Probabilistic Graphical Models

Madhavan Mukund

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

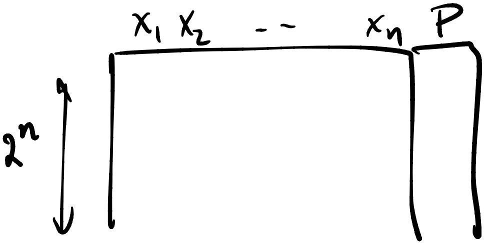
Data Mining and Machine Learning
August–December 2020

Conditional probabilities

- Boolean variables $x_1, x_2, \ldots, x_n$

Conditional probabilities

- Boolean variables $x_1, x_2, \ldots, x_n$
- Joint probabilities $P(v_1, v_2, \ldots, v_n)$
  - 2^n combinations of $x_1, x_2, \ldots, x_n$
  - $2^n - 1$ parameters



Conditional probabilities

- Boolean variables $x_1, x_2, \ldots, x_n$
- Joint probabilities $P(v_1, v_2, \ldots, v_n)$
  - 2^n combinations of $x_1, x_2, \ldots, x_n$
  - $2^n - 1$ parameters
- Naïve Bayes assumption — complete independence
  - $P(x_i = 1)$ for each x_i
  - n parameters

Conditional probabilities

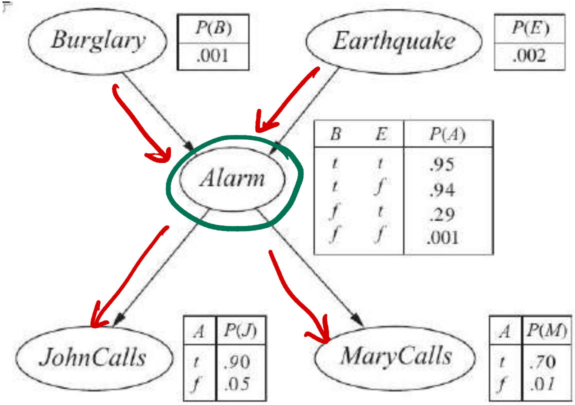
- Boolean variables $x_1, x_2, \ldots, x_n$
- Joint probabilities $P(v_1, v_2, \ldots, v_n)$
  - 2^n combinations of $x_1, x_2, \ldots, x_n$
  - $2^n - 1$ parameters
- Naïve Bayes assumption — complete independence
  - $P(x_i = 1)$ for each x_i
  - n parameters
- Can we strive for something in between?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

Probabilistic graphical models

- Judea Pearl [Turing Award 2011]
- Represent local dependencies using directed graph

Probabilistic graphical models

- Judea Pearl [Turing Award 2011]
- Represent local dependencies using directed graph
- 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!)

