GATE in Data Science & Artificial Intelligence

ARTIFICIAL INTELLIGENCE

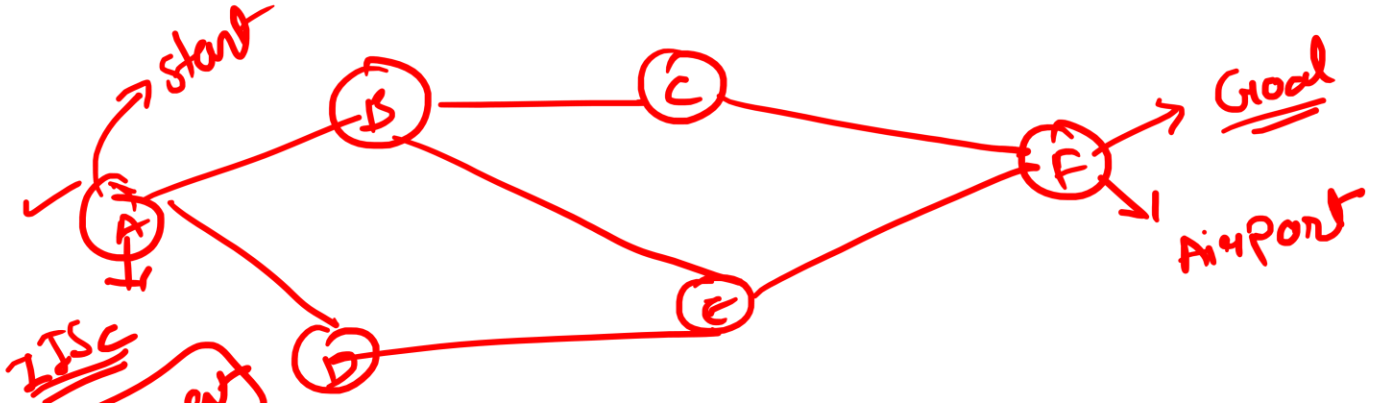
Introduction to reasoning under uncertainty

By:
Piyush Wairale
MTech (IIT Madras)
Instructor at IIT Madras BS in Data Science Degree

Real world
RTP

Optimal path

ideal



365km
2hr

2 sec
AI agent

AI agent

Self driving car → AI agent

Uncertainty

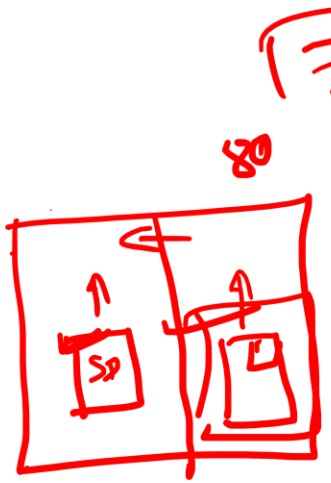
- Data is incomplete
- Data is noisy
- Future outcomes are uncertain

- Traffic
- Road issue
- Animal is coming
- Actions of other drivers

→ uncertainty

(1) Medical diagnosis
→ Cancer cells

Reasoning under uncertainty is a fundamental concept in artificial intelligence (AI) that involves making decisions or drawing conclusions when the information available is incomplete, ambiguous, or subject to change. Unlike deterministic systems, where outcomes are predictable and certain, AI systems often operate in environments where uncertainty is a significant factor.



⇒ Bayes theorem (conditional probability)

(Belief Network)

A graphical Probabilistic Models that represent a set of variable and their conditional dependies. using DAG

A (Disease)	B (Test)
T	+ve
T	+ve
T	-ve
F	-ve
F	+ve
F	+ve

① Conditional Independance

Bayesian Network

Nodes & Edges

DAG → Directed acyclic Graph

Nodes :- variable → CRV, DRV

Edges :- conditional dependecies

CPT — P(B)

P(B=+ve / A=T)	
P(B=-ve / A=T)	
P(B=+ve / A=F)	
P(B=-ve / A=F)	

Prior P(A) = 0.5

P(A=disease) = 0.5

P(A=not disease) = 1-0.5 = 0.5

A → Parent, Disease

B → Test

B is conditional dependent on A

P(B/A)

1. **Nodes (Variables):**
   - Each node in a Bayesian Network represents a random variable, which could be discrete or continuous. The variable can represent anything from observable quantities (e.g., sensor readings) to latent variables (e.g., underlying causes that are not directly observed).
2. **Edges (Dependencies):**
   - Directed edges (arrows) between nodes represent conditional dependencies between the variables. If there is a directed edge from node A to node B, it means that B is conditionally dependent on A.

3. **Conditional Probability Distributions (CPDs):**
   - Associated with each node is a conditional probability distribution that quantifies the effect of the parent nodes (those with directed edges leading to the node) on the node itself. For a node X_i with parents $Parents(X_i)$, the CPD specifies $P(X_i \mid Parents(X_i))$.

CPT

$P(B/A)$

4 Joint Probability Distribution:

- The joint probability distribution over all the variables in the network can be expressed as the product of the conditional probabilities for each variable given its parents:

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid Parents(X_i))$$

D (Disease) :- True/False

T (Test) :- [Positive/Negative]

S (Symptom) :- (Present/Absent)

(2) Symptom given Disease

$P(S = Present / D = True) = 0.7$

$P(S = Abs / D = False) = 0.9$

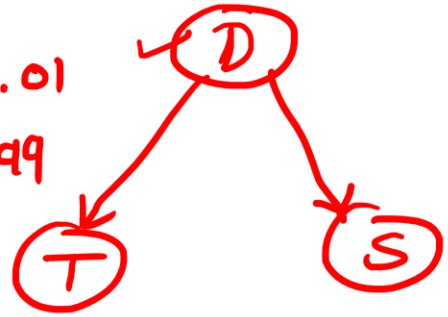
$P(S = Abs / D = True) = 1 - 0.7 = 0.3$

$P(S = Pres / D = False) = 1 - 0.9 = 0.1$

Prior Prob

$P(D = True) = 0.01$

$P(D = False) = 0.99$



CPT

(1) Test given Disease

$P(T = Pos / D = True) = 0.9$ (90% of chance of the test if disease is present (True))

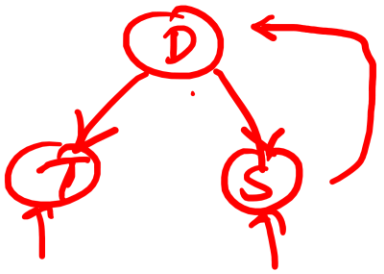
$P(T = Neg / D = False) = 0.8$ (80% chance of -ve test, if the disease is absent)

$P(T = Neg / D = True) = 1 - 0.9 = 0.1$

$P(T = Pos / D = False) = 1 - 0.8 = 0.2$

* Joint Prob Dist

$$P(D, T, S) = P(D) \times P(T/D) \times P(S/D)$$



D=? / T=+ve, S=Present

⇒ the prob that the person having a disease → D = True

$P(D=True, T=+ve, S=Pres) = P(D=True) \times P(T=+ve/D=True) \times P(S=Pre/D=True)$

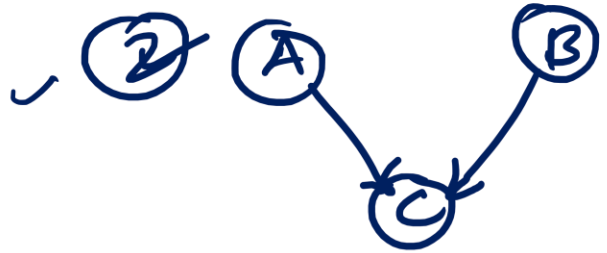
$= 0.01 \times 0.9 \times 0.7$

$= 0.0063$

② $P(D=True, T=-ve, S=Pres) = ?$

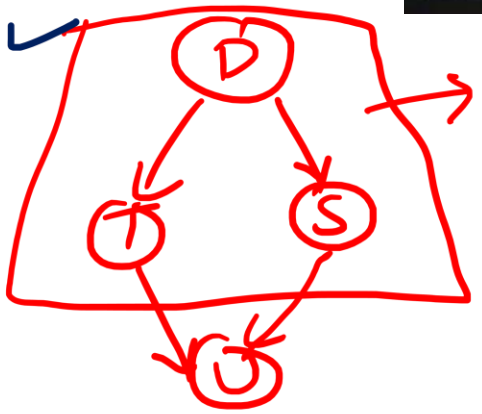
③ $P(D=True, T=-ve, S=-ve)$

$= ?$



$P(A, B, C) = ?$

1

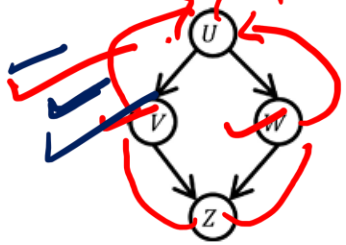


Grade DA 24

3

$P(A, B, C) = ?$

Q.64 Given the following Bayesian Network consisting of four Bernoulli random variables and the associated conditional probability tables:



	$P(\cdot)$
$U=0$	0.5
$U=1$	0.5

	$P(V=0\mid\cdot)$	$P(V=1\mid\cdot)$
$U=0$	0.5	0.5
$U=1$	0.5	0.5

	$P(W=0\mid\cdot)$	$P(W=1\mid\cdot)$
$U=0$	1	0
$U=1$	0	1

		$P(Z=0\mid\cdot)$	$P(Z=1\mid\cdot)$
$V=0$	$W=0$	0.5	0.5
$V=0$	$W=1$	1	0
$V=1$	$W=0$	1	0
$V=1$	$W=1$	0.5	0.5

The value of $P(U=1,V=1,W=1,Z=1)$ = _____ (rounded off to **three** decimal places).

Joint Probab :- for given network

$$P(U,V,W,Z) = P(U) \times P(V/U) \times P(W/U) \times P(Z/V,W)$$

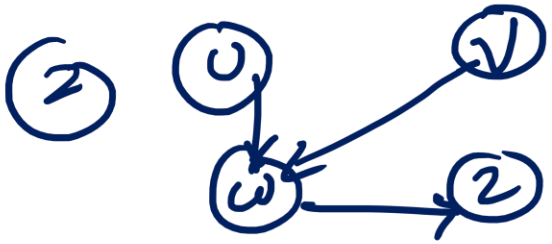
$$P(U=1,V=1,W=1,Z=1) = P(U=1) \times P(V=1/U=1) \times P(W=1/U=1) \times P(Z=1/V=1,W=1)$$

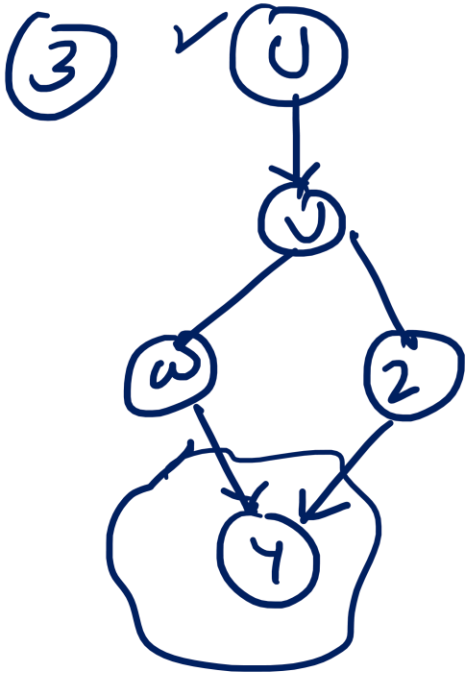
$$= 0.5 \times 0.5 \times 1 \times 0.5$$

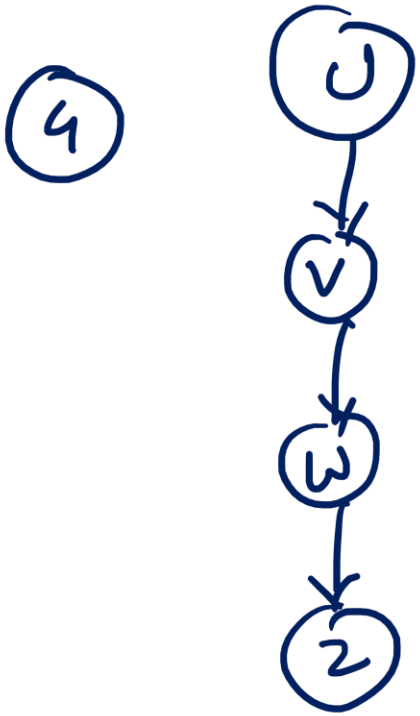
$$= 0.125$$

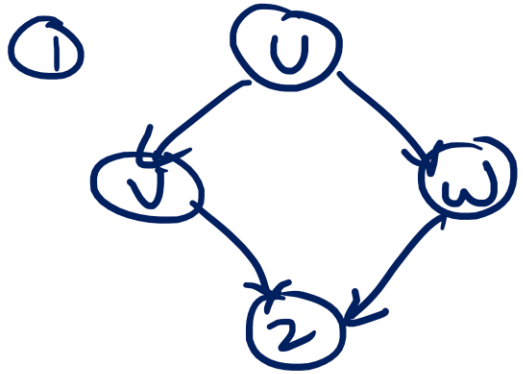
(1) $P(U=1,V=0,W=0,Z=1)$

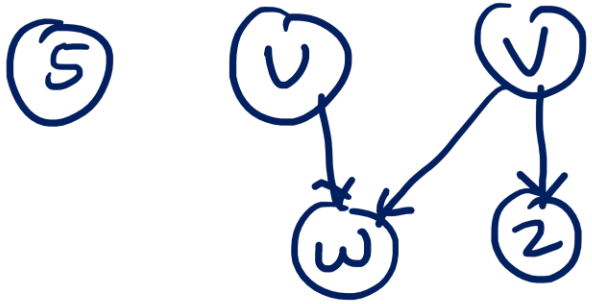
(2) $P(U=0,V=1,W=1,Z=1)$





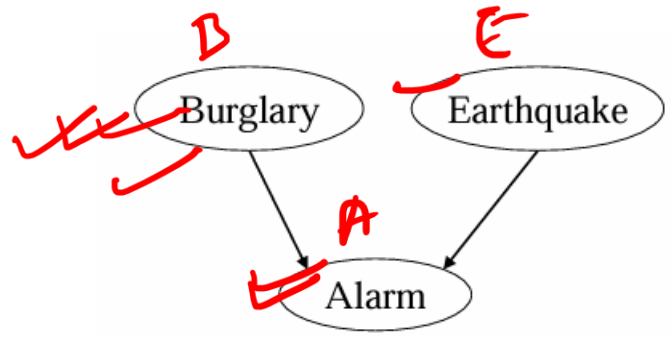






1. DAG
2. CPT
3. Required conditional Prob

Bayesian Network Example



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

$$= P(B=f) \cdot P(E=T) \cdot P(A=T|B=F,E=T)$$

$$= 0.999 \times 0.002 \times 0.29$$

$$= 1.$$

$B = false$

$E = True$

$A = True$

B	P(B)
false	0.999
true	0.001

E	P(E)
false	0.998
true	0.002

B	E	A	P(A|B,E)
false	false	false	0.999
false	false	true	0.001
false	true	false	0.71
false	true	true	0.29
true	false	false	0.06
true	false	true	0.94
true	true	false	0.05
true	true	true	0.95

