CS 161 Intro. To Artificial Intelligence

Week 8, Discussion 1D

Conditional Probability

Conditional Probability:

$$P(a|b) = \frac{P(a \wedge b)}{P(b)} \text{ if } P(b) \neq 0$$

- Product rule:

$$P(a \wedge b) = P(a|b)P(b) = P(b|a)P(a)$$

- Chain Rule (general product rule):

$$\begin{aligned} \mathbf{P}(X_1,\ldots,X_n) &= \mathbf{P}(X_1,\ldots,X_{n-1})\ \mathbf{P}(X_n|X_1,\ldots,X_{n-1}) \\ &= \mathbf{P}(X_1,\ldots,X_{n-2})\ \mathbf{P}(X_{n-1}|X_1,\ldots,X_{n-2})\ \mathbf{P}(X_n|X_1,\ldots,X_{n-1}) \\ &= \ldots \\ &= \prod_{i=1}^{n} \mathbf{P}(X_i|X_1,\ldots,X_{i-1}) \end{aligned}$$

← Note that:
$P(X_1,\ldots,X_n) = P(X_1 \wedge X_2 \wedge \cdots \wedge X_n)$

Independence

Independence:

- Independence: $A \perp B$
  - P(A|B) = P(A), or P(B|A) = P(B), or P(A,B) = P(A)P(B)
  - E.g. $$\mathbf{P}(Toothache, Catch, Cavity, Weather) = \mathbf{P}(Toothache, Catch, Cavity)\mathbf{P}(Weather)$$
- Conditional independence: $A \perp B|C$
  - P(A,B|C) = P(A|C)P(B|C), or P(A|B,C) = P(A|C)
  - This indicates: A and B are independent when the value of C is known and fixed
  - E.g. $Catch$ is **conditionally independent** of $Toothache$ given $Cavity$: $$\mathbf{P}(Catch|Toothache, Cavity) = \mathbf{P}(Catch|Cavity)$$ $$\begin{aligned}&\mathbf{P}(Toothache, Catch, Cavity)\\ &= \mathbf{P}(Toothache|Catch, Cavity)\mathbf{P}(Catch, Cavity)\\ &= \mathbf{P}(Toothache|Catch, Cavity)\mathbf{P}(Catch|Cavity)\mathbf{P}(Cavity)\\ &= \mathbf{P}(Toothache|Cavity)\mathbf{P}(Catch|Cavity)\mathbf{P}(Cavity)\end{aligned}$$

Probability Inference – Bayes Rule

Probability inference: inference the probability of one event

How to do it?

- Inference by enumeration
- Inference rule:
    - Bayes' Rule:

Bayes' Rule can be used in probablity inference when we have $P(b|a)$ but not $P(a|b)$.

Product rule $P(a \wedge b) = P(a|b)P(b) = P(b|a)P(a)$

$$\Rightarrow \textbf{Bayes' rule } P(a|b) = \frac{P(b|a)P(a)}{P(b)}$$

or in distribution form

$$\mathbf{P}(Y|X) = \frac{\mathbf{P}(X|Y)\mathbf{P}(Y)}{\mathbf{P}(X)} = \alpha\mathbf{P}(X|Y)\mathbf{P}(Y)$$

Bayes Rule

Useful for assessing **diagnostic** probability from **causal** probability:

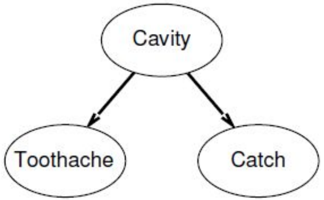
$$P(Cause|Effect) = \frac{P(Effect|Cause)P(Cause)}{P(Effect)}$$

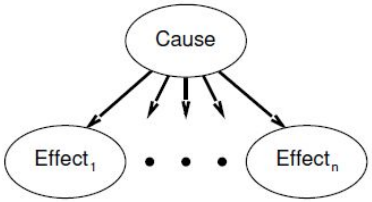
- **Naïve Bayes** model:

$$\mathbf{P}(Cause, Effect_1, \ldots, Effect_n) = \mathbf{P}(Cause)\prod_i \mathbf{P}(Effect_i|Cause)$$

Example:

$$\begin{aligned}&\mathbf{P}(Cavity|toothache \wedge catch)\\ =\;& \alpha\, \mathbf{P}(toothache \wedge catch|Cavity)\mathbf{P}(Cavity)\\ =\;& \alpha\, \mathbf{P}(toothache|Cavity)\mathbf{P}(catch|Cavity)\mathbf{P}(Cavity)\end{aligned}$$





Bayesian Network (BN) – Representation

Goal: Represent joint probability over a set of random variables

- Facilitate probability computation

Component:

- Graph Structure: a Directed Acyclic Graph (DAG)
  - Nodes: random variables (events)
  - Edges: $y \rightarrow x$ means y causes/influences x
- Local Probability Model
  - Represent the dependence of each variable on its parents
  - $y_1, y_2, \ldots, y_k \rightarrow x$: conditional probability $p(x \mid y_1, y_2, \ldots, y_k)$
  - Root variables: marginal probability

