INFO 510 Bayesian Modelling and Inference

Lecture 8 – Bayesian Networks and Inference methods

Dr. Kunal Arekar
College of Information
University of Arizona, Tucson

September 21, 2025

In previous lecture…

Graphical models in probabilistic inference

Types of graphical models – undirected, directed, DAGs, etc

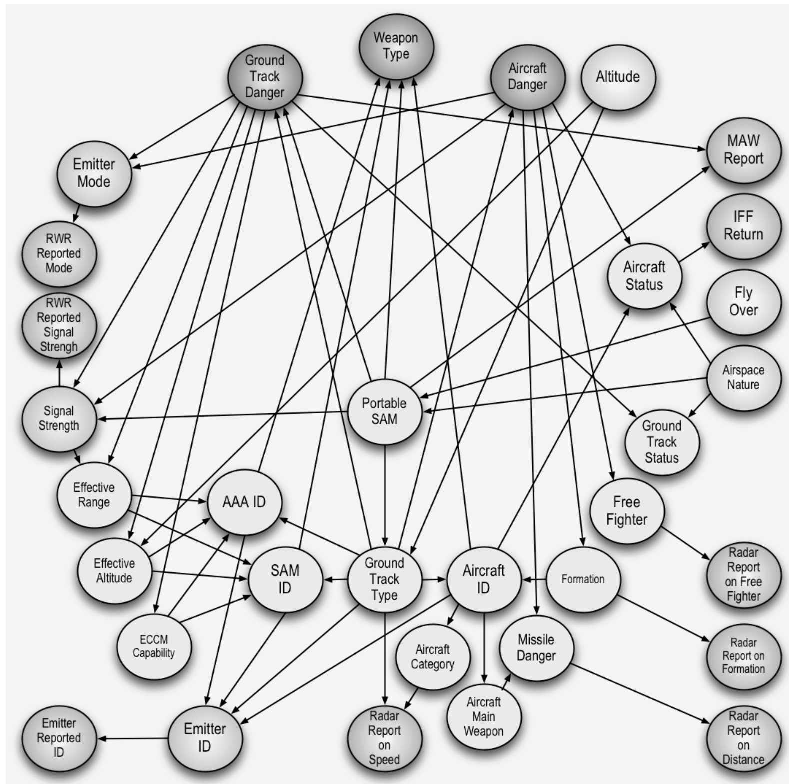
Graphical models used in Bayesian inference –
e.g., HMM, BN, etc (*all are different types of Bayesian Networks)

These models are instrumental in a wide range of applications, from medical diagnosis to machine learning

Inference methods in Bayesian networks

Bayesian networks

Also known as Bayesian Belief Network (BBN)

