CSE 150A-250A AI: Probabilistic Methods

Lecture 4

Fall 2025

Trevor Bonjour
Department of Computer Science and Engineering
University of California, San Diego

Slides adapted from previous versions of the course (Prof. Lawrence, Prof. Alvarado, Prof. Berg-Kirkpatrick)

Agenda

Review

Conditional probability tables

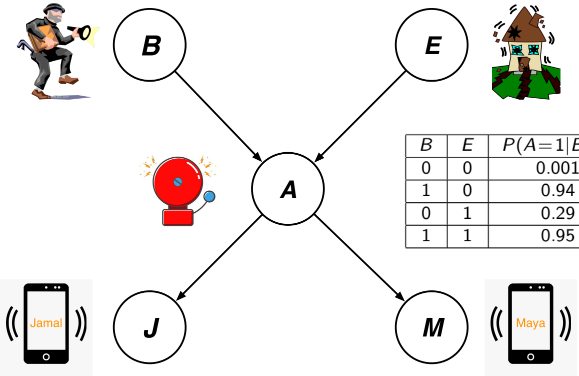
d-separation and examples

Review

Alarm example

$P(B=1) = 0.001$

$P(E=1) = 0.002$



B	E	$P(A=1\|B,E)$
0	0	0.001
1	0	0.94
0	1	0.29
1	1	0.95

A	$P(J=1\|A)$
0	0.05
1	0.9

A	$P(M=1\|A)$
0	0.01
1	0.7

A **belief network** (BN) is a directed acyclic graph (DAG) in which:

Belief networks

A **belief network** (BN) is a directed acyclic graph (DAG) in which:

1. Nodes represent random variables.

Belief networks

A **belief network** (BN) is a directed acyclic graph (DAG) in which:

1. Nodes represent random variables.
2. Edges represent (direct) dependencies.

A **belief network** (BN) is a directed acyclic graph (DAG) in which:

1. Nodes represent random variables.
2. Edges represent (direct) dependencies.
3. Conditional probability tables (CPTs) describe how each node depends on its parents.

BN = DAG + CPTs

Marginal and conditional independence in DAGs

- Missing edges encode assumptions of independence:

Marginal and conditional independence in DAGs

- Missing edges encode assumptions of independence:

$$P(X_i|X_1, \ldots, X_{i-1}) \;=\; P(X_i|\mathrm{pa}(X_i))$$

where $\mathrm{pa}(X_i) \subseteq \{X_1, \ldots, X_{i-1}\}$ denotes the **parents** of node X_i.

Marginal and conditional independence in DAGs

- Missing edges encode assumptions of independence:

$$P(X_i|X_1,\dots,X_{i-1}) \;=\; P(X_i|\mathrm{pa}(X_i))$$

where $\mathrm{pa}(X_i) \subseteq \{X_1,\dots,X_{i-1}\}$ denotes the **parents** of node X_i.

In words: Each variable is conditionally independent of its non-descendants given it's parents.

Marginal and conditional independence in DAGs

- **Missing edges encode assumptions of independence:**

$$P(X_i|X_1, \ldots, X_{i-1}) \; = \; P(X_i|\mathrm{pa}(X_i))$$

where $\mathrm{pa}(X_i) \subseteq \{X_1, \ldots, X_{i-1}\}$ denotes the **parents** of node X_i.

In words: Each variable is conditionally independent of its non-descendants given it's parents.

- **Alarm example:**

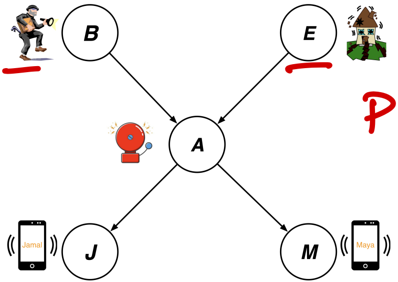
- Missing edges encode assumptions of independence:

$$P(X_i|X_1,\ldots,X_{i-1}) \;=\; P(X_i|\mathrm{pa}(X_i))$$

where $\mathrm{pa}(X_i) \subseteq \{X_1,\ldots,X_{i-1}\}$ denotes the **parents** of node X_i.

In words: Each variable is conditionally independent of its non-descendants given it's parents.

- Alarm example:



$$P(B,E,A,J,M) = P(B)\cdot P(E) \, P(A|B,E)\cdot P(J|A) \, P(M|A)$$

Marginal and conditional independence in DAGs

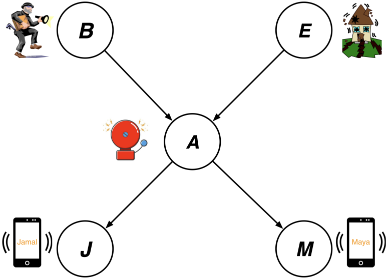
- Missing edges encode assumptions of independence:

$$P(X_i|X_1,\dots,X_{i-1}) \;=\; P(X_i|\mathrm{pa}(X_i))$$

where $\mathrm{pa}(X_i) \subseteq \{X_1,\dots,X_{i-1}\}$ denotes the **parents** of node X_i.

In words: Each variable is conditionally independent of its non-descendants given it's parents.

- Alarm example:



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

Marginal and conditional independence in DAGs

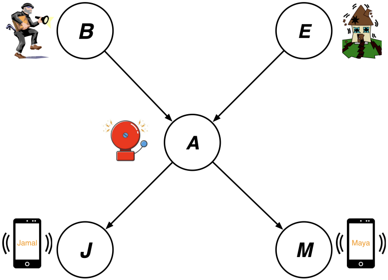
- **Missing edges encode assumptions of independence:**

$$P(X_i | X_1, \ldots, X_{i-1}) \;=\; P(X_i | \mathrm{pa}(X_i))$$

where $\mathrm{pa}(X_i) \subseteq \{X_1, \ldots, X_{i-1}\}$ denotes the **parents** of node X_i.

In words: Each variable is conditionally independent of its non-descendants given it's parents.

- **Alarm example:**

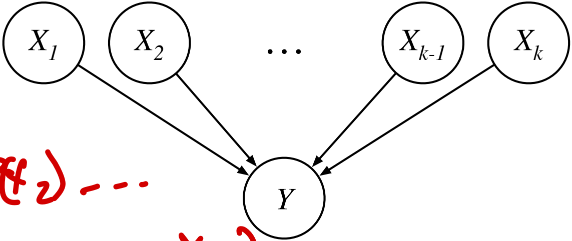


$$
\begin{aligned}
P(E) &= P(E|B) \\
P(J|A) &= P(J|A,B,E) \\
P(M|A) &= P(M|A,B,E,J)
\end{aligned}
$$

These are true no matter what CPTs are attached to the nodes in the DAG.

Conditional probability tables

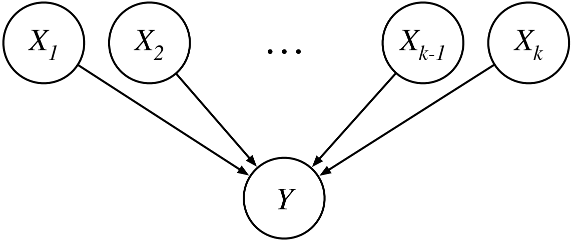
Representing CPTs



$P(x_1)P(x_2)\ldots$

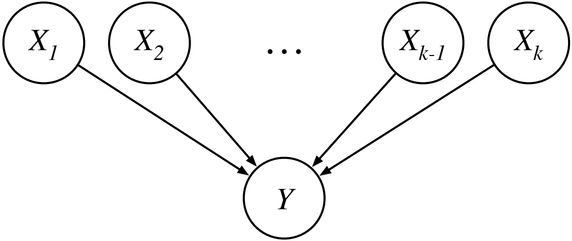
$P(y \mid x_1 \ldots x_k)$

Representing CPTs



- How to represent $P(Y|X_1, X_2, \ldots, X_k)$?

Representing CPTs



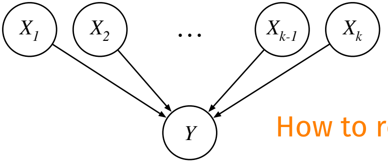
- **How to represent $P(Y|X_1, X_2, \ldots, X_k)$?**

- **Simplest case:**

  Suppose $X_i \in \{0, 1\}$, $Y \in \{0, 1\}$ are **binary** random variables.

  How to represent $P(Y = 1|X_1, X_2, \ldots, X_k)$?

Types of CPTs



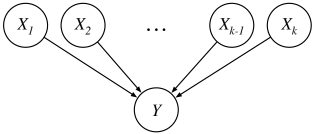
How to represent $P(Y=1|X_1, X_2, \ldots, X_k)$?