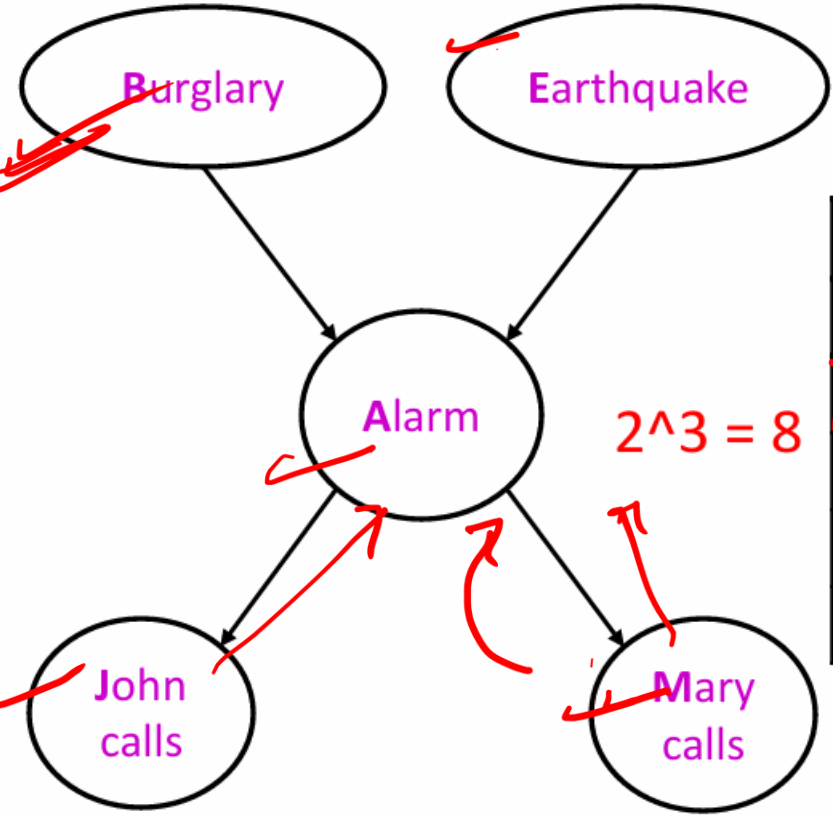
Here (1) $B = True$, $E = false$, $A = True$

(2) $B = F$, $E = t$, $A = True$

(3) $B = F$, $E = T$, $A = False$

Bayesian Network Example

Q



2

P(B)	
true	false
0.001	0.999

2

P(E)	
true	false
0.002	0.998

$P(J|A,B,E)$

2^3 = 8

B	E	P(A|B,E)	
		true	false
true	true	0.95	0.05
true	false	0.94	0.06
false	true	0.29	0.71
false	false	0.001	0.999

A	P(J|A)	
	true	false
true	0.9	0.1
false	0.05	0.95

2^2 = 4

A	P(M|A)	
	true	false
true	0.7	0.3
false	0.01	0.99

2^2 = 4

$P(B,E,A,J,M)$

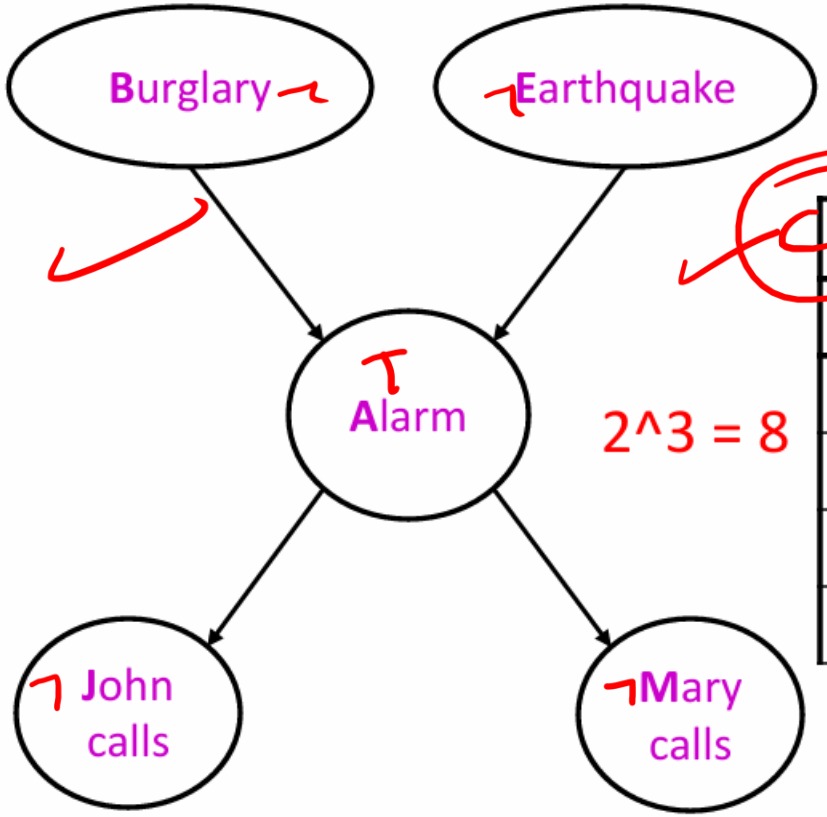
$= P(B) \cdot P(E) \times P(A|B,E) \times P(J|A) \times P(M|A)$

$= 0.001 \times 0.002 \times 0.95 \times 0.9 \times 0.7$

$=$

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

Bayesian Network Example



P(B)	
true	false
0.001	0.999

P(E)	
true	false
0.002	0.998

B	E	P(A|B,E)	
		true	false
true	true	0.95	0.05
true	false	0.94	0.06
false	true	0.29	0.71
false	false	0.001	0.999

A	P(J|A)	
	true	false
true	0.9	0.1
false	0.05	0.95

A	P(M|A)	
	true	false
true	0.7	0.3
false	0.01	0.99

2

2

2^3 = 8

2^2 = 4

2^2 = 4

HWE

① P(B, ¬E, A, ¬J, ¬M)

② P(¬B, ¬E, A, J, M)

③ P(B, E, ¬A, J, ¬M)

HWF

⇒ equivalent english statement for ① ② & ③

Bayesian Network Example

$P(x, y, z, u)$

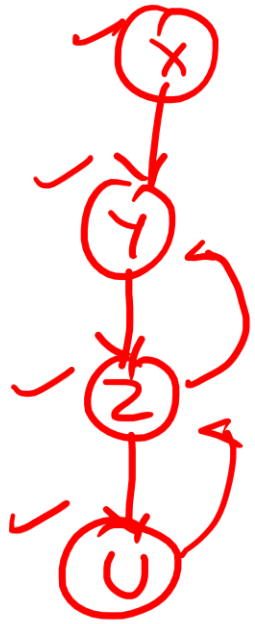
$= P(x) \cdot P(y/x) \cdot P(z/y, x)$

$\times P(u/z, y, x)$

by using chain rule

$= P(x) \cdot P(y/x) \cdot P(z/y)$

$P(u/z)$



Bayesian Network Example

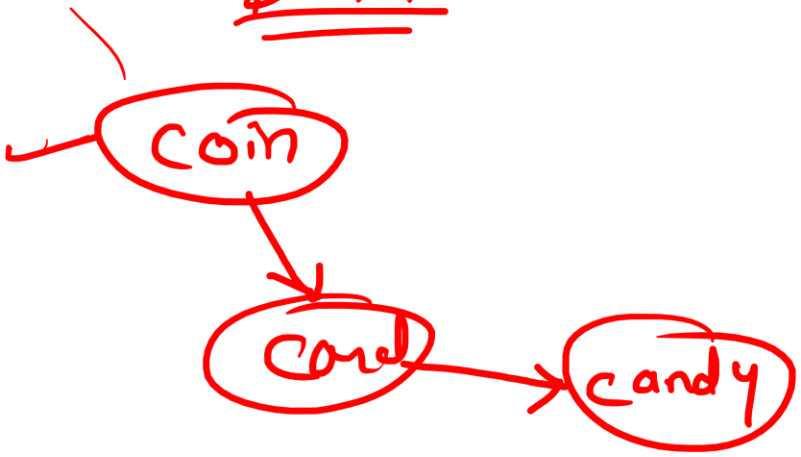
CPT

Coin	P(Coin)
tails	0.5
heads	0.5

Coin	Card	P(Card | Coin)
tails	black	0.6
tails	red	0.4
heads	black	0.3
heads	red	0.7

Card	Candy	P(Candy | Card)
black	1	0.5
black	2	0.2
black	3	0.3
red	1	0.1
red	2	0.3
red	3	0.6

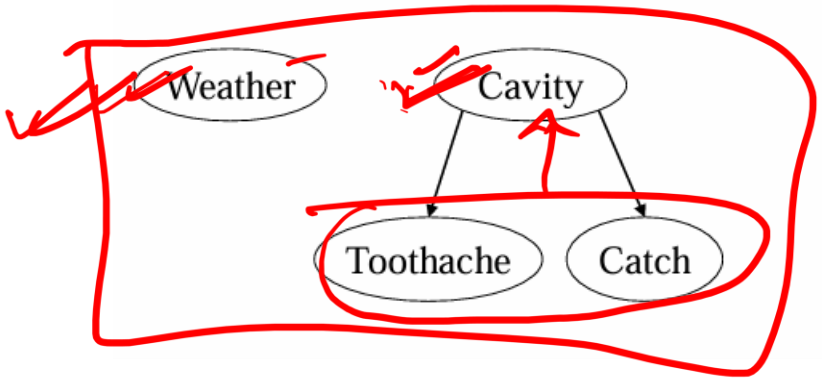
DAG



What does the DAG for this Bayes net look like?

$P(C, Card, Coin) = P(C) \cdot (P Card/Coin) \times P(Candy/Card)$

Bayesian Network Example



① weather & cavity are absolut independent

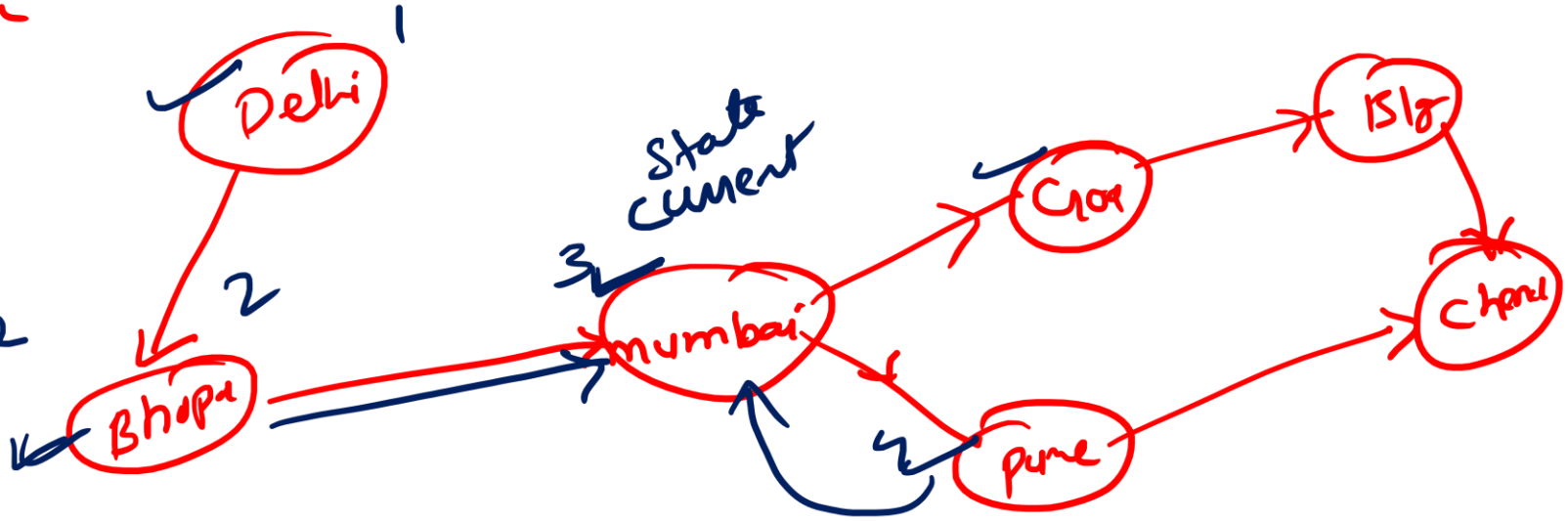
② Toothache and catch are conditional independent given cavity

$P(W, C, T, \text{catch}) = P(W) \cdot P(C) \cdot P(T/C) \times P(\text{catch}/\text{cavity})$

$= P(C, T, \text{catch}) \times P(\text{weather})$

$P(\text{cavity}/\text{Tooth}, \text{catch})$

* Bayesian Network



(1) Today

$P(Rain) = 0.8$

$P(Sunny) = 0.2$

$P(Rain = tom)$

$P(m/B, D)$

$X_4 = X_3$

$P(D, B, M, P) = P(D) \cdot P(B/D) \times P(m/B,D) \times P(Pune/m, B, D)$

$P(P/M)$

A **Markov Chain** is a mathematical system that undergoes transitions from one state to another within a finite or countable number of possible states. It is a type of stochastic process that satisfies the **Markov property**, which states that the future state depends only on the current state and not on the sequence of events that preceded it.