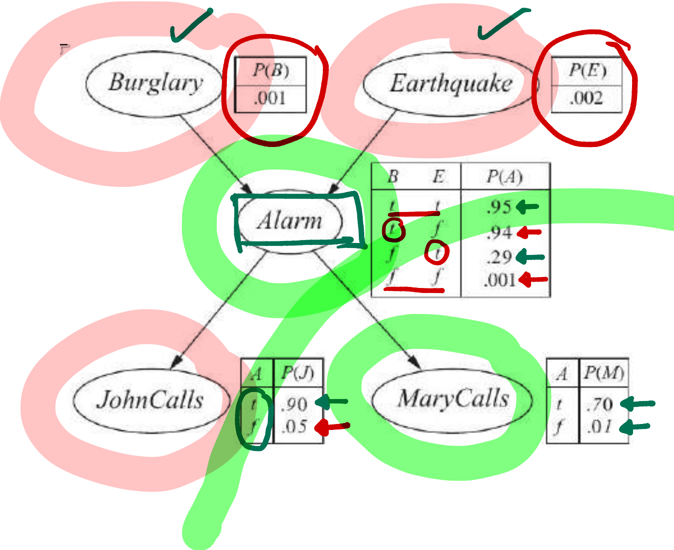


Probabilistic graphical models

- Each node has a local (conditional) probability table

$P(J \mid A)$

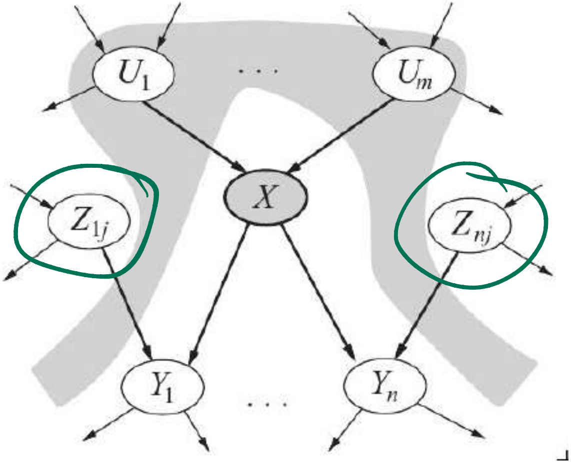
$P(A \mid J)$



Probabilistic graphical models

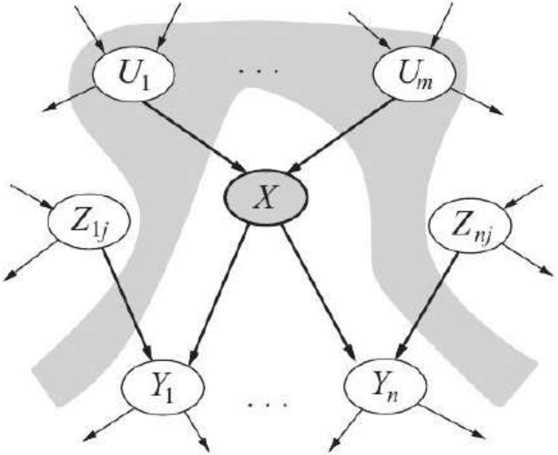
- Each node has a local (conditional) probability table
- Fundamental assumption:
  A node is conditionally independent of non-descendants, given its parents

Semantics



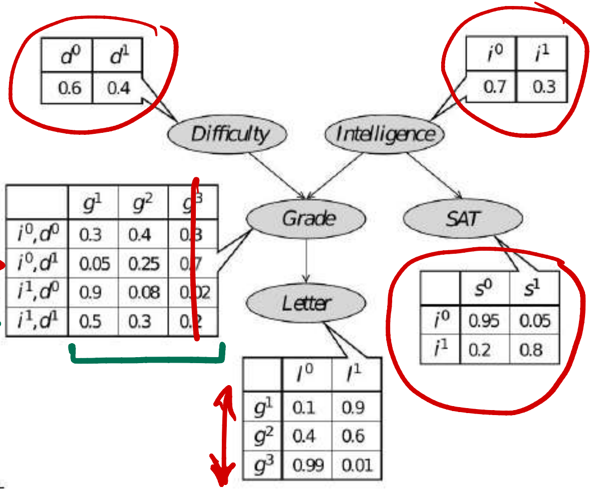
Probabilistic graphical models

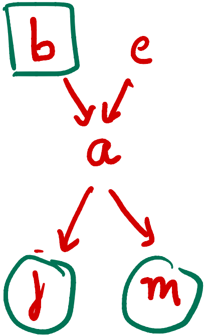
- Each node has a local (conditional) probability table
- Fundamental assumption:
  A node is conditionally independent of non-descendants, given its parents
- Graph is a DAG, no cyclic dependencies



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





- 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: $P(A, B) = P(A \mid B)P(B)$

  $P(A|B) = \frac{P(A,B)}{P(B)}$
- $P(x_1, x_2, \ldots, x_n) = P(x_1 \mid x_2, \ldots, x_n)$ $P(x_2, \ldots, x_n)$

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

Chain Rule

Evaluating a network

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

Evaluating a network

- $P(x_1, x_2, \ldots, x_n) = P(x_1 \mid x_2, \ldots, x_n) P(x_2 \mid x_3, \ldots, x_n) \cdots P(x_{n-1} \mid x_n) P(x_n)$
- Can choose any ordering of $x_1, x_2, \ldots, x_n$