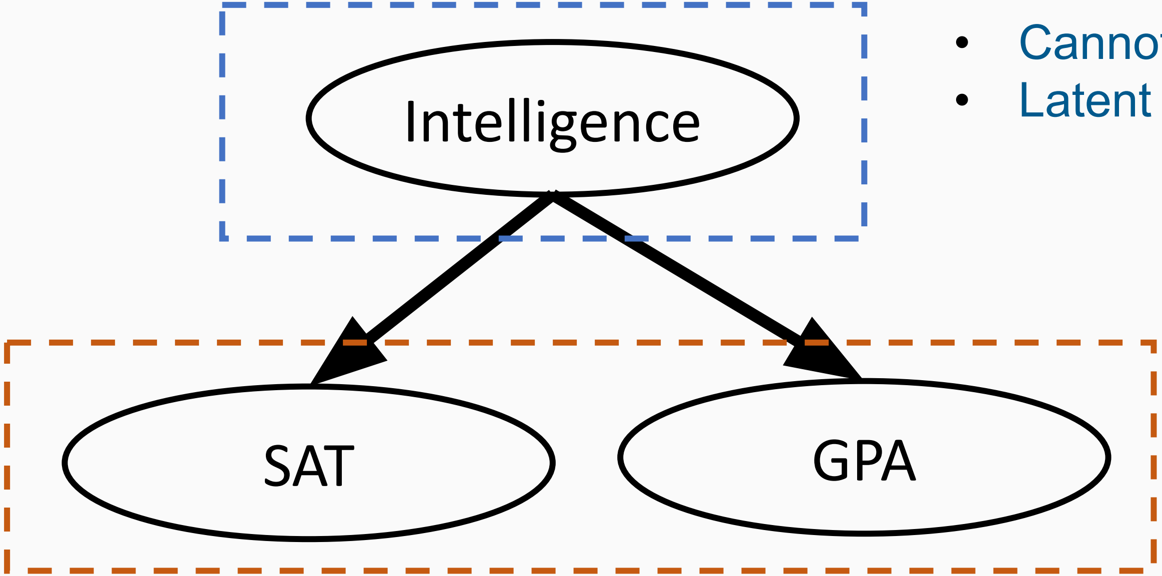
BN – Graph Structure

Scenario: A company wants to hire an intelligent student.

- But intelligence cannot be directly measured.
- But the company may have access to the student's SAT and GPA score.

- Based on the observable evidences (SAT and GPA), company can try to infer whether this student is intelligent or not.

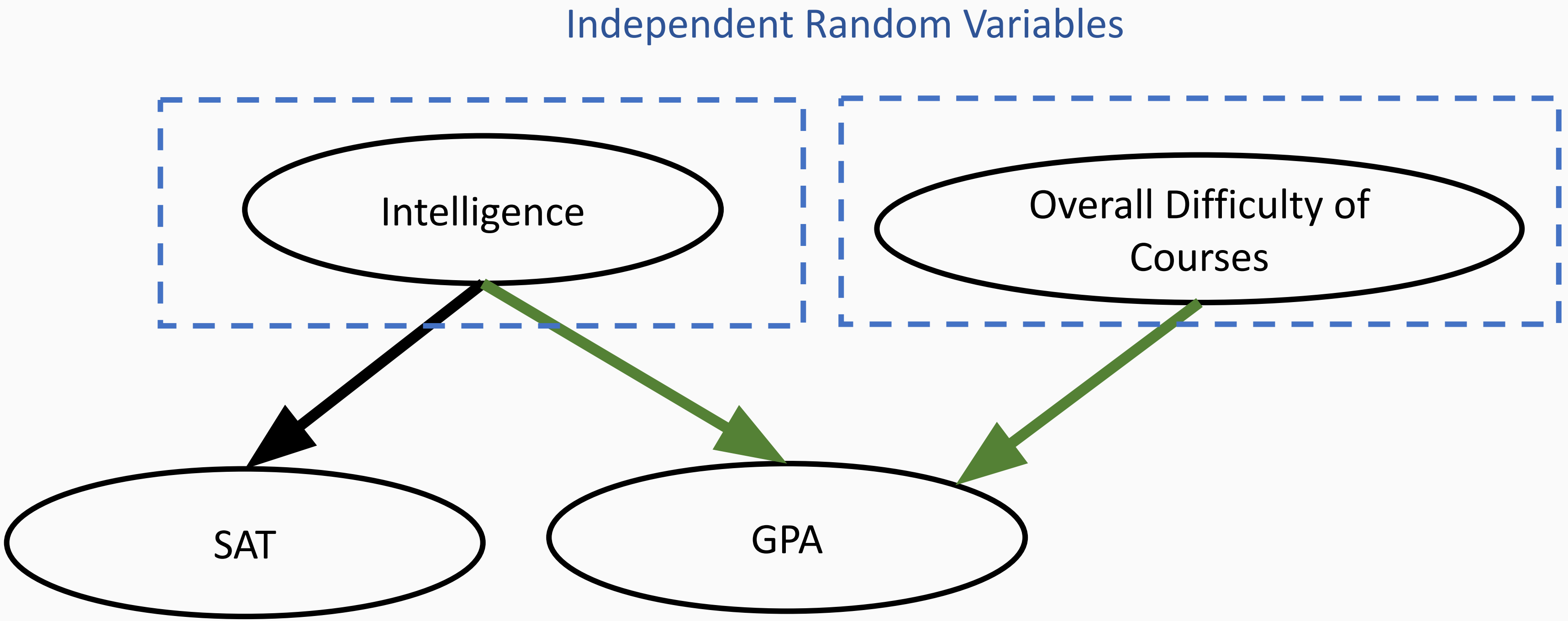
BN – Graph Structure



- Cannot be directly measured
- Latent factor of its children

Can be directly observed

BN – Graph Structure



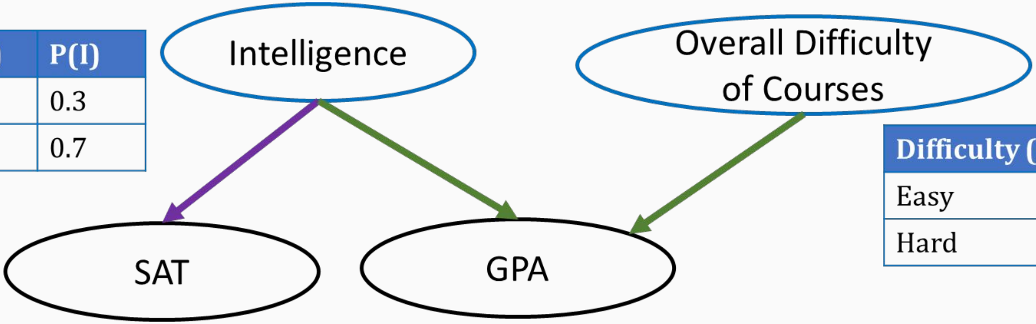
P(GPA | Intelligence, Difficulty of Courses)