Key Components:

1. **States:** The set of possible conditions in which the system can exist.
2. **Transition Probabilities:** The probabilities of moving from one state to another.
3. **Initial State Distribution:** The probability distribution over the states at the start of the process.

X_{n+1}
X_n
X_{n-1}
X_{n-2}
X_0

Markov Property

The basic property of a Markov chain is that *only the most recent point in the trajectory affects what happens next.*

This is called the *Markov Property.*
It means that X_{t+1} *depends upon* X_t, *but it does not depend upon* $X_{t-1}, \ldots, X_1, X_0$.

We formulate the Markov Property in mathematical notation as follows:

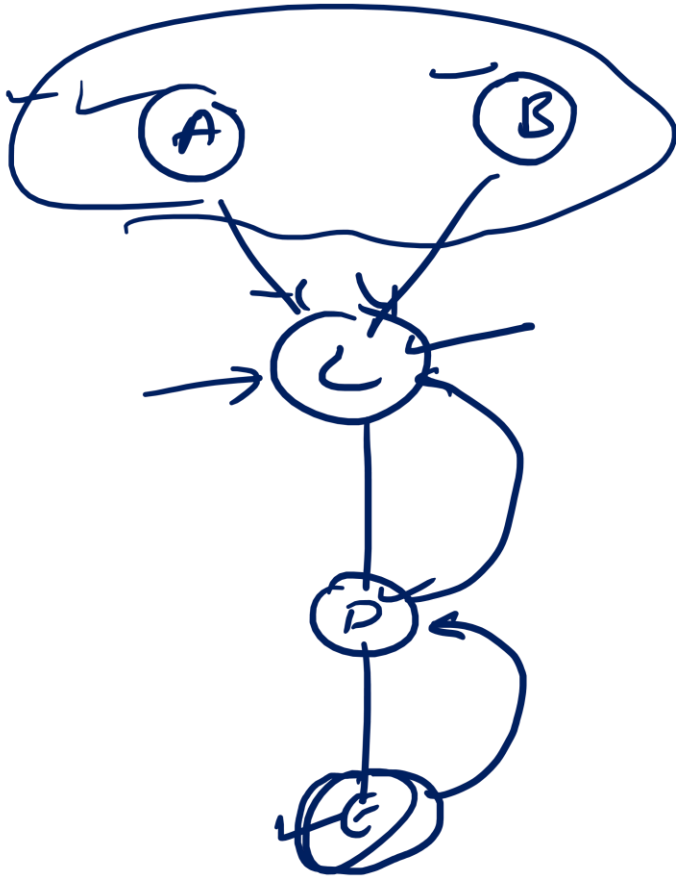
$$\mathbb{P}(X_{t+1} = s \,|\, X_t = s_t, X_{t-1} = s_{t-1}, \ldots, X_0 = s_0) \;=\; \mathbb{P}(X_{t+1} = s \,|\, X_t = s_t),$$

for all $t = 1, 2, 3, \ldots$ and for all states $s_0, s_1, \ldots, s_t, s$.

Explanation:

$\mathbb{P}(X_{t+1} = s \;|\; X_t = s_t, X_{t-1} = s_{t-1}, X_{t-2} = s_{t-2}, \ldots, X_1 = s_1, X_0 = s_0)$

- $\mathbb{P}(X_{t+1} = s)$: *distribution of X_{t+1}*
- $X_t = s_t$: *depends on X_t*
- $X_{t-1} = s_{t-1}, X_{t-2} = s_{t-2}, \ldots, X_1 = s_1, X_0 = s_0$: *but whatever happened before time t doesn't matter.*



Example weather forecasting

Today:- Day 1 → Current State

If it sunny today, there is chance of 80% that it will be sunny tomorrow

If it Rainy today, there is chance of 60% that it will rain tomorrow

T States Sunny (S), Rainy (R)

Tomorrow → Transition Proba :-

Day 2

① $P(S_{tom} / S_{today}) = 0.8$

② $P(R_{tom} / S_{today}) = 0.2$

③ $P(S_{tom} / R_{today}) = 0.4$

④ $P(R_{tom} / R_{today}) = 0.6$

20 Day 4

Day after tomorrow

Day 3

If it Sunny tomorrow

$P(Sunny=DAT / Sunny_{tom}) = 0.8 \times 0.8$

$= 0.64$

If it Rainy tomorrow

$P(Sunny=DAT / R=tom) = 0.4 \times 0.2$

$= 0.08$

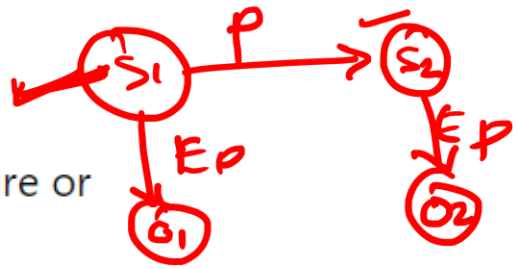
Total Prob of Sunny 2 day from now

$P(Sun\ in\ 2\ day) = 0.64 + 0.08$

$= 0.72$

Hidden Markov Model (HMM)

A **Hidden Markov Model (HMM)** is an extension of a Markov Chain where the states are not directly observable (hidden). Instead, each state produces an observable output (or emission) according to a probability distribution.



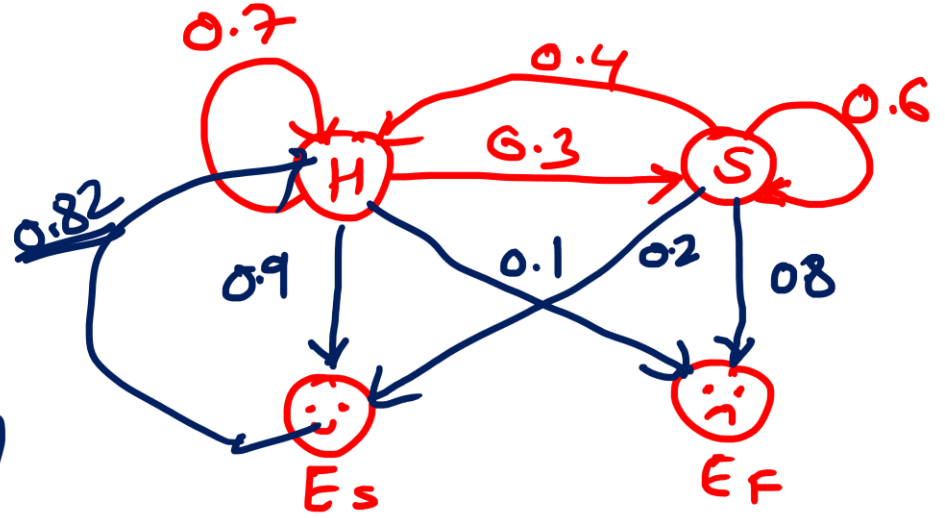
Key Components:

1. **States:** The hidden states of the system (e.g., underlying weather conditions like high pressure or low pressure).
2. **Transition Probabilities:** The probabilities of transitioning from one hidden state to another.
3. **Emission Probabilities:** The probabilities of observing a certain output given the current hidden state.
4. **Initial State Distribution:** The probability distribution over the hidden states at the start.

(*) We want to guess your bf/gf mood based on their behavior.
the mood can be Happy or Sad, but you can't directly observe their mood, but you see their actions like Smiling or Frowning.

⇒ Hidden state (1) Happy (H) (2) Sad (S)

⇒ Observation :- Smiling (Smile) ☺
Frowning (frown) ☹



Q You observe that your gf/bf is Smiling today.
What can you infer about their mood

⇒ $(P(\text{Smile})/H) = 0.9$

$(P(\text{Smile}/S)) = 0.2$

* Initial Probs :- $P(H) = 0.5$ $P(S) = 0.5$

$P(😊/H) = \frac{P(S \cap H \cap H)}{P(H)}$

- $P(H \cap 😊) = 0.5 \times 0.9 = 0.45$ =

✓ $P(S \cap 😊) = 0.5 \times 0.2 = 0.1$

* Normalize Probs :-

① Total prob that you bf|gf is smiling $= 0.45 + 0.1 = 0.55$

$P(H|😊) = \frac{0.45}{0.55} = 0.82$ ✓

$P(S|😊) = \frac{0.1}{0.55} = 0.18$

Hidden Markov Model (HMM)

Hidden Markov Models (HMMs) are a type of statistical model used to represent systems that are assumed to be a Markov process with unobservable (hidden) states. HMMs are particularly useful in scenarios where you have observable data that is believed to be influenced by hidden states, such as in speech recognition, natural language processing, and bioinformatics.

Three fundamental algorithms associated with HMMs are:

1. **Forward Algorithm**
2. **Viterbi Algorithm**
3. **Baum-Welch Algorithm**

1. Forward Algorithm

The Forward Algorithm is used to compute the **probability of a sequence of observations** given a Hidden Markov Model. It helps determine how likely a particular sequence of observed events is, given the model parameters.

Decoding

2. Viterbi Algorithm

The Viterbi Algorithm is used to find the **most likely sequence of hidden states** (called the Viterbi path) that could have generated a given sequence of observations. This is useful in applications like speech recognition, where you want to determine the most probable sequence of spoken words given an audio signal.

3. Baum-Welch Algorithm

- The **Baum-Welch Algorithm** is a special case of the **Expectation-Maximization (EM) algorithm** used to find the unknown parameters of an HMM. It is used for **training** HMMs when the state transition probabilities and emission probabilities are not known.

- The Baum-Welch Algorithm, also known as the Expectation-Maximization (EM) algorithm for HMMs, is used to **learn the parameters** (transition probabilities, emission probabilities, and initial state probabilities) of an HMM from a set of observed sequences when the states are unknown. It finds the parameters that maximize the likelihood of the observed sequences.

Summary of Each Algorithm

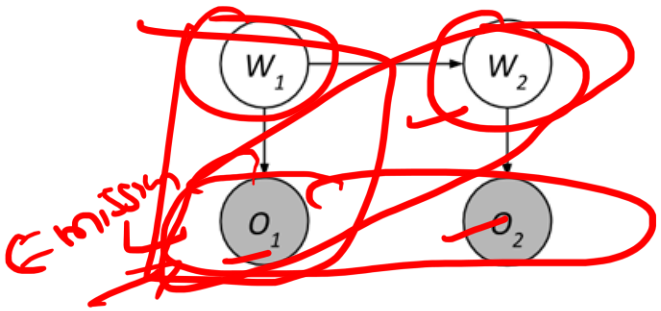
- **Forward Algorithm**: Computes the probability of an observation sequence given the HMM parameters.
- **Viterbi Algorithm**: Finds the most likely sequence of hidden states for a given observation sequence.
- **Baum-Welch Algorithm**: Estimates the HMM parameters from observed sequences using an iterative approach.

Inference in HMMs:

The two main tasks are:

1. **Decoding:** Given a sequence of observations, determine the most likely sequence of hidden states. This is often done using the **Viterbi algorithm**.
2. **Learning:** Estimate the model parameters (transition and emission probabilities) given a set of observed sequences. This is typically done using the **Baum-Welch algorithm**.

Consider the following Hidden Markov Model. O_1 and O_2 are supposed to be shaded.



W_1	$P(W_1)$
0	0.3
1	0.7

W_t	W_{t+1}	$P(W_{t+1}\|W_t)$
0	0	0.4
0	1	0.6
1	0	0.8
1	1	0.2

W_t	O_t	$P(O_t\|W_t)$
0	a	0.9
0	b	0.1
1	a	0.5
1	b	0.5

Suppose that we observe $O_1 = a$ and $O_2 = b$.
Using the forward algorithm, compute the probability distribution $P(W_2|O_1 = a, O_2 = b)$ one step at a time.

$P(w_2 = 1 / w_1 = 0) = 0.6$

$t = 2$, $O_2 = b$, $P(O_2 = b / w_2 = 0)$

$P(w_2, O_1 = a, O_2 = b) = P(O_2/w_2) \cdot P(w_2, O_1)$

$= P(O_2 = b/w_2) \cdot P(w_2, O_1 = a)$

$P(w_1, O_1 = a) = P(w_1) \cdot P(O_1/w_1)$

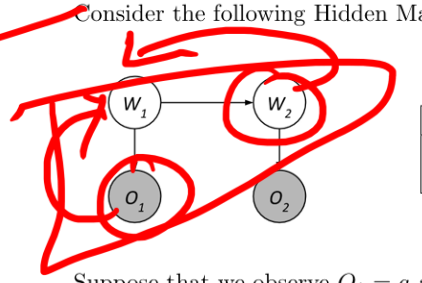
$P(w_1 = 0, O_1 = a) = P(w_1 = 0) \cdot P(O_1 = a/w_1 = 0)$

$= 0.3 \times 0.9 = 0.27$

$P(w_1 = 1, O_1 = a) = P(w_1 = 1) \cdot P(O_1 = a/w_1 = 1)$

$= 0.7 \times 0.5 = 0.35$

Consider the following Hidden Markov Model. O_1 and O_2 are supposed to be shaded.



W_1	$P(W_1)$
0	0.3
1	0.7

W_t	W_{t+1}	$P(W_{t+1}\|W_t)$
0	0	0.4
0	1	0.6
1	0	0.8
1	1	0.2

W_t	O_t	$P(O_t\|W_t)$
0	a	0.9
0	b	0.1
1	a	0.5
1	b	0.5

Suppose that we observe $O_1 = a$ and $O_2 = b$.
Using the forward algorithm, compute the probability distribution $P(W_2|O_1 = a, O_2 = b)$ one step at a time.

