

## Lecture 20: 2 April, 2026

Madhavan Mukund

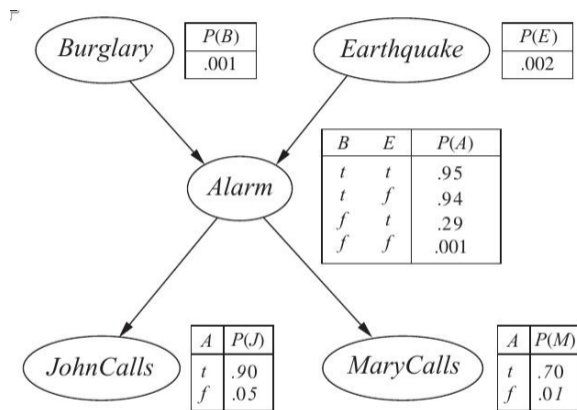
<https://www.cmi.ac.in/~madhavan>

Data Mining and Machine Learning  
January–April 2026

# Conditional probabilities

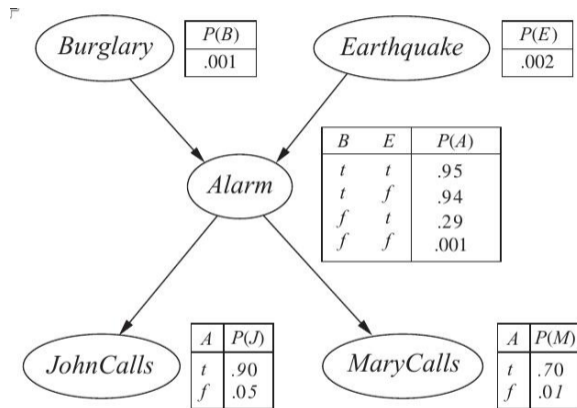
- Boolean variables  $x_1, x_2, \dots, x_n$
- Joint probabilities  $P(v_1, v_2, \dots, v_n)$ 
  - $2^n$  combinations of  $x_1, x_2, \dots, x_n$
  - $2^n - 1$  parameters
- Naïve Bayes assumption — complete independence
  - $P(x_i = 1)$  for each  $x_i$
  - $n$  parameters
- Can we strive for something in between?
  - “Local” dependencies between some variables

- Represent local dependencies using directed graph
- Each node has a local (conditional) probability table
- Example: Burglar alarm
  - Pearl's house has a burglar alarm
  - Neighbours John and Mary call if they hear the alarm
  - John is prone to mistaking ambulances etc for the alarm
  - Mary listens to loud music and sometimes fails to hear the alarm
  - The alarm may also be triggered by an earthquake (California!)



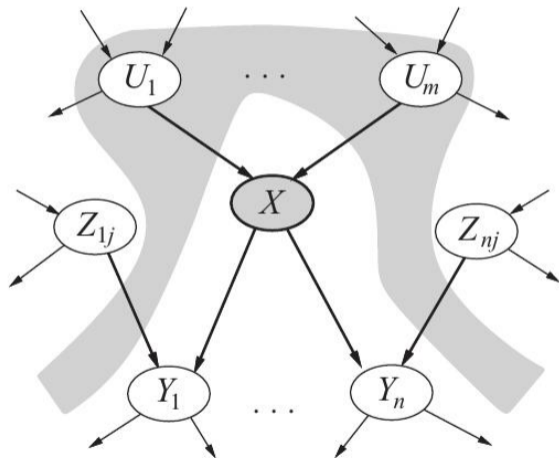
# Probabilistic graphical models

- Graph is a DAG, no cyclic dependencies



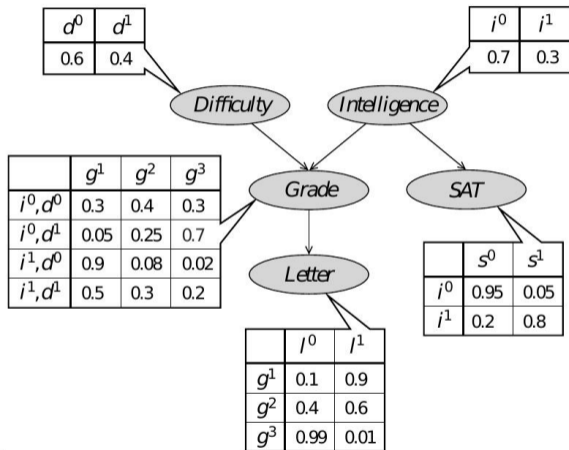
# Probabilistic graphical models

- Graph is a DAG, no cyclic dependencies
- Fundamental assumption:  
A node is conditionally independent of non-descendants, given its parents