BN– Full Representation

Components of BN:

- Directed, acyclic graph (**DAG**)
- Conditional distribution for each node given its parents (e.g. $\mathbf{P}(X_i|Parents(X_i))$
  - Often represented as conditional probability tables (**CPTs**)

Intelligence (I)	P(I)
True	0.3
False	0.7



Difficulty (D)	P(D)
Easy	0.6
Hard	0.4

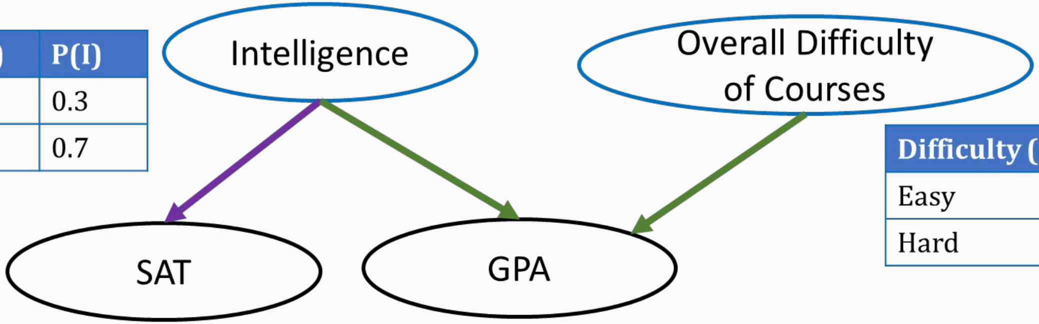
I	SAT	P(SAT|I)
True	High	0.7
True	Low	0.3
False	High	0.4
False	Low	0.6

I	D	GPA	P(GPA|I,D)
True	Easy	High	0.8
True	Easy	Low	0.2
True	Hard	High	0.6
...	...	...	...

BN - Independence

- Independence: I(Intelligence, ∅, Difficulty)
- Parents: GPA – Intelligence, Difficulty; SAT – Intelligence
- Descendants: Intelligence – SAT, GPA; Difficulty – GPA
- Non-descendants: nodes that are **not descendant nor parents.** E.g. non-descendant of GPA is SAT

Intelligence (I)	P(I)
True	0.3
False	0.7



Difficulty (D)	P(D)
Easy	0.6
Hard	0.4

I	SAT	P(SAT|I)
True	High	0.7
True	Low	0.3
False	High	0.4
False	Low	0.6

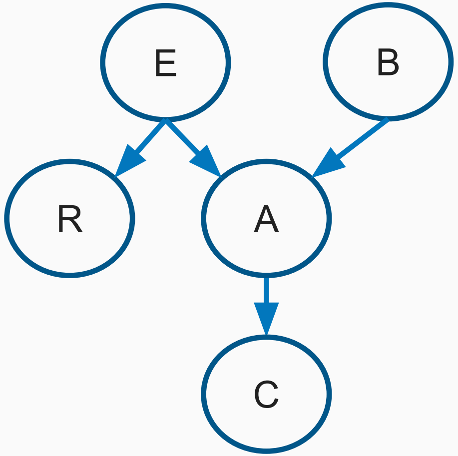
I	D	GPA	P(GPA|I,D)
True	Easy	High	0.8
True	Easy	Low	0.2
True	Hard	High	0.6
…	…	…	…

Markovian Assumption

BN satisfies **local Markov property**:

- A node is conditionally independent of its non-descendants given its parents.
- It can be represented as: **I (V, Parents(V), Non-Descendants(V))**

- E.g.
  - I (C, A, B E R)
  - I (R, E, A B C)
  - I (A, B E, R)
  - I (B, ø, E R)
  - I (E, ø, B)



Joint Probability

BN models the following **joint probability**:

$$P(X_1, X_2, \dots, X_N) = \prod_{i=1}^{N} P(X_i | \text{parents of } X_i)$$

Reason:

- Without loss of generality, assume $X_1, X_2, \dots, X_N$ is a topological ordering
- **Chain rule**:

$$P(X_1, X_2, \dots, X_N) = \prod_{i=1}^{N} P(X_i | X_1, X_2, \dots, X_{i-1})$$

- $P(X_i | X_1, X_2, \dots, X_n) = P(X_i | \text{parents of } X_i)$
    - Topological ordering => parents are in $X_1, X_2, \dots, X_{i-1}$
    - **Markovian assumption** => given parents, a variable is X_i is independent of other variables in $X_1, X_2, \dots, X_{i-1}$

