Lecture 22: 15 April, 2025

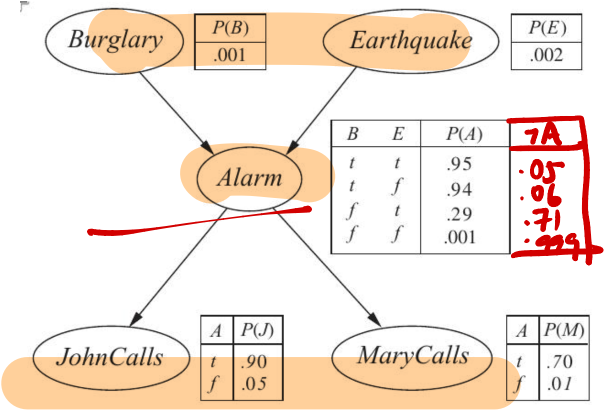
Madhavan Mukund

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

Data Mining and Machine Learning
January–April 2025

- Underlying DAG, no cyclic dependencies
- Each node has a local (conditional) probability table

$P(A \mid B, E)$



Evaluating a network

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

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

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

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
- Using $P(x_1, x_2, \ldots, x_n) = P(x_1 \mid x_2, \ldots, x_n) P(x_2 \mid x_3, \ldots, x_n) \cdots P(x_{n-1} \mid x_n) P(x_n)$ and writing variables in topolological sort order,

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

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
- Using $P(x_1, x_2, \ldots, x_n) = P(x_1 \mid x_2, \ldots, x_n) P(x_2 \mid x_3, \ldots, x_n) \cdots P(x_{n-1} \mid x_n) P(x_n)$ and writing variables in topolological sort order,

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

- Why is computing $P(b,m,j)$ enough? Should we not compute $P(b \mid m,j)$?

① $P(b \mid m, j) = \dfrac{P(b, m, j)}{P(m, j)}$ ✓ ?

$P(A \wedge B) = P(A \mid B) \cdot P(B)$

② $P(\neg b \mid m, j) = \dfrac{P(\neg b, m, j)}{P(m, j)}$

$P(A \mid B) = \dfrac{P(A \wedge B)}{P(B)}$

$P(b \mid m, j) = \alpha \cdot P(b, m, j)$ ✓

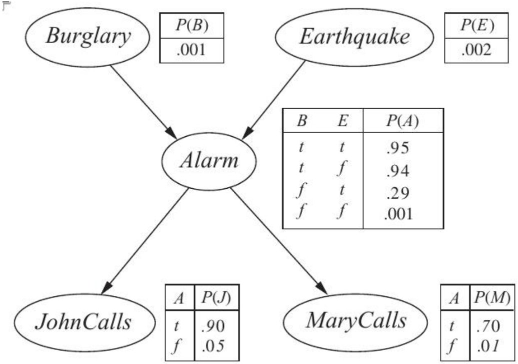
$P(\neg b \mid m, j) = \alpha \cdot P(\neg b, m, j)$ ✓

Conditional independence

- $x \perp y$ — x and y are independent
  - $P(x \wedge y) = P(x) \cdot P(y)$

$P(x = b \mid y = a) = P(x = b)$

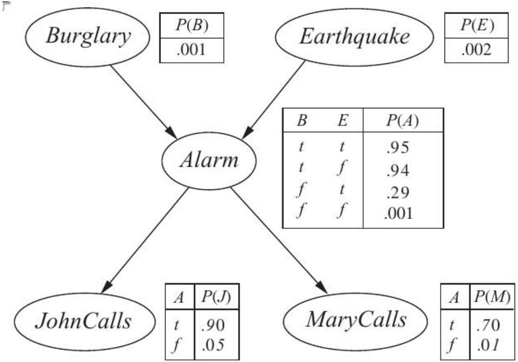
For any b, a



- $x \perp y$ — 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
  - $P(x \wedge y \mid z) = P(x \mid z) \cdot P(y \mid z)$

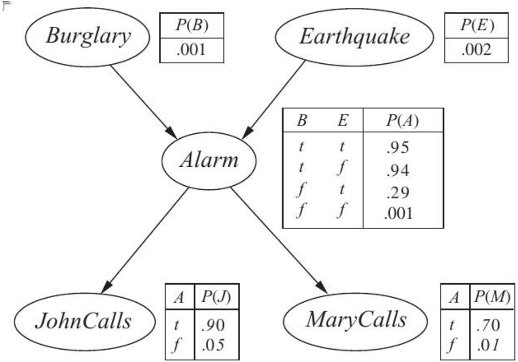
$P(x \mid y, z) = P(x \mid z)$

Is $M \perp B \mid A$?