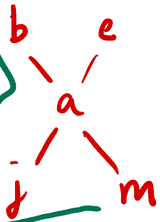
Evaluating a network

- $P(x_1, x_2, \ldots, x_n) = P(x_1 \mid x_2, \ldots, x_n) P(x_2 \mid x_3, \ldots, x_n) \cdots P(x_{n-1} \mid x_n) P(x_n)$
- Can choose any ordering of $x_1, x_2, \ldots, x_n$
- Use topological ordering in a Bayesian network

Evaluating a network

- $P(x_1, x_2, \ldots, x_n) = P(x_1 \mid x_2, \ldots, x_n)P(x_2 \mid x_3, \ldots, x_n) \cdots P(x_{n-1} \mid x_n)P(x_n)$
- Can choose any ordering of $x_1, x_2, \ldots, x_n$
- Use topological ordering in a Bayesian network
- $P(m, j, a, b, e) = \underline{P(m \mid a)}\underline{P(j \mid a)}\underline{P(a \mid b, e)}\underline{P(b)}P(e)$

$$\underline{P(m \mid \cancel{j}, a, \cancel{b}, \cancel{e})} \cdot \underline{P(j \mid a}, \cancel{b}, \cancel{e}) \cdot \underline{P(a \mid b, e)} \cdot \underline{P(b \mid \cancel{e})} \cdot \underline{P(e)}$$



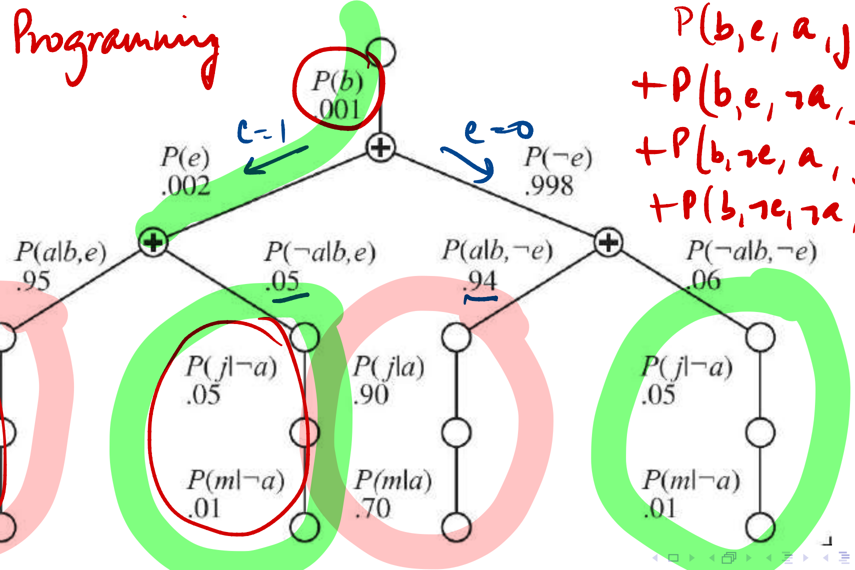
Evaluating a network

- $P(x_1, x_2, \ldots, x_n) = P(x_1 \mid x_2, \ldots, x_n) P(x_2 \mid x_3, \ldots, x_n) \cdots P(x_{n-1} \mid x_n) P(x_n)$
- Can choose any ordering of $x_1, x_2, \ldots, x_n$
- Use topological ordering in a Bayesian network
- $P(m, j, a, b, e) = P(m \mid a) P(j \mid a) P(a \mid b, e) P(b) P(e)$
- $P(m, j, b) = \sum_{a=0}^{1} \sum_{e=0}^{1} P(m \mid a) P(j \mid a) P(a \mid b, e) P(b) \underline{P(e)}$

Evaluating a network

- $P(x_1, x_2, \ldots, x_n) = P(x_1 \mid x_2, \ldots, x_n) P(x_2 \mid x_3, \ldots, x_n) \cdots P(x_{n-1} \mid x_n) P(x_n)$
- Can choose any ordering of $x_1, x_2, \ldots, x_n$
- Use topological ordering in a Bayesian network
- $P(m, j, a, b, e) = P(m \mid a) P(j \mid a) P(a \mid b, e) P(b) P(e)$
- $P(m, j, b) = \sum_{a=0}^{1} \sum_{e=0}^{1} P(m \mid a) P(j \mid a) P(a \mid b, e) P(b) P(e)$
- $P(m, j, b) = P(b) \sum_{e=0}^{1} P(e) \sum_{a=0}^{1} P(m \mid a) P(j \mid a) \underline{P(a \mid b, e)}$

