

Intro To Artificial Intelligence

Olu Gbadebo

December 26th, 2016

As taught by Sebastain Thrun and Peter Norvic

What is Artificial Intelligence? It is an **Intelligent Agent**. An Intelligent Agent interacts with a certain environment to which it receives data from and send some data back to. The Agent uses **sensors** to “see” the environment and extracts information from the environment’s state. It then uses **actuators** to affect the state of the environment. When the Agent receives data, it goes through a **decision state** where it runs a function that analyzes the data and make a decision based on the data. The decision it makes is what the **attractors** will carry out on the environment.

The process of the Agent carrying out these (see, decide, act) actions repeatedly is called the **Perception Action Cycle**. In other words, the Perception Action Cycle is simply that the Agent continuously asks the environment for some data, processes the data, makes a decision and sends its decision to the environment.

Applications of A.I.

Artificial Intelligence has been used in several different areas of life. Some of the areas that AI has been successfully used in are:

1. Finance
2. Robotics
3. Games
4. Medicine
5. The Web

Finance

There is a huge use of AI in financing and in making trading decision. AI in finance is called a **Trading Agent**. The environment of a trading agent might be a stock market or a bond market or a commodities market. The agent can read the news, follow certain events, and analyze datasets, which are its form of sensors. The decisions the agent can make are either to buy or sell (in other words, trade).

Trading agents have been used to look at data over time, and those who use trading agents have made good amount of profits.

Robotics

Artificial Intelligence also has good history in robotics. AI agents in robotics can be actual robots. The agents' sensors include cameras, microphone and tactile sensor (touch). The way robots agent impact the environment is to move, touch or speak with humans.

Games

AI has been used hugely in games. For example, AI has been used in chess game to play against humans. The AI agent, **Game Agent**, reads your moves, analyzes it and then makes it own moves.

There are two reasons of all reasons that game agents are used in games. The first is to play against you with purpose of making you lose and

ultimately make you feel like a better player. The second reason is to make game play feel more natural and real in a way that characters in the games are smarter.

Medicine

A **Diagnostic Agent** is used in medicine to make “diagnostic” decisions. The agent observes you by making measurements like blood pressure and heart signal, and, in most cases, communicates its decision with a doctor who then intervenes the final decision. There are many other versions of diagnostics agent.

The Web

A web crawler is a type of AI on the web. The crawling agent goes to the World Wide Web and retrieves web pages, arranges the web pages and then show the best web pages. This agent is widely used by companies such as Google and Microsoft.

Terminology

Terminologies relating to environments are as follows:

1. Fully and Partial Observable
2. Deterministic and Stochastic
3. Discrete and Continuous
4. Benign and Adversarial

Fully and Partial Observable

An environment is considered a **Fully Observable environment** if what the agent senses at any point in time is completely sufficient to make an optimal decision. For example, in many card games where all the cards are visible, the momentary sites of all the cards is really sufficient for the AI to make a decision. On the contrary, a **Partial Observable environment** is where the AI needs some memory to make the best decision. For example,

in the game of poker where the cards are hidden, the AI would need some previous plays as data to make some predictions.

In a Perception Action cycle, a Fully Observable environment would have an agent that has full access to the state whereas a Partially Observable environment would have an agent that has access to some parts of the state and also have memorized data.

Deterministic and Stochastic

The **Deterministic** environment is where the agent's actions uniquely determine the outcome, like in chess where moves can be predetermined and the one moves affects the next. But a **Stochastic** environment where predictions can't be made based on random events, like a game that involves throwing a dice.

Discrete and Continuous

In **Discrete** environment, there are finitely many action choices for the agent to make. Well in **Continuous** environment, *you guessed it!*, there are infinitely many actions to make. Examples are chess game with finite positions on the board and dart with infinite many ways of throwing the dart (different angles and accelerations).

Benign and Adversarial

Benign environment doesn't have an objective that contradicts your own objective. Its goal is not to *hunt you down* like the weather which just does its thing without trying to hurt you. It thrives to achieve a goal that doesn't affect its user. Whereas an **Adversarial** environment exists to *make you not exist* like in a chess game where your opponent is sweating hard to make you lose.