2^k

Some possibilities:

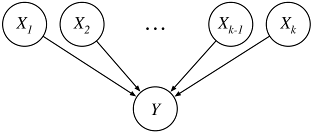
1. Tabular
2. Logical / Deterministic
3. Noisy-OR
4. Sigmoid

1. Tabular CPT



X_1	X_2	$\cdots$	X_k	$P(Y=1 \mid X_1, X_2, \ldots, X_k)$
0	0	$\cdots$	0	0.1
1	0	$\cdots$	0	0.6
0	1	$\cdots$	0	0.3
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
1	1	$\cdots$	1	0.2

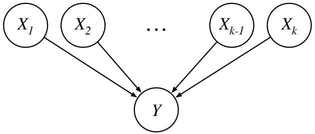
1. Tabular CPT



X_1	X_2	$\cdots$	X_k	$P(Y=1 \mid X_1, X_2, \ldots, X_k)$
0	0	$\cdots$	0	0.1
1	0	$\cdots$	0	0.6
0	1	$\cdots$	0	0.3
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
1	1	$\cdots$	1	0.2

A lookup table can exhaustively enumerate a conditional probability for every configuration of parents.

1. Tabular CPT

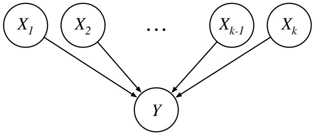


X_1	X_2	$\cdots$	X_k	$P(Y=1 \mid X_1, X_2, \ldots, X_k)$
0	0	$\cdots$	0	0.1
1	0	$\cdots$	0	0.6
0	1	$\cdots$	0	0.3
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
1	1	$\cdots$	1	0.2

A lookup table can exhaustively enumerate a conditional probability for every configuration of parents.

Pro Able to model arbitrarily complicated dependence.

1. Tabular CPT



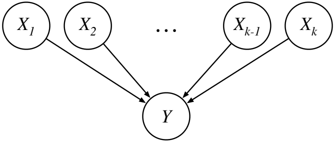
X_1	X_2	$\cdots$	X_k	$P(Y=1 \mid X_1, X_2, \ldots, X_k)$
0	0	$\cdots$	0	0.1
1	0	$\cdots$	0	0.6
0	1	$\cdots$	0	0.3
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
1	1	$\cdots$	1	0.2

A lookup table can exhaustively enumerate a conditional probability for every configuration of parents.

Pro Able to model arbitrarily complicated dependence.

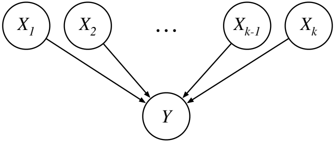
Con A table with 2^k rows is too unwieldy for large k.

2. Logical / Deterministic CPT



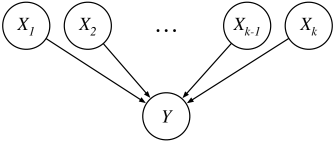
CPTs can also mimic the behavior of logical circuits.

2. Logical / Deterministic CPT



CPTs can also mimic the behavior of logical circuits.

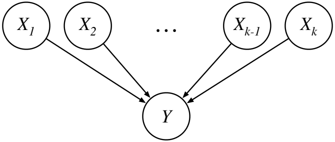
AND gate



CPTs can also mimic the behavior of logical circuits.

AND gate

$$P(Y=1|X_1, X_2, \ldots, X_k) = \prod_{i=1}^{k} X_i$$

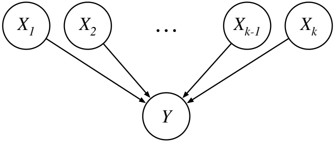


CPTs can also mimic the behavior of logical circuits.

AND gate

$$P(Y=1|X_1, X_2, \ldots, X_k) = \prod_{i=1}^{k} X_i$$

OR gate



CPTs can also mimic the behavior of logical circuits.

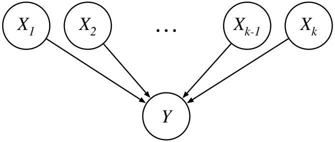
AND gate

$$P(Y=1|X_1, X_2, \ldots, X_k) = \prod_{i=1}^{k} X_i$$

OR gate

$$P(Y=0|X_1, X_2, \ldots, X_k) = \prod_{i=1}^{k} (1 - X_i)$$

2. Logical / Deterministic CPT

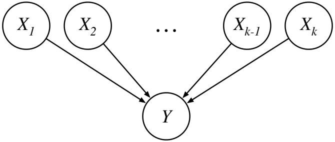


CPTs can also mimic the behavior of logical circuits.

AND gate $$P(Y=1|X_1, X_2, \ldots, X_k) = \prod_{i=1}^{k} X_i$$

OR gate $$P(Y=0|X_1, X_2, \ldots, X_k) = \prod_{i=1}^{k} (1 - X_i)$$

Pro Compact representation for large k.



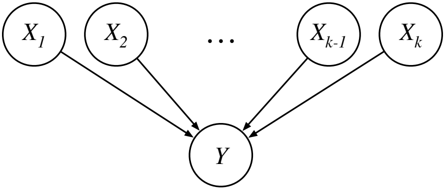
CPTs can also mimic the behavior of logical circuits.

AND gate $$P(Y=1|X_1, X_2, \ldots, X_k) = \prod_{i=1}^{k} X_i$$

OR gate $$P(Y=0|X_1, X_2, \ldots, X_k) = \prod_{i=1}^{k} (1 - X_i)$$

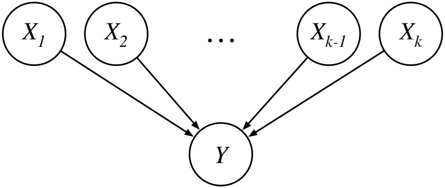
Pro Compact representation for large k.

Con No model of uncertainty.

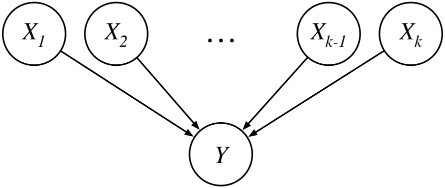


X_1
X_2
…
X_{k-1}
X_k
Y

3. Noisy-OR CPT



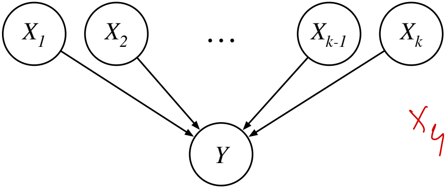
Use k numbers $p_i \in [0, 1]$ to parameterize all 2^k entries in the CPT:



Use k numbers $p_i \in [0,1]$ to parameterize all 2^k entries in the CPT:

$$P(Y=0|X_1, X_2, \ldots, X_k) \quad = \quad \prod_{i=1}^{k} (1 - p_i)^{X_i}$$

$(1 - X_i)$

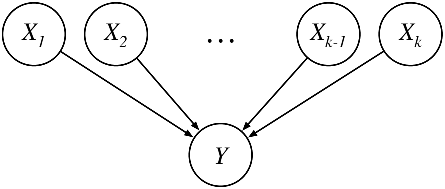


Use k numbers $p_i \in [0,1]$ to parameterize all 2^k entries in the CPT:

$$P(Y=0|X_1, X_2, \ldots, X_k) = \prod_{i=1}^{k} (1-p_i)^{X_i}$$

$$P(Y=1|X_1, X_2, \ldots, X_k) = 1 - \prod_{i=1}^{k} (1-p_i)^{X_i}$$

3. Noisy-OR CPT



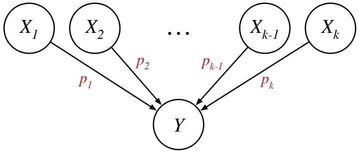
Use k numbers $p_i \in [0,1]$ to parameterize all 2^k entries in the CPT:

$$P(Y=0|X_1, X_2, \ldots, X_k) = \prod_{i=1}^{k} (1-p_i)^{X_i}$$

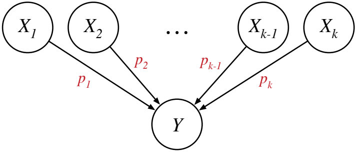
$$P(Y=1|X_1, X_2, \ldots, X_k) = 1 - \prod_{i=1}^{k} (1-p_i)^{X_i}$$

But why is this called Noisy-OR?