Dynamic Programming



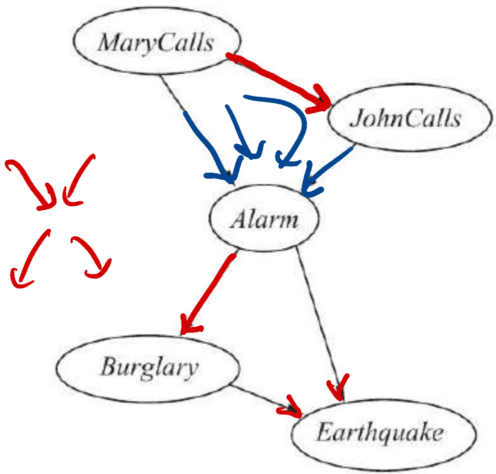
$P(b, e, a, j, m)$
$+ P(b, e, \neg a, j, m)$
$+ P(b, \neg e, a, j, m)$
$+ P(b, \neg e, \neg a, j, m)$

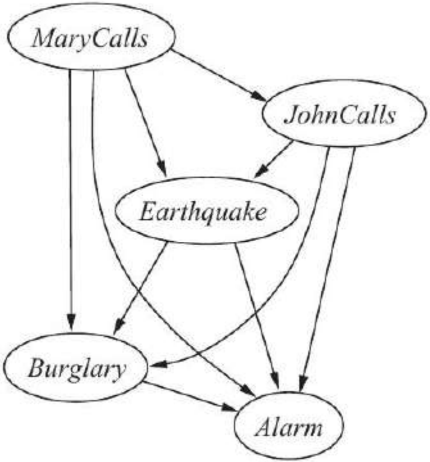
Exact inference

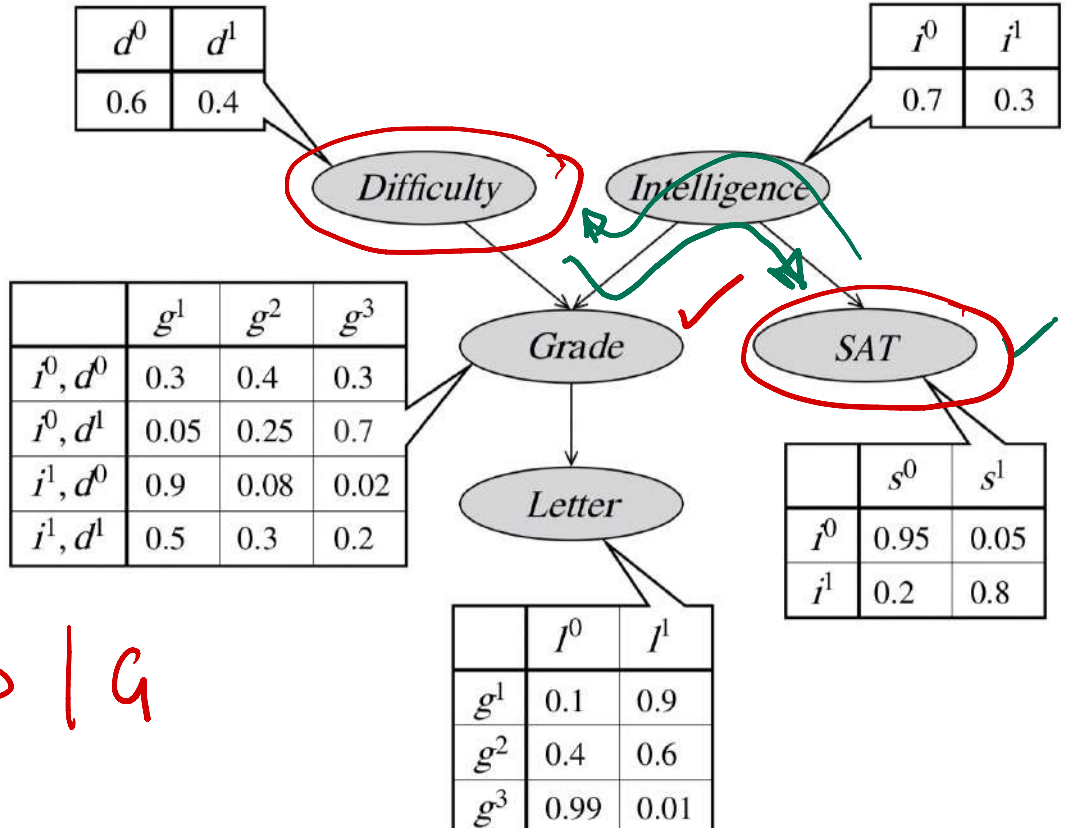
$P(x, y, z)$ — #P complete

Alternative networks

m, j, a, b, e







d^0	d^1
0.6	0.4

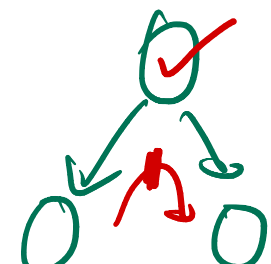