$P(W_2 | O_1 = a)$

$P(W_2 = 0, O_1 = a) =$

$P(W_2 = 1, O_1 = a) =$

$$P(W_2 = 0 | O_1 = a) = \sum_{W_1=0,1} P(W_1, O_1 = a) \cdot P(W_2 = 0 | W_1) \rightarrow \text{total pm}$$

$$= P(W_1 = 0, O_1 = a) \cdot P(W_2 = 0 | W_1 = 0) + P(W_1 = 1, O_1 = a) \cdot P(W_2 = 0 | W_1 = 1)$$

$$= 0.27 \times 0.4 + 0.35 \times 0.8 = 0.388$$

$$P(W_2 = 1, O_1 = a) = \sum_{W_1=0,1} P(W_1, O_1 = a) \cdot P(W_2 = 1 | W_1) \rightarrow \text{total pm}$$

$$= P(W_1 = 0, O_1 = a) \cdot P(W_2 = 1 | W_1 = 0) + P(W_1 = 1, O_1 = a) \cdot P(W_2 = 1 | W_1 = 1)$$

$$= 0.27 \times 0.6 + 0.35 \times 0.2 = 0.232$$

$$P(\omega_2, o_1 = a, o_2 = b) = P(\omega_2, o_1 = a) \cdot P(o_2^{=b} / \omega_2)$$

① $\omega_2 = 0$

$$P(\omega_2 = 0, o_1 = a, o_2 = b) = P(\omega_2 = 0, o_1 = a) \cdot P(o_2 = b / \omega_2 = 0)$$

$$= 0.388 \times 0.1 = 0.0388$$

② $\omega_2 = 1$

$$P(\omega_2 = 1, o_1 = a, o_2 = b) = P(\omega_2 = 1, o_1 = a) \cdot P(o_2 = b / \omega_2 = 1)$$

$$= 0.232 \times 0.5 = 0.116$$

$\Rightarrow$ $P(\omega_2 \mid o_1 = a, o_2 = b)$ by Normality

① $P(\omega_2 = 0 \mid o_1 = a, o_2 = b) = \dfrac{0.0388}{0.0388 + 0.116} \approx 0.25$

$$\Rightarrow P(\omega_2 = 1 \mid o_1 = a, o_2 = b) = P(\omega_2 = 1, o_1 = a, o_2 = b)$$

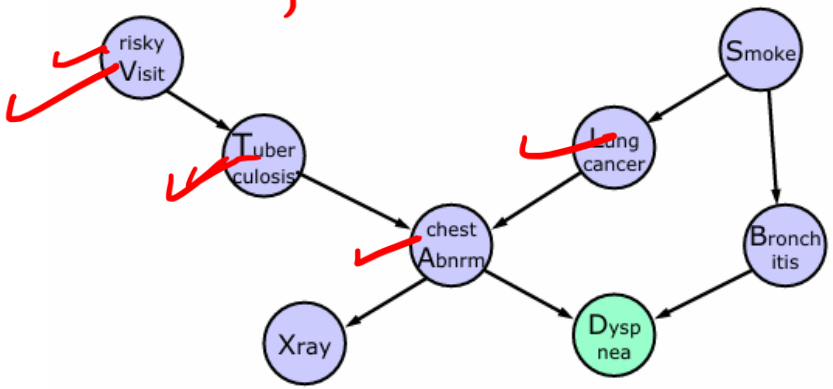
$$= \frac{0.116}{0.388 + 0.116} = 0.25$$

Probabilistic inference

Probabilistic inference involves computing probabilities and making predictions based on a probabilistic model. It is widely used in fields such as statistics, machine learning, artificial intelligence, and data science to draw conclusions from data that may have uncertainty or noise.

Probabilistic inference can be broadly divided into two types: exact inference and approximate inference.

$P(X = t \mid e) = ?$



Computing $P(X = x|e)$ in a GM is an NP-hard problem, which means that there is not generalized efficient algorithm for inference given an arbitrary PGM, query and evidence nodes. However, for some families of models, there are polynomial time algorithms for inference. Another solution to deal with the hardness of the problem is finding approximate solutions. The following is a list of some of the exact and approximate algorithms on graphical models.

Exact Inference

Exact inference refers to the process of computing the exact probability distribution or the exact values of interest from a probabilistic model. This involves finding the precise solution to a problem, considering all possible scenarios or configurations of the random variables.

Examples of Exact Inference:

- **Bayesian Networks:** Exact inference in Bayesian networks involves computing posterior probabilities of certain variables given evidence. Algorithms like Variable Elimination and the Junction Tree Algorithm are commonly used for this purpose.
- **Markov Chains:** For discrete-time Markov chains, exact inference involves calculating steady-state probabilities or finding the probability of being in a particular state at a given time.
- **Hidden Markov Models (HMMs):** The Forward-Backward algorithm is used for exact inference to compute the posterior probabilities of hidden states given observed data.

Approximate Inference

Approximate inference is used when exact inference is computationally infeasible. It involves finding an approximate solution that is close enough to the exact result. Approximate inference methods trade off some amount of precision for computational efficiency.

Examples of Approximate Inference:

- **Sampling Methods:** Techniques like Markov Chain Monte Carlo (MCMC) methods, including the Metropolis-Hastings algorithm and Gibbs sampling, generate samples from the posterior distribution to approximate it.
- **Variational Inference:** This method approximates a complex probability distribution by a simpler distribution and then minimizes the divergence between the two. The Expectation-Maximization (EM) algorithm and Variational Autoencoders (VAEs) are examples of models using variational inference.
- **Loopy Belief Propagation:** This is an approximate inference technique used in graphical models like Bayesian networks and Markov Random Fields. It iteratively updates beliefs or marginal probabilities but does not guarantee convergence to the exact solution.
- **Expectation Propagation:** A method that approximates the posterior distribution by matching moments between the true distribution and an approximating distribution.

* Bayes Theorem

$$P(B_i/A) = \frac{P(B_i) \times P(A/B_i)}{P(A)}$$

A = 0,1 B = 0,1

A → B

$P(A,B) = P(A) \cdot P(B/A)$

$P(A=0, B=1) = P(A=0) \cdot P(B=1/A=0)$

→ $P(A=0, B=0)$

→ $P(A=1, B=0)$

$P(A=1, B=1)$

Theorem of total Probability

$$P(A) = \sum_{i=1}^{n} P(B_i) \cdot P(A/B_i)$$

$P(A) = P(B=0) \cdot P(A/B=0) + P(B=1) \cdot P(A/B=1)$

$P(B) = \sum_{A=1}^{n} P(A_i) \cdot P(B/A_i)$

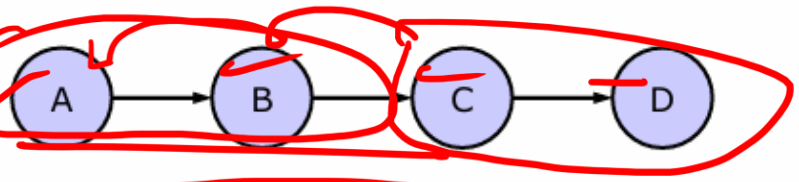
$= P(A=0) \cdot P(B=0/A=0) + P(A=1) \cdot P(B=0/A=1)$

$+ P(A=0) \cdot P(B=1/A=0) + P(A=1) \cdot P(B=1/A=1)$

$\sum_{i=0} P(B=0/A_i) \cdot P(A_i)$

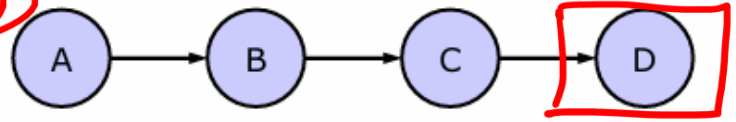
$\sum_{i=0} P(B=1/A_i) \cdot P(A_i)$

$P(A, B, C, D) = P(A) \cdot P(B/A) \cdot P(C/B) \cdot P(D/C)$



$$\Pr(d) = \sum_{ABC} \Pr(a,b,c,d)$$

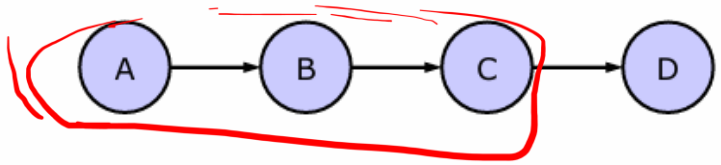
$P(d) =$ $P(d|c)\,P(c)$



$$\Pr(d) = \sum_{ABC} \Pr(a,b,c,d)$$

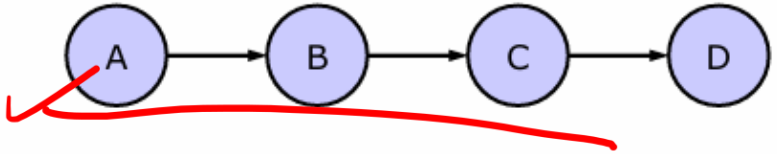
$$= \sum_{ABC} \Pr(d \mid c) \Pr(c \mid b) \Pr(b \mid a) \Pr(a)$$

$$= \sum_{C} \sum_{B} \sum_{A} \Pr(d \mid c) \Pr(c \mid b) \Pr(b \mid a) \Pr(a)$$



$$\Pr(d) = \sum_{ABC} \Pr(a,b,c,d)$$

$$= \sum_{ABC} \Pr(d \mid c) \Pr(c \mid b) \Pr(b \mid a) \Pr(a)$$



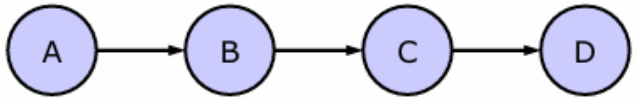
$$\Pr(d) = \sum_{ABC} \Pr(a,b,c,d)$$

$$= \sum_{ABC} \Pr(d \mid c)\Pr(c \mid b)\Pr(b \mid a)\Pr(a)$$

$$= \sum_{C}\sum_{B}\sum_{A} \Pr(d \mid c)\Pr(c \mid b)\Pr(b \mid a)\Pr(a)$$

$$= \sum_{C} \Pr(d \mid c) \sum_{B} \Pr(c \mid b) \sum_{A} \Pr(b \mid a)\Pr(a)$$

Summing out

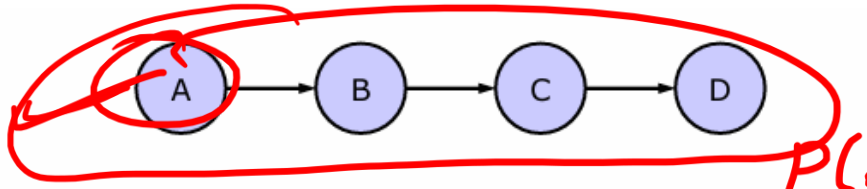


$$\Pr(d) = \sum_C \Pr(d \mid c) \sum_B \Pr(c \mid b) \sum_A \Pr(b \mid a) \Pr(a)$$

$$\begin{bmatrix} \Pr(b_1 \mid a_1)\Pr(a_1) & \Pr(b_1 \mid a_2)\Pr(a_2) \\ \Pr(b_2 \mid a_1)\Pr(a_1) & \Pr(b_2 \mid a_2)\Pr(a_2) \end{bmatrix}$$

$$\Pr(d) = \sum_C \Pr(d \mid c) \sum_B \Pr(c \mid b) \sum_A \Pr(b \mid a) \Pr(a)$$

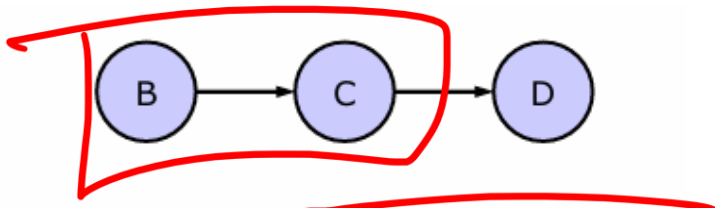
$$\begin{bmatrix} \sum_A \Pr(b_1 \mid a) \Pr(a) \\ \sum_A \Pr(b_2 \mid a) \Pr(a) \end{bmatrix}$$



$$\Pr(d) = \sum_C \Pr(d \mid c) \sum_B \Pr(c \mid b) \sum_A \Pr(b \mid a) \Pr(a)$$

