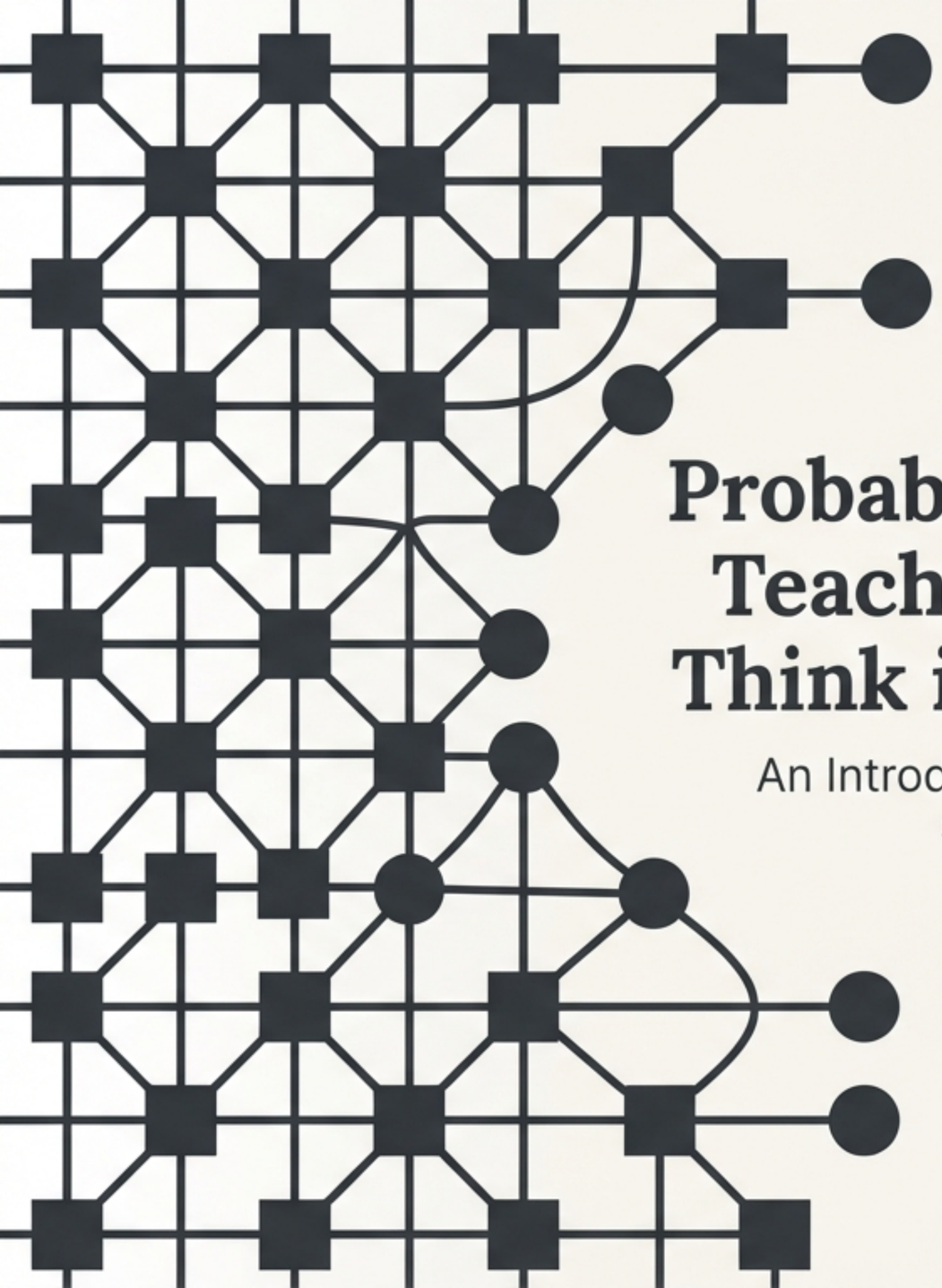
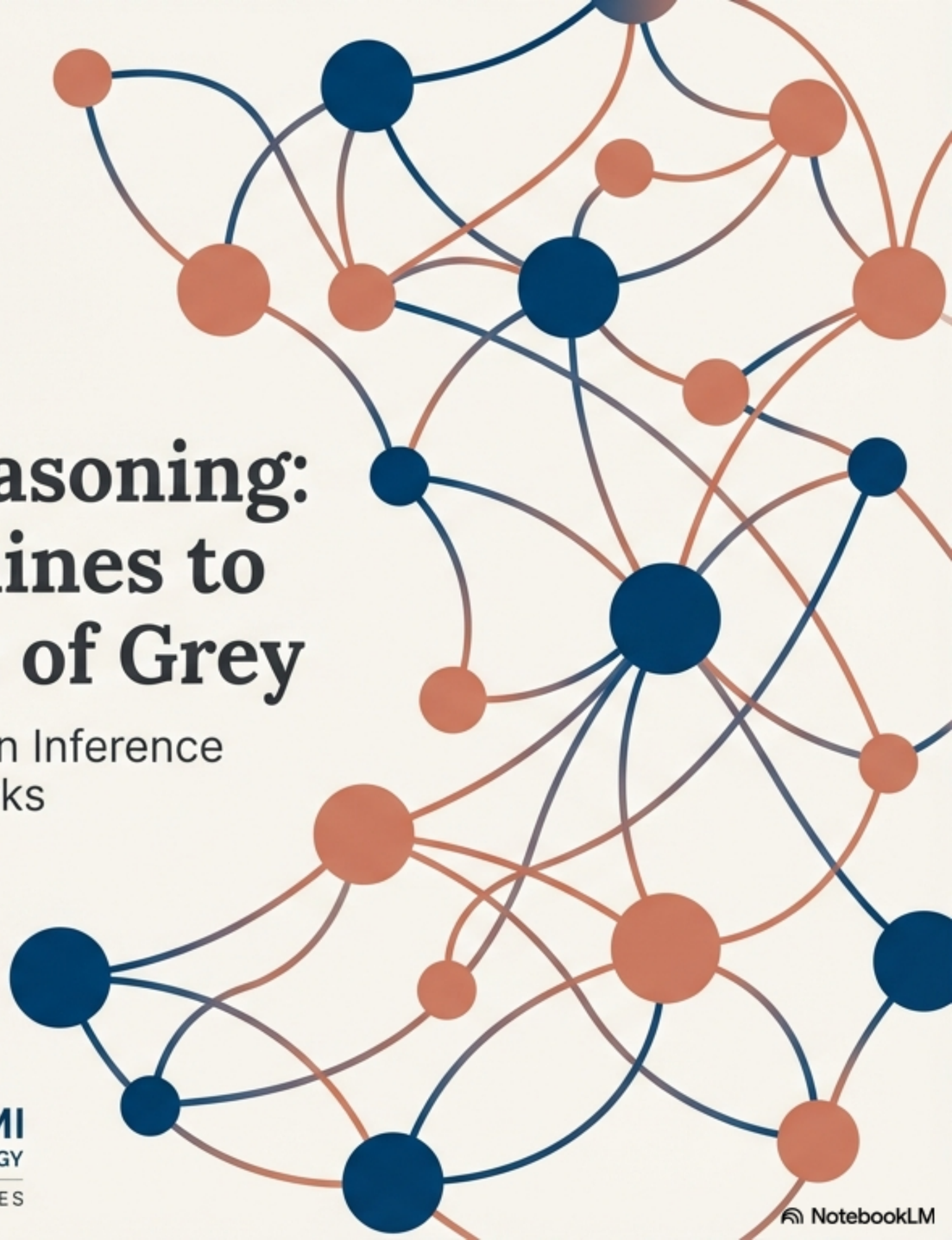