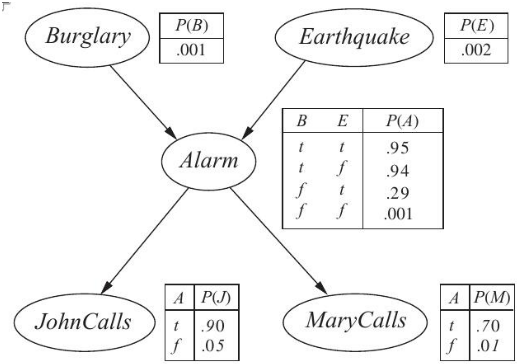
Conditional independence

- $x \perp y$ — 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* ($j \perp m$)?



Conditional independence

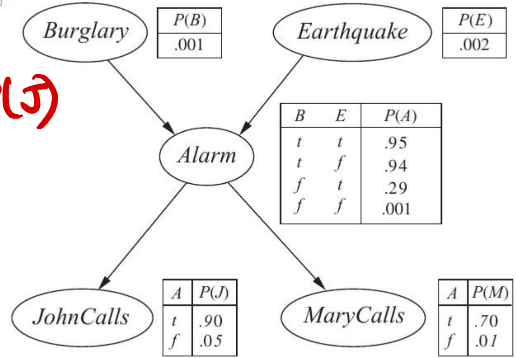
- $x \perp y$ — x and y are independent
  - $P(x \wedge y) = P(x) \cdot P(y)$
- $x \perp y \mid z$
  - x and y are independent given z
  - $P(x \wedge y \mid z) = P(x \mid z) \cdot P(y \mid z)$
- Is *JohnCalls* independent of *MaryCalls* ($j \perp m$)?
  - No — value of j tells us something about value of m and vice versa



Conditional independence

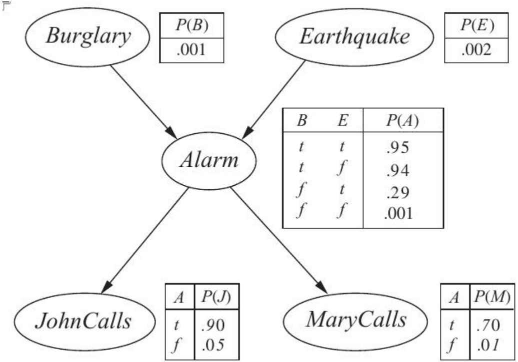
- $x \perp y$ — x and y are independent
  - $P(x \wedge y) = P(x) \cdot P(y)$
- $x \perp y \mid z$
  - x and y are independent given z
  - $P(x \wedge y \mid z) = P(x \mid z) \cdot P(y \mid z)$
- Is *JohnCalls* independent of *MaryCalls* ($j \perp m$)?
  - No — value of j tells us something about value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

$P(J|M) = P(J)$



Conditional independence

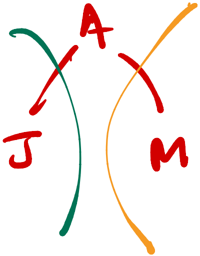
- $x \perp y$ — x and y are independent
  - $P(x \wedge y) = P(x) \cdot P(y)$
- $x \perp y \mid z$
  - x and y are independent given z
  - $P(x \wedge y \mid z) = P(x \mid z) \cdot P(y \mid z)$
- Is *JohnCalls* independent of *MaryCalls* ($j \perp m$)?
  - No — value of j tells us something about value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