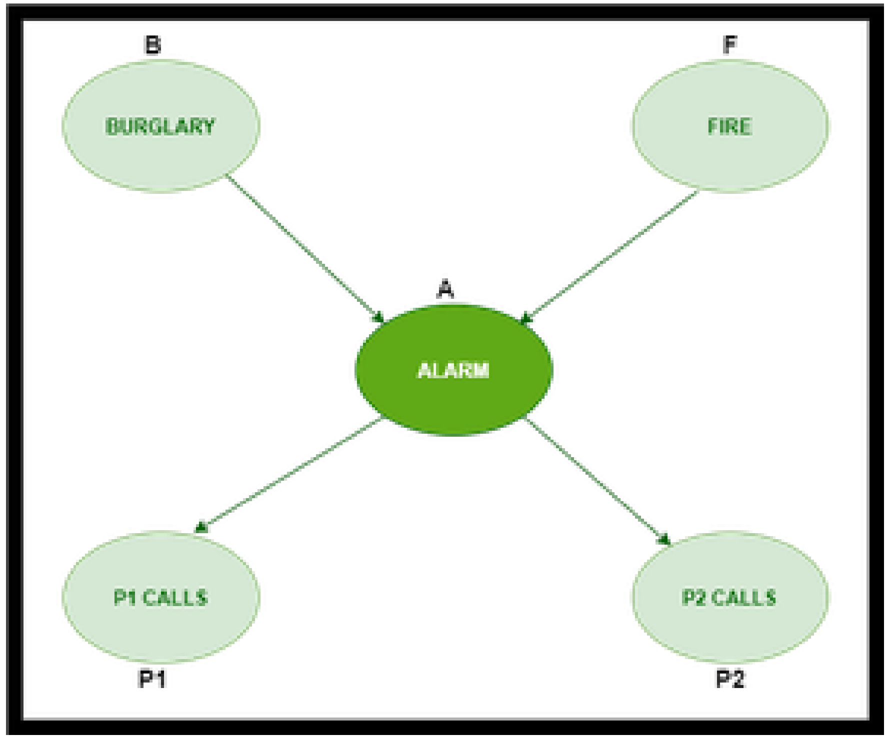


- valuable tools for understanding and solving problems involving uncertain events
- Also known as Bayes networks, belief networks, decision networks, or Bayesian models

Question

Fires and burglaries are independent events that will cause an alarm to go off. Suppose you hear an alarm. How does hearing on the radio that there's a fire change your beliefs?

- it increases the probability of burglary
- it decreases the probability of burglary
- it does not change the probability of burglary



‘P1’ is true, ‘P2’ is true when the alarm ‘A’ rang, but no burglary ‘B’ and no fire ‘F’ has occurred.

P (P1, P2, A, ~B, ~F)

Burglary 'B' –

- P (B=T) = 0.001
- P (B=F) = 0.999)

Fire 'F'

- P (F=T) = 0.002
- P (F=F) = 0.998

Alarm 'A' –

B	F	P (A=T)	P (A=F)
T	T	0.95	0.05
T	F	0.94	0.06
F	T	0.29	0.71
F	F	0.001	**0.999**

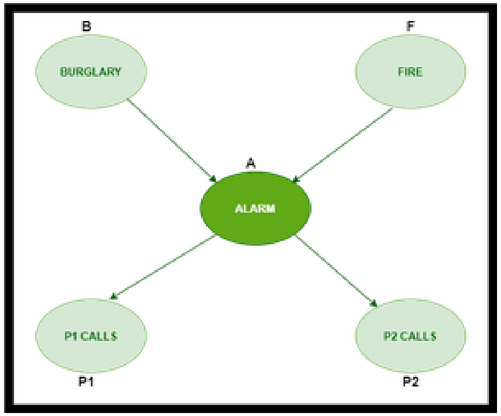
Person 'P1' –

A	P (P1=T)	P (P1=F)
T	**0.95**	0.05
F	0.05	0.95

Person 'P2' –

A	P (P2=T)	P (P2=F)
T	**0.80**	0.20
F	0.01	0.99

‘P1’ is true, ‘P2’ is true when the alarm ‘A’ rang, but no burglary ‘B’ and fire ‘F’ has occurred.



P (P1, P2, A, ~B, ~F)

P (P1, P2, A, ~B, ~F) = P (P1/A) * P (P2/A) * P (A/~B~F) * P (~B) * P (~F)

= 0.95 * 0.80 * 0.001 * 0.999 * 0.998

= 0.00075

Inference in Bayesian networks

Bayesian Networks (BNs) are graphical models for probabilistic inference, representing a set of variables and their conditional dependencies via a directed acyclic graph (DAG).

What is inference?

Inference in Bayesian Networks involves answering probabilistic queries about the network

Most common types of inference:

- **Marginalization**: Determining the probability distribution of a subset of variables, ignoring the values of all other variables
- **Conditional Probability**: Computing the probability distribution of a subset of variables given evidence observed on other variables

Mathematically, $P(X|E = e)$

What is an inference?

In the simplest form, inference is answering questions like *What is the probability of a hidden variable, given some observed data?*

E.g.,

- a card game with some cards are hidden
- you know the rules of the game (the structure)
- you can see some cards (the observed data)
- make an educated guess about what the hidden cards

Exact inference - calculating all possible card combinations, considering all the rules, to figure out exactly what those hidden cards are.