Probabilistic Reasoning: Teaching Machines to Think in Shades of Grey

An Introduction to Bayesian Inference and Belief Networks



RAJALAKSHMI INSTITUTE OF TECHNOLOGY

BELIEVE IN THE POSSIBILITIES

The world is uncertain; purely logical agents are brittle and quickly overwhelmed.

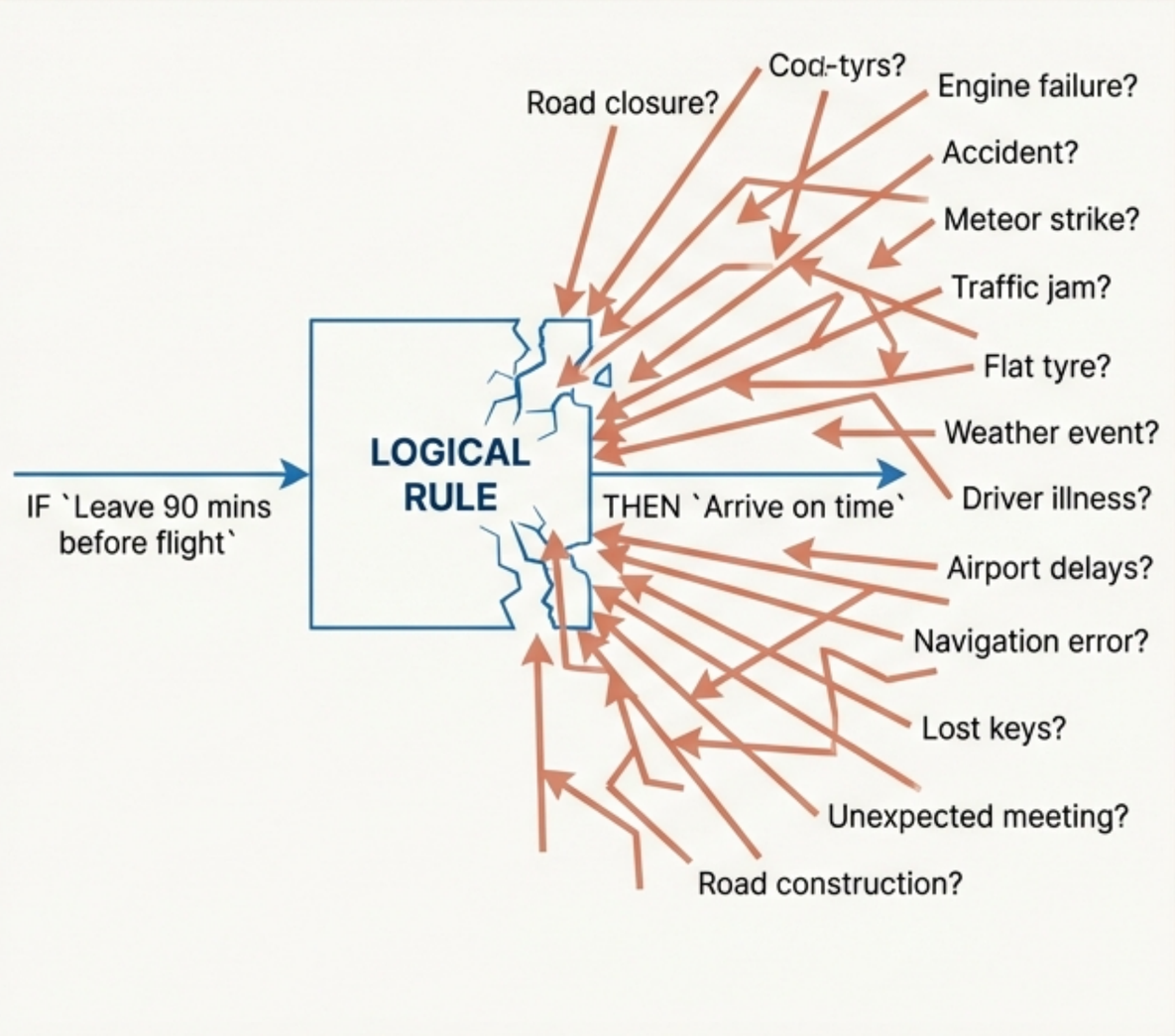
AI agents must act in environments that are partially observable and non-deterministic. Keeping track of every possibility leads to unwieldy belief states.