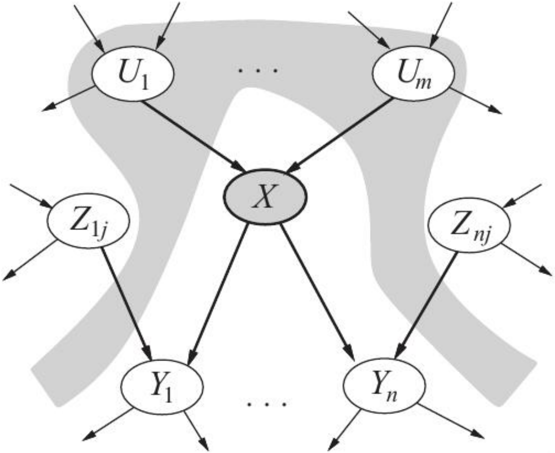


Probabilistic graphical models

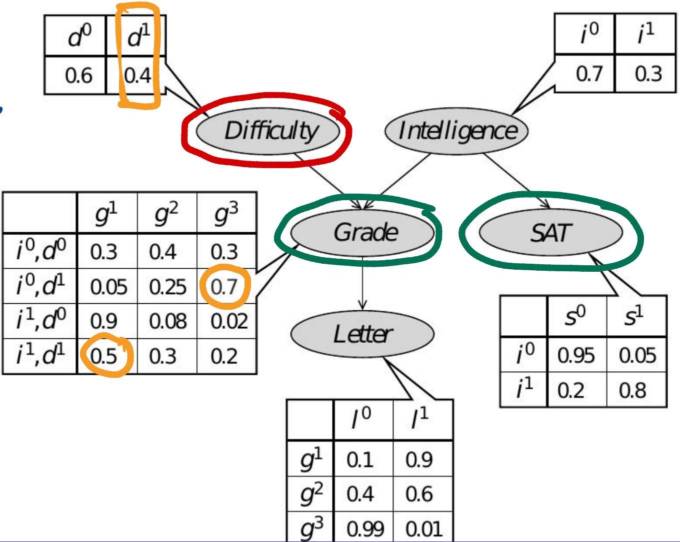
- Fundamental assumption

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





- *SAT* ⊥ *Grade* | ~~*Difficulty*~~ ? Intelligence



d^0	d^1
0.6	0.4

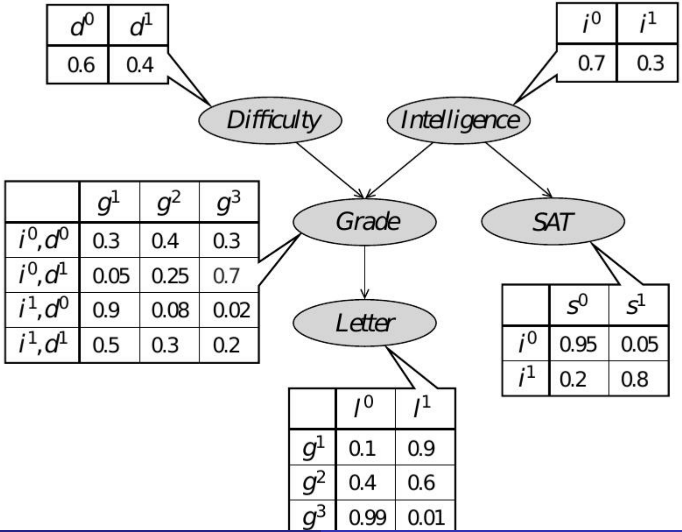
i^0	i^1
0.7	0.3

	g^1	g^2	g^3
i^0,d^0	0.3	0.4	0.3
i^0,d^1	0.05	0.25	0.7
i^1,d^0	0.9	0.08	0.02
i^1,d^1	0.5	0.3	0.2

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

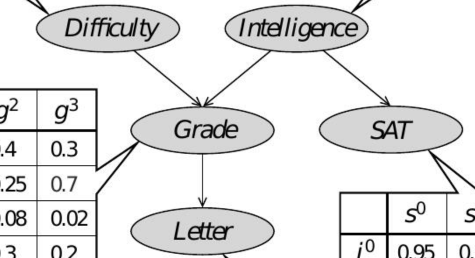
	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^3	0.99	0.01

- *SAT* $\perp$ *Grade* | *Difficulty* ?
  - No



Student example

- *SAT* $\perp$ *Grade* | *Difficulty* ?
  - No
- Can we calculate conditional independence from the graph?



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

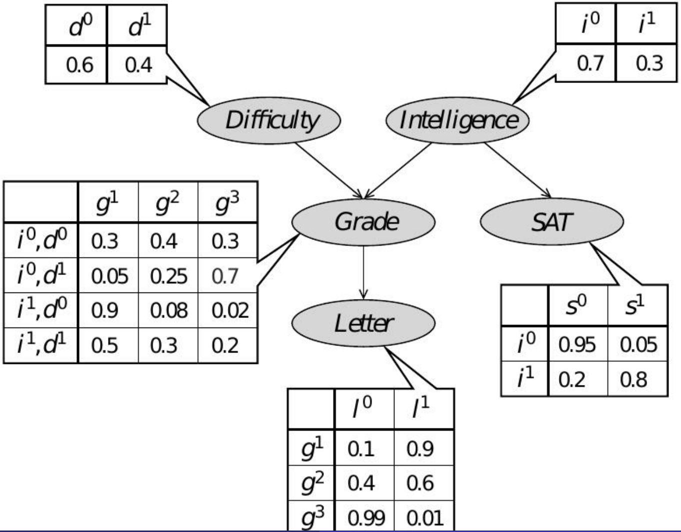
	g^1	g^2	g^3
i^0,d^0	0.3	0.4	0.3
i^0,d^1	0.05	0.25	0.7
i^1,d^0	0.9	0.08	0.02
i^1,d^1	0.5	0.3	0.2