# Student example

- Example due to Nir Friedman and Daphne Koller
- Student asks teacher for a reference letter
- Teacher has forgotten the student, so letter is entirely based on student's grade in the course



# Evaluating a network

- John and Mary call Pearl. What is the probability that there has been a burglary?

- $P(b, m, j)$ , where  $b$ : burglary,  $j$ : John calls,  $m$ : Mary calls

- $P(b, m, j) = \sum_{a=0}^1 \sum_{e=0}^1 P(b, j, m, a, e)$ , where  $a$ : alarm rings,  $e$ : earthquake

- Bayes Rule:  $P(A, B) = P(A | B)P(B)$

- $P(x_1, x_2, \dots, x_n) = P(x_1 | x_2, x_3, \dots, x_n)P(x_2, x_3, \dots, x_n)$

- Applied recursively, this gives us the **chain rule**

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

# Evaluating a network

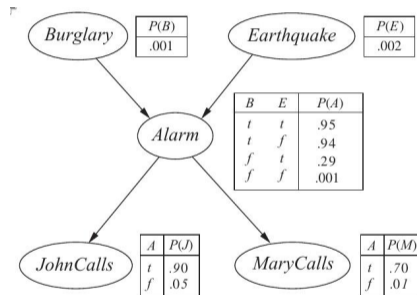
- $P(x_1, x_2, \dots, x_n) = P(x_1 | x_2, \dots, x_n)P(x_2 | x_3, \dots, x_n) \cdots P(x_{n-1} | x_n)P(x_n)$

- Can choose any ordering of  $x_1, x_2, \dots, x_n$

- Use topological ordering in a Bayesian network

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

- $P(m, j, b) =$   
 $\sum_{a=0}^1 \sum_{e=0}^1 P(m | a)P(j | a)P(a | b, e)P(b)P(e)$





# Evaluating a network

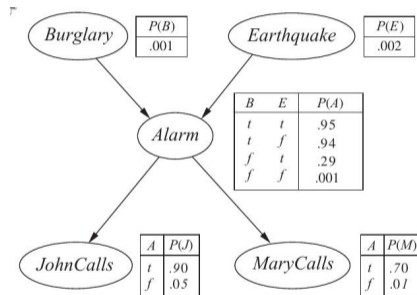
- $P(x_1, x_2, \dots, x_n) = P(x_1 | x_2, \dots, x_n)P(x_2 | x_3, \dots, x_n) \cdots P(x_{n-1} | x_n)P(x_n)$

- Can choose any ordering of  $x_1, x_2, \dots, x_n$

- Use topological ordering in a Bayesian network

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

- $P(m, j, b) =$   
 $\sum_{e=0}^1 \sum_{a=0}^1 P(m | a)P(j | a)P(a | b, e)P(b)P(e)$



# Evaluating a network

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

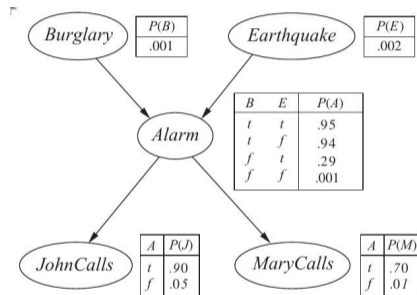
- Can choose any ordering of  $x_1, x_2, \dots, x_n$