The Qualification Problem:

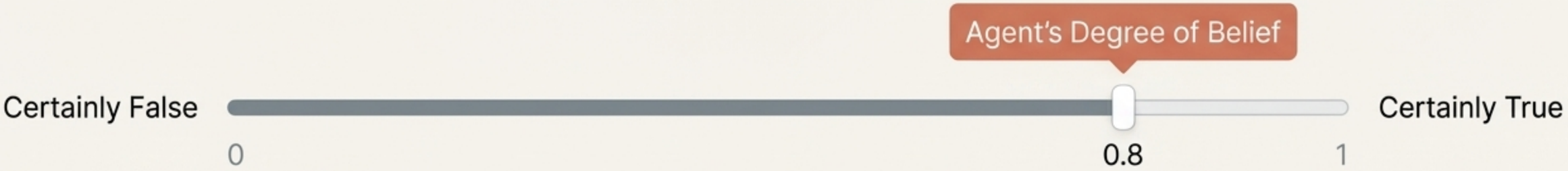
Logical rules require a near-infinite list of exceptions. A plan to drive to the airport works *unless* the car breaks down, there's an accident, a road closure, a meteor strike, etc. It is impossible to conclude with certainty that a plan will succeed.

Why Logic Fails in Complex Domains:

- **Laziness**: Too much work to list every exception and antecedent.
- **Theoretical Ignorance**: No complete theory exists for most domains (e.g., medicine).
- **Practical Ignorance**: We can't run every necessary test on a patient or system.



Instead of certainty, we use probability to represent a degree of belief.



Probability theory allows an agent to manage uncertainty by assigning a numerical degree of belief (from 0 to 1) to propositions. This is not a weaker form of logic; it's a richer language that enables rational decision-making under uncertainty. To make rational choices, we combine belief with preference using **Decision Theory**.

Decision Theory = Probability Theory + Utility Theory. An agent is rational if it chooses the action with the **Maximum Expected Utility (MEU)**, which balances the likelihood of an outcome with its desirability.

A formal language to express and manipulate degrees of belief.

Random Variable: A variable representing an uncertain quantity (e.g., `Weather`, `Cavity`). It has a domain of possible values.

Probability Distribution: Assigns a probability to every possible value of a random variable. The sum of probabilities must be 1. For example: **P**(Weather) = <0.6, 0.1, 0.29, 0.01> for the values <sun, rain, cloud, snow>.

Prior Probability `P(a)`: The initial degree of belief in a proposition `a` in the absence of any other information.

Conditional Probability `P(a|b)`: The updated degree of belief in `a`, `*given* that all we know is `b`. It is defined by the **Product Rule**: $P(a \wedge b) = P(a|b)P(b)$.

The joint distribution is a complete model of the domain, but it scales exponentially.

The **full joint probability distribution** specifies the probability for every possible combination of variable values in the domain (a complete "possible world").

From it, we can compute any probability for any proposition by **marginalisation** (summing out the variables we are not interested in).

The Curse of Dimensionality: For a domain with N boolean variables, the joint distribution table requires $\mathbf{2^N}$ entries.

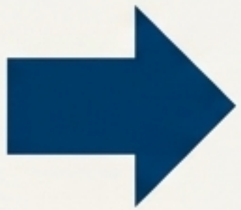
Adding a single `Weather` variable (4 values) to our 3-variable dental world (`Cavity`, `Toothache`, `Catch`) multiplies the table size by 4 (from 8 to 32 entries).

This is intractable for real-world problems.

P(Cavity, Toothache, Catch)

0.108	0.012
0.012	0.072
0.072	0.008
0.008	0.016
0.016	0.064
0.144	0.144
0.576	0.576

8 entries



P(Cavity, Toothache, Catch, Weather)

Cavity	0.108	0.012	0.072	0.108
Cavity	0.012	0.072	0.008	0.072
Cavity	0.072	0.008	0.016	0.016
Cavity	0.008	0.016	0.064	0.064
Cavity				0.144
Cavity				0.576
Cavity				0.008
Cavity				0.064
Cavity	0.012	0.008	0.064	0.144
Cavity	0.012	0.008	0.044	0.576
Cavity	0.072	0.072	0.008	0.144
Weathy	0.008	0.064	0.076	0.576
Weather				

32 entries... and growing

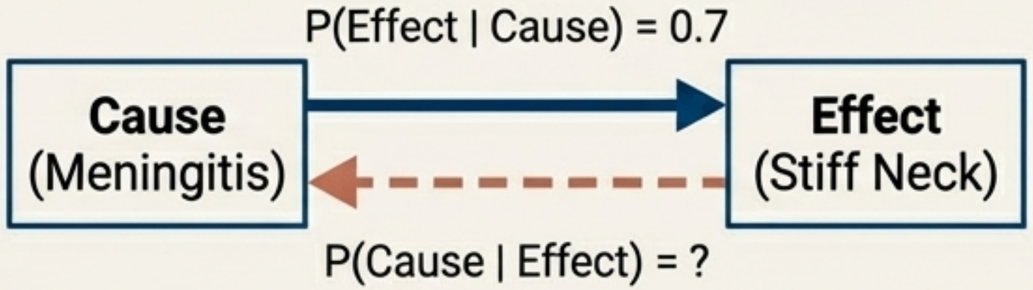
Bayes' Rule lets us update our beliefs by inverting conditional probabilities.

$$P(b|a) = [P(a|b) * P(b)] / P(a)$$

Posterior = (Likelihood * Prior) / Evidence

Derived from the product rule, Bayes' Rule is the foundation of most modern AI systems for probabilistic inference.