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^3	0.99	0.01

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

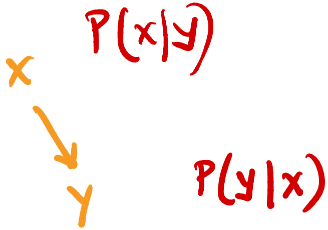


Conditional independence

- How does dependence "flow" through a network?

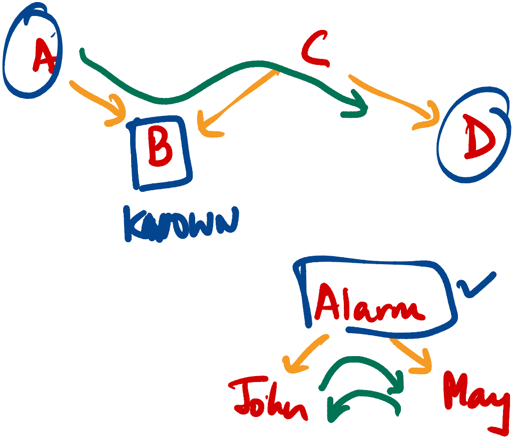
Conditional independence

- How does dependence “flow” through a network?
- For neighbouring nodes, dependence flows both ways
  - If $x \to y$, knowing x tells us about y and vice versa



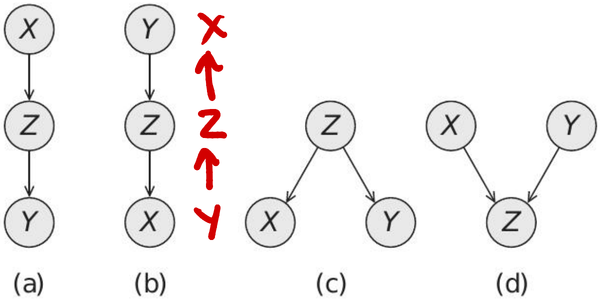
Conditional independence

- How does dependence "flow" through a network?
- For neighbouring nodes, dependence flows both ways
  - If $x \to y$, knowing x tells us about y and vice versa
- Examine trails between nodes
  - Paths in the underlying undirected graph

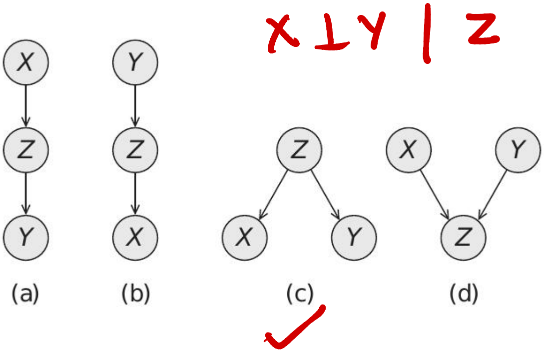


Conditional independence

- How does dependence "flow" through a network?
- For neighbouring nodes, dependence flows both ways
  - If $x \rightarrow y$, knowing x tells us about y and vice versa
- Examine trails between nodes
  - Paths in the underlying undirected graph
- Basic trails — (undirected) paths of length 2
  - Four basic trails

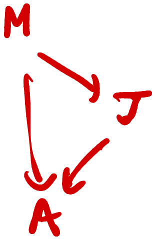


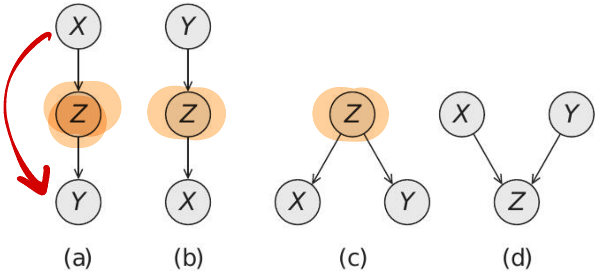
Basic trails



Basic trails

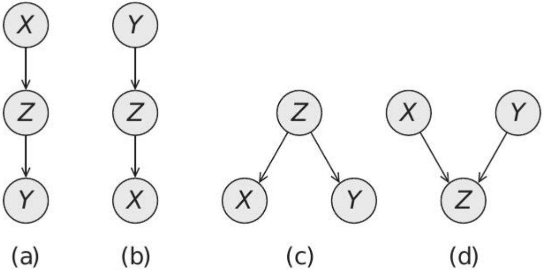
- (a), (b) and (c): Z blocks flow between X and Y, by semantics of Bayesian networks