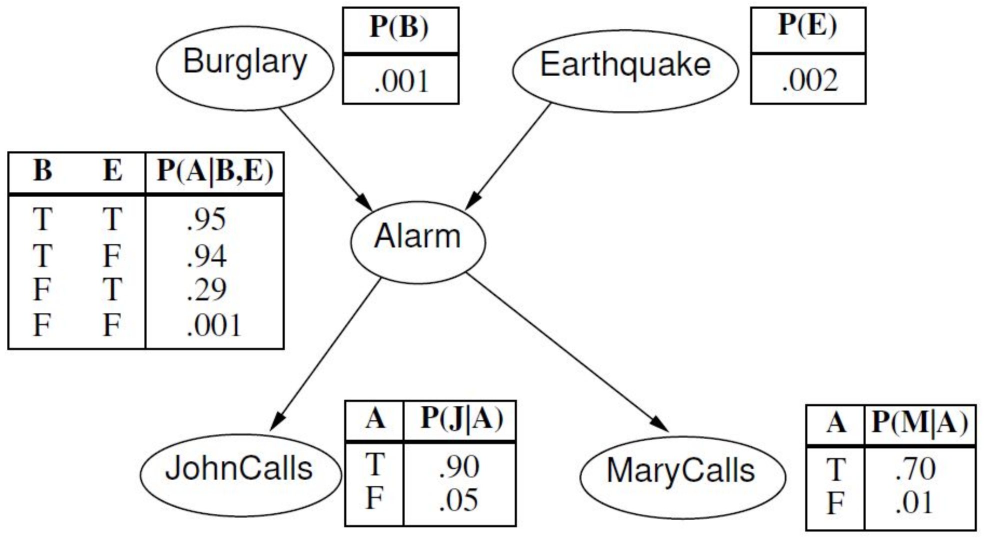
Joint Probability - Example

I'm at work, neighbor John calls to say my alarm is ringing, but neighbor Mary doesn't call. Sometimes it's set off by minor earthquakes.

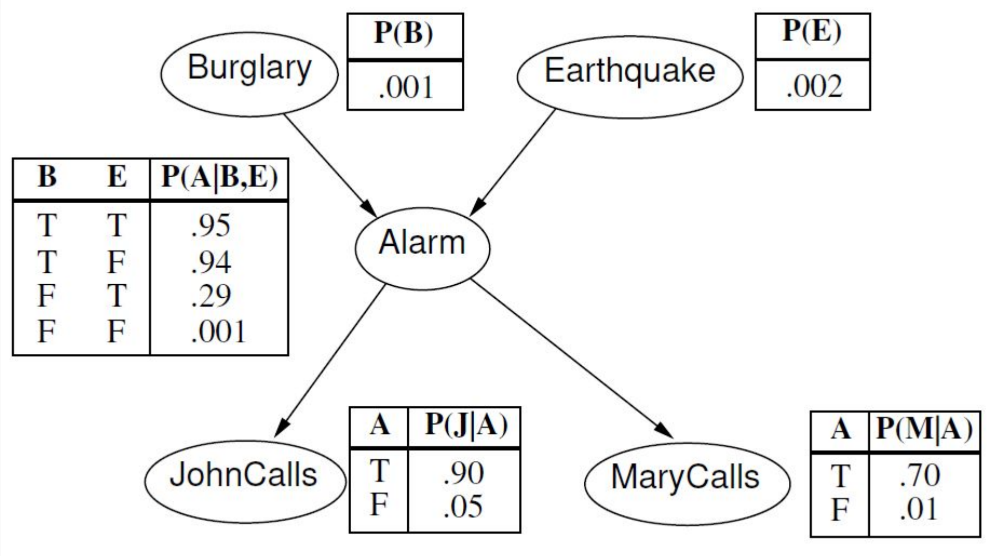
Variables: $Burglar$, $Earthquake$, $Alarm$, $JohnCalls$, $MaryCalls$

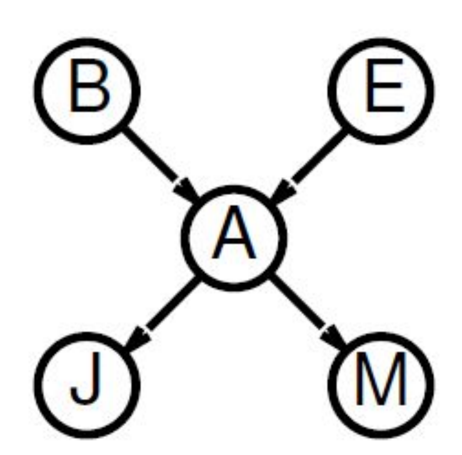
Network topology reflects "causal" knowledge:

- A burglar can set the alarm off
- An earthquake can set the alarm off
- The alarm can cause Mary to call
- The alarm can cause John to call



Joint Probability - Example





e.g., $P(j \land m \land a \land \neg b \land \neg e)$

$$\begin{aligned} &= P(j|a)P(m|a)P(a|\neg b, \neg e)P(\neg b)P(\neg e) \\ &= 0.9 \times 0.7 \times 0.001 \times 0.999 \times 0.998 \\ &\approx 0.00063 \end{aligned}$$

D-separation

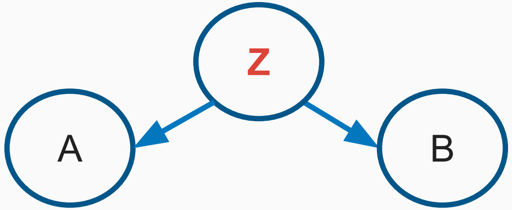
D-separation is a graphical test of independence: **I (A, Z, B)** if A and B are **d-separated** given Z.