$f_1(b)$

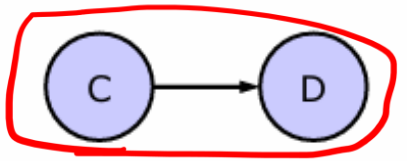
$$f_1(c) = \sum_B P(c \mid b) \cdot f_1(b)$$



$$\Pr(d) = \sum_C \Pr(d \mid c) \sum_B \Pr(c \mid b) f_1(b)$$

$P(C)$

$f_2(c)$

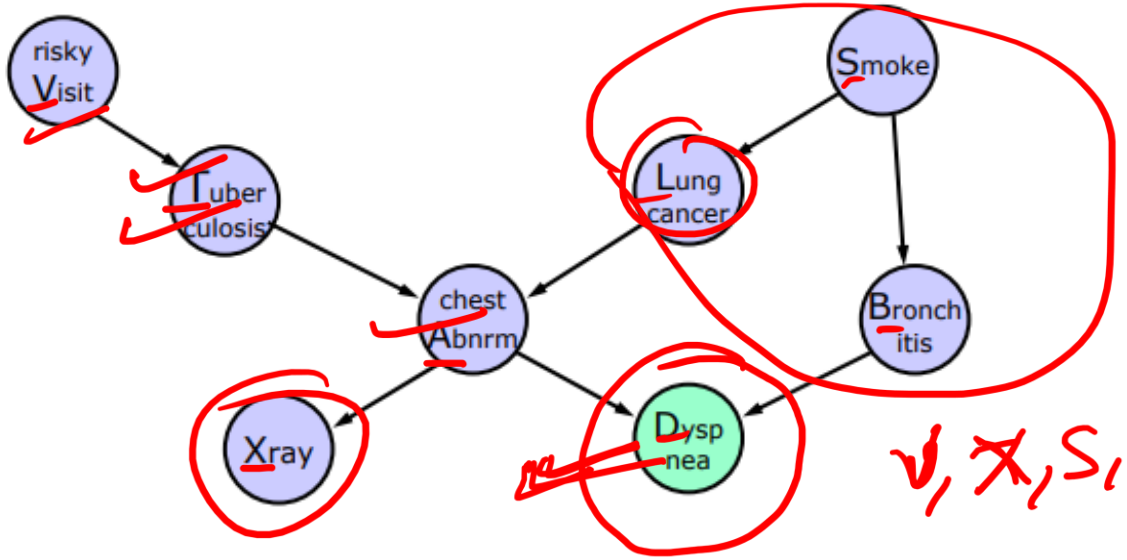


$P(C)$

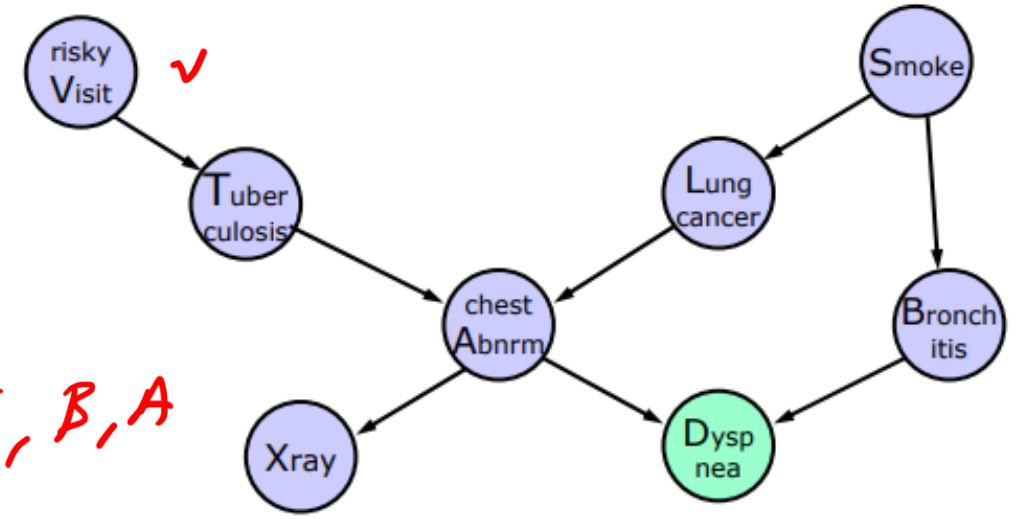
$$\Pr(d) = \sum_C \Pr(d \mid c) f_2(c)$$

Variable Elimination Example 2

Network



V, X, S, T, J, B, A



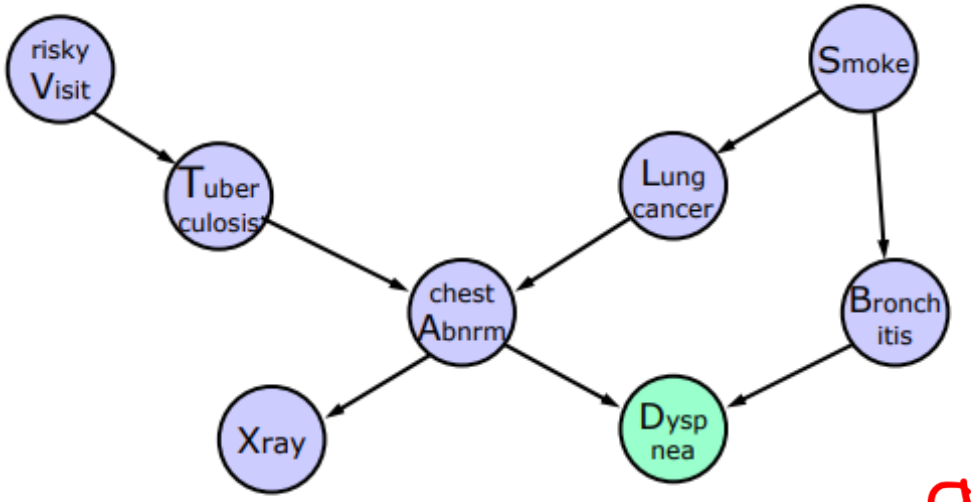
$P(V, T, X, A, L, S, B, D)$

$$\Pr(d) = \sum_{A,B,L,T,S,X,V} \Pr(d \mid a,b)\Pr(a \mid t,l)\Pr(b \mid s)\Pr(l \mid s)\Pr(s)\Pr(x \mid a)\Pr(t \mid v)\Pr(v)$$

Factor T → $f_1(t)$

$P(T)$

Variable Elimination Example 2



$$\Pr(d) = \sum_{A,B,L,T,S,X} \Pr(d \mid a,b)\Pr(a \mid t,l)\Pr(b \mid s)\Pr(l \mid s)\Pr(s)\Pr(x \mid a)\sum_{V}\Pr(t \mid v)\Pr(v)$$

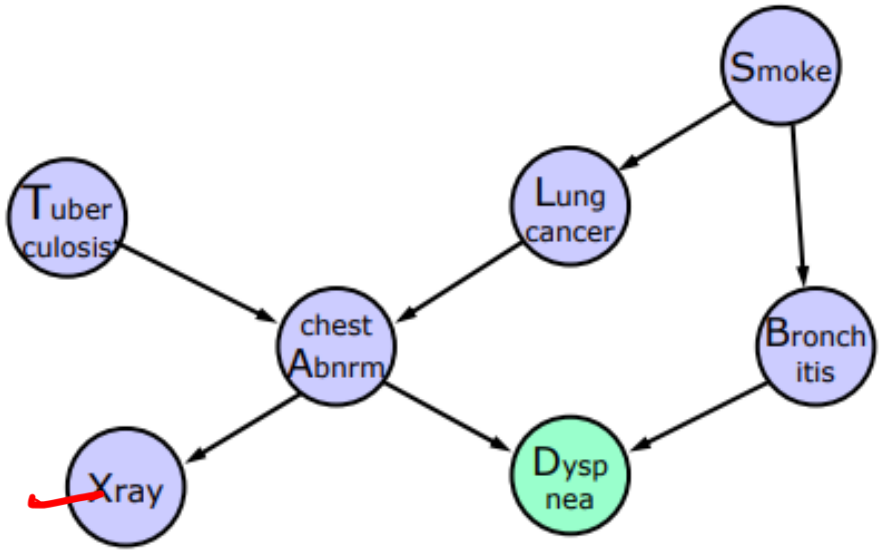
$f_1(t)$

a1 a2

x1 x2

CPT

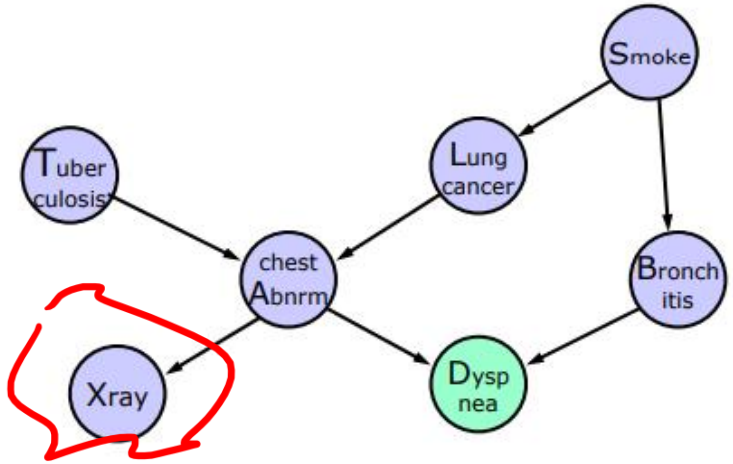
a1 x1



$$\Pr(d) = \sum_{A,B,L,T,S,X} \Pr(d \mid a,b)\Pr(a \mid t,l)\Pr(b \mid s)\Pr(l \mid s)\Pr(s)\Pr(x \mid a) f_1(t)$$

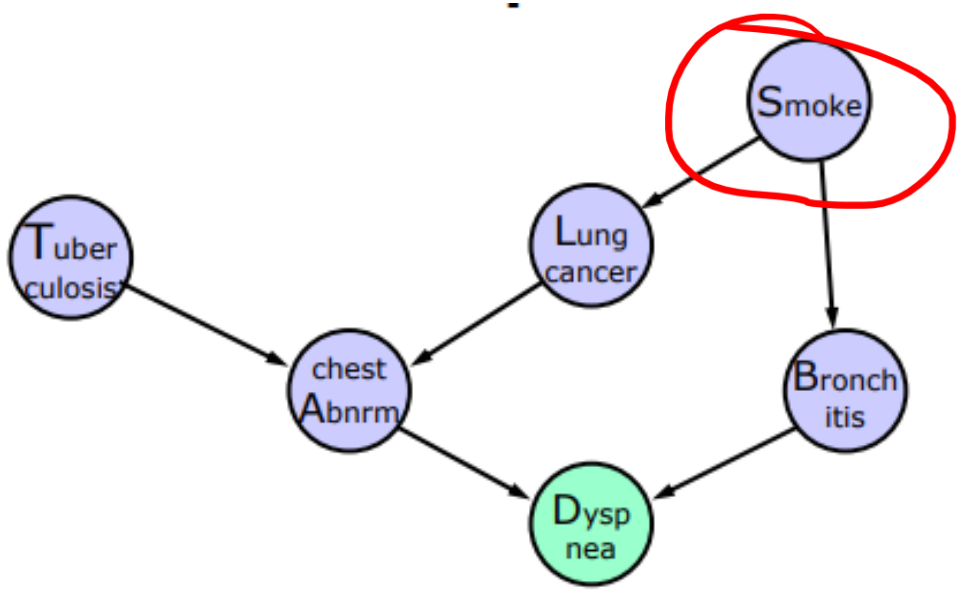
$\sum_x P(x|a) = 1$

Variable Elimination Example 2



$$\Pr(d) = \sum_{A,B,L,T,S} \Pr(d \mid a,b)\Pr(a \mid t,l)\Pr(b \mid s)\Pr(l \mid s)\Pr(s) f_1(t) \sum_{X} \Pr(x \mid a)$$

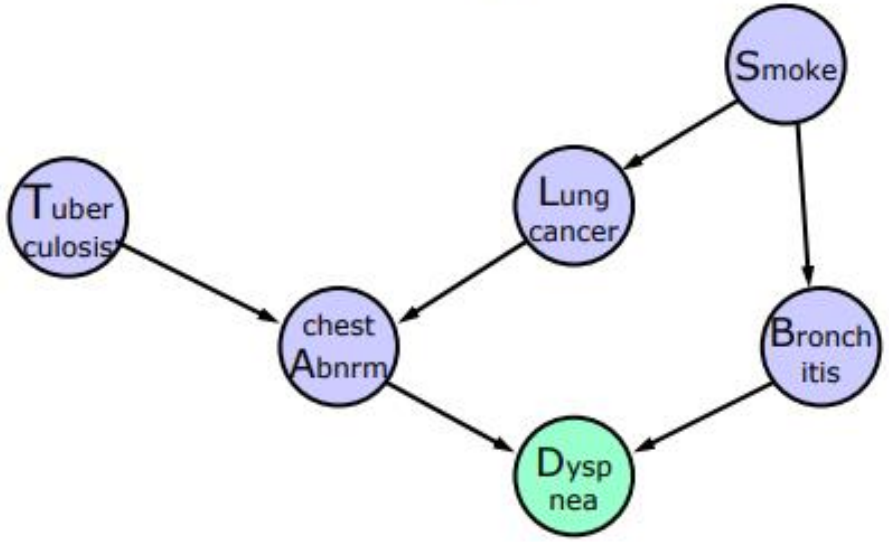
$$\sum_{X} \Pr(x \mid a) = 1$$



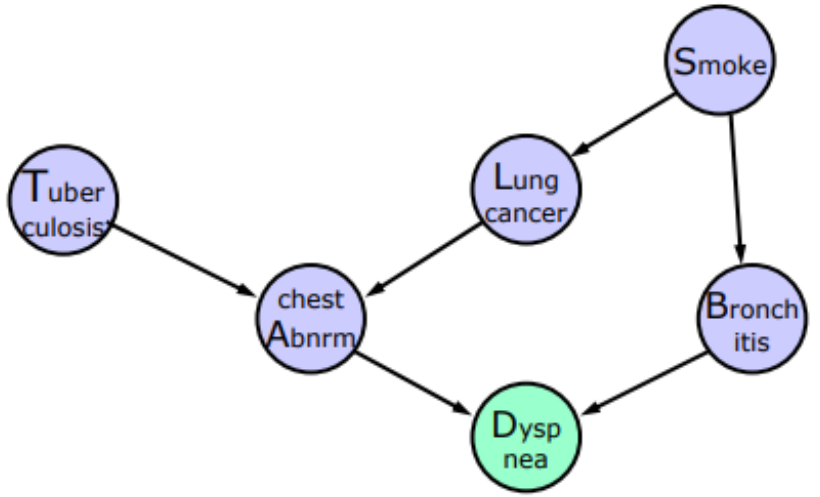
$$\Pr(d) = \sum_{A,B,L,T,S} \Pr(d \mid a,b)\Pr(a \mid t,l)\Pr(b \mid s)\Pr(l \mid s)\Pr(s) f_1(t)$$

$$\sum_{S} P(b|s) \cdot P(l|s) \cdot P(s) \cdot f_2(b,l)$$

Variable Elimination Example 2

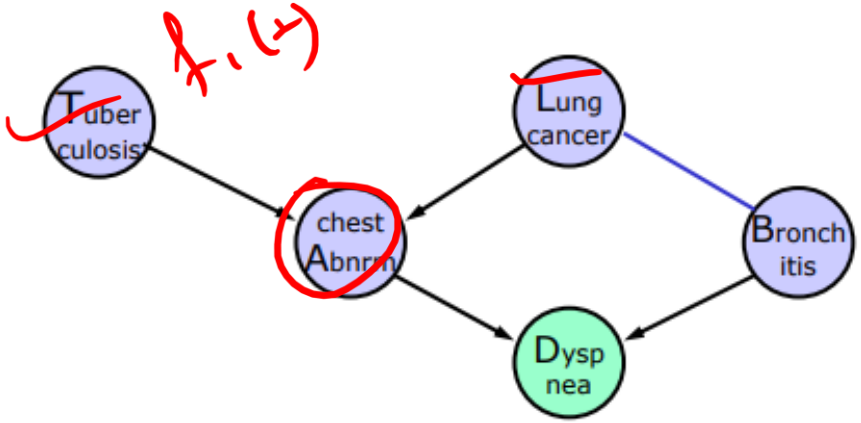


$$\Pr(d) = \sum_{A,B,L,T} \Pr(d \mid a,b) \Pr(a \mid t,l) f_1(t) \sum_{S} \Pr(b \mid s) \Pr(l \mid s) \Pr(s)$$



$$\Pr(d) = \sum_{A,B,L,T} \Pr(d \mid a,b) \Pr(a \mid t,l) f_1(t) \underbrace{\sum_{S} \Pr(b \mid s) \Pr(l \mid s) \Pr(s)}_{f_2(b,l)}$$