Diagnostic vs. Causal Reasoning: We often have knowledge in the causal direction P(effect|cause) but need to perform diagnostic reasoning P(cause|effect).



Example: Meningitis & Stiff Neck:
A doctor knows causal info: P(stiff neck | meningitis) = 0.7.
They also know priors: P(meningitis) = 1/50,000 and P(stiff neck) = 0.01.
Using Bayes' rule, they can diagnose: P(meningitis | stiff neck) = (0.7 * 1/50000) / 0.01 = 0.0014.
The probability is low because the prior for a stiff neck is much higher than the prior for meningitis.

Naïve Bayes simplifies inference by assuming effects are conditionally independent given the cause.

The model is 'naïve' because it assumes all 'effect' variables are independent of each other, given the 'cause' variable. This is often technically false but works surprisingly well in practice.

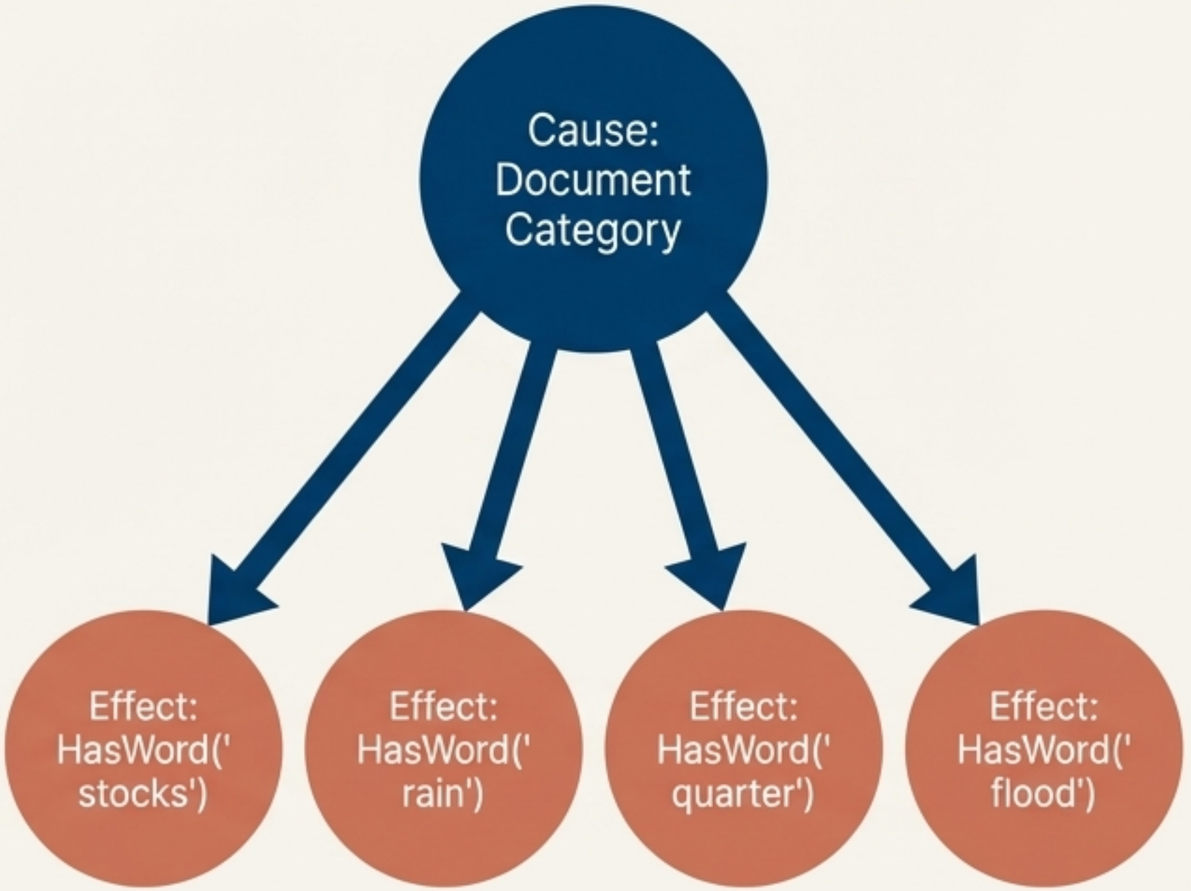
Structure: A single cause node with arrows pointing to multiple, separate effect nodes. This simplifies the joint distribution to:

$P(\text{Cause}, \text{Effect}_1, \ldots, \text{Effect}_n) = P(\text{Cause}) \prod P(\text{Effect}_i|\text{Cause})$

Classic Application: Text Classification.

- **Cause:** The document category (e.g., 'Business', 'Weather').
- **Effects:** The presence or absence of specific words (e.g., 'stocks', 'rain').

Widely used for tasks like spam filtering, where it provides a fast and effective classification despite the violated independence assumption (e.g., the words 'first' and 'quarter' are not independent).