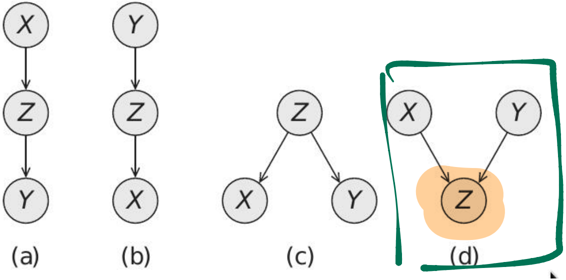
Basic trails

- (a), (b) and (c): Z blocks flow between X and Y, by semantics of Bayesian networks
- In (d), knowing Z allows influence to flow



Basic trails

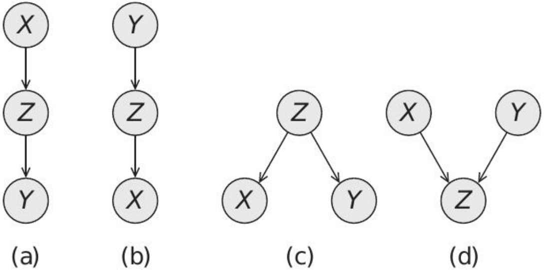
- (a), (b) and (c): Z blocks flow between X and Y, by semantics of Bayesian networks
- In (d), knowing Z allows influence to flow
  - Z: Car does not start
    X: Low Battery, Y: No Fuel



Know Z enable flow

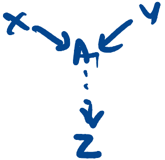
Basic trails

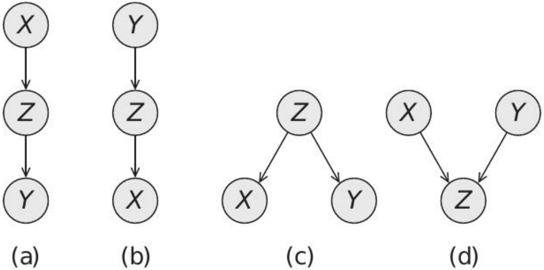
- (a), (b) and (c): Z blocks flow between X and Y, by semantics of Bayesian networks
- In (d), knowing Z allows influence to flow
  - Z: Car does not start
    X: Low Battery, Y: No Fuel
  - Z: Grass is wet
    X: Overnight rain, Y: Sprinkler ran

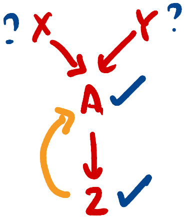


Basic trails

- (a), (b) and (c): Z blocks flow between X and Y, by semantics of Bayesian networks
- In (d), knowing Z allows influence to flow
  - Z: Car does not start
    X: Low Battery, Y: No Fuel
  - Z: Grass is wet
    X: Overnight rain, Y: Sprinkler ran
  - Simplest form of V-structure



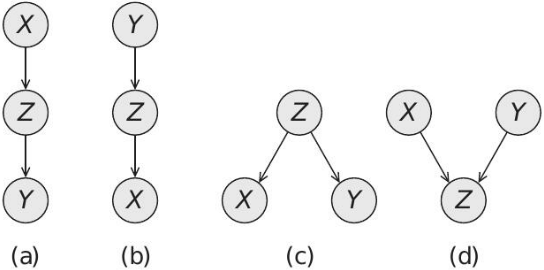




$X \perp Y \mid Z$

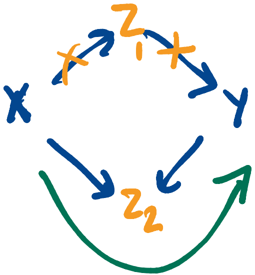
D-Separation

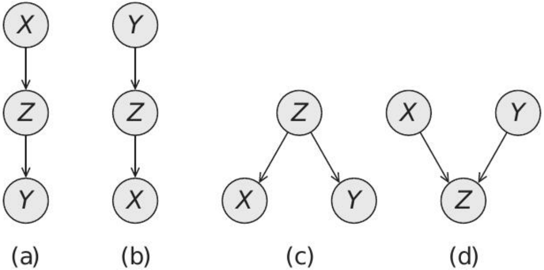
- Check if $X \perp Y \mid Z$



D-Separation

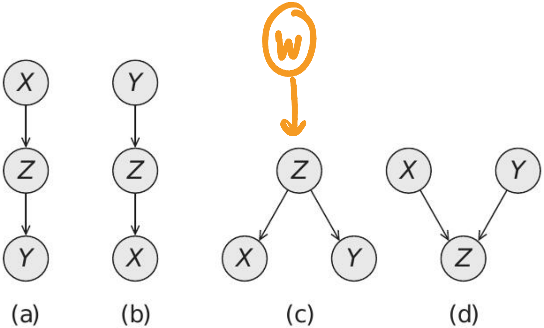
- Check if $X \perp Y \mid Z$
- Dependence should be blocked on every trail from X to Y