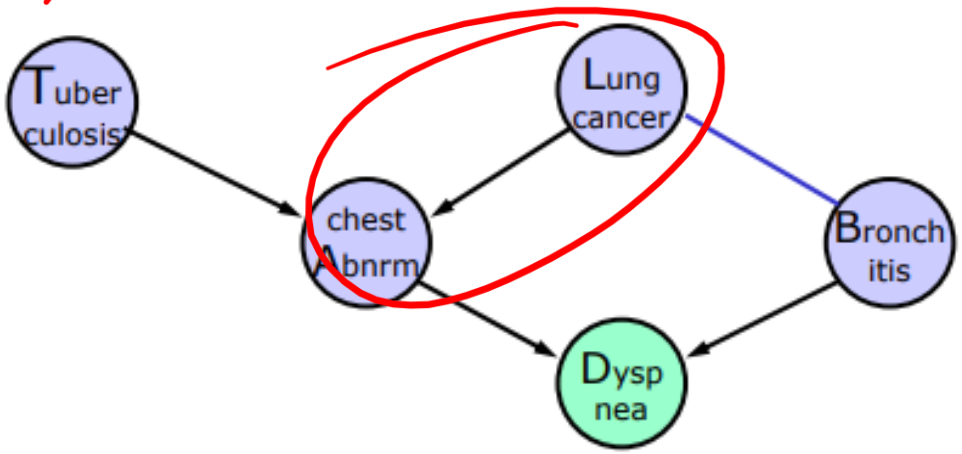
Variable Elimination Example 2



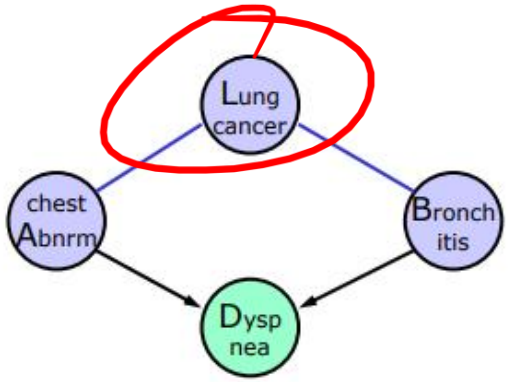
$$\Pr(d) = \sum_{A,B,L,T} \Pr(d \mid a,b) \Pr(a \mid t,l) f_1(t) f_2(b,l)$$

$$\Pr(d) = \sum_{A,B,L} \Pr(d \mid a,b) f_2(b,l) \sum_{T} \Pr(a \mid t,l) f_1(t)$$

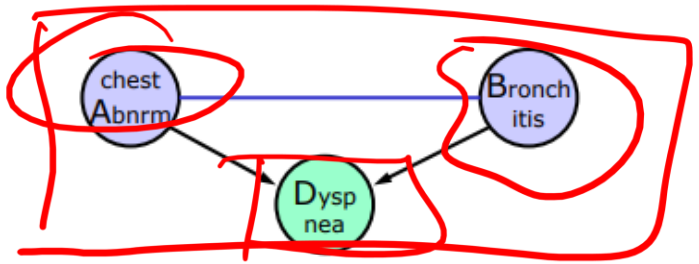
$f_3(a,l)$



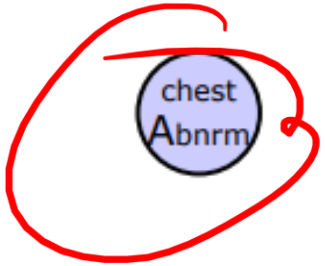
Variable Elimination Example 2



$$\Pr(d) = \sum_{A,B} \Pr(d \mid a,b) \underbrace{\sum_{L} f_2(b,l) f_3(a,l)}_{f_4(a,b)}$$



$$\Pr(d) = \sum_{A} \underbrace{\sum_{B} \Pr(d \mid a,b) f_4(a,b)}_{f_5(a)}$$



$P(d/a,b) \cdot P(a,b)$

$$\Pr(d) = \sum f_5(a)$$

A factor is a function from a set of random variables to a number. Formally, let f denote a factor and let X_1 to X_j denote the variables in the factor.

A factor is an abstract concept. It can represent a joint probability or a conditional probability. For example, a factor with two variables X_1 and X_2 can represent $P(X_1 \wedge X_2)$ — this is a joint probability. It can also represent $P(X_1|X_2)$ — this is a conditional probability. A third possibility is representing $P(X_1 and X_3 = v_3|X_2)$. For the third probability, although X_3 appears, it is not a variable since we already assigned a value to it. When you are given a factor, the meanings of the values may not be obvious. For instance, if the factor does not represent a joint distribution, the values may not sum to 1.

For the variable elimination algorithm, we will define a factor for every variable/node in the Bayesian network. The initial factor for each variable/node captures the conditional probability distribution for that variable/node.

$f(E, R)$:

E	R	val
t	t	0.9
t	f	0.1
f	t	0.0002
f	f	0.9998

Example: Let's look at an example. f_1 is a factor with three variables.

CPT

$f_1(X, Y, Z)$:

X	Y	Z	val
t	t	t	0.1
t	t	f	0.9
t	f	t	0.2
t	f	f	0.8
f	t	t	0.4
f	t	f	0.6
f	f	t	0.3
f	f	f	0.7

X=t

① Restriction

$f_1(x=t, Y=f, z=f) = 0.2$

Y=f

$f_3(z)$ =

Z	val
t	0.2
f	0.8

z=t

$f_4() = 0.2$

$f_2(Y, Z)$:

Y	Z	val
t	t	0.1
t	f	0.9
f	t	0.2
f	f	0.8

Example: Let's look at an example of sum out. The factor f1 has three variables: X, Y, and Z. What happens if we sum out Y? Since we summed out Y, Y will disappear from the factor. The new factor f2 has X and Z only.

$f_1(X, Y, Z)$:

X	Y	Z	val
t	t	t	0.03
t	t	f	0.07
t	f	t	0.54
t	f	f	0.36
f	t	t	0.06
f	t	f	0.14
f	f	t	0.48
f	f	f	0.32

② Sum out

addition of val corresponding to same x & z value only

for some combn of x & z, we are going to add value

$f_2(X, Z)$:

X	Z	val
t	t	0.57
t	f	0.43
f	t	0.54
f	f	0.46

We have two factors. f1 has variables X and Y. f2 has variables Y and Z. Let's multiply them together.

f_1:

X	Y	val
t	t	0.1
t	f	0.9
f	t	0.2
f	f	0.8

f_2:

Y	Z	val
t	t	0.3
t	f	0.7
f	t	0.6
f	f	0.4

$f_1 \times f_2$:

X	Y	Z	val
t	t	t	0.03
t	t	f	0.07
t	f	t	0.54
t	f	f	0.36
f	t	t	0.06
f	t	f	0.14
f	f	t	0.48
f	f	f	0.32

③ multiply

$f_1 \cup f_2 = XY \cup YZ$

$= X, Y, Z$

Natural Join in SQL

$f_1 \times f_2 (X=t, Y=t, Z=f)$

$= 0.07$

④ Normalisation

Example: Suppose that we want to normalize the factor f1. By normalizing, we will produce a new factor f2. Normalize does not change the size of the factor. So f2 has the same size as f1. Each value in f2 is the original value divided by the sum of all the values in f1. The sum is 0.8. So f2 contains 0.25 and 0.75.

f_1:

Y	val
t	0.2
f	0.6

$\frac{0.2}{0.2+0.6} = \frac{2}{8}$

$= \frac{0.6}{0.8}$

$=$

f_2:

Y	val
t	0.25
f	0.75

①

Sampling

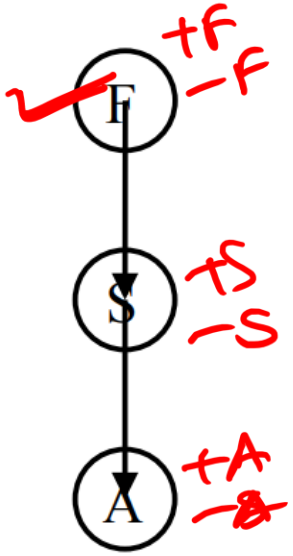
Approximate inference using sampling refers to a set of methods used to estimate probability distributions, especially in probabilistic models like Bayesian networks, where exact inference is computationally expensive or infeasible.

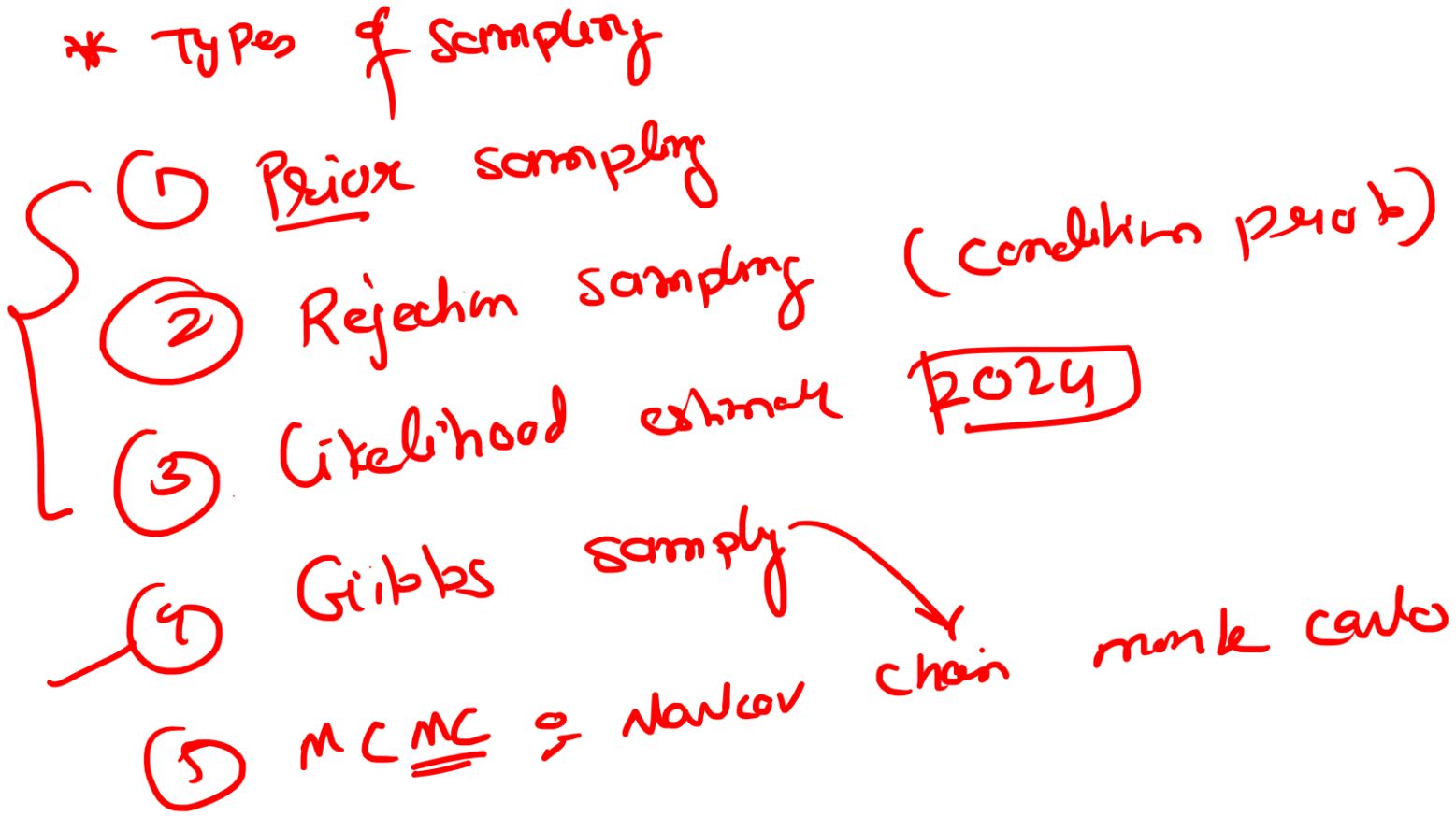
These methods generate samples from the probability distribution and use these samples to estimate the desired probabilities.

Since the process only approximates the true distribution, it is called *approximate inference*.

Approximate Inference

- Simulation has a name: sampling
- Sampling is a hot topic in machine learning, and it's really simple
- Basic idea:
  - Draw N samples from a sampling distribution S
  - Compute an approximate posterior probability
  - Show this converges to the true probability P
- Why sample?
  - Learning: get samples from a distribution you don't know
  - Inference: getting a sample is faster than computing the right answer (e.g. with variable elimination)





Prior Sampling

Prior sampling is a method used to generate samples from a **Bayesian Network** without any conditions or evidence. The goal is to sample values for the variables based on the **joint probability distribution** that the Bayesian network represents.