i^0	i^1
0.7	0.3

	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

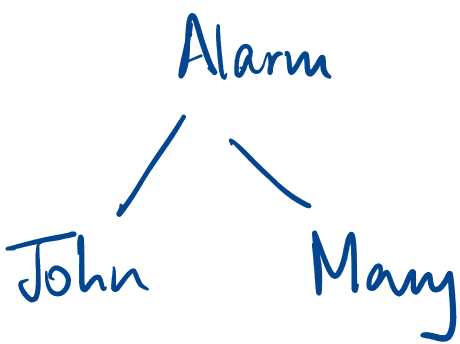
Is $S \perp D \mid G$



Conditional Independence

$X \perp Y$ – X & Y are independent

$X \perp Y \mid Z$ – X, Y independent if Z is known

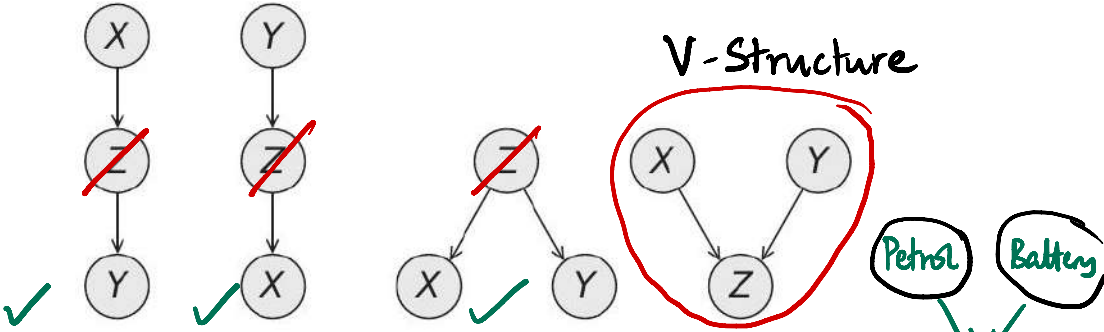


$J \perp M$ ✗

$J \perp M \mid A$

Basic trails

$X \perp Y \mid Z$?



(a) (b) (c) (d)