Noisy-OR CPT (con't)

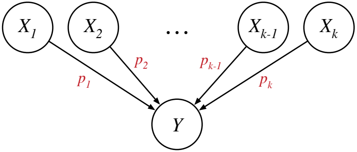


Noisy-OR CPT (con't)



$$P(Y=1|X_1, X_2, \ldots, X_k) \;=\; 1 - \prod_{i=1}^{k}(1-p_i)^{X_i}$$

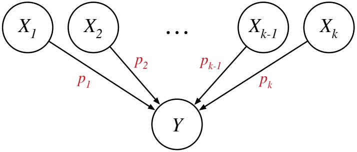
Noisy-OR CPT (con't)



$$P(Y=1|X_1, X_2, \ldots, X_k) \; = \; 1 - \prod_{i=1}^{k}(1 - p_i)^{X_i}$$

- When all parents are equal to zero:

Noisy-OR CPT (con't)

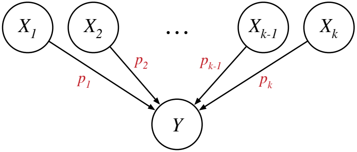


$$P(Y=1|X_1, X_2, \ldots, X_k) \;=\; 1 - \prod_{i=1}^{k}(1-p_i)^{X_i}$$

- When all parents are equal to zero:

  $P(Y=1|X_1=0, X_2=0, \ldots, X_k=0)$

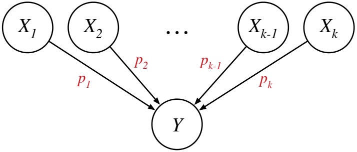
Noisy-OR CPT (con't)



$$P(Y=1|X_1, X_2, \ldots, X_k) \;=\; 1 - \prod_{i=1}^{k}(1-p_i)^{X_i}$$

- When all parents are equal to zero:
  $$P(Y=1|X_1=0, X_2=0, \ldots, X_k=0) \;=\; 1-\prod_{i=1}^{k}(1-p_i)^0$$

Noisy-OR CPT (con't)

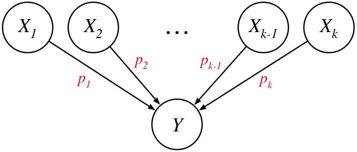


$$P(Y=1|X_1, X_2, \ldots, X_k) \ = \ 1 - \prod_{i=1}^{k}(1-p_i)^{X_i}$$

- When all parents are equal to zero:

$$P(Y=1|X_1=0, X_2=0, \ldots, X_k=0) \ = \ 1-\prod_{i=1}^{k}(1-p_i)^0 \ = \ 1-\prod_{i=1}^{k}(1)$$

Noisy-OR CPT (con't)

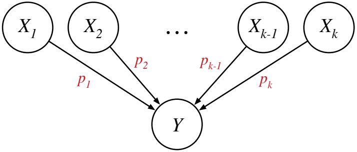


$$P(Y=1|X_1, X_2, \ldots, X_k) \; = \; 1 - \prod_{i=1}^{k}(1-p_i)^{X_i}$$

- When all parents are equal to zero:

$$P(Y=1|X_1=0, X_2=0, \ldots, X_k=0) \; = \; 1 - \prod_{i=1}^{k}(1-p_i)^0 \; = \; 1 - \prod_{i=1}^{k}(1) \; = \; 0$$

Noisy-OR CPT (con't)



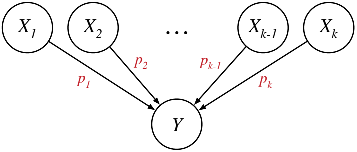
$$P(Y=1|X_1,X_2,\ldots,X_k) \;=\; 1-\prod_{i=1}^{k}(1-p_i)^{X_i}$$

- When all parents are equal to zero:

  $$P(Y=1|X_1=0,X_2=0,\ldots,X_k=0) \;=\; 1-\prod_{i=1}^{k}(1-p_i)^0 \;=\; 1-\prod_{i=1}^{k}(1) \;=\; 0$$

- When exactly one parent X_j is equal to one:

Noisy-OR CPT (con't)



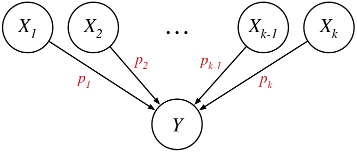
$$P(Y=1|X_1,X_2,\ldots,X_k) \;=\; 1-\prod_{i=1}^{k}(1-p_i)^{X_i}$$

- When all parents are equal to zero:

$$P(Y=1|X_1=0,X_2=0,\ldots,X_k=0) \;=\; 1-\prod_{i=1}^{k}(1-p_i)^0 \;=\; 1-\prod_{i=1}^{k}(1) \;=\; 0$$

- When exactly one parent X_j is equal to one:

$$P(Y=1|X_1=0,\ldots,X_{j-1}=0,X_j=1,X_{j+1}=0,\ldots,X_k=0)$$



$$P(Y=1|X_1, X_2, \ldots, X_k) \; = \; 1 - \prod_{i=1}^{k} (1 - p_i)^{X_i}$$

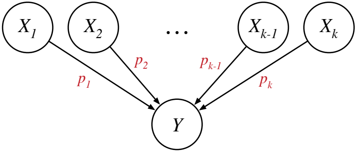
- When all parents are equal to zero:

$$P(Y=1|X_1=0, X_2=0, \ldots, X_k=0) \; = \; 1 - \prod_{i=1}^{k} (1-p_i)^0 \; = \; 1 - \prod_{i=1}^{k} (1) \; = \; 0$$

- When exactly one parent X_j is equal to one:

$$\begin{aligned} &P(Y=1|X_1=0, \ldots, X_{j-1}=0, X_j=1, X_{j+1}=0, \ldots, X_k=0) \\ &\quad = \; 1 - (1-p_1)^0 \cdots (1-p_{j-1})^0 (1-p_j)^1 (1-p_{j+1})^0 \cdots (1-p_k)^0 \end{aligned}$$

Noisy-OR CPT (con't)



$$P(Y=1|X_1,X_2,\ldots,X_k) \;=\; 1-\prod_{i=1}^{k}(1-p_i)^{X_i}$$

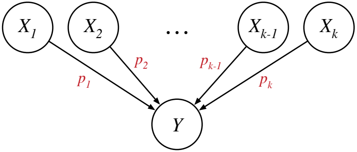
- When all parents are equal to zero:

$$P(Y=1|X_1=0,X_2=0,\ldots,X_k=0) \;=\; 1-\prod_{i=1}^{k}(1-p_i)^0 \;=\; 1-\prod_{i=1}^{k}(1) \;=\; 0$$

- When exactly one parent X_j is equal to one:

$$\begin{aligned} &P(Y=1|X_1=0,\ldots,X_{j-1}=0,X_j=1,X_{j+1}=0,\ldots,X_k=0) \\ &\quad= 1-(1-p_1)^0\cdots(1-p_{j-1})^0(1-p_j)^1(1-p_{j+1})^0\cdots(1-p_k)^0 \\ &\quad= 1-(1-p_j) \end{aligned}$$

Noisy-OR CPT (con't)



$$P(Y=1|X_1, X_2, \ldots, X_k) = 1 - \prod_{i=1}^{k}(1-p_i)^{X_i}$$

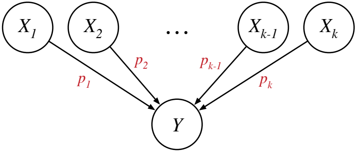
- When all parents are equal to zero:

$$P(Y=1|X_1=0, X_2=0, \ldots, X_k=0) = 1 - \prod_{i=1}^{k}(1-p_i)^0 = 1 - \prod_{i=1}^{k}(1) = 0$$

- When exactly one parent X_j is equal to one:

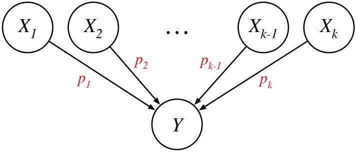
$$\begin{aligned}
&P(Y=1|X_1=0, \ldots, X_{j-1}=0, X_j=1, X_{j+1}=0, \ldots, X_k=0) \\
&\quad = 1 - (1-p_1)^0 \cdots (1-p_{j-1})^0 (1-p_j)^1 (1-p_{j+1})^0 \cdots (1-p_k)^0 \\
&\quad = 1 - (1-p_j) \\
&\quad = p_j
\end{aligned}$$

Noisy-OR CPT (con't)



$$P(Y=1|X_1, X_2, \ldots, X_k) \;=\; 1 - \prod_{i=1}^{k}(1-p_i)^{X_i}$$

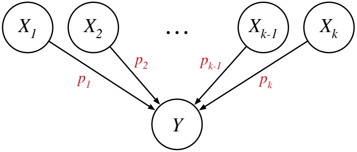
Noisy-OR CPT (con't)



$$P(Y=1|X_1, X_2, \ldots, X_k) \;=\; 1 - \prod_{i=1}^{k}(1-p_i)^{X_i}$$

- Modeling uncertainty

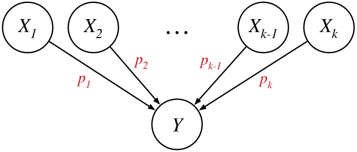
Noisy-OR CPT (con't)



$$P(Y=1|X_1, X_2, \dots, X_k) \; = \; 1 - \prod_{i=1}^{k}(1-p_i)^{X_i}$$

- **Modeling uncertainty**
  Intuitively, $p_i \in [0,1]$ is the probability that $X_i=1$ **by itself** triggers $Y=1$.