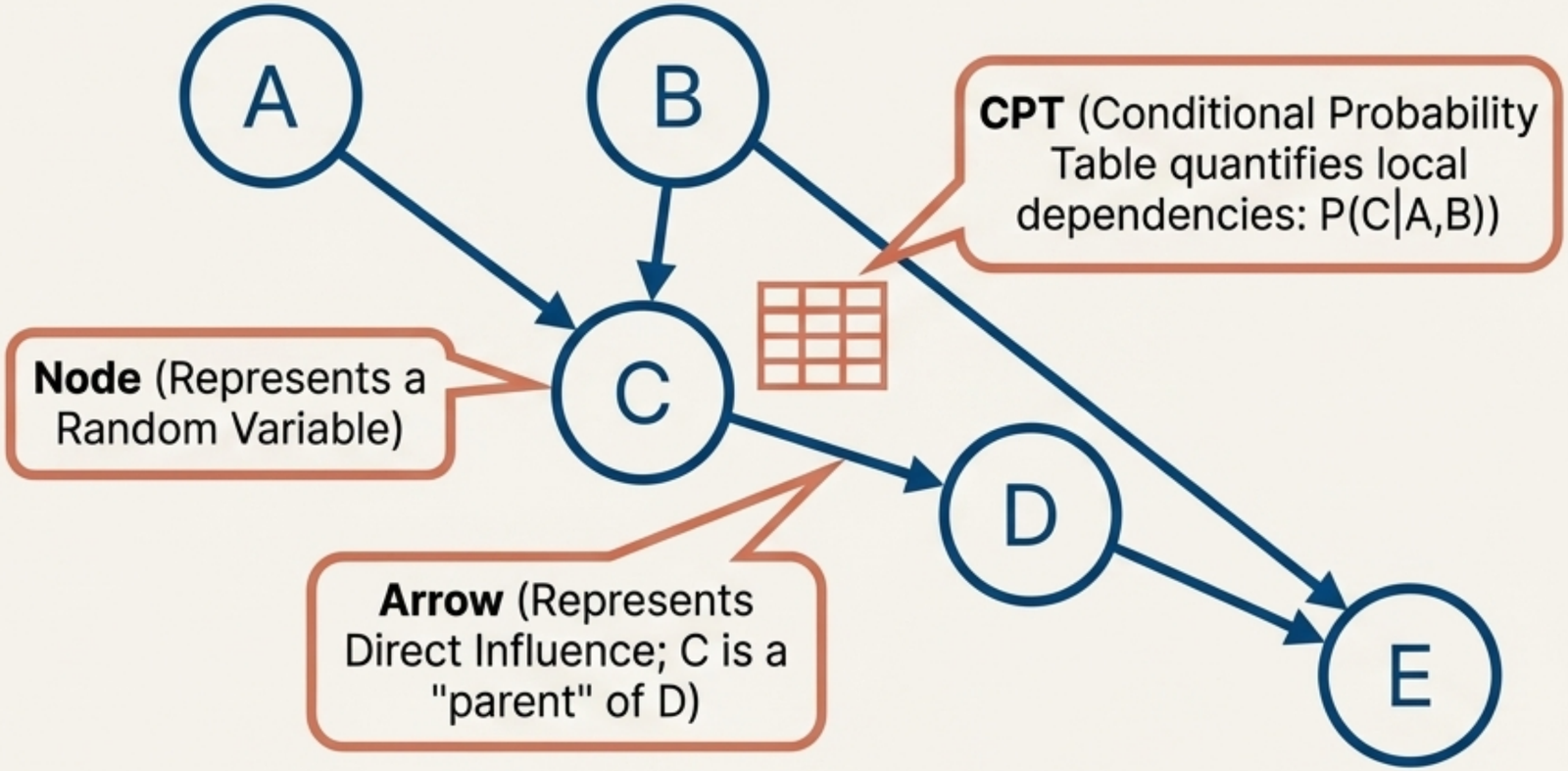


Bayesian Networks represent complex relationships between variables efficiently and intuitively.

A **Bayesian Network** (or Belief Network) is a directed acyclic graph (DAG) that provides a compact representation of the full joint distribution.

1. **Nodes**: Represent random variables.

2. **Arrows**: Represent direct dependencies. An arrow from X to Y means X is a **“parent”** of Y and has a **direct influence** on it.



3. Absence of an arrow: Signifies **conditional independence**. This is the key to the network’s efficiency.

Each node has a **Conditional Probability Table (CPT)** that quantifies the effect of its parents on it.

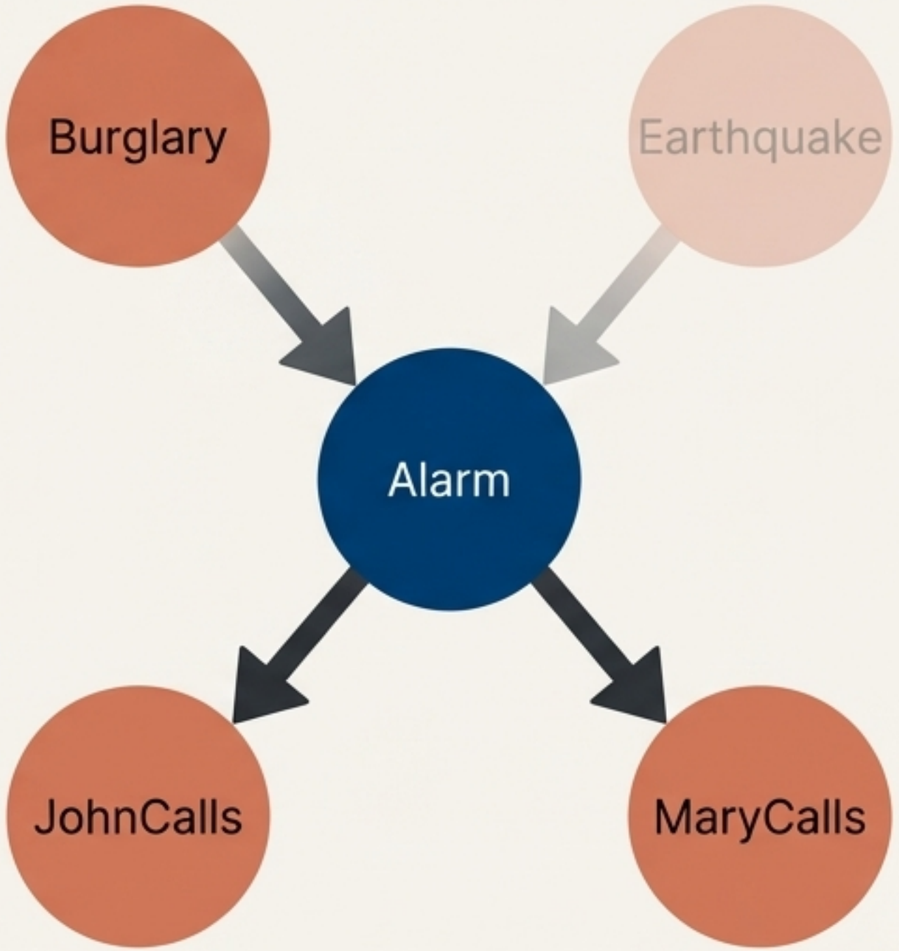
A Bayesian Network elegantly models a real-world scenario with multiple causes and effects.