- **dsep(A, Z, B)** iff every path between a node in A and a node in B is blocked by Z
- A path is **blocked** by Z iff at least one valve on the path is <u>**closed**</u> given Z

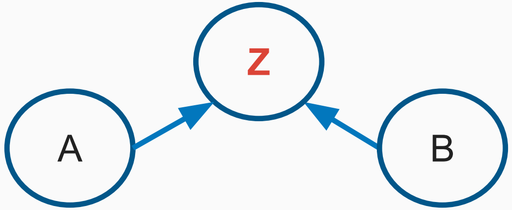
Three types of valve:

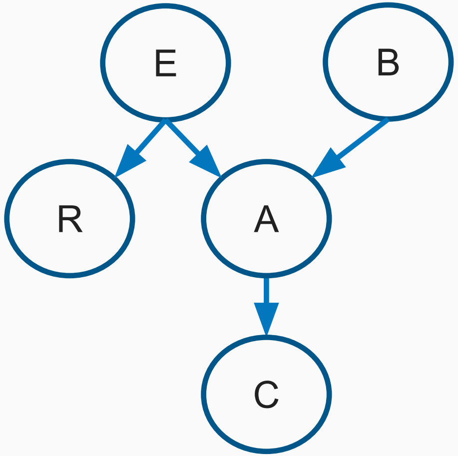
- Sequential:

- Divergent:



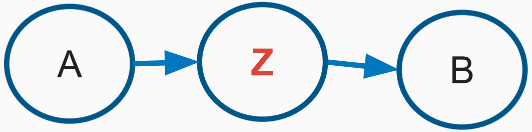
- Convergent:



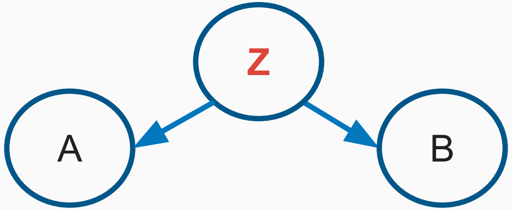


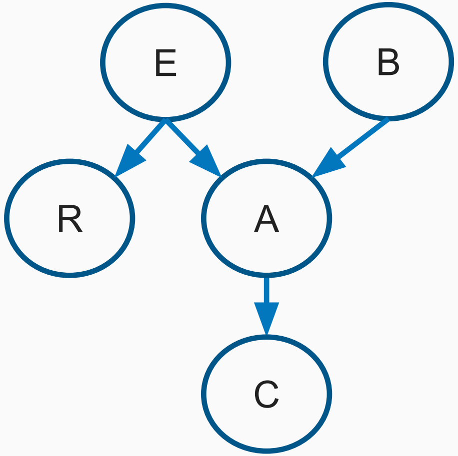
D-separation

- A sequential valve is **closed** iff **variable Z is known**

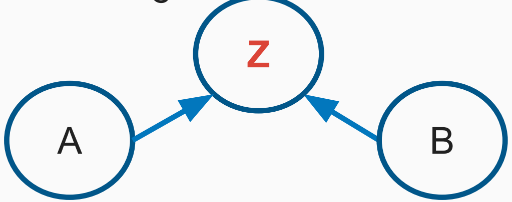


- A divergent valve is **closed** iff **variable Z is known**





- A convergent valve is **closed** iff **neither variable Z nor any of its descendant is known**



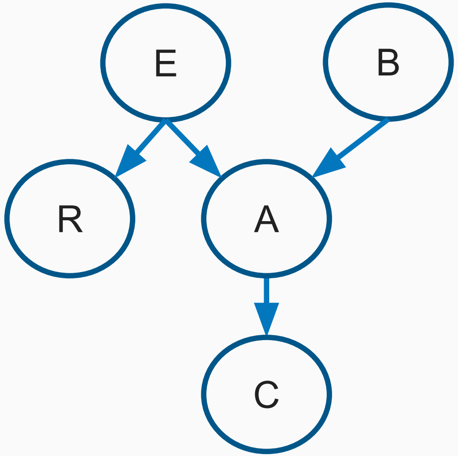
D-separation - Example

Example:

Z = {E, C} → known

Q: dsep(B, E C, R)? → Do we have I(B, E C, R)?

Q: dsep(E, B C, R)? → Do we have I(E, B C, R)?



D-separation - Example

Example:

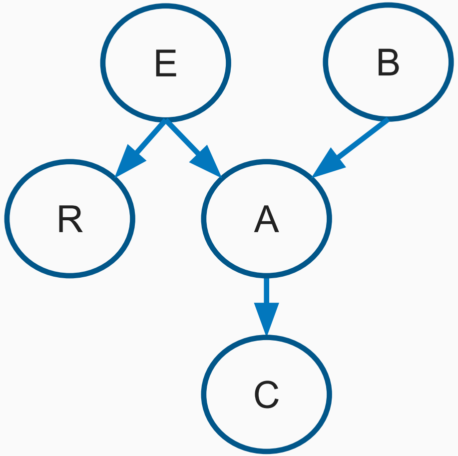
Z = {E, C} known

Q: dsep(B, E C, R)? Do we have I(B, E C, R)?

- E closes the only path

Q: dsep(E, B C, R)? Do we have I(E, B C, R)?

- C opens the only path



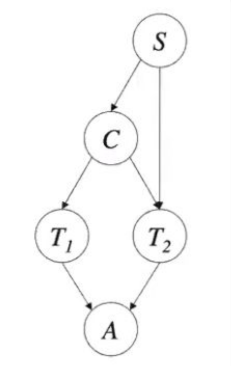
BN Inference - Queries

- Prior marginal: for any variable, compute its distribution
  - E.g.