Noisy-OR CPT (con't)

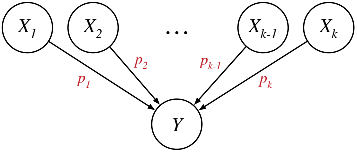


$$P(Y=1|X_1, X_2, \ldots, X_k) \;=\; 1 - \prod_{i=1}^{k}(1-p_i)^{X_i}$$

- **Modeling uncertainty**
  Intuitively, $p_i \in [0,1]$ is the probability that $X_i = 1$ **by itself** triggers $Y = 1$.

- **Logical OR as special case**

Noisy-OR CPT (con't)

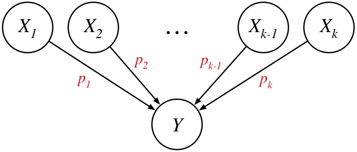


$$P(Y=1|X_1, X_2, \ldots, X_k) \;=\; 1 - \prod_{i=1}^{k}(1-p_i)^{X_i}$$

- **Modeling uncertainty**
  Intuitively, $p_i \in [0,1]$ is the probability that $X_i = 1$ **by itself** triggers $Y = 1$.

- **Logical OR as special case**
  We recover a logical OR gate by taking the limit $p_i \rightarrow 1$ for all parents $i = 1, 2, \ldots, k$.

Noisy-OR CPT (con't)



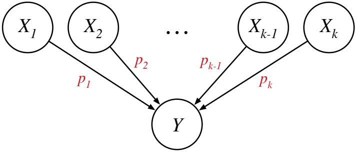
$$P(Y=1|X_1, X_2, \ldots, X_k) \;=\; 1 - \prod_{i=1}^{k}(1-p_i)^{X_i}$$

- **Modeling uncertainty**
  Intuitively, $p_i \in [0,1]$ is the probability that $X_i = 1$ **by itself** triggers $Y = 1$.

- **Logical OR as special case**
  We recover a logical OR gate by taking the limit $p_i \to 1$ for all parents $i = 1, 2, \ldots, k$.

- **Canonical application**

Noisy-OR CPT (con't)



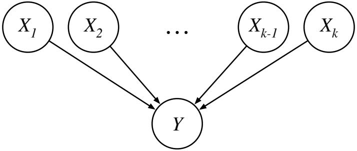
$$P(Y=1|X_1, X_2, \ldots, X_k) \;=\; 1 - \prod_{i=1}^{k}(1-p_i)^{X_i}$$

- **Modeling uncertainty**
  Intuitively, $p_i \in [0,1]$ is the probability that $X_i=1$ **by itself** triggers $Y=1$.

- **Logical OR as special case**
  We recover a logical OR gate by taking the limit $p_i \to 1$ for all parents $i = 1, 2, \ldots, k$.

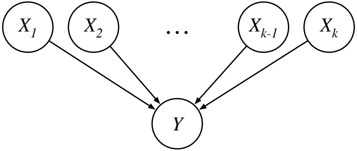
- **Canonical application**
  The parents $\{X_i\}_{i=1}^{k}$ are diseases, and the child Y is a symptom. The more diseases, the more likely is the symptom.

4. Sigmoid CPT



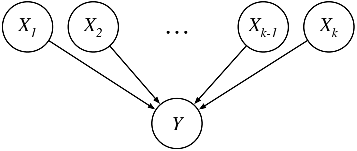
Use k real numbers $\theta_i \in \Re$ to parameterize all 2^k entries in the CPT:

4. Sigmoid CPT



Use k real numbers $\theta_i \in \Re$ to parameterize all 2^k entries in the CPT:

$$P(Y=1|X_1, X_2, \ldots, X_k) = \sigma\left(\sum_{i=1}^{k} \theta_i X_i\right)$$

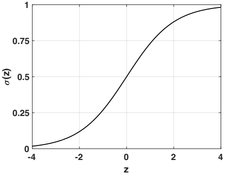


Use k real numbers $\theta_i \in \Re$ to parameterize all 2^k entries in the CPT:

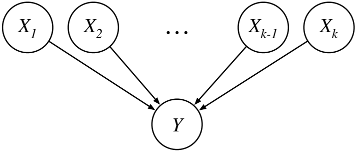
$$P(Y=1|X_1, X_2, \ldots, X_k) \;=\; \sigma\left(\sum_{i=1}^{k} \theta_i X_i\right)$$

The function on the right hand side is called the **sigmoid** function:

$$\sigma(z) \;=\; \frac{1}{1+e^{-z}}$$

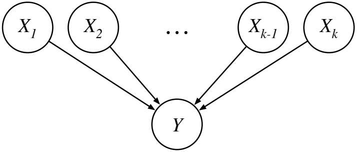


4. Sigmoid CPT (con't)



$$P(Y=1|X_1, X_2, \ldots, X_k) \;=\; \sigma\left(\sum_{i=1}^{k} \theta_i X_i\right)$$

Other uses of sigmoid functions:

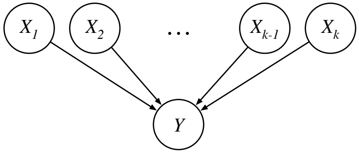


$$P(Y=1|X_1, X_2, \ldots, X_k) \; = \; \sigma\left(\sum_{i=1}^{k} \theta_i X_i\right)$$

Other uses of sigmoid functions:

- Activation function in neural nets
- Inverse of the link function for logistic regression

4. Sigmoid CPT (con't)

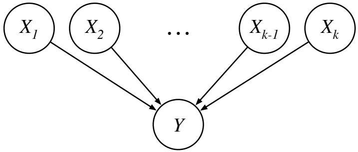


$$P(Y=1|X_1, X_2, \ldots, X_k) \; = \; \sigma\left(\sum_{i=1}^{k} \theta_i X_i\right)$$

Other uses of sigmoid functions:

- Activation function in neural nets
- Inverse of the link function for logistic regression

Properties:



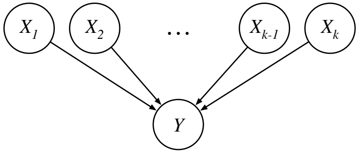
$$P(Y=1|X_1, X_2, \ldots, X_k) \;=\; \sigma\left(\sum_{i=1}^{k} \theta_i X_i\right)$$

Other uses of sigmoid functions:

- Activation function in neural nets
- Inverse of the link function for logistic regression

Properties:

- If $\theta_i > 0$, then $X_i=1$ favors $Y=1$.



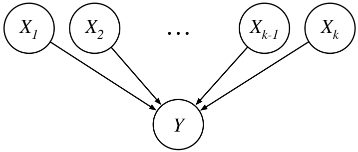
$$P(Y=1|X_1, X_2, \ldots, X_k) \;=\; \sigma\left(\sum_{i=1}^{k} \theta_i X_i\right)$$

Other uses of sigmoid functions:

- Activation function in neural nets
- Inverse of the link function for logistic regression

Properties:

- If $\theta_i > 0$, then $X_i = 1$ favors $Y = 1$.
- If $\theta_i < 0$, then $X_i = 1$ inhibits $Y = 1$.



$$P(Y=1|X_1, X_2, \ldots, X_k) \;=\; \sigma\left(\sum_{i=1}^{k} \theta_i X_i\right)$$

Other uses of sigmoid functions:

- Activation function in neural nets
- Inverse of the link function for logistic regression

Properties:

- If $\theta_i > 0$, then $X_i = 1$ favors $Y = 1$.
- If $\theta_i < 0$, then $X_i = 1$ inhibits $Y = 1$.
- These effects can mix in a sigmoid CPT (unlike noisy-OR).

d-separation and examples

Conditional independence

- What we’ve already seen

Conditional independence

- What we've already seen

  A node X_i is conditionally independent of its non-parent ancestors given its parents:

$$P(X_i|X_1, X_2, \ldots, X_{i-1}) \;=\; P(X_i|\mathrm{pa}(X_i))$$

Conditional independence

- **What we've already seen**

  A node X_i is conditionally independent of its non-parent ancestors given its parents:

  $$P(X_i | X_1, X_2, \ldots, X_{i-1}) = P(X_i | \mathrm{pa}(X_i))$$

- **What we can ask more generally**

  Let X, Y, and E refer to disjoint *sets* of nodes in a BN. When is X conditionally independent of Y given E?

Conditional independence

- **What we've already seen**

  A node X_i is conditionally independent of its non-parent ancestors given its parents:

$$P(X_i | X_1, X_2, \ldots, X_{i-1}) \;=\; P(X_i | \mathrm{pa}(X_i))$$

- **What we can ask more generally**

  Let X, Y, and E refer to disjoint *sets* of nodes in a BN. When is X conditionally independent of Y given E?