How it works:

1. **Start at the root nodes** (i.e., nodes with no parents) and sample values for these variables according to their prior probabilities.
2. **Move through the network** from parents to children, sampling values for each variable based on its conditional probability distribution given the sampled values of its parents.
3. **Repeat the process** until you have generated the desired number of samples.

The generated samples can be used to estimate marginal probabilities by counting occurrences of events.

Q9

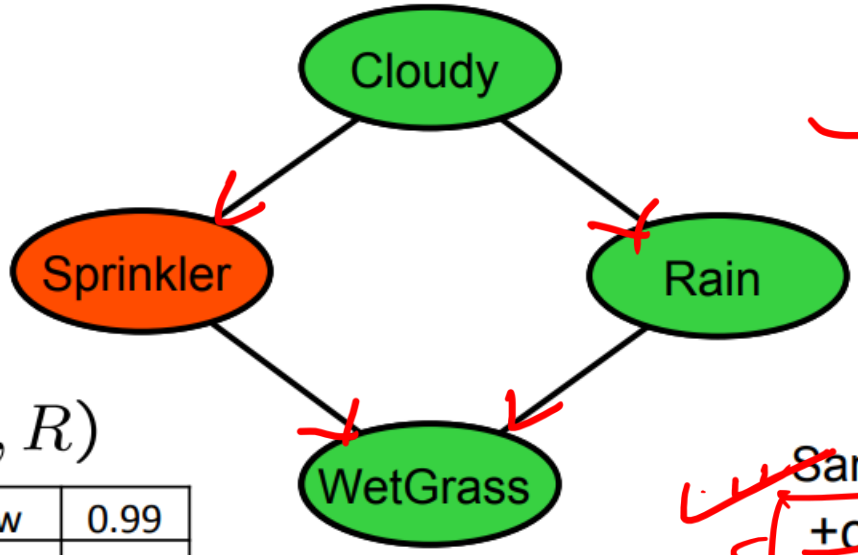
Im

$P(C)$

+c	0.5
-c	0.5

$P(S|C)$

+c	+s	0.1
	-s	0.9
-c	+s	0.5
	-s	0.5



$P(R|C)$

+c	+r	0.8
	-r	0.2
-c	+r	0.2
	-r	0.8

$P(W|S,R)$

+s	+r	+w	0.99
		-w	0.01
	-r	+w	0.90
		-w	0.10
-s	+r	+w	0.90
		-w	0.10
	-r	+w	0.01
		-w	0.99

P(-s) = ?

P(+w) = ?

P(+c | +s, +r)

Samples:

+c, -s, +r, +w

+c, +s, +r, +w

-c, +s, +r, -w

+c, -s, +r, +w

-c, -s, -r, +w

$P(+C) = P(C = +c)$

$= \frac{\text{No. of sample having } +C}{\text{Total No. of sample}}$

$= \frac{3}{5}$

$= 0.6$

$P(-s, +r) = \frac{2}{5}$

$= 0.4$

Sampling

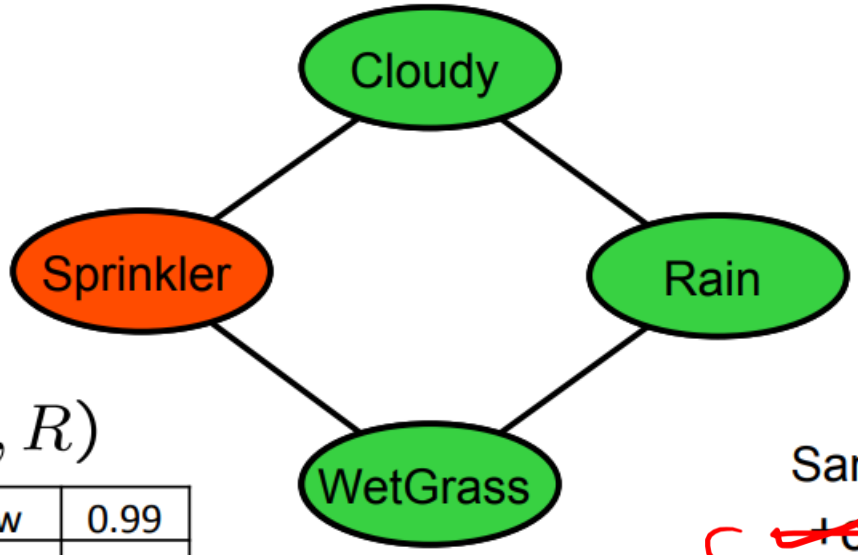
Rejection sampling improves on prior sampling by dealing more effectively with cases where there is evidence (i.e., observed variables). It can be used to estimate conditional probabilities, $P(X = x \mid Y = y)$.

- **Setup**: You have observed some variables (evidence) and want to sample from the conditional distribution given that evidence.
- **Procedure**:
  1. **Generate Prior Samples**: Like in prior sampling, generate samples from the full joint distribution.
  2. **Reject Inconsistent Samples**: After generating a sample, check if it matches the observed evidence (i.e., check if the sample's value for Y equals the observed value y). If the sample is consistent with the evidence, keep it. Otherwise, discard (reject) it.
  3. **Repeat**: Continue generating samples until you have enough that are consistent with the evidence.
  4. **Estimate Probabilities**: Once you have enough samples, estimate probabilities by looking at the proportion of samples that satisfy the desired conditions. For example, to estimate $P(X = x \mid Y = y)$, calculate the fraction of samples where both $X = x$ and $Y = y$, then normalize it to the number of samples where $Y = y$.

Rejection Sampling

$P(C)$

+c	0.5
-c	0.5



$P(S|C)$

+c	+s	0.1
	-s	0.9
-c	+s	0.5
	-s	0.5

$P(R|C)$

+c	+r	0.8
	-r	0.2
-c	+r	0.2
	-r	0.8

$P(W|S,R)$

+s	+r	+w	0.99
		-w	0.01
	-r	+w	0.90
		-w	0.10
-s	+r	+w	0.90
		-w	0.10
	-r	+w	0.01
		-w	0.99

Samples:

~~+c, -s, +r, +w~~

+c, +s, +r, +w

-c, +s, +r, -w

~~+c, -s, +r, +w~~

~~-c, -s, -r, +w~~

$P(+c \mid +s, +r) = \frac{1}{2}$

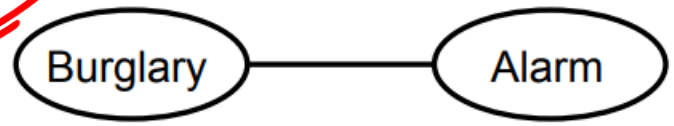
$= \frac{\text{No. of sample having } +c, +s, +r}{\text{No. of sample having } +s, +r}$

$P(+c \mid +r, +w)$

$= ?$

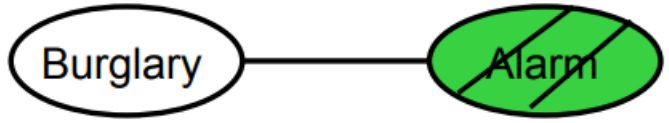
Likelihood Weighting

- Problem with rejection sampling:
  - If evidence is unlikely, you reject a lot of samples
  - You don't exploit your evidence as you sample
  - Consider P(B|+a)



-b, -a
-b, a
-b, -a
-b, -a
+b, +a

- Idea: fix evidence variables and sample the rest



-b +a
-b, +a
-b, +a
-b, +a
+b, +a

- Problem: sample distribution not consistent!
- Solution: weight by probability of evidence given parents

Likelihood Weighting

Likelihood weighting is a sampling technique used in probabilistic graphical models (such as Bayesian networks) to estimate conditional probabilities.
It is particularly useful when you have evidence (observed variables) and want to avoid the inefficiencies of methods like rejection sampling, which discards many samples.

In rejection sampling, if evidence is rare, many samples will be rejected, making it inefficient. Likelihood weighting addresses this issue by assigning weights to samples based on the likelihood of the evidence, rather than rejecting them.

Likelihood Weighting

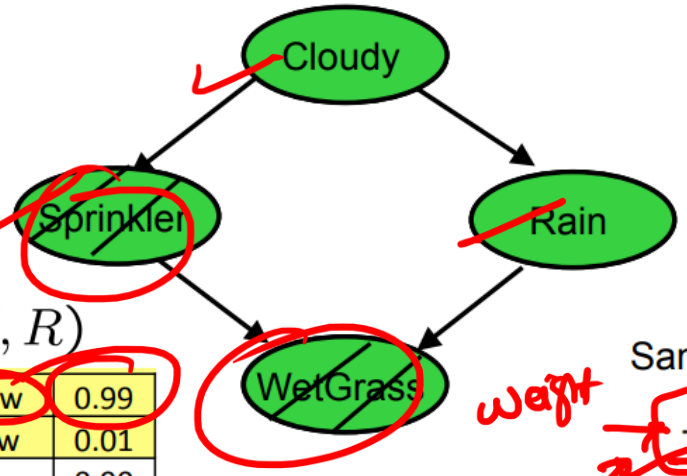
$P(+c \mid +s, +w)$

Evidence : +s, +w

$P(\text{Evidence} \mid \text{parent})$

$P(C)$

+c	0.5
-c	0.5



$P(S|C)$

+c	+s	0.1
	-s	0.9
-c	+s	0.5
	-s	0.5

$P(R|C)$

+c	+r	0.8
	-r	0.2
-c	+r	0.2
	-r	0.8

$P(W|S,R)$

+s	+r	+w	0.99
		-w	0.01
	-r	+w	0.90
		-w	0.10
-s	+r	+w	0.90
		-w	0.10
	-r	+w	0.01
		-w	0.99

Samples:

+c, +s, +r, +w

…

weight

$= P(+s \mid +c) \cdot P(+w \mid +s, +r) \times 1$

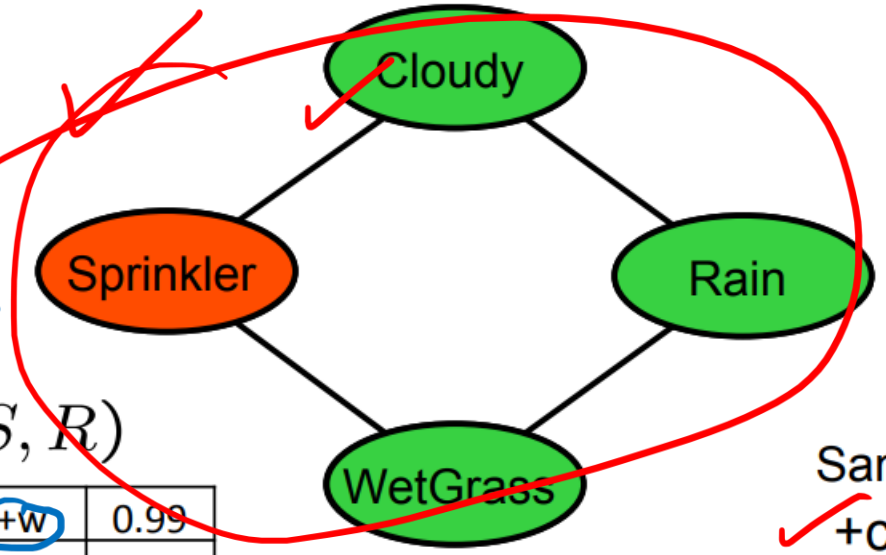
$= 0.1 \times 0.99 = 0.099$

$w = 1.0 \times 0.1 \times 0.99$

$= P(C) \times P(S|C) \times P(R|C) \times P(W|S,R)$

$P(C)$

+c	0.5
-c	0.5



$P(S|C)$

+c	+s	0.1
	-s	0.9
-c	+s	0.5
	-s	0.5

$P(R|C)$

+c	+r	0.8
	-r	0.2
-c	+r	0.2
	-r	0.8

$P(W|S,R)$

+s	+r	+w	0.99
		-w	0.01
	-r	+w	0.90
		-w	0.10
-s	+r	+w	0.90
		-w	0.10
	-r	+w	0.01
		-w	0.99

$P(+c|-s,+w)$ → evidence

$w_1 = P(+c) \cdot P(-s|+c) \times P(+r|+c) \times P(+w|-s,+r)$

$= 0.9 \times 0.2 = 0.18$

$w_2 = P(-s|-c) \times P(+w|-s,-r)$

$= 0.5 \times 0.01 = 0.005$

Samples:

+c, -s, +r, +w

~~+c, +s, +r, +w~~

~~-c, +s, +r, -w~~

+c, -s, +r, +w s1, $w_1 = 0.18$

-c, -s, -r, +w s2 $w_2 = 0.005$

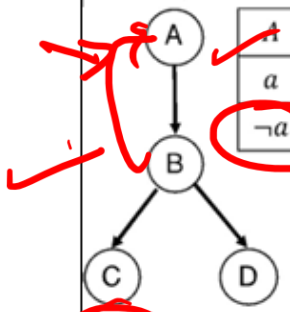
$P(+c|-s,+w) = \frac{0.18}{0.18 + 0.005}$

Q.32

Consider the Bayes Net containing four Boolean random variables (A, B, C, D), with the following convention:

$A = True \Rightarrow A = a$, and $A = False \Rightarrow A = \neg a$

and similarly for the other variables. The conditional probability tables for the nodes in the network are also indicated in the figure. The following samples were generated through likelihood weighting:



A	$P(A)$
a	1/5
$\neg a$	4/5

A	B	$P(B\|A)$
a	b	1/5
$\neg a$	b	1/2
a	$\neg b$	4/5
$\neg a$	$\neg b$	1/2

B	C	$P(C\|B)$
b	c	1/4
$\neg b$	c	3/4
b	$\neg c$	2/5
$\neg b$	$\neg c$	3/5

B	D	$P(D\|B)$
b	d	1/2
$\neg b$	d	4/5
b	$\neg d$	1/2
$\neg b$	$\neg d$	1/5

$s_1: (\neg a, \neg b, \neg c, \neg d)$; $s_2: (\neg a, b, \neg c, \neg d)$; $s_3: (\neg a, \neg b, \neg c, d)$; $s_4: (\neg a, b, \neg c, d)$