Problem solving techniques

A problem in AI can be broken down into a number of components.

1. Initial State: This is the state that the agent starts out with.

2. Action: A function, `action(s)`, that takes in a state of the agent and returns a set of actions `{a1, a2, a3, ... , an}`. For some agents, they might have the same actions in all states and in other agents they might have different actions in all states.
3. Result: A function, `result(s, a)`, that takes in a state and an action and delivers as output a new state.
4. Goal Test: A function, `goaltest(s)`, which takes a state and returns a boolean stating whether the state is the goal or not.
5. Path Cost: A function, `pathcost(s1 \xrightarrow{a} s2 \xrightarrow{a} ...)` which takes a sequence of state-action transactions and returns a number `n` which is the cost of that action.

Searches can be used in AI to solve AI problems. For example, provided a map, how can you find the best route from a starting position to a destination on the map? A search algorithm can help solve the problem. Some search algorithms used in AI are :

1. Tree Search
2. Breadth First Search
3. Graph Search
4. Depth First Search
5. Uniform Cost Search (Cheapest Cost)
6. A* Search

I won't write in details what these searches are. I know I should. You can find the [lectures](#) on Udacity's Intro to AI, "Problem Solving" section

Problem with problem-solving

These problem-solving techniques only work if the following conditions are true:

- The environment must be **Fully Observable**: We must be able to know what initial state we start out with.

- The environment must be **Known**: We must know the set of actions that can be performed.
- The environment must be **Discrete**: There must be a finite actions to choose from.
- The environment must be **Deterministic**: We have to know the result of taking an action.
- The environment must be **Static**: There must be nothing else that can change the environment except our own actions.

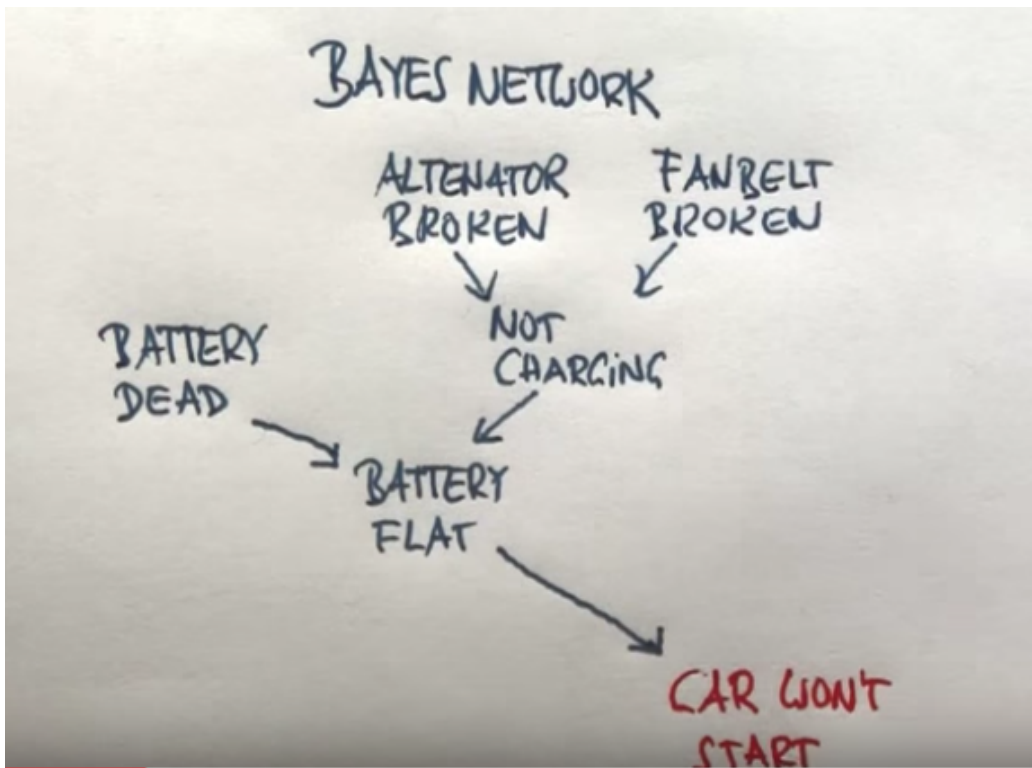
Probability in AI: Bayes Network

Bayes Network is a probability technique that is widely used almost in all fields of smart computing systems such as diagnostics, predictions, machine learning. It's also used in fields like finance, robotics and internal departments at Google. Bayes Network also serves as the building block for some advance AI techniques including particle filters, hidden MArkov model, MDPs, POMDPs, kalman filters and many others.