$X \perp\!\!\!\perp Y \mid E$

$\perp$

$$\text{When is} \quad \left\{ \begin{array}{lcl} P(X|E,Y) & = & P(X|E) \\ P(Y|E,X) & = & P(Y|E) \\ P(X,Y|E) & = & P(X|E)\,P(Y|E) \end{array} \right\} \quad ?$$

Conditional independence

- **What we've already seen**

  A node X_i is conditionally independent of its non-parent ancestors given its parents:

$$P(X_i | X_1, X_2, \ldots, X_{i-1}) \;=\; P(X_i | \mathrm{pa}(X_i))$$

- **What we can ask more generally**

  Let X, Y, and E refer to disjoint *sets* of nodes in a BN. When is X conditionally independent of Y given E?

$$\text{When is } \left\{ \begin{array}{rcl} P(X|E,Y) & = & P(X|E) \\ P(Y|E,X) & = & P(Y|E) \\ P(X,Y|E) & = & P(X|E)\,P(Y|E) \end{array} \right\} \text{ ?}$$

- **Above is special case**

$X_i \perp Y \mid pa(X_i)$

$$X = \{X_i\}, \quad E = \mathrm{pa}(X_i) \quad Y = \{X_1, X_2, \ldots, X_{i-1}\} - \mathrm{pa}(X_i)$$

Base Cases

d-separation in DAGs

d-separation = direction-dependent separation

d-separation in DAGs

d-separation = direction-dependent separation

- Motivation

  How is conditional independence in a BN encoded by the structure of its DAG?

d-separation = direction-dependent separation

- **Motivation**

  How is conditional independence in a BN encoded by the structure of its DAG?

- **Theorem**

  $P(X, Y|E) = P(X|E)\,P(Y|E)$ if and only if every *path* from a node in X to a node in Y is *blocked* by E.

d-separation in DAGs

d-separation = direction-dependent separation

- **Motivation**

  How is conditional independence in a BN encoded by the structure of its DAG?

- **Theorem**

  $P(X, Y|E) = P(X|E)\,P(Y|E)$ if and only if every *path* from a node in X to a node in Y is *blocked* by E.

What counts as a path, and when is it blocked?

Paths in DAGs

- Definition

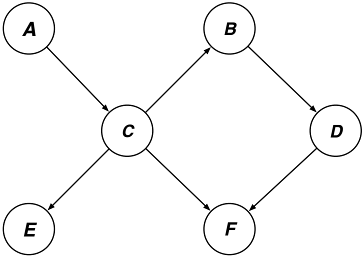
  A **path** is any sequence of nodes connected by edges (*regardless of their directionalities*); it is also assumed that no nodes repeat.

Paths in DAGs

- Definition

  A **path** is any sequence of nodes connected by edges (*regardless of their directionalities*); it is also assumed that no nodes repeat.

- Examples

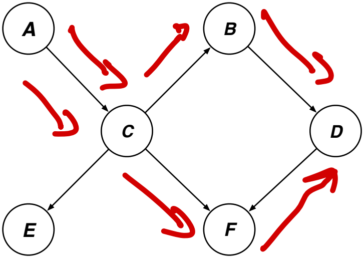


Paths in DAGs

- Definition

  A **path** is any sequence of nodes connected by edges (*regardless of their directionalities*); it is also assumed that no nodes repeat.

- Examples



Two ? paths from A to D:

(1) $A \rightarrow C \rightarrow B \rightarrow D$

(2) $A \rightarrow C \rightarrow F \leftarrow D$

Blocked paths

- **Definition**

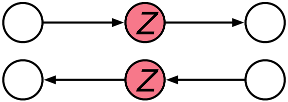
  A path π is **blocked** by a set of nodes E if there exists a node $Z \in \pi$ for which one of the three following conditions hold.

Blocked paths

- **Definition**

  A path π is **blocked** by a set of nodes E if there exists a node $Z \in \pi$ for which one of the three following conditions hold.

(1) $Z \in E$



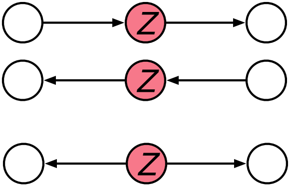
edges **align**

- Definition

  A path π is **blocked** by a set of nodes E if there exists a node $Z \in \pi$ for which one of the three following conditions hold.

(1) $Z \in E$ — edges **align**

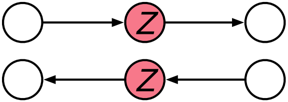
(2) $Z \in E$ — edges **diverge**



- Definition

A path π is **blocked** by a set of nodes E if there exists a node $Z \in \pi$ for which one of the three following conditions hold.

(1) $Z \in E$

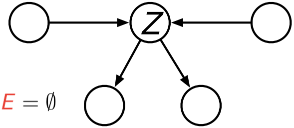


edges **align**

(2) $Z \in E$

edges **diverge**

(3) $Z \notin E$



edges **converge**

$\text{descendants}(Z) \cap E = \emptyset$

d-separation

- **Theorem**

  $P(X, Y|E) = P(X|E)\,P(Y|E)$ if and only if every *path* from a node in X to a node in Y is *blocked* by E.

d-separation

- **Theorem**

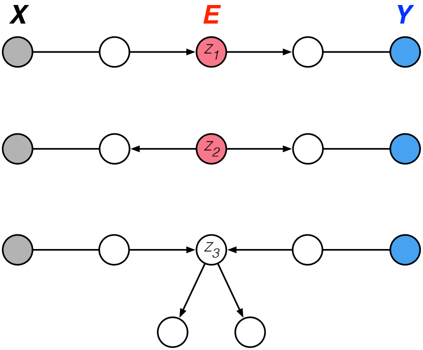
  $P(X, Y|E) = P(X|E)\, P(Y|E)$ if and only if every *path* from a node in X to a node in Y is *blocked* by E.

- **Intuition**

- Theorem

  $P(X, Y|E) = P(X|E)\,P(Y|E)$ if and only if every *path* from a node in X to a node in Y is *blocked* by E.

- Intuition

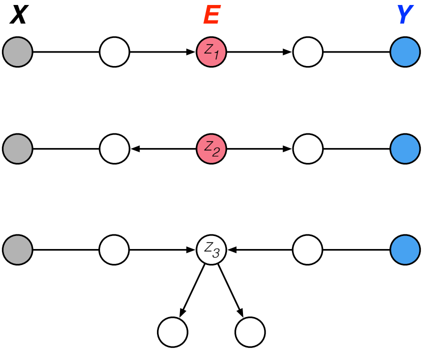


d-separation

- **Theorem**

  $P(X, Y|E) = P(X|E)\,P(Y|E)$ if and only if every *path* from a node in X to a node in Y is *blocked* by E.

- **Intuition**

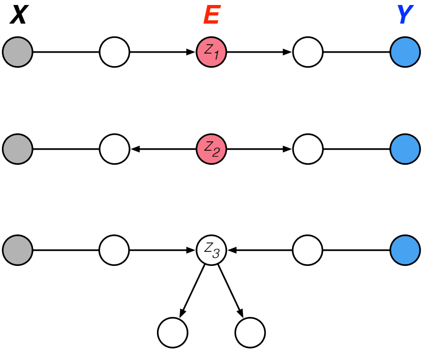


$Z_1 \in E$ is an intervening event in a causal chain

- **Theorem**

  $P(X, Y|E) = P(X|E)\,P(Y|E)$ if and only if every *path* from a node in X to a node in Y is *blocked* by E.

- **Intuition**



$Z_1 \in E$ is an intervening event in a causal chain

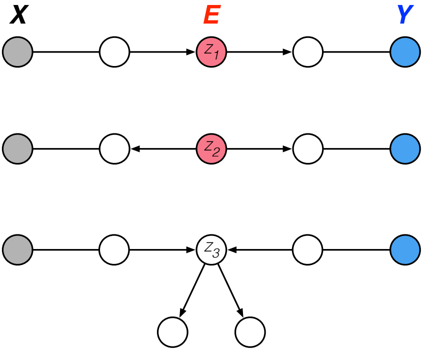
$Z_2 \in E$ is a common explanation or cause

d-separation

- **Theorem**

  $P(X, Y|E) = P(X|E)\,P(Y|E)$ if and only if every *path* from a node in X to a node in Y is *blocked* by E.

- **Intuition**



$Z_1 \in E$ is an intervening event in a causal chain

$Z_2 \in E$ is a common explanation or cause

$Z_3 \notin E, \mathrm{desc}(Z_3) \cap E = \emptyset$ is an unobserved common effect

d-separation

- **Theorem**

  $P(X, Y|E) = P(X|E)\,P(Y|E)$ if and only if every *path* from a node in X to a node in Y is *blocked* by E.

- **Theorem**

  $P(X, Y|E) = P(X|E)\,P(Y|E)$ if and only if every *path* from a node in X to a node in Y is *blocked* by E.