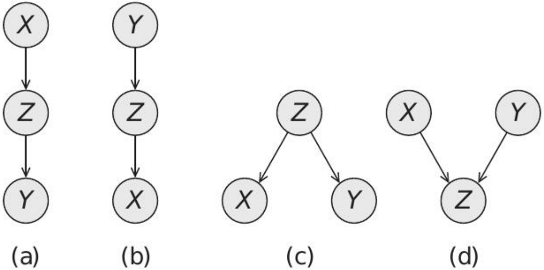
D-Separation

- Check if $X \perp Y \mid Z$
- Dependence should be blocked on every trail from X to Y
  - Each undirected path from X to Y is a sequence of basic trails



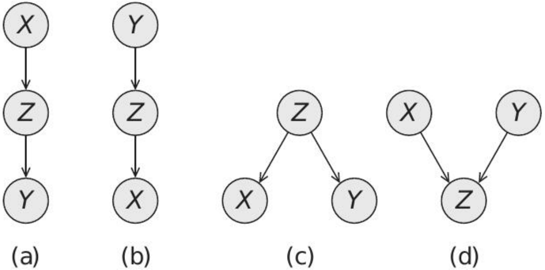
D-Separation

- Check if $X \perp Y \mid Z$
- Dependence should be blocked on every trail from X to Y
  - Each undirected path from X to Y is a sequence of basic trails
  - For (a), (b), (c), need Z present



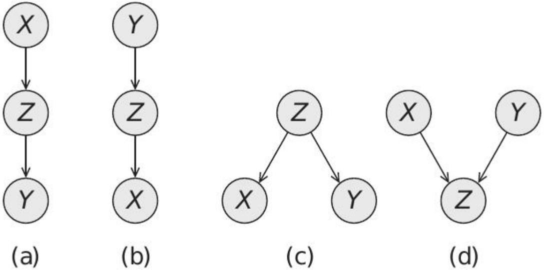
D-Separation

- Check if $X \perp Y \mid Z$
- Dependence should be blocked on every trail from X to Y
  - Each undirected path from X to Y is a sequence of basic trails
  - For (a), (b), (c), need Z present
  - For (d), need Z absent



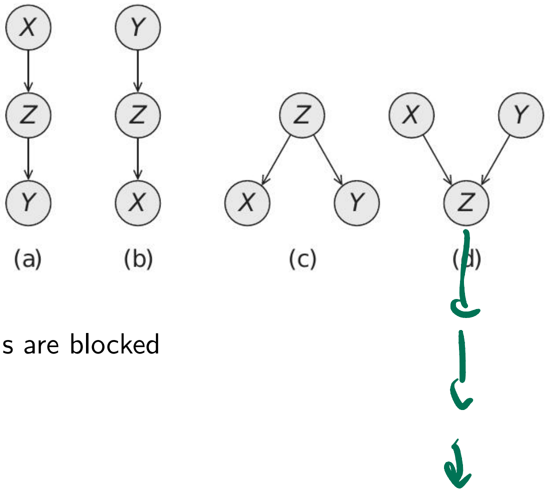
D-Separation

- Check if $X \perp Y \mid Z$
- Dependence should be blocked on every trail from X to Y
  - Each undirected path from X to Y is a sequence of basic trails
  - For (a), (b), (c), need Z present
  - For (d), need Z absent
  - In general, V-structure includes descendants of the bottom node



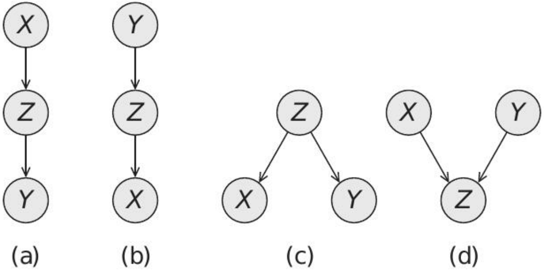
D-Separation

- Check if $X \perp Y \mid Z$
- Dependence should be blocked on every trail from X to Y
  - Each undirected path from X to Y is a sequence of basic trails
  - For (a), (b), (c), need Z present
  - For (d), need Z absent
  - In general, V-structure includes descendants of the bottom node
- x and y are D-separated given z if all trails are blocked