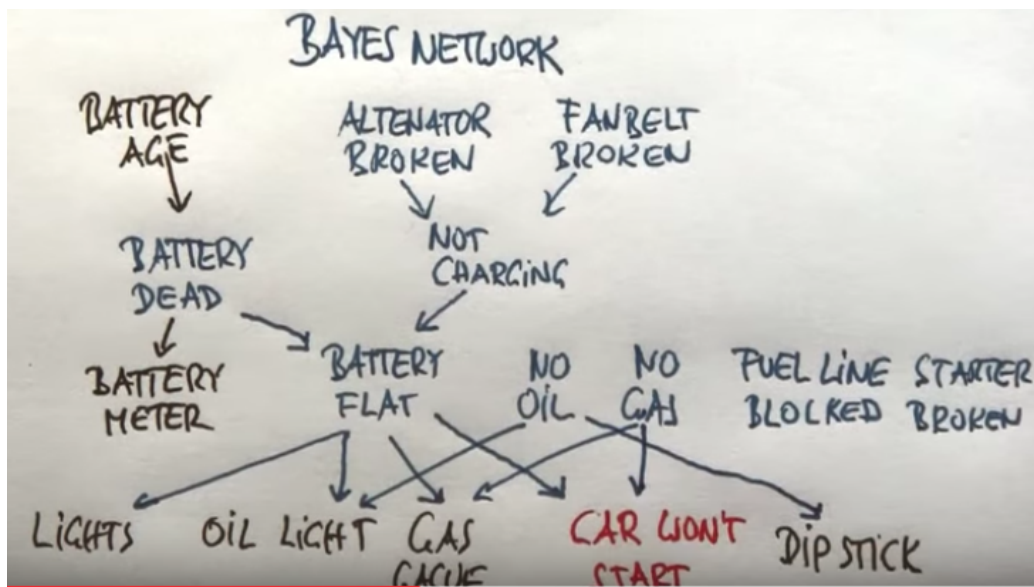
Problem in red, Causes in blue, Diagnostic in black/gray

To understand Bayes Network, we will use the following example. Suppose you find out in the morning that your car won't start, there are many reasons such situation could occur. One is that your battery is flat. For a flat battery, there are other causes such as battery is dead or the battery is not charging. You see the "chain reaction" here? If the battery is not charging, it could be that the alternator is broken or the fanbelt is broken.

Analyzing the Bayes Network, you can see that there are many different causes for the car not to start. The question is, "can we diagnose the problem?"



One diagnostic tool is a battery meter, which would either support your belief, or not, that the battery is the sole cause. Another reason to think it might be your battery is the battery's age. Older batteries tend to die more often. Other diagnostic approach is to inspect the light, oil light, gas gauge, and you could dip into the engine oil with a dipstick. This approach relates to other reasons why the car won't start, like no oil, no gas, fuel line blocked, or broken starter. *They are all connected!* The flat battery will cause the lights not to work, and would have effect on the oil light and gas gauge. The dipstick and the oil light would indicate whether or not there is oil in the engine. And of course, the car won't start if there's no gas, which would be indicated by the gas gauge.



A Bayes Network is graph composed of nodes. Nodes can correspond to known or unknown events, typically called **Random Variables**. These nodes are connected by arcs, and the arcs suggests that the child of the arc is affected by the parent either in a deterministic way or in a probabilistic way. Therefore, the Bayes Network graph structure provides a probability distribution in the space of all the given variables (nodes).