Inference methods in Bayesian networks

$$P(\mathbf{Y} \mid \mathbf{E} = \mathbf{e}) = \frac{P(\mathbf{Y}, \mathbf{e})}{P(\mathbf{e})} = \frac{\sum_w P(\mathbf{y}, \mathbf{e}, \mathbf{w})}{\sum_{y,w} P(\mathbf{e})}$$

Problem of Inference

*The effectiveness of inference depends on the network structure

#Sum out (marginalise) nuisance variables and then divide

y^1	e^1	w^1	0.18
y^1	e^1	w^2	0.09
y^1	e^2	w^1	0.01
y^1	e^2	w^2	0.37
y^2	e^1	w^1	0.13
y^2	e^1	w^2	0.16
y^2	e^2	w^1	0.05
y^2	e^2	w^2	0.01

$P(y, e, w)$

$\rightarrow \sum_w P(y, e, w) \rightarrow$

y^1	e^1	0.27
y^1	e^2	0.37
y^2	e^1	0.29
y^2	e^2	0.06

$P(y, e)$

y^1	e^1	0.27
y^1	e^2	0.37
y^2	e^1	0.29
y^2	e^2	0.06

$P(y, e)$

$\rightarrow \sum_y P(y, e) \rightarrow$

e^1	0.57
e^2	0.43

$P(e)$

$$P(y^1 \mid e^1) = \frac{P(y^1, e^1)}{P(e^1)} = \frac{0.27}{0.57} = 0.47$$

Note that $P(y^1 \mid e^1)$ is **not the same thing** as $P(y^1, e^1)$

Exact inference methods

Exact inference refers to methods that provide precise, mathematically derived results based on the underlying probability distributions

crucial in fields where precision and reliability are critical

Different algorithms:

- The elimination algorithm (variable elimination)
- Message-passing algorithm (sum-product, belief propagation)
- The junction tree algorithms

Exact inference is not always feasible

Leads to exponential blow-up of joint distributions

This makes exact inference in graphical models NP-hard

To overcome this – approximate inference

Inference methods

Exact inference methods

Exact inference refers to methods that provide precise, mathematically derived results based on the underlying probability distributions.

- The elimination algorithm
- Message-passing algorithm (sum-product, belief propagation)
- The junction tree algorithms

Approximate inference techniques

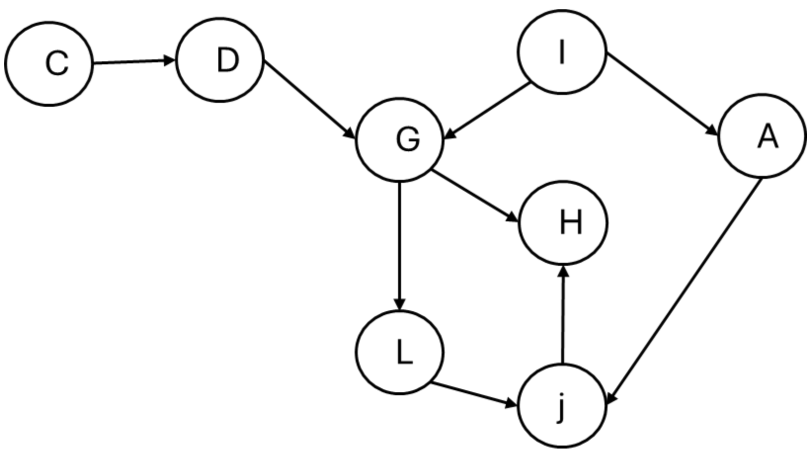
methods that provide estimates or approximations of the true distribution or parameters, often when exact inference is computationally infeasible.

- Stochastic simulation / sampling methods
- Markov chain Monte Carlo methods
- Variational algorithms

The elimination algorithm (variable elimination)

Variable elimination can be thought of like solving a maze. You remove pathways that don't lead anywhere useful, step by step, until you find the solution.

1. $P(A) = \sum_B P(A, B)$
2. $\sum_C \boxed{P(A)} P(B, C) = \boxed{P(A)} \sum_C P(B, C)$



Calculate P(J)

$$P(C, D, I, G, A, L, J, H) = P(C)P(D \mid C)P(I) \\ P(G \mid I, D)P(A \mid I)P(L \mid G) \\ P(J \mid L, A)P(H \mid G, J)$$