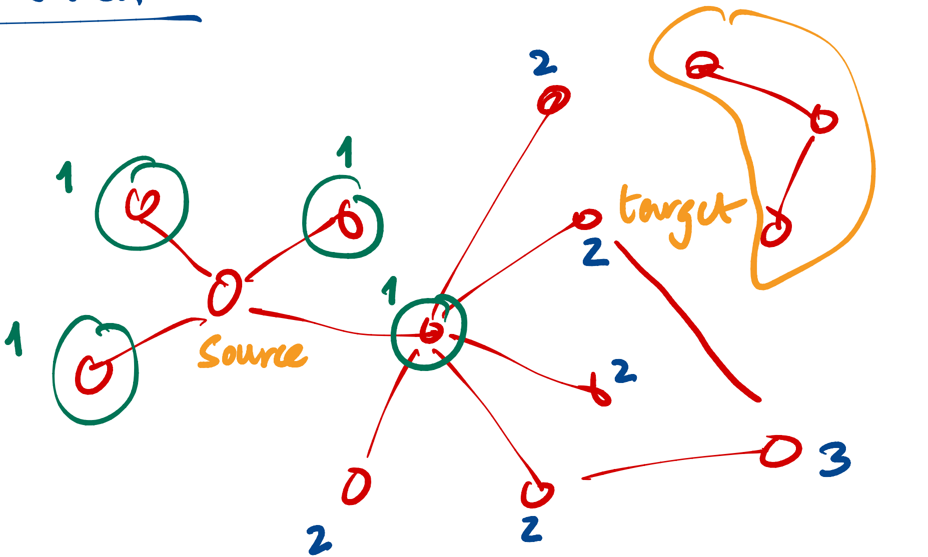
D-Separation

- Check if $X \perp Y \mid Z$
- Dependence should be blocked on every trail from X to Y
  - Each undirected path from X to Y is a sequence of basic trails
  - For (a), (b), (c), need Z present
  - For (d), need Z absent
  - In general, V-structure includes descendants of the bottom node



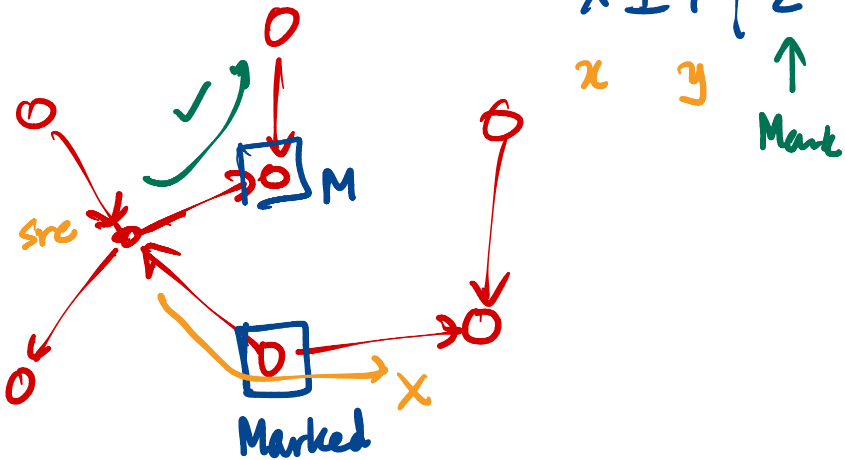
- x and y are D-separated given z if all trails are blocked
- Variation of breadth first search (BFS) to check if y is reachable from x through some trail

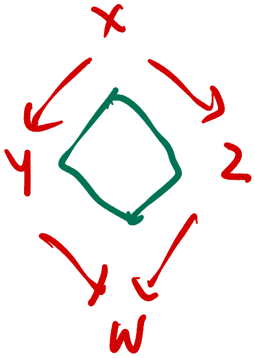
Breadth first search



BFS

$X \perp Y \mid Z$

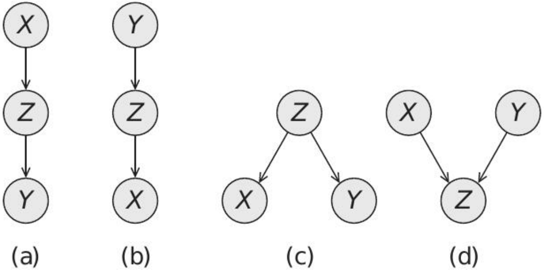




Undirected cycles are possible

D-Separation

- Check if $X \perp Y \mid Z$
- Dependence should be blocked on every trail from X to Y
  - Each undirected path from X to Y is a sequence of basic trails
  - For (a), (b), (c), need Z present
  - For (d), need Z absent
  - In general, V-structure includes descendants of the bottom node