- **Proof** (not given)

- **Theorem**

  $P(X, Y|E) = P(X|E)\,P(Y|E)$ if and only if every *path* from a node in X to a node in Y is *blocked* by E.

- **Proof** (not given)

  The proof of the theorem is non-trivial.
  You are **not** responsible for its proof.

d-separation

- **Theorem**

  $P(X, Y|E) = P(X|E)\,P(Y|E)$ if and only if every *path* from a node in X to a node in Y is *blocked* by E.

- **Proof** (not given)

  The proof of the theorem is non-trivial.
  You are **not** responsible for its proof.

- **How useful is the theorem?**

- **Theorem**

  $P(X, Y|E) = P(X|E)\, P(Y|E)$ if and only if every *path* from a node in X to a node in Y is *blocked* by E.

- **Proof** (not given)

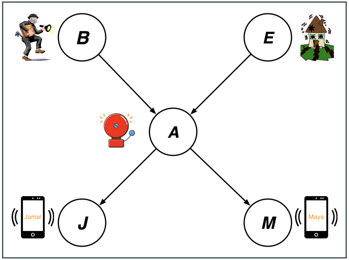
  The proof of the theorem is non-trivial.
  You are **not** responsible for its proof.

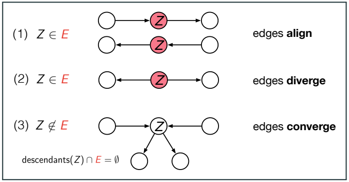
- **How useful is the theorem? Very!**

  There are efficient algorithms to test d-separation in large BNs.
  You should become skilled at these tests in simple BNs.

Alarm example

A. TRUE or B. FALSE?





Alarm example

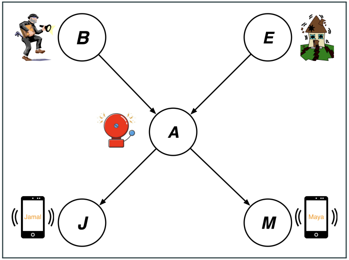
A. TRUE or B. FALSE?

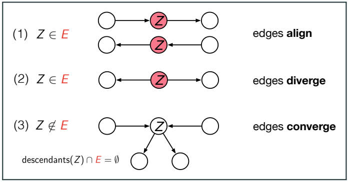
1. $P(B|A, M) \stackrel{?}{=} P(B|A)$

$X = \{B\}$

$Y = \{M\}$

$E = \{A\}$



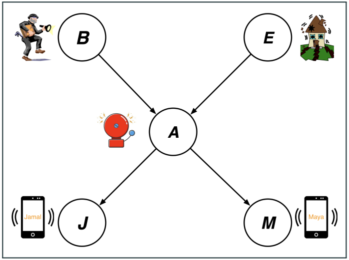


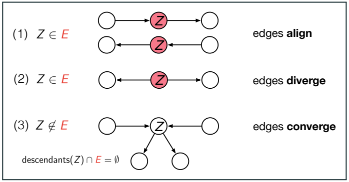
Alarm example

A. TRUE or B. FALSE?

1. $P(B|A, M) \stackrel{?}{=} P(B|A)$

   The evidence is $\{A\}$.





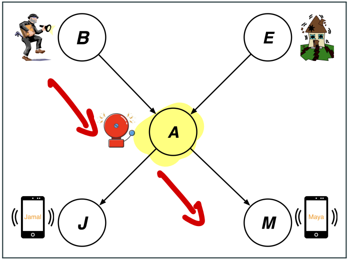
Alarm example

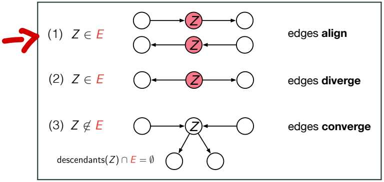
A. TRUE or B. FALSE?

1. $P(B|A, M) \stackrel{?}{=} P(B|A)$

   The evidence is $\{A\}$.

   There is one path $B \rightarrow A \rightarrow M$.



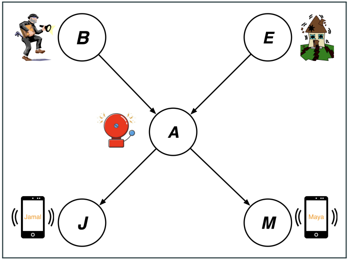


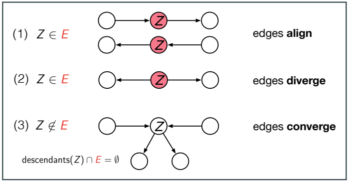
Alarm example

A. TRUE or B. FALSE?

1. $P(B|A, M) \stackrel{?}{=} P(B|A)$

   The evidence is $\{A\}$.
   There is one path $B \rightarrow A \rightarrow M$.
   Node A satisfies condition (1).



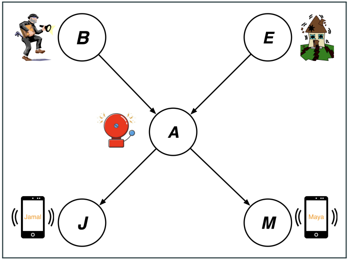


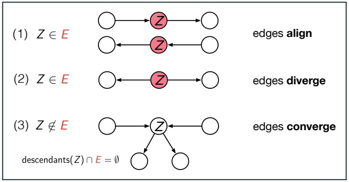
Alarm example

A. TRUE or B. FALSE?

1. $P(B|A, M) \stackrel{?}{=} P(B|A)$

   The evidence is $\{A\}$.
   There is one path $B \to A \to M$.
   Node A satisfies condition (1).
   The statement is **true**.





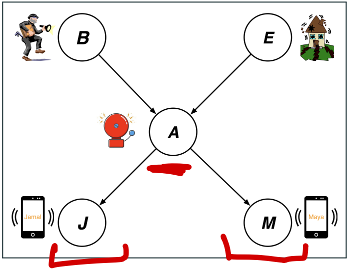
Alarm example

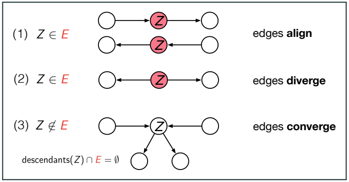
A. TRUE or B. FALSE?

1. $P(B|A, M) \stackrel{?}{=} P(B|A)$

   The evidence is $\{A\}$.
   There is one path $B \rightarrow A \rightarrow M$.
   Node A satisfies condition (1).
   The statement is **true**.

2. $P(J, M|A) \stackrel{?}{=} P(J|A)\,P(M|A)$





Alarm example

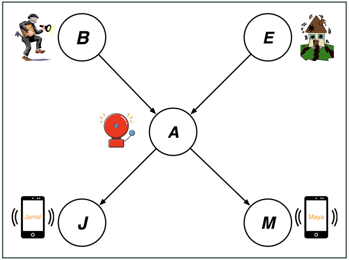
A. TRUE or B. FALSE?

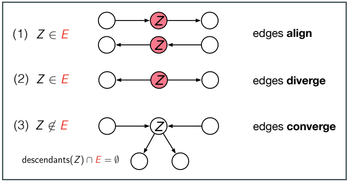
1. $P(B|A, M) \stackrel{?}{=} P(B|A)$

   The evidence is $\{A\}$.
   There is one path $B \rightarrow A \rightarrow M$.
   Node A satisfies condition (1).
   The statement is **true**.

2. $P(J, M|A) \stackrel{?}{=} P(J|A)\, P(M|A)$

   The evidence is $\{A\}$.





Alarm example

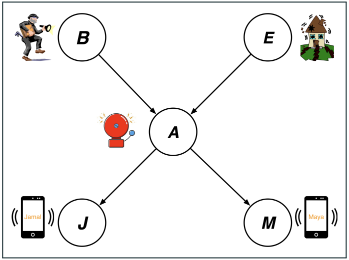
A. TRUE or B. FALSE?

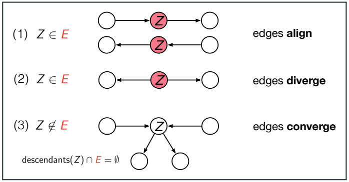
1. $P(B|A, M) \stackrel{?}{=} P(B|A)$

   The evidence is $\{A\}$.
   There is one path $B \to A \to M$.
   Node A satisfies condition (1).
   The statement is **true**.

2. $P(J, M|A) \stackrel{?}{=} P(J|A)\,P(M|A)$

   The evidence is $\{A\}$.
   There is one path $J \leftarrow A \to M$.





Alarm example

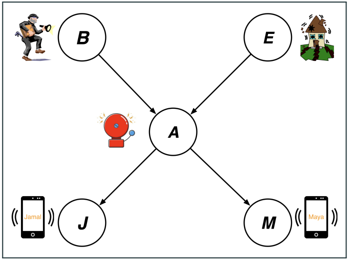
A. TRUE or B. FALSE?