From BN semantics

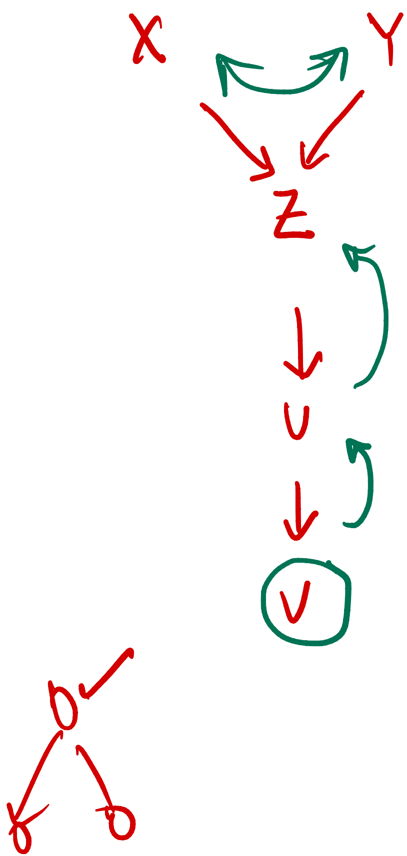
Knowing Z creates info flow

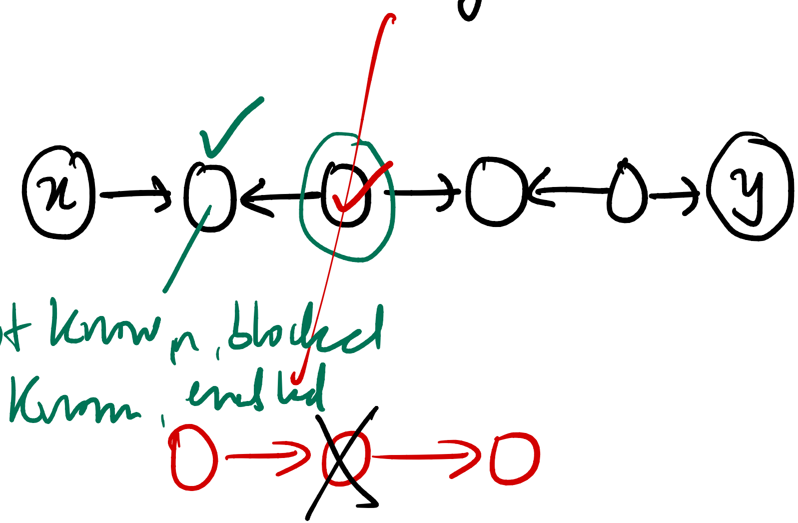
In general,

sets of nodes X, Y, Z

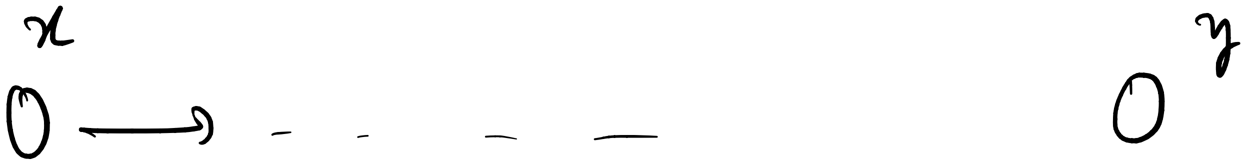
Want $X \perp Y \mid Z$?

$x \qquad y$

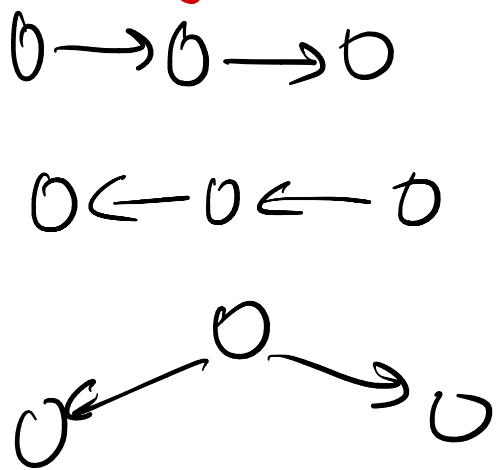


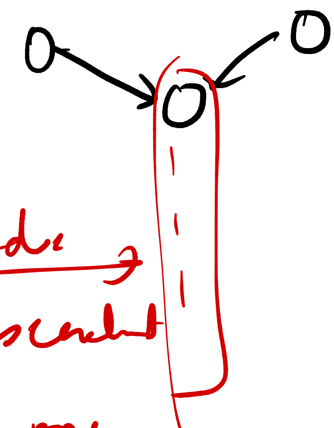


"Trails"



If in Z, blocked





If node or descendant then open else blocked

Active Trail - allows info to flow

$X \perp Y \mid Z$ - check all trails from X to Y are inactive

Clever way, using BFS

PGM structure

- DAG
- Semantics

Evaluate - tree, dynamic programming

Conditional Independence via active trails