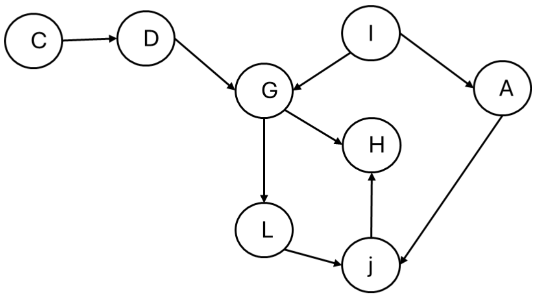
As a set of factors each with a scope:

$$P(C, D, I, G, A, L, J, H) = \phi_C(C)\phi_D(C, D)\phi_I(I) \\ \phi_G(G, I, D)\phi_A(A, I)\phi_L(L, G) \\ \phi_J(J, L, A)\phi_H(H, G, J)$$

Order of elimination: **C D I H G A L**

$$P(J) = \sum_{C,D,I,H,G,A,L} \phi_C(C)\phi_D(C,D)\phi_I(I)\phi_G(G,I,D)\phi_A(A,I)$$
$$\phi_L(L,G)\phi_J(J,L,A)\phi_H(H,G,J)$$

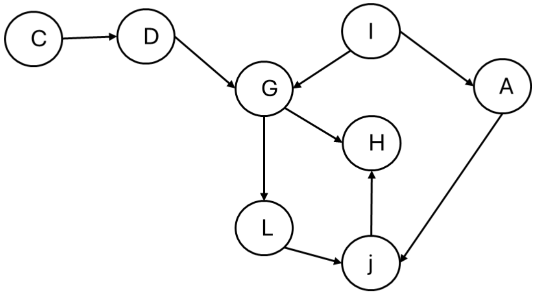
$$= \sum_{L} \sum_{A} \phi_J(J,L,A) \sum_{G} \phi_L(L,G) \sum_{H} \phi_H(H,G,J)$$
$$\sum_{I} \phi_I(I)\phi_A(A,I) \sum_{D} \phi_G(G,I,D) \sum_{C} \phi_C(C)\phi_D(C,D)$$



Order of elimination: **C D I H G A L**

$$P(J) = \sum_{C,D,I,H,G,A,L} \phi_C(C)\phi_D(C,D)\phi_I(I)\phi_G(G,I,D)\phi_A(A,I)$$
$$\phi_L(L,G)\phi_J(J,L,A)\phi_H(H,G,J)$$

$$= \sum_{L}\sum_{A} \phi_J(J,L,A) \sum_{G} \phi_L(L,G) \sum_{H} \phi_H(H,G,J)$$
$$\sum_{I} \phi_I(I)\phi_A(A,I) \sum_{D} \phi_G(G,I,D) \sum_{C} \phi_C(C)\phi_D(C,D)$$



$$= \sum_{L}\sum_{A} \phi_J(J,L,A) \sum_{G} \phi_L(L,G) \sum_{H} \phi_H(H,G,J)$$
$$\sum_{I} \phi_I(I)\phi_A(A,I) \sum_{D} \phi_G(G,I,D) \sum_{C} \psi_1(C,D)$$

$$= \sum_{L}\sum_{A} \phi_J(J,L,A) \sum_{G} \phi_L(L,G) \sum_{H} \phi_H(H,G,J)$$
$$\sum_{I} \phi_I(I)\phi_A(A,I) \sum_{D} \phi_G(G,I,D)\, \tau_1(D)$$

$$= \sum_L \sum_A \phi_J(J,L,A) \sum_G \phi_L(L,G) \sum_H \phi_H(H,G,J)$$
$$\sum_I \phi_I(I)\phi_A(A,I) \sum_D \phi_G(G,I,D)\, \tau_1(D)$$

$$= \sum_L \sum_A \phi_J(J,L,A) \sum_G \phi_L(L,G) \sum_H \phi_H(H,G,J)$$
$$\sum_I \phi_I(I)\phi_A(A,I) \sum_D \phi_G(G,I,D)\tau_1(D)$$

$$= \sum_L \sum_A \phi_J(J,L,A) \sum_G \phi_L(L,G) \sum_H \phi_H(H,G,J)$$
$$\sum_I \phi_I(I)\phi_A(A,I) \sum_D \psi_2(G,I,D)$$

$$= \sum_L \sum_A \phi_J(J,L,A) \sum_G \phi_L(L,G) \sum_H \phi_H(H,G,J)$$
$$\sum_I \phi_I(I)\phi_A(A,I)\, \tau_2(G,I)$$

= …… ……… ……… ………… …………..