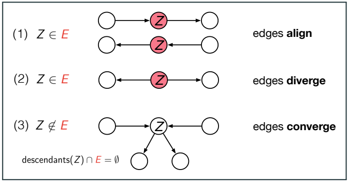
1. $P(B|A, M) \stackrel{?}{=} P(B|A)$

   The evidence is $\{A\}$.
   There is one path $B \rightarrow A \rightarrow M$.
   Node A satisfies condition (1).
   The statement is **true**.

2. $P(J, M|A) \stackrel{?}{=} P(J|A)\,P(M|A)$

   The evidence is $\{A\}$.
   There is one path $J \leftarrow A \rightarrow M$.
   Node A satisfies condition (2).





Alarm example

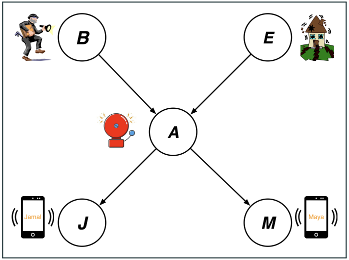
A. TRUE or B. FALSE?

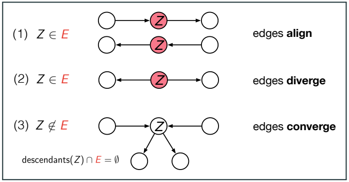
1. $P(B|A, M) \stackrel{?}{=} P(B|A)$

   The evidence is $\{A\}$.
   There is one path $B \rightarrow A \rightarrow M$.
   Node A satisfies condition (1).
   The statement is **true**.

2. $P(J, M|A) \stackrel{?}{=} P(J|A)\,P(M|A)$

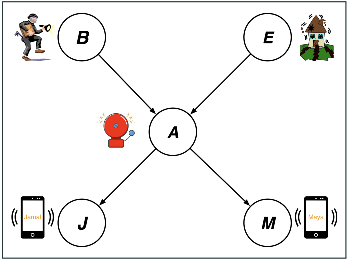
   The evidence is $\{A\}$.
   There is one path $J \leftarrow A \rightarrow M$.
   Node A satisfies condition (2).
   The statement is **true**.

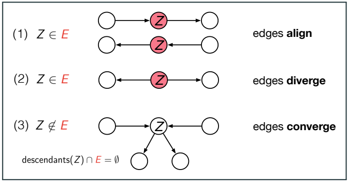




Alarm example (con't)

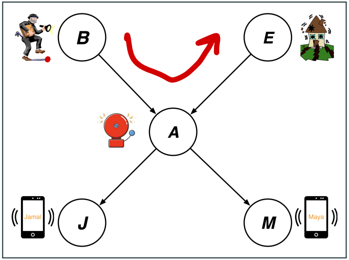
A. TRUE or B. FALSE?

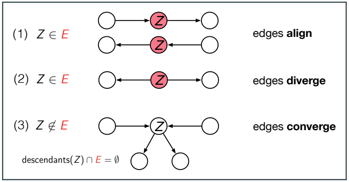




A. TRUE or B. FALSE?

3. $P(B) \stackrel{?}{=} P(B|E)$



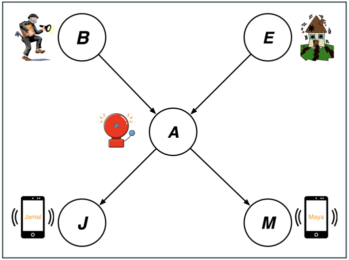


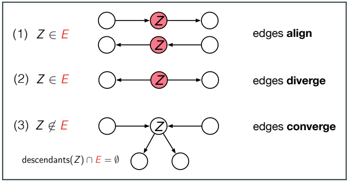
Alarm example (con't)

A. TRUE or B. FALSE?

3. $P(B) \stackrel{?}{=} P(B|E)$

   The evidence is $\{\}$.





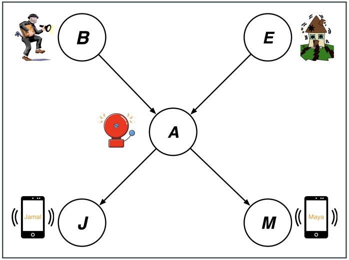
Alarm example (con't)

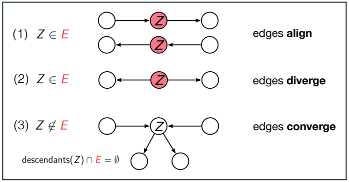
A. TRUE or B. FALSE?

3. $P(B) \stackrel{?}{=} P(B|E)$

   The evidence is $\{\}$.

   There is one path $B \rightarrow A \leftarrow E$.



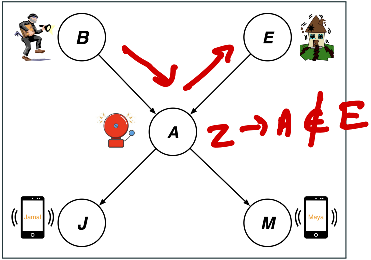


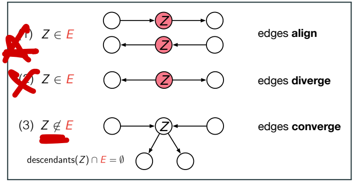
Alarm example (con't)

A. TRUE or B. FALSE?

3. $P(B) \stackrel{?}{=} P(B|E)$

   The evidence is $\{\}$.
   There is one path $B \rightarrow A \leftarrow E$.
   Node A satisfies condition (3).



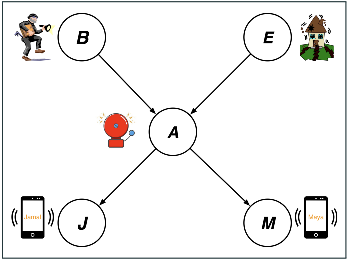


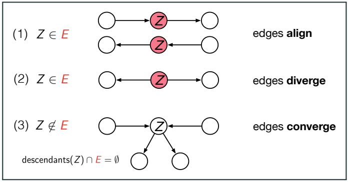
Alarm example (con't)

A. TRUE or B. FALSE?

3. $P(B) \stackrel{?}{=} P(B|E)$

   The evidence is $\{\}$.
   There is one path $B \rightarrow A \leftarrow E$.
   Node A satisfies condition (3).
   The statement is **true**.





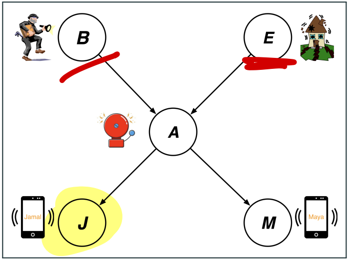
Alarm example (con't)

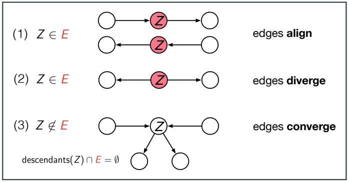
A. TRUE or B. FALSE?

3. $P(B) \stackrel{?}{=} P(B|E)$

   The evidence is $\{\}$.
   There is one path $B \to A \leftarrow E$.
   Node A satisfies condition (3).
   The statement is **true**.

4. $P(B|M) \stackrel{?}{=} P(B|M, E)$





Alarm example (con't)

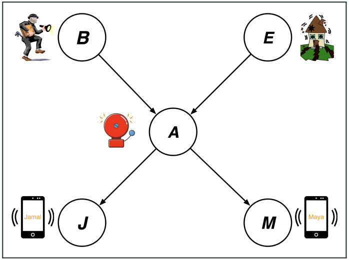
A. TRUE or B. FALSE?

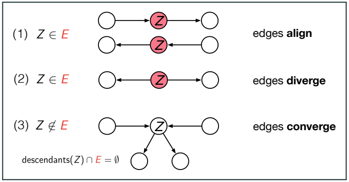
3. $P(B) \stackrel{?}{=} P(B|E)$

   The evidence is $\{\}$.
   There is one path $B \rightarrow A \leftarrow E$.
   Node A satisfies condition (3).
   The statement is **true**.

4. $P(B|M) \stackrel{?}{=} P(B|M, E)$

   The evidence is $\{M\}$.





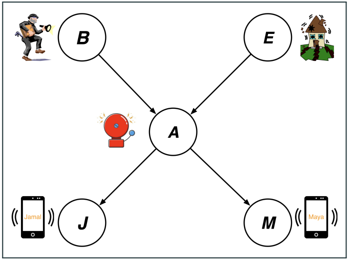
A. TRUE or B. FALSE?

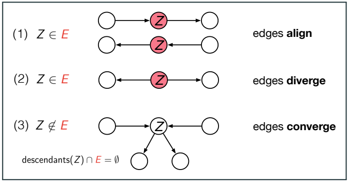
3. $P(B) \stackrel{?}{=} P(B|E)$

   The evidence is $\{\}$.
   There is one path $B \rightarrow A \leftarrow E$.
   Node A satisfies condition (3).
   The statement is **true**.