Scenario: You have a new burglar alarm. It is reliable but can also be triggered by minor earthquakes. Two neighbours, John and Mary, have promised to call if they hear the alarm.

Variables: `Burglary`, `Earthquake`, `Alarm`, `JohnCalls`, `MaryCalls`.

Dependencies (The Structure):

- The `Alarm`'s state is directly influenced by `Burglary` and `Earthquake`.
- `JohnCalls` and `MaryCalls` are influenced *only* by the `Alarm`. They are not affected directly by a burglary or earthquake.
- This means John and Mary's calls are **conditionally independent** of Burglary/Earthquake, *given* the state of the Alarm.



CPTs provide the quantitative details for the network's qualitative structure.

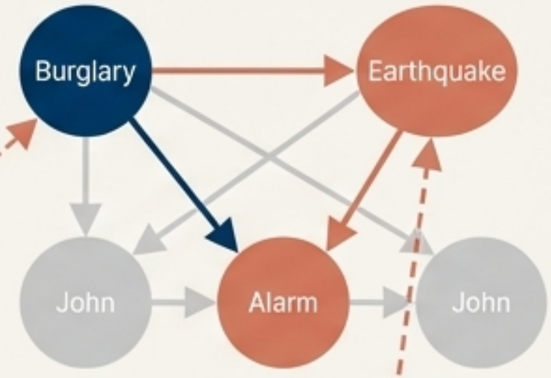
Each node's CPT specifies the probability of its value for every possible combination of its parents' values.

The **Burglary** and **Earthquake** nodes have no parents, so their CPTs are simple prior probabilities:

P(B=true) = 0.001
P(E=true) = 0.002

The Alarm node's CPT has 4 entries, specifying P(Alarm | Burglary, Earthquake) for all four true/false combinations of its parents.

The full joint probability is the product of the relevant conditional probabilities from the CPTs:



CPT: P(Alarm | Burglary, Earthquake)

Burglary	Earthquake	P(Alarm=true)
true	true	0.95
true	false	0.94
false	true	0.29
false	false	0.001

CPT: P(JohnCalls | Alarm)

Alarm	P(JohnCalls=true)
true	0.90
false	0.05

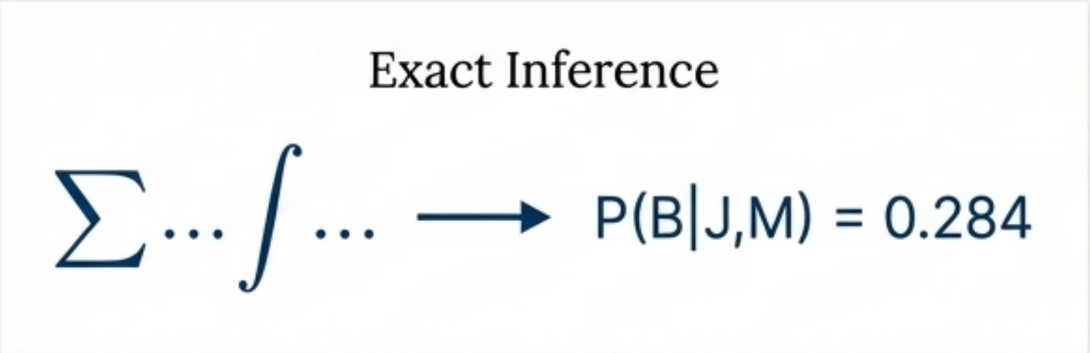
$$P(J,M,A,\neg B,\neg E) = P(J|A) * P(M|A) * P(A|\neg B,\neg E) * P(\neg B) * P(\neg E)$$