= …… ……… ……… ………… …………..

$$= \sum_L \tau_7(J,L)$$

$$= \tau_7(J)$$

We can summarise the run of VE neatly in a table:

Step	Var Eli	Factors used	Variable Involved	New Factor
1	C	$\phi_C(C), \phi_D(D, C)$	C,D	$\tau_1(D)$
2	D	$\phi_G(G, I, D), \tau_1(D)$	G,I,D	$\tau_2(G, I)$
3	I	$\phi_I(I), \phi_A(A, I), \tau_2(G, I)$	G,A,I	$\tau_3(G, A)$
4	H	$\phi_H(H, G, J)$	H,G,J	$\tau_4(G, J)$
5	G	$\tau_4(G, J), \tau_3(G, A), \phi_L(L, G)$	G,J,L,A	$\tau_5(J, L, A)$
6	A	$\tau_5(J, L, A), \phi_J(J, L, A)$	J,L,A	$\tau_6(J, L)$
7	L	$\tau_6(J, L)$	J,L	$\tau_7(J)$

Here is another run of VE with a different elimination ordering:

Step	Var Eli	Factors used	Variable Involved	New Factor
1	G	$\phi_G(G, I, D), \phi_L(L, G), \phi_H(H, G, J)$	G,I,D,L,J,H	$\tau_1(I, D, L, J, H)$
2	I	$\phi_I(I), \phi_A(A, I), \tau_1(I, D, L, A, J, H)$	A,I,D,L,J,H	$\tau_2(D, L, A, J, H)$
3	A	$\phi_J(J, L, A), \tau_2(D, L, A, J, H)$	D,L,A,J,H	$\tau_3(D, L, J, H)$
4	L	$\tau_3(D, L, J, H)$	D,L,J,H	$\tau_4(D, J, H)$
5	H	$\tau_4(D, J, H)$	D,J,H	$\tau_5(D, J)$
6	C	$\phi_C(C), \phi_D(D, C)$	D,J,C	$\tau_6(D)$
7	D	$\tau_5(D, J), \tau_6(D)$	D,J	$\tau_7(J)$

Message-passing algorithm (sum-product belief propagation)

iterative process in which neighboring variables “talk” to each other, passing messages such as: “I (variable x3) think that you (variable x2) belong in these states with various likelihoods…”

After enough iterations, this series of conversations is likely to converge to a consensus that determines the marginal probabilities of all the variables.

Estimated marginal probabilities are called beliefs.

By leveraging the local structure of a graphical model, we can reduce the complexity of inference

It works by passing messages between nodes in a graph until the beliefs (marginal probabilities) about the variables converge.

The Belief Propagation Algorithm

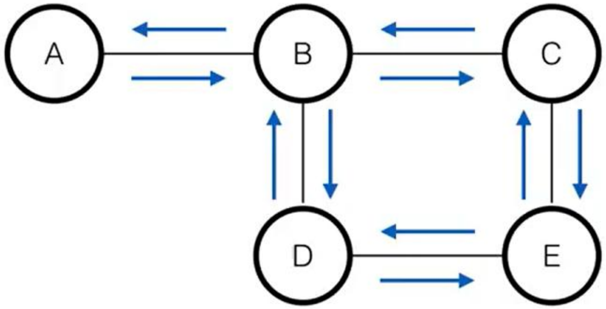
an iterative, message-passing process that allows nodes in a graph to communicate information about their beliefs (probabilities) with each other

Messages: Each node sends a "message" to its neighbors. These messages represent the current state of belief about the node's variable, taking into account the information it has received from other neighbors

Belief: The belief at a node is the estimate of the marginal probability of that variable. It is computed by combining all incoming messages

Update Rule: The way messages are computed and beliefs are updated is determined by the rules of probability, specifically marginalization

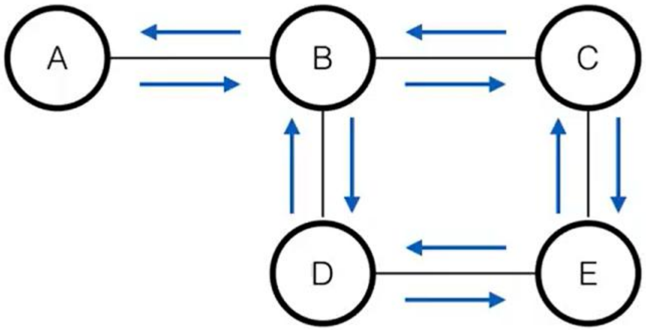
sum-product belief propagation



Passing messages between variables in a network.

