_q|x_v,\theta) = \sum_{x_u} P(x_q, x_u|x_v)$$

Inference on PGMs

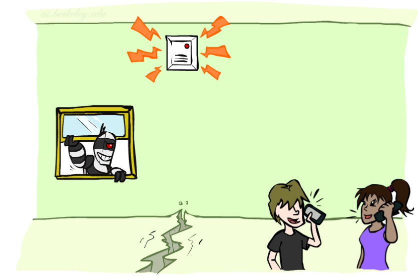
Burglary

B	P(B)
+b	0.001
-b	0.999



E	P(E)
+e	0.002
-e	0.998

Earthquake



A	J	P(J|A)
+a	+j	0.9
+a	-j	0.1
-a	+j	0.05
-a	-j	0.95

A	M	P(M|A)
+a	+m	0.7
+a	-m	0.3
-a	+m	0.01
-a	-m	0.99

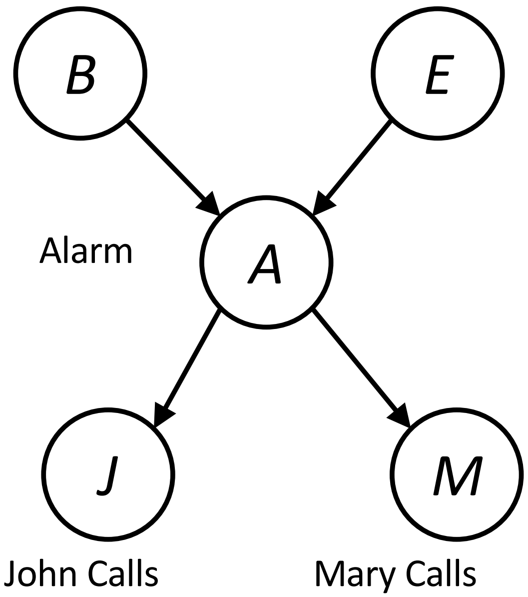
B	E	A	P(A|B,E)
+b	+e	+a	0.95
+b	+e	-a	0.05
+b	-e	+a	0.94
+b	-e	-a	0.06
-b	+e	+a	0.29
-b	+e	-a	0.71
-b	-e	+a	0.001
-b	-e	-a	0.999

$$P(+b, -e, +a, -j, +m) =$$
$$P(+b)P(-e)P(+a| + b, -e)P(-j| + a)P(+m| + a) =$$

Inference on PGMs

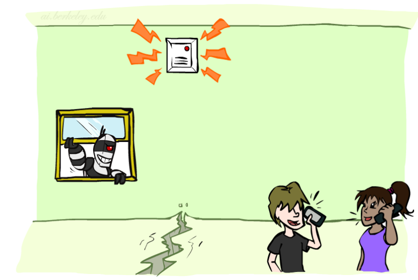
Burglary

B	P(B)
+b	0.001
-b	0.999



E	P(E)
+e	0.002
-e	0.998

Earthquake



A	J	P(J|A)
+a	+j	0.9
+a	-j	0.1
-a	+j	0.05
-a	-j	0.95

A	M	P(M|A)
+a	+m	0.7
+a	-m	0.3
-a	+m	0.01
-a	-m	0.99

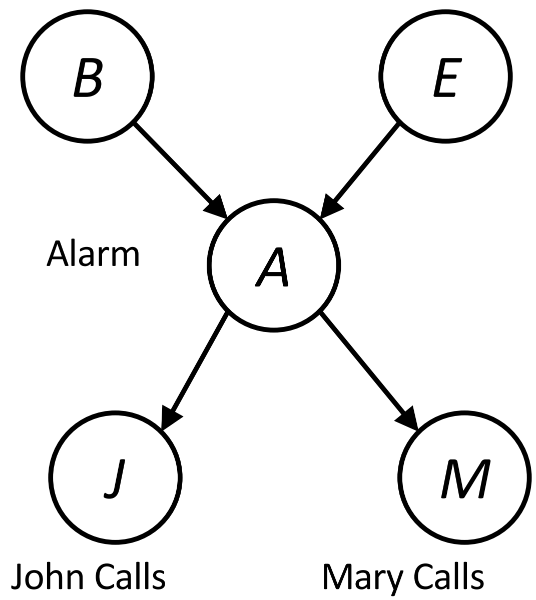
B	E	A	P(A|B,E)
+b	+e	+a	0.95
+b	+e	-a	0.05
+b	-e	+a	0.94
+b	-e	-a	0.06
-b	+e	+a	0.29
-b	+e	-a	0.71
-b	-e	+a	0.001
-b	-e	-a	0.999

$$P(+b, -e, +a, -j, +m) =$$
$$P(+b)P(-e)P(+a|+b, -e)P(-j|+a)P(+m|+a) =$$
$$0.001 \times 0.998 \times 0.94 \times 0.1 \times 0.7$$

Inference on PGMs

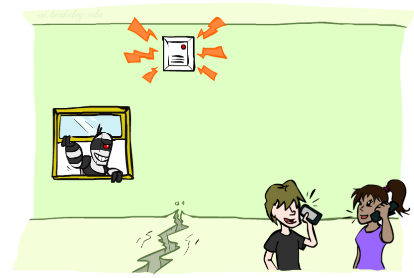
Burglary

B	P(B)
+b	0.001
-b	0.999



E	P(E)
+e	0.002
-e	0.998

Earthquake



A	J	P(J|A)
+a	+j	0.9
+a	-j	0.1
-a	+j	0.05
-a	-j	0.95

A	M	P(M|A)
+a	+m	0.7
+a	-m	0.3
-a	+m	0.01
-a	-m	0.99

B	E	A	P(A|B,E)
+b	+e	+a	0.95
+b	+e	-a	0.05
+b	-e	+a	0.94
+b	-e	-a	0.06
-b	+e	+a	0.29
-b	+e	-a	0.71
-b	-e	+a	0.001
-b	-e	-a	0.999

$$P(B \mid +j, +m) = \frac{P(B, +j, +m)}{P(+j, +m)} = ?$$

** Inference vs. Learning

- Inference:
  - Compute $P(x_h|x_v, \theta)$
  - Parameters are assumed to be known
- Learning (parameter estimation):
  - Compute MLE or MAP estimate of the parameters

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \prod_{i=1}^{N} P(\boldsymbol{x}_{i,v};\ \theta) = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{N} \log P(\boldsymbol{x}_{i,v};\ \theta)$$

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \left\{ \prod_{i=1}^{N} P\left(\boldsymbol{x}_{i,v} \mid \theta\right) \right\} P(\theta) = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{N} \log P(\boldsymbol{x}_{i,v} \mid \theta) + \log P(\theta)$$

Today

- Probabilistic graphical models (PGMs)
  - An efficient way to encode conditional independence
  - From PGMs, we can decompose a joint probability much efficiently

- Probabilistic Inference on PGMs

- Independence in PGMs

Markov Blanket

- Definition:
  - The smallest set of "observed" nodes that renders a node t conditionally independent of all the other nodes in the graph.
- Markov blanket in DAG is:
  - Parents
  - Children
  - Co-parents (other nodes that are also parents of the children)

Markov Blanket

- Each node is conditionally independent of all others given its Markov blanket:
  - parents + children + children's parents