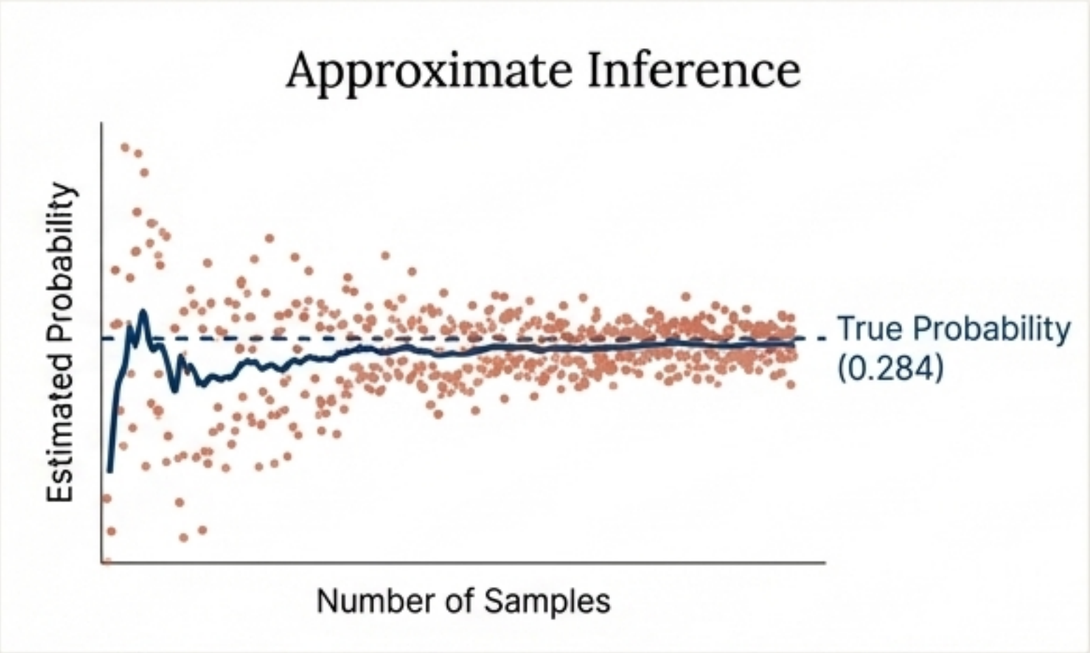
Inference is the process of querying the network to compute probabilities given evidence.

Query: Answering questions like 'What is the probability of a `Burglary`, given that both `John` and `Mary` have called?" This is written as:
`P(Burglary | JohnCalls=true, MaryCalls=true)`

Exact Inference: Algorithms like *variable elimination* compute the precise probability by summing over all hidden (unobserved) variables. This is NP-hard in the worst case and can be intractable for large networks.

Approximate Inference: For complex networks, we use methods like *Markov Chain Monte Carlo* (MCMC) sampling. These algorithms generate a large number of random samples from the network's distribution to *estimate* the true probability, trading some accuracy for tractability.





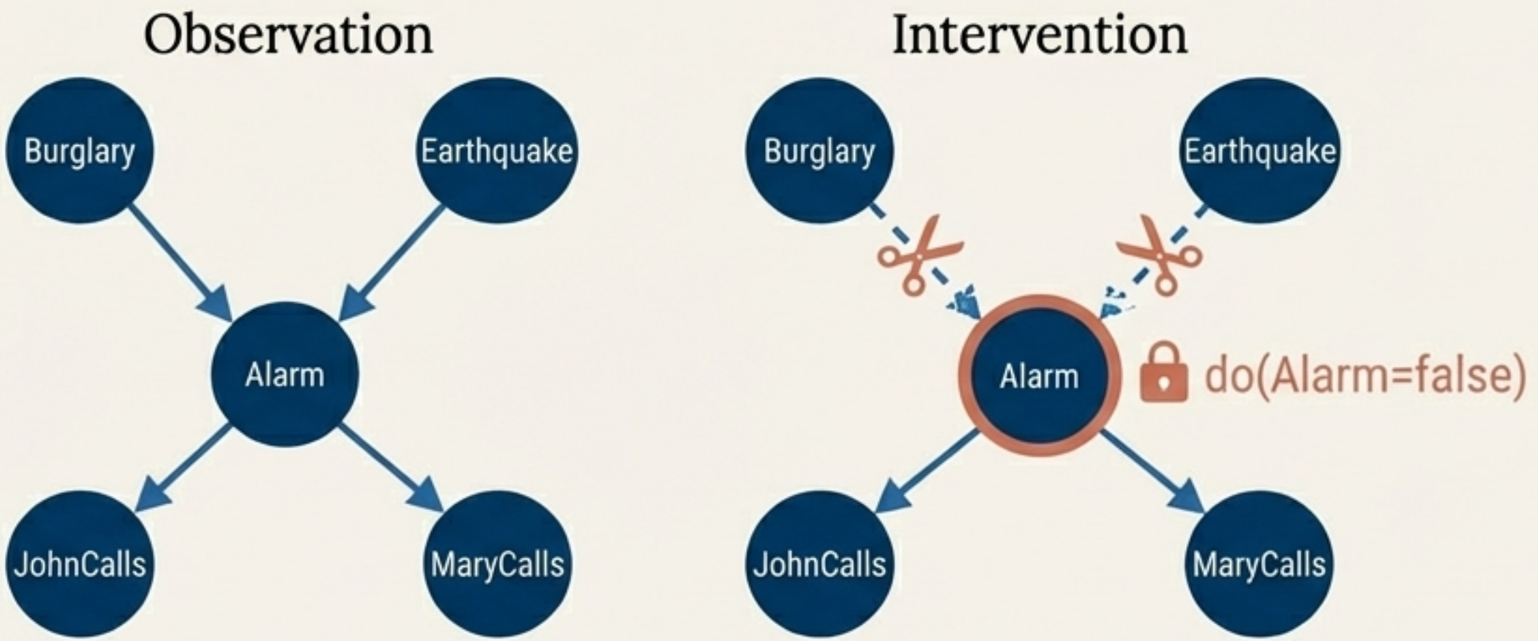
Causal networks add a powerful assumption: arrows represent true causal mechanisms.

A standard Bayesian Network represents dependencies. `A → B` can be probabilistically equivalent to `A ← B`. They capture correlations.

A **Causal Network** adds the semantic requirement that arrows must flow from cause to effect.

This allows us to predict the effect of **interventions**, represented by the `do(X=x). For example: for example: 'What is the probability of `JohnCalls` if I "*force* the alarm to be silent?" (do(Alarm=false)).

Mathematically, this is modelled by "cutting" the links from the parents of the intervened node (`Burglary` and `Earthquake`) to that node (Alarm`) and setting its value, simulating a direct manipulation of the system.