- Use topological ordering in a Bayesian network

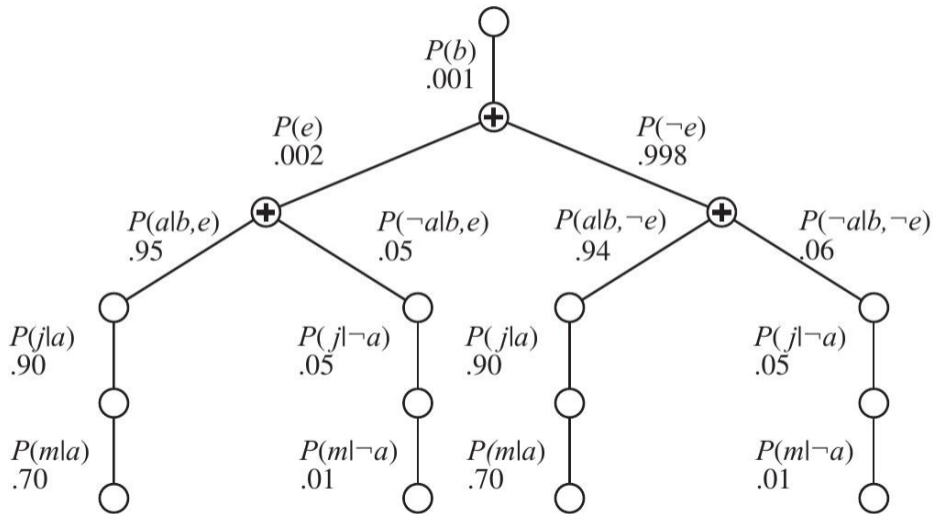
- $P(m, j, a, b, e) =$   
 $P(m | j, a, b, e)P(j | a, b, e)P(a | b, e)P(b | e)P(e)$   
 $= P(m | a)P(j | a)P(a | b, e)P(b)P(e)$

- $P(m, j, b) =$   
$$\sum_{e=0}^1 \sum_{a=0}^1 P(m | a)P(j | a)P(a | b, e)P(b)P(e)$$

- $P(m, j, b) = P(b) \sum_{e=0}^1 P(e) \sum_{a=0}^1 P(m | a)P(j | a)P(a | b, e)$

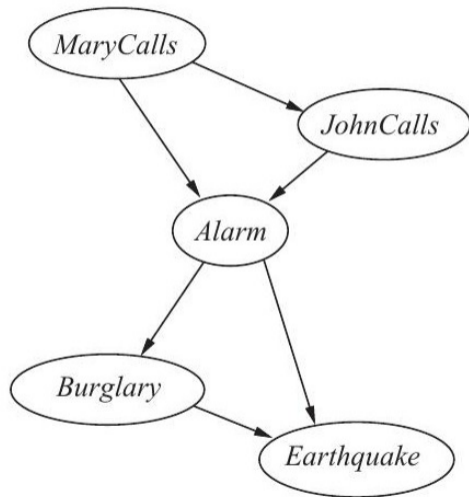


# Evaluation tree



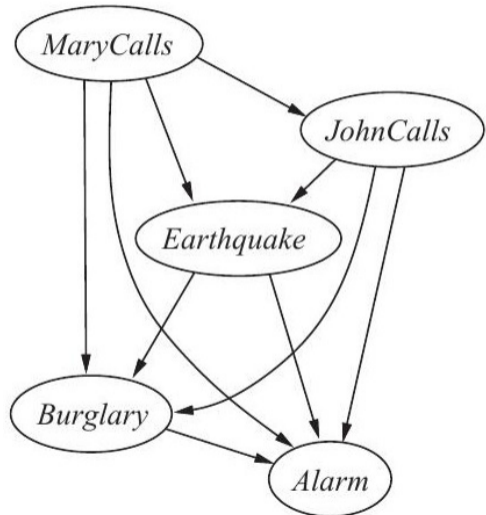
# Designing the Bayesian network

- Need to choose node ordering wisely to get a compact Bayesian network
- Ordering *MaryCalls*, *JohnCalls*, *Alarm*, *Burglary*, *Earthquake* produces this network



# Designing the Bayesian network

- Need to choose node ordering wisely to get a compact Bayesian network
- Ordering *MaryCalls*, *JohnCalls*, *Alarm*, *Burglary*, *Earthquake* produces this network
- Ordering *MaryCalls*, *JohnCalls*, *Earthquake*, *Burglary*, *Alarm* is even worse
- **Causal model** (causes to effects) works better than **diagnostic model** (effects to causes)