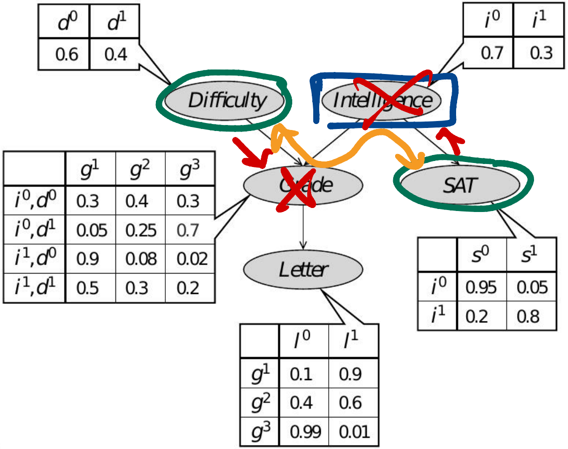
- x and y are D-separated given z if all trails are blocked

$x \perp y \mid z$

- Variation of breadth first search (BFS) to check if y is reachable from x through some trail
- Extends to sets — each $x \in X$ is D-separated from each $y \in Y$

Conditional independence, example

- Is SAT independent of Difficulty given Intelligence?
  - Yes, Difficulty – Grade – Intelligence – SAT trail is blocked at Grade (V-structure) and Intelligence



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

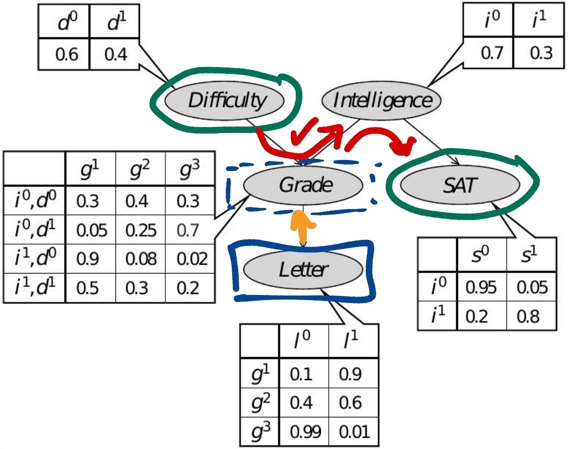
	g^1	g^2	g^3
i^0,d^0	0.3	0.4	0.3
i^0,d^1	0.05	0.25	0.7
i^1,d^0	0.9	0.08	0.02
i^1,d^1	0.5	0.3	0.2

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^3	0.99	0.01

Conditional independence, example

- Is SAT independent of Difficulty given Intelligence?
  - Yes, Difficulty – Grade – Intelligence – SAT trail is blocked at Grade (V-structure) and Intelligence
- Is SAT independent of Difficulty given Letter?
  - No, Difficulty – Grade – Intelligence – SAT trail is open
  - Letter is known, hence something about Grade is known (V-structure)
  - Intelligence is not known



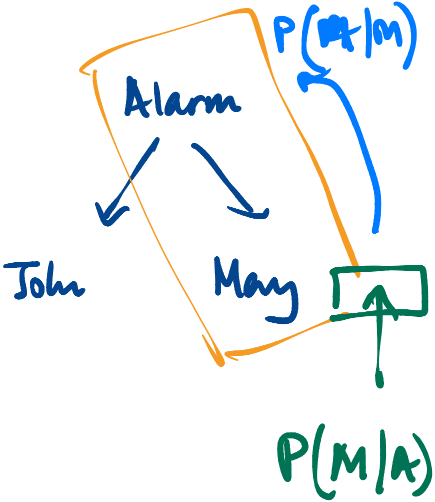
d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

	g^1	g^2	g^3
i^0,d^0	0.3	0.4	0.3
i^0,d^1	0.05	0.25	0.7
i^1,d^0	0.9	0.08	0.02
i^1,d^1	0.5	0.3	0.2

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^3	0.99	0.01



$$P(A \mid M)$$

$$P(J \mid A=1) \cdot P(A=1 \mid M)$$

$$P(J = A = 0) \quad P(A=0 \mid M)$$