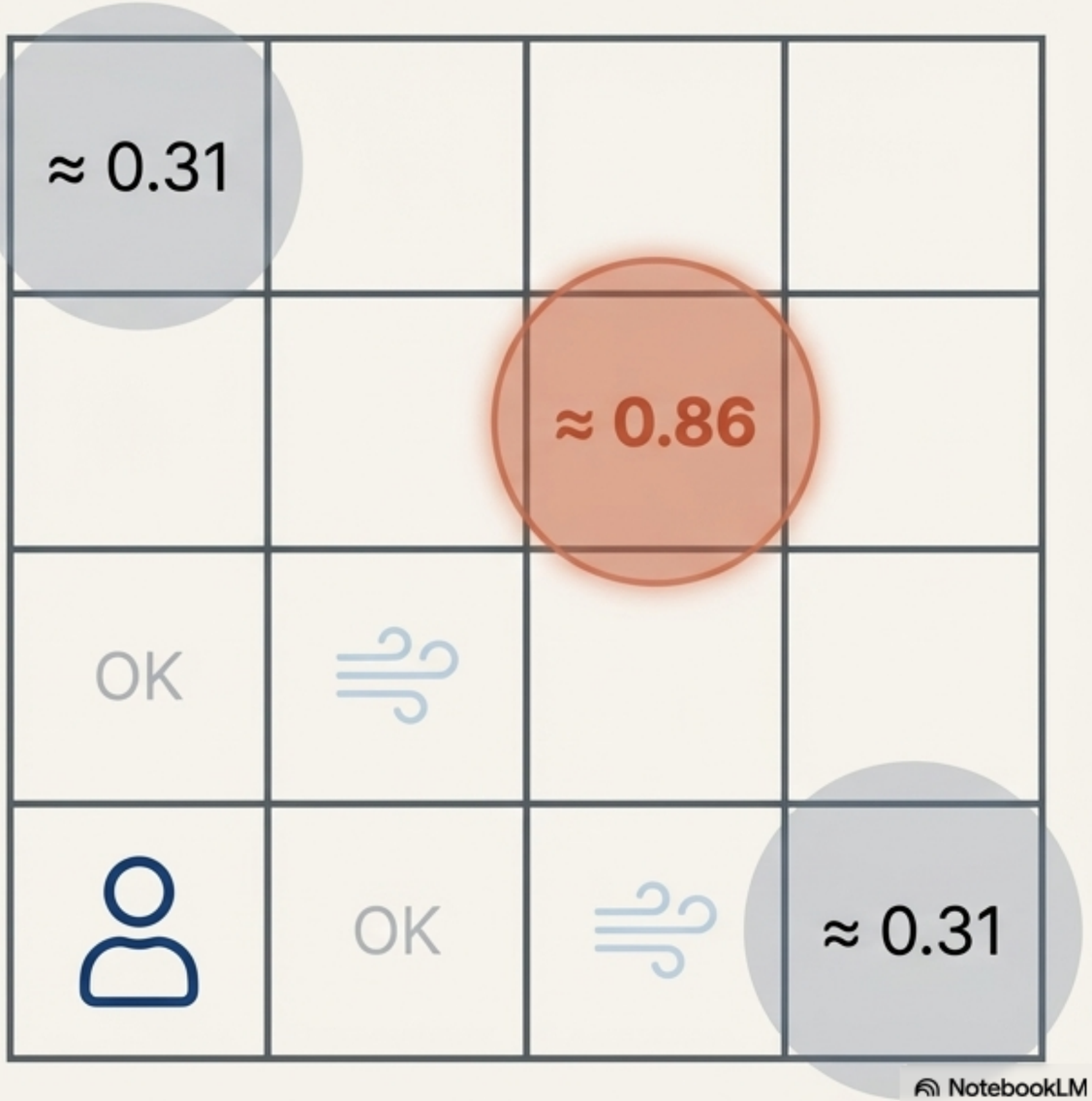
Probabilistic reasoning allows an agent to make intelligent choices where logic fails.

The Dilemma for Logic: The logical agent knows a pit *might* be in squares [1,3], [2,2], or [3,1] based on breezes in [1,2] and [2,1]. It cannot distinguish the risk and must choose randomly.

The Probabilistic Agent's Approach:

1. Defines random variables for pits (P_{ij}) and known breezes (b).
2. Uses the rules of the world (pits cause breezes in adjacent squares) to define the dependencies—a Bayesian Network.
3. Performs inference to calculate $P(P_{ij} \mid \text{known}, b)$ for each frontier square.

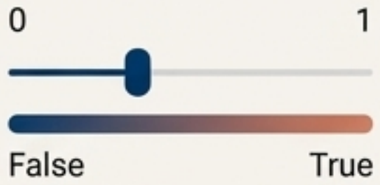
The Decisive Result: The agent computes that P(Pit in [2,2]) ≈ 0.86, while P(Pit in [1,3]) ≈ 0.31. The agent now has a clear, rational basis to avoid [2,2], an insight unattainable with pure logic.



From brittle rules to fluid beliefs, probabilistic reasoning is the foundation of modern intelligent systems.

The Problem

The real world's uncertainty breaks purely logical agents due to issues like the qualification problem.

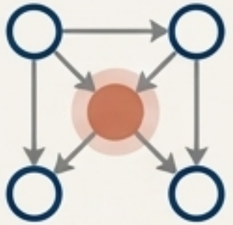


The Foundation

Probability theory provides a formal language to quantify belief, enabling rational decision-making via Maximum Expected Utility.

The Engine

Bayes' Rule provides the core mechanism for updating beliefs based on new evidence, turning causal knowledge into diagnostic power.



The Framework

Bayesian Networks offer a compact, intuitive, and powerful way to model complex domains by representing conditional conditional independencies, which enables efficient inference.

Believe in the Possibilities

Department of Artificial Intelligence and Data Science

RAJALAKSHMI
INSTITUTE OF TECHNOLOGY