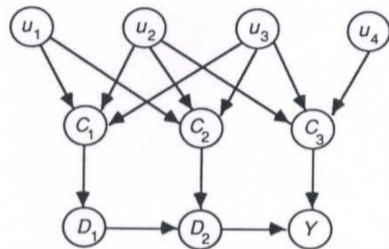


# Complexity of exact inference

- Exact inference of Bayesian networks is NP-complete
- Boolean formula in **Conjunctive Normal Form (CNF)**
  - Boolean variables  $\{v_1, v_2, \dots, v_n\}$
  - A **literal**  $u_i$  is either  $v_i$  or  $\neg v_i$
  - A **clause** is a disjunction of literals  $u_{j_1} \vee u_{j_2} \vee \dots \vee u_{j_k}$
  - A CNF formula is a conjunction of clauses  $C_1 \wedge C_2 \wedge \dots \wedge C_m$
- **SAT** — given a formula in CNF, is there an assignment to variables that makes the formula true?
- **3-SAT** — SAT where each clause has exactly 3 literals
- Both SAT and 3-SAT are **NP-complete**
  - No known efficient algorithm — try all possible valuations

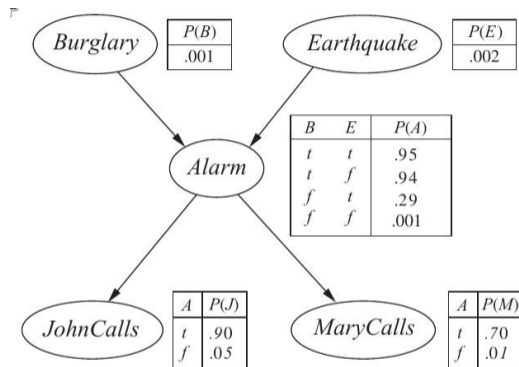
# Reducing 3-SAT to exact inference

- Convert a 3-CNF formula into a Bayesian network
- Top layer: one node for each literal  $u_i$
- Middle layer: one node for each clause  $C_j$ 
  - Parents are three variables whose literals are in  $C_j$
  - Conditional probability table for  $C_j$  has 8 rows, for all possible valuations of 3 variables
  - $P(C_j = 1) = 0$  for row where each input literal is false,  $P(C_j = 1) = 1$  for remaining 7 rows
- Bottom row builds up  $C_1 \wedge \dots \wedge C_m$  one clause at a time
- $P(Y = 1) > 0$  iff original 3-CNF formula is satisfiable



# Conditional independence

- $x \perp y$  —  $x$  and  $y$  are independent
  - $P(x \wedge y) = P(x) \cdot P(y)$
- $x \perp y \mid z$ 
  - $x$  and  $y$  are independent given  $z$
  - $P(x \wedge y \mid z) = P(x \mid z) \cdot P(y \mid z)$
- Is *JohnCalls* independent of *MaryCalls* ( $j \perp m$ )?
  - No — value of  $j$  tells us something about value of  $m$  and vice versa
- Is *JohnCalls* independent of *MaryCalls* given *Alarm* ( $j \perp m \mid a$ )?
  - Yes — by semantics of network, local independence

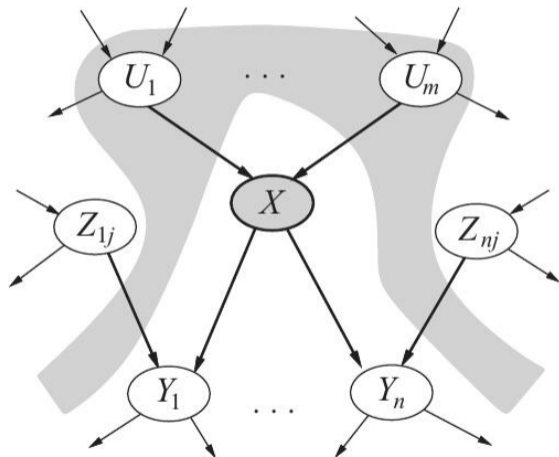




# Probabilistic graphical models

## ■ Fundamental assumption

A node is conditionally independent of non-descendants, given its parents



# Student example

- $SAT \perp Grade \mid Difficulty$  ?

- No

- Can we calculate conditional independence from the graph?

- In general, check if  $X \perp Y \mid Z$  for sets of variables  $X, Y, Z$