C = yes	3.2%
C = No	96.8%

☐ This basically means if pick up a person from population, its prob. of getting disease C

  - Can be computed from joint prob. table Pr(S, C, T1, T2, A)



S	θ_s
male	.55
female	.45

S	C	$\theta_{c\|s}$
male	yes	.05
male	no	.95
female	yes	.01
female	no	.99

C	T_1	$\theta_{t_1\|c}$
yes	+ve	.80
yes	−ve	.20
no	+ve	.20
no	−ve	.80

S	C	T_2	$\theta_{t_2\|c,s}$
male	yes	+ve	.80
male	yes	−ve	.20
male	no	+ve	.20
male	no	−ve	.80
female	yes	+ve	.95
female	yes	−ve	.05
female	no	+ve	.05
female	no	−ve	.95

T_1	T_2	A	$\theta_{a\|t_1,t_2}$
+ve	+ve	yes	1
+ve	+ve	no	0
+ve	−ve	yes	0
+ve	−ve	no	1
−ve	+ve	yes	0
−ve	+ve	no	1
−ve	−ve	yes	1
−ve	−ve	no	0

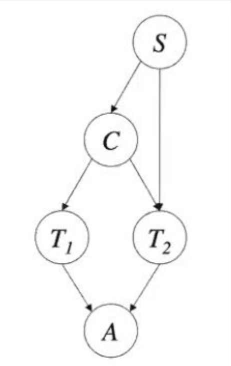
- Posterior marginal: given evidence, compute a variable's distribution
  - E.g. Given β: {T1 = +ve, T2 = +ve}

C = yes	45.3%
C = No	54.7%

  - Can be computed from joint prob. table Pr(S, C, A| β)

BN Inference - Queries

- Most probable explanation (MPE):
  - Find the most probable instantiation of **all remaining vars.** given evidence
  - E.g. Given A = Yes, what's the most probable instantiation of S, C, T1, T2?
    - In this scenario, we can find Pr(C=No, S=female, T1=-ve, T2=-ve) ~ 47%



- Maximum a posteriori hypothesis (MAP):
  - Similar to MPE, but only find the most probable instantiation of **a set of vars.** given evidence
  - E.g. Given A = Yes, what's the most likely states of S and C?
    - We can find Pr(C=No, S=male) ~ 49.3%
  - MAP is a general case of MPE, an algorithm solving MAP can work for MPE too if we pass in all remaining vars.

S	θ_s
male	.55
female	.45

S	C	$\theta_{c\|s}$
male	yes	.05
male	no	.95
female	yes	.01
female	no	.99

C	T_1	$\theta_{t_1\|c}$
yes	+ve	.80
yes	−ve	.20
no	+ve	.20
no	−ve	.80

S	C	T_2	$\theta_{t_2\|c,s}$
male	yes	+ve	.80
male	yes	−ve	.20
male	no	+ve	.20
male	no	−ve	.80
female	yes	+ve	.95
female	yes	−ve	.05
female	no	+ve	.05
female	no	−ve	.95

T_1	T_2	A	$\theta_{a\|t_1,t_2}$
+ve	+ve	yes	1
+ve	+ve	no	0
+ve	−ve	yes	0
+ve	−ve	no	1
−ve	+ve	yes	0
−ve	+ve	no	1
−ve	−ve	yes	1
−ve	−ve	no	0

BN Inference - Algorithms

Goal: Compute answers to queries without constructing joint probability tables, otherwise it's not practical.

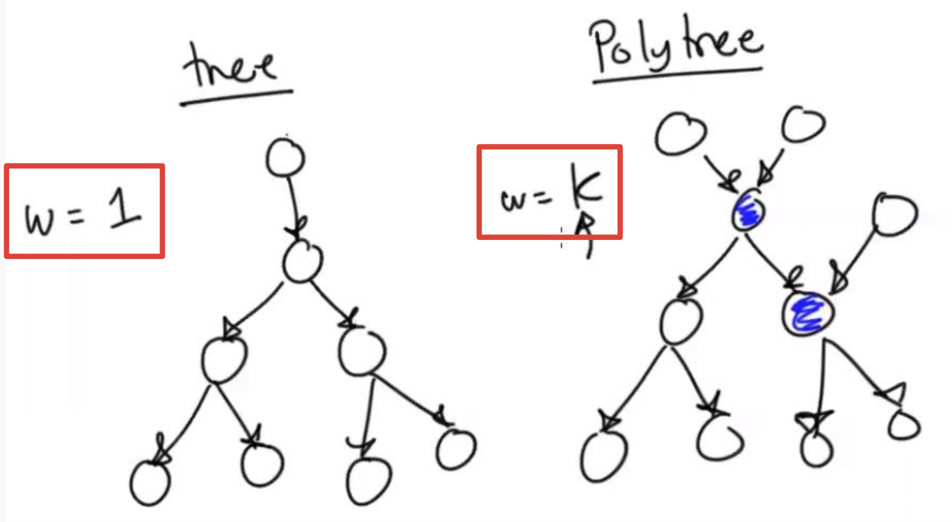
- Two categories of algorithms:
  - Variable elimination
  - Conditioning

- Complexity of algorithms depend on the topology of the graph:
  - n: # of variables; d: # of values (e.g. d=2 for binary values); w: treewidth
  - Complexity of answering prior/posterior marginals = $\boldsymbol{O}(\boldsymbol{n} \cdot \boldsymbol{d}^{\boldsymbol{w}})$

BN Inference - Algorithms

- Complexity of algorithms depend on the topology of the graph:
  - n: # of variables; d: # of values (e.g. d=2 for binary values); w: treewidth
  - Complexity of answering prior/posterior marginals = $\boldsymbol{O}(\boldsymbol{n} \cdot \boldsymbol{d}^{\boldsymbol{w}})$

- Treewidth of 2 special types of network:



- **k**: maximum # of parents a node can have

- Polytree is also called **singly-connected network**
  - multiple parents per node
  - underlying undirected tree

- (General) DAG is also called **multiply-connected network** (more general Bayesian network)

Weighted Model Counting

- **Weighted Model Counting** (WMC):
  - If a world satisfy a formula, we say it's a **model** for the formula
  - Model counting is counting the satisfiable models for a formula
    - E.g. Δ = (A v B) ^ -C.
    - We then have: SAT □ Yes/No, #SAT □ 3 models
  - WMC is counting models according to their given weights
    - The weight of a world/row is the product of weights assigned to its literals
    - E.g. WMC: 0.04 + 0.1 + 0.00 = 0.1
  - If we compile the formula to a smooth, decomposable and deterministic NNF circuit, we can do WMC in linear time!