Estimate the likelihood weight of each sample and thereby estimate

$P(b|\neg a, \neg c)$

DA sample 24

CPT

P (evidence / parent)

$\neg a, \neg c$

$P(A,B,C,D) = P(A) \times P(B|A) \times P(C|B) \times P(D|B)$

$s_1 :- (\neg a, \neg b, \neg c, \neg d)$

$w_1 = P(\neg a) \times P(\neg b|\neg a) \times P(\neg c|\neg b) \times P(\neg d|\neg b)$

$= \frac{4}{5} \times \frac{3}{5} = \frac{12}{25} = 0.48$

$s_2 :- \neg a, b, \neg c, \neg d.$

$w_2 = P(\neg a) \times P(\neg c|b)$

$= \frac{4}{5} \times \frac{2}{5} = \frac{8}{25} = 0.32$

$s_3 :- (\neg a, \neg b, \neg c, d)$

$w_3 :- P(\neg a) \times P(\neg c|\neg b)$

$= \frac{4}{5} \times \frac{3}{5} = 0.48$

$w_4 = P(\neg a) \cdot P(\neg c|b) = \frac{4}{5} \times \frac{2}{5} = 0.32$

(A)	$s_1: 0.48, s_2: 0.32, s_3: 0.48, s_4: 0.32, P(b\|\neg a, \neg c) = 0.4$
(B)	$s_1: 0.48, s_2: 0.32, s_3: 0.48, s_4: 0.32, P(b\|\neg a, \neg c) = 0.64$
(C)	$s_1: 0.32, s_2: 0.48, s_3: 0.48, s_4: 0.32, P(b\|\neg a, \neg c) = 0.64$
(D)	$s_1: 0.48, s_2: 0.32, s_3: 0.32, s_4: 0.32, P(b\|\neg a, \neg c) = 0.4$

$$P(b \mid \neg a, \neg c) = \frac{P(b, \neg a, \neg c)}{P(\neg a, \neg c)}$$

query — evidence

$$= \frac{\text{Sum of wt. having 'b' (sample)}}{\text{Summation of wt. of all sample for given evidence}}$$

$$= \frac{0.32 + 0.32}{0.48 + 0.32 + 0.48 + 0.32}$$

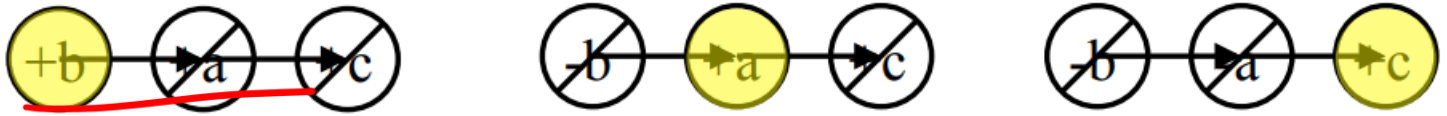
$$= \frac{0.64}{1.6} = \frac{64}{160}$$

$$= \frac{8}{20} = \frac{4}{10} = 0.4$$

Markov Chain Monte Carlo*

- Idea: instead of sampling from scratch, create samples that are each like the last one.

- Gibbs Sampling: resample one variable at a time, conditioned on the rest, but keep evidence fixed.



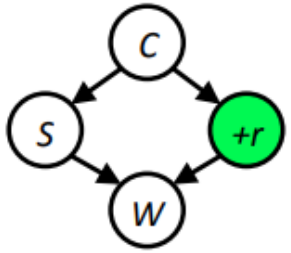
- Properties: Now samples are not independent (in fact they're nearly identical), but sample averages are still consistent estimators!

- What's the point: both upstream and downstream variables condition on evidence.

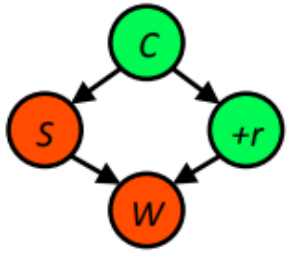
Gibbs Sampling Example

$$P(S|+r)$$

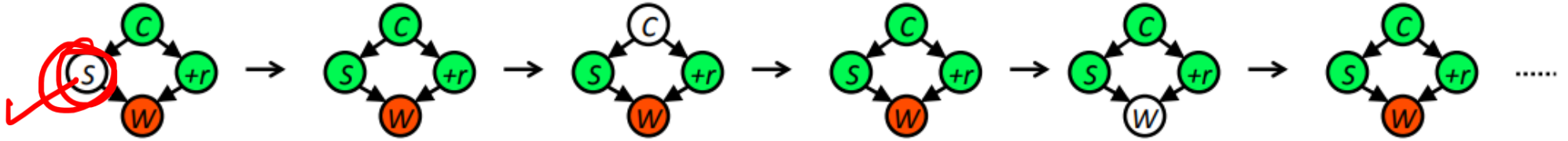
- Step 1: Fix evidence
  - R = +r



- Step 2: Initialize other variables
  - Randomly



- Steps 3: Repeat
  - Choose a non-evidence variable X
  - Resample X from P(X | all other variables)

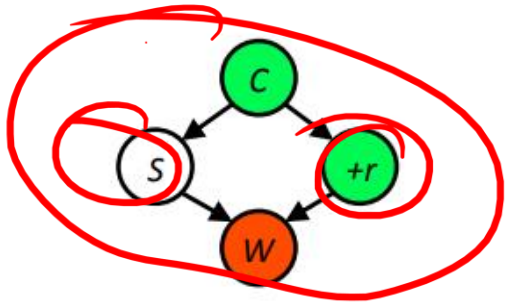


Sample from $P(S|+c,-w,+r)$ Sample from $P(C|+s,-w,+r)$ Sample from $P(W|+s,+c,+r)$

Sampling One Variable

- Sample from P(S | +c, +r, -w)

$$
\begin{aligned}
P(S|+c,+r,-w) &= \frac{P(S,+c,+r,-w)}{P(+c,+r,-w)} \\
&= \frac{P(S,+c,+r,-w)}{\sum_s P(s,+c,+r,-w)} \\
&= \frac{P(+c)P(S|+c)P(+r|+c)P(-w|S,+r)}{\sum_s P(+c)P(s|+c)P(+r|+c)P(-w|s,+r)} \\
&= \frac{P(+c)P(S|+c)P(+r|+c)P(-w|S,+r)}{P(+c)P(+r|+c)\sum_s P(s|+c)P(-w|s,+r)} \\
&= \frac{P(S|+c)P(-w|S,+r)}{\sum_s P(s|+c)P(-w|s,+r)}
\end{aligned}
$$



- Many things cancel out – only CPTs with S remain!
- More generally: only CPTs that have resampled variable need to be considered, and joined together

MCMC- Gibbs Sampling

Consider a Bayesian network with three binary variables X_1, X_2, and X_3, where each variable depends on the others in some way (e.g., a chain structure). We want to estimate $P(X_1 = 1 \mid X_3 = 1)$.

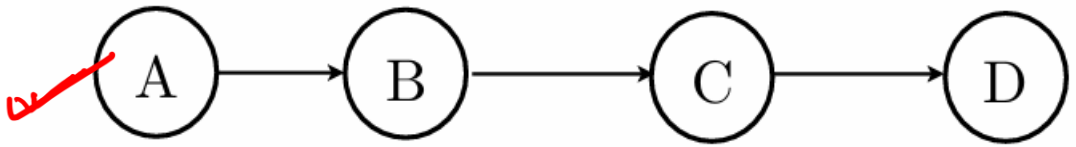
Steps:

1. **Initialize**: Start with arbitrary values for X_1, X_2, and X_3, say $(X_1 = 0, X_2 = 1, X_3 = 1)$.
2. **Iteratively Sample**:
   - Sample X_1 from its conditional distribution $P(X_1 \mid X_2, X_3)$.
   - Sample X_2 from $P(X_2 \mid X_1, X_3)$.
   - Sample X_3 from $P(X_3 \mid X_1, X_2)$.

   Keep repeating these steps. Since we want to compute the conditional probability given $X_3 = 1$, we always fix $X_3 = 1$ and only update X_1 and X_2.

3. **Burn-in Period**: Discard the first few samples to allow the chain to "forget" the initial values.
4. **Collect Samples**: After the burn-in period, collect samples of X_1 and X_2.
5. **Estimate**: Use the collected samples to estimate $P(X_1 = 1 \mid X_3 = 1)$.

Assume the following Bayes' net, and the corresponding distributions over the variables in the Bayes' net:



$P(A)$	
$-a$	3/4
$+a$	1/4

$P(B\|A)$		
$-a$	$-b$	2/3
$-a$	$+b$	1/3
$+a$	$-b$	4/5
$+a$	$+b$	1/5

$P(C\|B)$		
$-b$	$-c$	1/4
$-b$	$+c$	3/4
$+b$	$-c$	1/2
$+b$	$+c$	1/2

$P(D\|C)$		
$-c$	$-d$	1/8
$-c$	$+d$	7/8
$+c$	$-d$	5/6
$+c$	$+d$	1/6

(a) You are given the following samples:

$+a$	$+b$	$-c$	$-d$
$+a$	$-b$	$+c$	$-d$
$-a$	$+b$	$+c$	$-d$
$-b$	$+c$	$-d$	

$+a$	$-b$	$-c$	$+d$
$+a$	$+b$	$+c$	$-d$
$-a$	$+b$	$-c$	$+d$
$+a$	$-b$	$+c$	$-d$

(i) Assume that these samples came from performing Prior Sampling, and calculate the sample estimate of $P(+c)$.

5/8

(ii) Now we will estimate $P(+c \mid +a, -d)$. Above, clearly cross out the samples that would **not** be used when doing Rejection Sampling for this task, and write down the sample estimate of $P(+c \mid +a, -d)$ below.

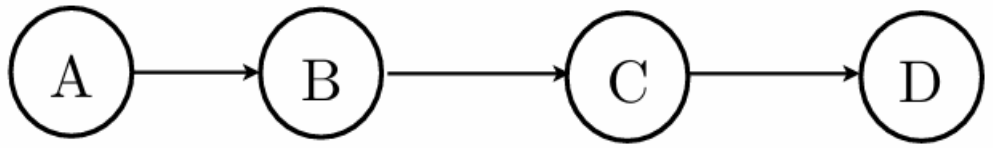
2/3

$P(+c) = \frac{\text{No. of samples having } +c}{\text{Total no. of samples}} = \frac{5}{8}$

$P(+c \mid +a, -d) = \frac{\text{No. of samples having } +c}{\text{No. of samples satisfying given evidence}} = \frac{2}{3}$

Example: Sampling

Assume the following Bayes' net, and the corresponding distributions over the variables in the Bayes' net:



$P(A)$	
$-a$	3/4
$+a$	1/4

$P(B\|A)$		
$-a$	$-b$	2/3
$-a$	$+b$	1/3
$+a$	$-b$	4/5
$+a$	$+b$	1/5

$P(C\|B)$		
$-b$	$-c$	1/4
$-b$	$+c$	3/4
$+b$	$-c$	1/2
$+b$	$+c$	1/2

$P(D\|C)$		
$-c$	$-d$	1/8
$-c$	$+d$	7/8
$+c$	$-d$	5/6
$+c$	$+d$	1/6

(b) Using Likelihood Weighting Sampling to estimate $P(-a \mid +b, -d)$, the following samples were obtained. Fill in the weight of each sample in the corresponding row.

Sample				Weight
$-a$	$+b$	$+c$	$-d$	$P(+b \mid -a)P(-d \mid +c) = 1/3 * 5/6 = 5/18 = 0.277$
$+a$	$+b$	$+c$	$-d$	$P(+b \mid +a)P(-d \mid +c) = 1/5 * 5/6 = 5/30 = 1/6 = 0.17$
$+a$	$+b$	$-c$	$-d$	$P(+b \mid +a)P(-d \mid -c) = 1/5 * 1/8 = 1/40 = 0.025$
$-a$	$+b$	$-c$	$-d$	$P(+b \mid -a)P(-d \mid -c) = 1/3 * 1/8 = 1/24 = 0.042$

Example: Sampling

(c) From the weighted samples in the previous question, estimate $P(-a \mid +b, -d)$.

$$\frac{5/18+1/24}{5/18+5/30+1/40+1/24} = 0.625$$

(d) Which query is better suited for likelihood weighting, $P(D \mid A)$ or $P(A \mid D)$? Justify your answer in one sentence.

$P(D \mid A)$ is better suited for likelihood weighting sampling, because likelihood weighting conditions only on upstream evidence.

Example: Sampling

(e) Recall that during Gibbs Sampling, samples are generated through an iterative process. Assume that the only evidence that is available is $A = +a$. Clearly fill in the circle(s) of the sequence(s) below that could have been generated by Gibbs Sampling.

1. $A = +a$ → Fixed
   evidence is fixed
2. one variable at a time

Sequence 1

1 :	$+a$	$-b$	$-c$	$+d$
2 :	$+a$	$-b$	$-c$	$+d$
3 :	$+a$	$-b$	$+c$	$+d$

Sequence 2

1 :	$+a$	$-b$	$-c$	$+d$
2 :	$+a$	$-b$	$-c$	$-d$
3 :	$-a$	$-b$	$-c$	$+d$

Sequence 3

1 :	$+a$	$-b$	$-c$	$+d$
2 :	$+a$	$-b$	$-c$	$-d$
3 :	$+a$	$+b$	$-c$	$-d$

Sequence 4

1 :	$+a$	$-b$	$-c$	$+d$
2 :	$+a$	$-b$	$-c$	$-d$
3 :	$+a$	$+b$	$-c$	$+a$

Gibbs sampling updates one variable at a time and never changes the evidence.

The first and third sequences have at most one variable change per row, and hence could have been generated from Gibbs sampling. In sequence 2, the evidence variable A changed. In sequence 4, the second and third samples have both B and D changing.