4. $P(B|M) \stackrel{?}{=} P(B|M, E)$

   The evidence is $\{M\}$.
   There is one path $B \rightarrow A \leftarrow E$.





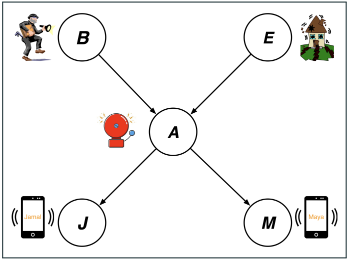
A. TRUE or B. FALSE?

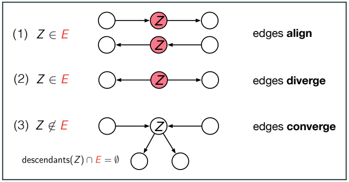
3. $P(B) \stackrel{?}{=} P(B|E)$

   The evidence is $\{\}$.
   There is one path $B \rightarrow A \leftarrow E$.
   Node A satisfies condition (3).
   The statement is **true**.

4. $P(B|M) \stackrel{?}{=} P(B|M, E)$

   The evidence is $\{M\}$.
   There is one path $B \rightarrow A \leftarrow E$.
   Note that $M \in \mathrm{desc}(A)$.





Alarm example (con't)

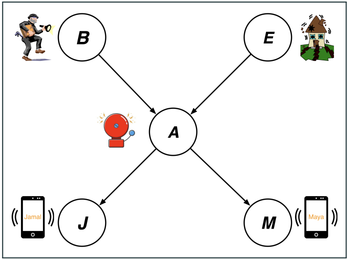
A. TRUE or B. FALSE?

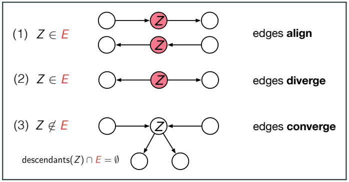
3. $P(B) \stackrel{?}{=} P(B|E)$

   The evidence is $\{\}$.
   There is one path $B \rightarrow A \leftarrow E$.
   Node A satisfies condition (3).
   The statement is **true**.

4. $P(B|M) \stackrel{?}{=} P(B|M, E)$

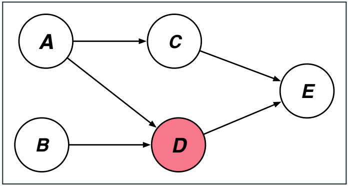
   The evidence is $\{M\}$.
   There is one path $B \rightarrow A \leftarrow E$.
   Note that $M \in \mathrm{desc}(A)$.
   The statement is **false**.

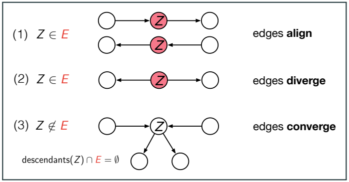




Loopy example

A. TRUE or B. FALSE?

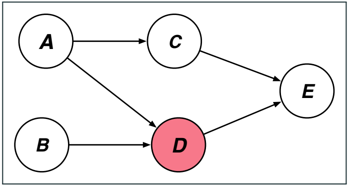


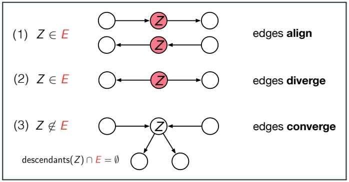


Loopy example

A. TRUE or B. FALSE?

5. $P(B|D, E) \stackrel{?}{=} P(B|D)$



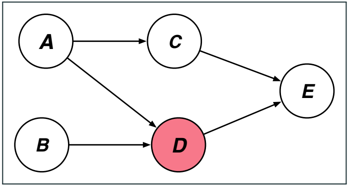


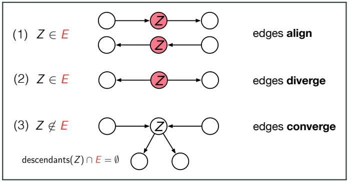
Loopy example

A. TRUE or B. FALSE?

5. $P(B|D, E) \stackrel{?}{=} P(B|D)$

   The evidence is $\{D\}$.





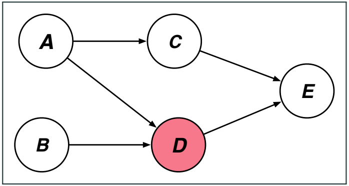
Loopy example

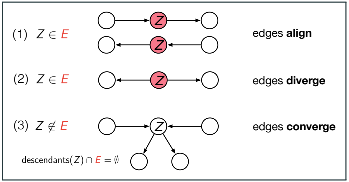
A. TRUE or B. FALSE?

5. $P(B|D, E) \stackrel{?}{=} P(B|D)$

   The evidence is $\{D\}$.

   There are two paths from B to E.





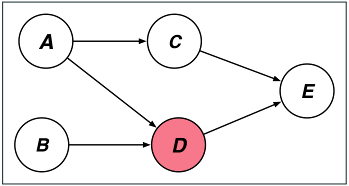
Loopy example

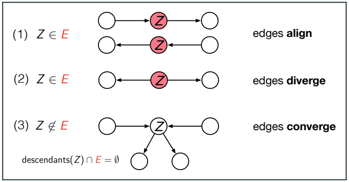
A. TRUE or B. FALSE?

5. $P(B|D,E) \stackrel{?}{=} P(B|D)$

   The evidence is $\{D\}$.
   There are two paths from B to E.

   Path $B \rightarrow D \rightarrow E$
   is blocked by node D,
   satisfying condition (1).





Loopy example

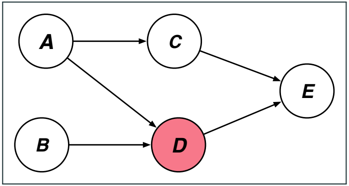
A. TRUE or B. FALSE?

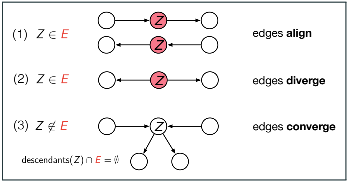
5. $P(B|D, E) \stackrel{?}{=} P(B|D)$

   The evidence is $\{D\}$.
   There are two paths from B to E.

   Path $B \rightarrow D \rightarrow E$
   is blocked by node D,
   satisfying condition (1).

   Path $B \rightarrow D \leftarrow A \rightarrow C \rightarrow E$
   is not blocked by any node.





A. TRUE or B. FALSE?

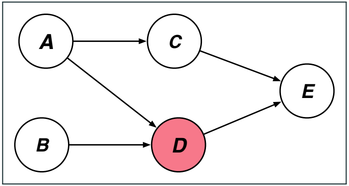
5. $P(B|D,E) \stackrel{?}{=} P(B|D)$

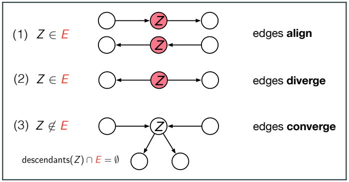
The evidence is $\{D\}$.
There are two paths from B to E.

Path $B \to D \to E$
is blocked by node D,
satisfying condition (1).

Path $B \to D \leftarrow A \to C \to E$
is not blocked by any node.

The statement is **false**.



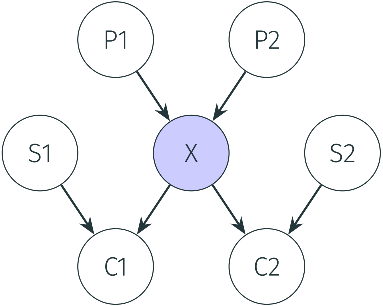


Markov Blanket

A Markov Blanket B_X of node X consists of **parents** of X, **children** of X and **"spouses"** (other parents of children of X, but not X) of X.

Markov Blanket

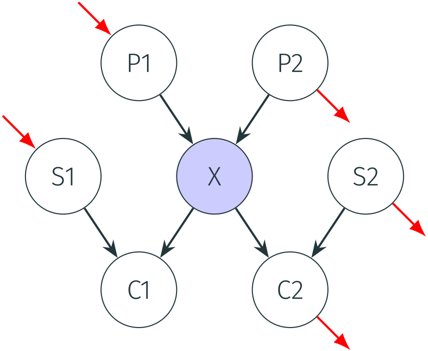
A Markov Blanket B_X of node X consists of **parents** of X, **children** of X and **"spouses"** (other parents of children of X, but not X) of X.



Every variable is conditionally independent of any other variable given it's Markov Blanket.

Markov Blanket

A Markov Blanket B_X of node X consists of **parents** of X, **children** of X and **"spouses"** (other parents of children of X, but not X) of X.



Every variable is conditionally independent of any other variable given it's Markov Blanket.

That's all folks!