Messages are the functions of the variables receiving messages

$$m_{s\to t}(x_t) := \sum_{x_s}\left(\phi_{st}(x_s, x_t)\prod_{u\in \text{neighbors}(s)\setminus t} m_{u\to s}(x_s)\right)$$

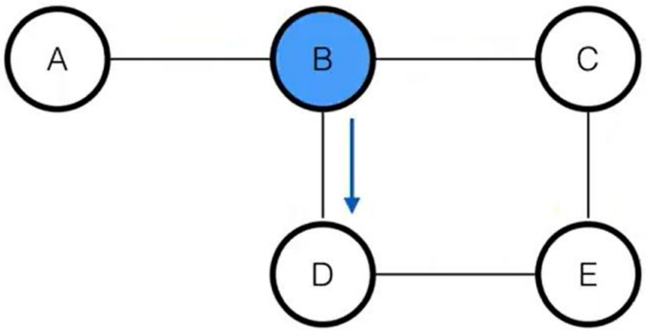


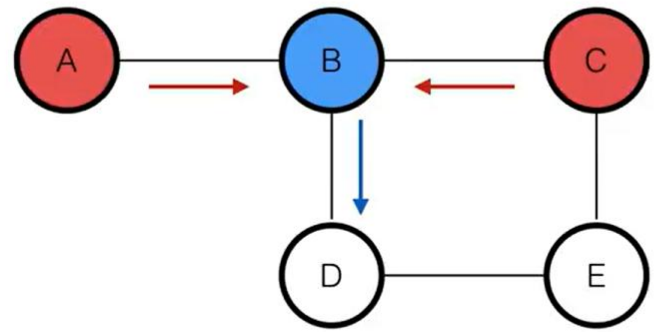
$$m_{s \to t}(x_t) := \sum_{x_s} \left(\phi_{st}(x_s, x_t) \prod_{u \in \text{neighbors}(s) \setminus t} m_{u \to s}(x_s) \right)$$

Compute all messages between all of the adjacent variables

Compute beliefs; these are estimated marginal probabilities of each individual variable by multiplying together all the incoming messages.

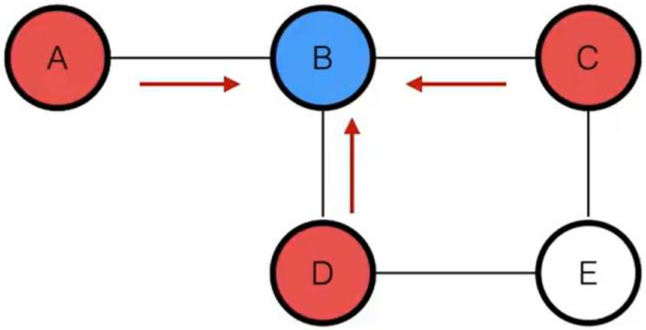
$$b_t(x_t) \propto \prod_{s \in \text{neighbors}(t)} m_{t \to s}(x_s)$$





$$m_{s\to t}(x_t) := \sum_{x_s} \left(\phi_{st}(x_s, x_t) \prod_{u \in \text{neighbors}(s) \setminus t} m_{u \to s}(x_s) \right)$$

$$m_{B\to D}(x_D) = \sum_{x_B} \phi(x_B, x_C) \times m_{A\to B}(x_B) \times m_{C\to B}(x_B)$$



Compute beliefs or marginal probabilities

$$b_t(x_t) \propto \prod_{s \in \text{neighbors}(t)} m_{t \to s}(x_s)$$

$$b_B(x_B) \propto (m_{A \to B}(x_B))(m_{C \to B}(x_B))(m_{D \to B}(x_B))$$

BP always converges to the exact marginal probabilities at each node after a finite number of iterations

Therefore, BP can achieve exact inference in tree-structured graphs or graphs without loops

When BP is applied to graphs with cycles (loopy graphs), it's called loopy belief propagation (LBP)

The LBP algorithm may not converge, and even if it does, the results may not represent the exact marginals; provides approximate inference

Challenges and Applications of exact inference

Computational Challenges:
Curse of dimensionality.
E.g., Card game with 50 hidden cards (too many combinations).