A	B	C	↓ weights
t	t	t	.08
t	t	f	.04
t	f	t	.10
t	f	f	.10
f	t	t	.00
f	t	f	.00
f	f	t	.42
f	f	f	.06

Reducing Probabilistic Inference to WMC

$$\Delta = \begin{cases} A \Leftrightarrow P_1 \\ \neg A \Leftrightarrow P_2 \\ A \wedge B \Leftrightarrow P_3 \\ A \wedge \neg B \Leftrightarrow P_4 \\ \neg A \wedge B \Leftrightarrow P_5 \\ \neg A \wedge \neg B \Leftrightarrow P_6 \\ A \wedge C \Leftrightarrow P_7 \\ A \wedge \neg C \Leftrightarrow P_8 \\ \neg A \wedge C \Leftrightarrow P_9 \\ \neg A \wedge \neg C \Leftrightarrow P_{10} \end{cases}$$

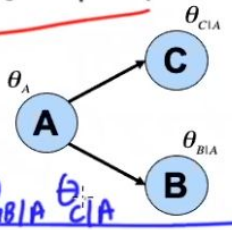
weights

* one model of Δ

$A \; \neg B \; C \; P_1 \; P_4 \; P_7 \; \neg P_2 \; \neg P_3 \; \neg P_5 \; \neg P_6 \; \neg P_8 \; \neg P_9 \; \neg P_{10}$

$w(A)\,w(\neg B)\,w(C)\,w(P_1) \ldots$

$w(\text{model}) = w(P_1)\,w(P_4)\,w(P_7) = \theta_A\,\theta_{\neg B|A}\,\theta_{C|A}$



A	B	C	Pr(.)
T	T	T	$\theta_A \theta_{B\mid A} \theta_{C\mid A}$
T	T	F	$\theta_A \theta_{B\mid A} \theta_{\neg C\mid A}$
T	F	T	$\theta_A \theta_{\neg B\mid A} \theta_{C\mid A}$
T	F	F	$\theta_A \theta_{\neg B\mid A} \theta_{\neg C\mid A}$
F	T	T	$\theta_{\neg A} \theta_{B\mid \neg A} \theta_{C\mid \neg A}$
F	T	F	$\theta_{\neg A} \theta_{B\mid \neg A} \theta_{\neg C\mid \neg A}$
F	F	T	$\theta_{\neg A} \theta_{\neg B\mid \neg A} \theta_{C\mid \neg A}$
F	F	F	$\theta_{\neg A} \theta_{\neg B\mid \neg A} \theta_{\neg C\mid \neg A}$

8 world

* weights of literals

- $w(A) = w(\neg A) = w(B) = w(\neg B) = w(C) = w(\neg C) = 1$
- $w(P_i) = \theta_i$ number
  - $w(P_1) = \theta_A$
  - $w(P_4) = \theta_{\neg B|A}$
  - $w(P_7) = \theta_{C|A}$
- $w(\neg P_i) = 1$

$w(\neg P_i) = 1$

A, B, C — Boolean variables

P_1	θ_A
P_2	$\theta_{\neg A}$
P_3	$\theta_{B\mid A}$
P_4	$\theta_{\neg B\mid A}$
P_5	$\theta_{B\mid \neg A}$
P_6	$\theta_{\neg B\mid \neg A}$
P_7	$\theta_{C\mid A}$
P_8	$\theta_{\neg C\mid A}$
P_9	$\theta_{C\mid \neg A}$
P_{10}	$\theta_{\neg C\mid \neg A}$

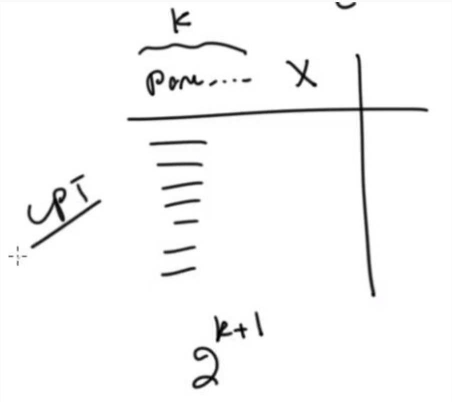
Parameters

BN Modeling

Compactness of BN:

If we have a BN with following parameters:

- n -- variables, k – max # parents per node, d – max # values per variable
- Complexity is $O(n \cdot d^{k+1})$ → size of BN



Size of BN is much smaller compared with that of joint probability table, which is $O(d^n)$.