Applications:
Medical diagnosis (inferring diseases from symptoms)
AI and robotics (decision-making under uncertainty)
Natural language processing (predicting sentence structure)

When to Use Approximate Inference:
Trade-offs between precision and computational cost.
When approximation becomes necessary in large, complex networks.

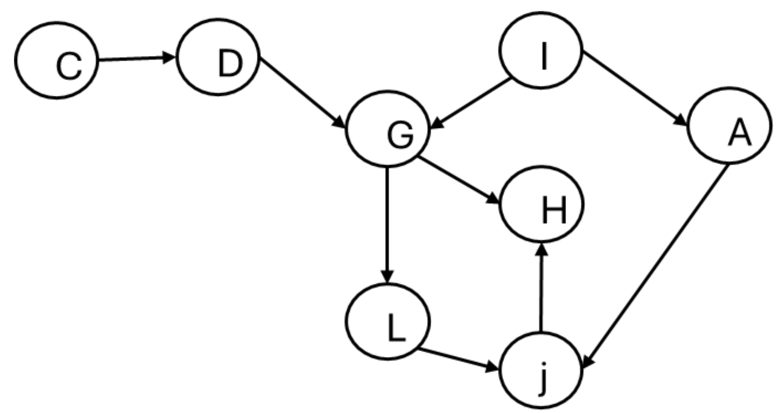
Exact inference methods play a crucial role in various fields where making precise probabilistic predictions or decisions based on observed data is critical.

Medical Diagnosis

making accurate diagnoses based on patient symptoms and test results is a critical task

Scenario - patient visits a doctor with a high fever and sore throat.
Possible causes - flu, a bacterial infection, or COVID-19
Need further information to make a diagnosis

The Bayesian network would have nodes representing each of these diseases and symptoms like fever, sore throat, cough, and test results (e.g., a positive PCR test).

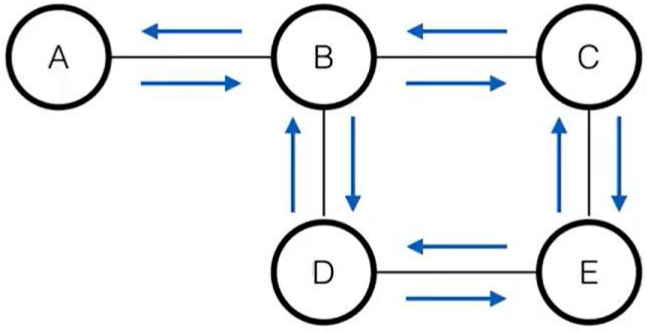


Inference Process:
Conduct rapid flu test – negative.

Using exact inference, the system can update the probabilities of the possible diseases.

If the test result lowers the probability of the flu, but a positive COVID-19 test is yet to be received, the system uses belief propagation to pass this updated information throughout the network.

As more symptoms and test results are incorporated, the system narrows down the most likely diagnosis.



Exact inference methods play a crucial role in various fields where making precise probabilistic predictions or decisions based on observed data is critical.

Fault Diagnosis in Engineering Systems

In complex engineering systems like aircraft engines, nuclear power plants, or even industrial robots, ensuring that the system is functioning correctly is paramount.

Here, Bayesian networks or other probabilistic models can be used to represent components and their interactions.

E.g., nuclear power plant where temperature sensors monitor the coolant system

Bayesian network – nodes as pumps, valves, and temperature levels

A spike in coolant temperature could indicate a potential fault in the cooling system, but it's unclear whether this is due to a valve malfunction or a faulty pump.

Using **variable elimination**, the system can analyze the data from multiple sensors and infer the most likely cause of the temperature spike.

By eliminating irrelevant variables (e.g., unrelated sensor readings), the system focuses on key components like the valve and the pump, eventually pinpointing the faulty one.

Once identified, technicians can fix the issue before a more serious failure occurs.