BN Modeling

Steps of modeling BN:

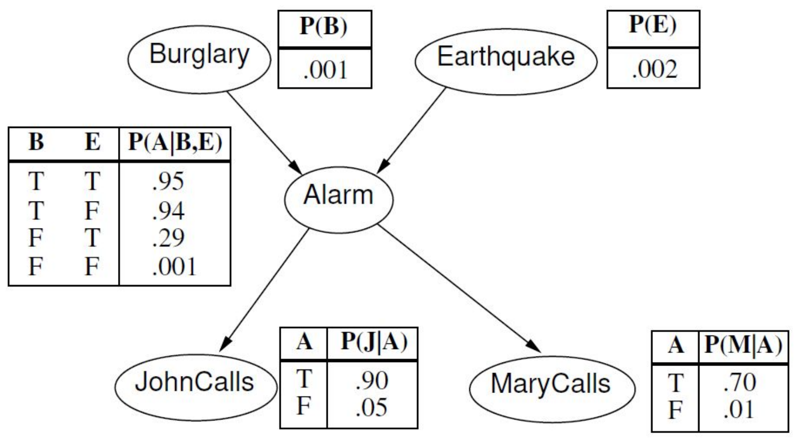
- Variables and values
- Edge
- CPTs

I'm at work, neighbor John calls to say my alarm is ringing, but neighbor Mary doesn't call. Sometimes it's set off by minor earthquakes.

Variables: $Burglar$, $Earthquake$, $Alarm$, $JohnCalls$, $MaryCalls$

Network topology reflects "causal" knowledge:

- A burglar can set the alarm off
- An earthquake can set the alarm off
- The alarm can cause Mary to call
- The alarm can cause John to call

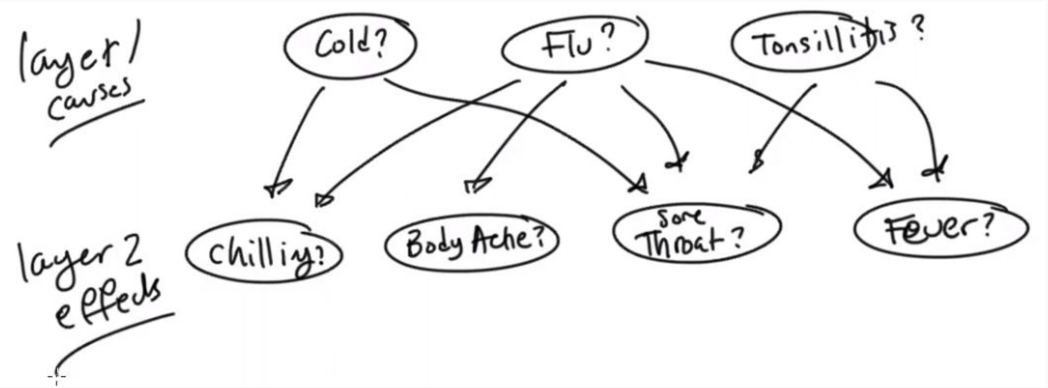


BN Modeling - Example

- We can also have query variables and evidence variables based on task
    - E.g. Diagnostic vs predictive tasks
- Ways to construct CPTs (where to obtain numbers):
    - Problem statement
    - Subjective beliefs
    - Learning from data

Example

The flu is an acute disease characterized by fever, body aches and pains, and can be associated with chilling and a sore throat. The cold is a bodily disorder popularly associated with chilling and can cause a sore throat. Tonsillitis is inflammation of the tonsils which leads to a sore throat and can be associated with fever.



BN Modeling

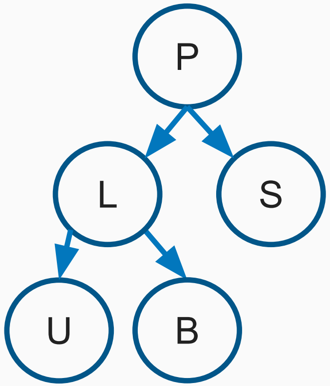
CPTs can also be estimated from medical records of previous patients

Case	*Cold?*	*Flu?*	*Tonsillitis?*	*Chilling?*	*Bodyache?*	*Sorethroat?*	*Fever?*
1	true	false	?	true	false	false	false
2	false	true	false	true	true	false	true
3	?	?	true	false	?	true	false
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

- Complete data (e.g. 2nd case): if **every** row is complete, dataset is complete □ efficient
- Incomplete data (e.g. 1st case): if **any** row is incomplete, dataset is incomplete
  - Use algorithm such as expectation maximization (EM) to find <u>maximum likelihood parameters</u>
- **BN structure + CPTs = BN**
  - May come out multiple BNs □ The BN with **max score** (computed by multiplying prob. assigned to all cases based on each BN) is better
  - It's called <u>maximum likelihood principle</u>

Sensitivity Analysis

> **Example**
>
> A few weeks after inseminating a cow, we have three possible tests to confirm pregnancy. The first is a scanning test which has a false positive of 1% and a false negative of 10%. The second is a blood test, which detects progesterone with a false positive of 10% and a false negative of 30%. The third test is a urine test, which also detects progesterone with a false positive of 10% and a false negative of 20%. The probability of a detectable progesterone level is 90% given pregnancy, and 1% given no pregnancy. The probability that insemination will impregnate a cow is 87%.



- Given S = -ve, B = -ve, U = -ve
- If we know:

P = yes	10.2%
P = No	89.8%

- **Sensitivity analysis** tells us what we can change in CPTs to change target our probability value
  - Useful in BN design process to debug the probability value of an existing network

Tips for HW8

- Install Java (JRE or JDK) and set up the Java path in environment variables
    - Environment setup: https://www.tutorialspoint.com/java/java_environment_setup.htm

- Install SamIam:
    - Download from:

      http://reasoning.cs.ucla.edu/samiam
    - For Windows:

      Amd64 for 64-bit system

      I386 for 32-bit system
    - See **tutorial videos** in:

      Online Help section