Probabilities

Too lazy to write

PROBABILITIES

$$P(H) = \frac{1}{2} \quad P(T) = \boxed{\frac{1}{2}}$$
$$P(H) = \frac{1}{4} \quad P(T) = \boxed{\frac{3}{4}}$$
$$1 - \frac{1}{4} = \frac{3}{4}$$
$$P(H, H, H) = \boxed{0.125} \quad P(H) = \frac{1}{2}$$
$$\frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{8}$$

PROBABILITY

$$X_i = \text{result of } i\text{-th coin flip} \quad X_i = \{H, T\}$$
$$P(X_1 = X_2 = X_3 = X_4) = \boxed{0.125} \quad P_i(H) = \frac{1}{2} \quad \forall i$$
$$\hookrightarrow \frac{1}{16} + \frac{1}{16} = \frac{1}{8}$$

$$P(\{X_1, X_2, X_3, X_4\} \text{ contains } \geq 3 \text{ H}) = \underline{0.3125}$$

$$5 \cdot \frac{1}{16} = 0.3125$$

$\begin{array}{cccc} H & H & H & H \\ H & H & H & T \\ H & H & T & H \\ H & T & H & H \\ T & H & H & H \end{array}$

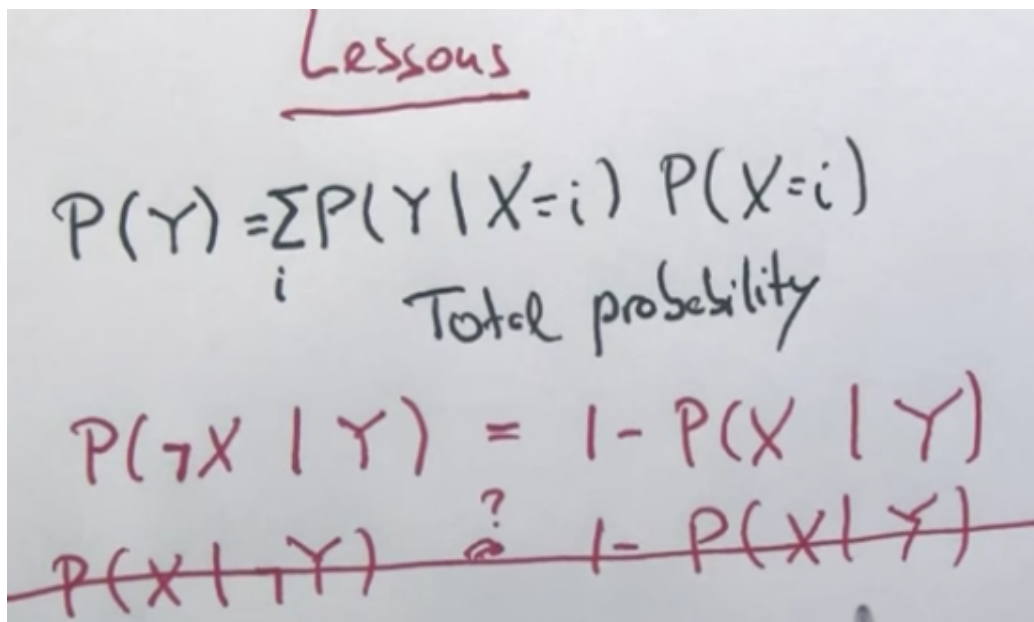
0:28

DEPENDENCE

$$P(X_1 = H) = \frac{1}{2} \begin{cases} \rightarrow H: P(X_2 = H | X_1 = H) = 0.9 \\ \rightarrow T: P(X_2 = T | X_1 = T) = 0.8 \end{cases}$$

$$P(X_2 = H) = \boxed{0.55}$$

$$\begin{aligned}
 P(X_2 = H) &= P(X_2 = H | X_1 = H) \cdot P(X_1 = H) \\
 &\quad + P(X_2 = H | X_1 = T) \cdot P(X_1 = T) \\
 &= 0.9 \cdot \frac{1}{2} + 0.2 \cdot \frac{1}{2} = 0.45 + 0.1
 \end{aligned}$$



I have to write something now!

If for some certain variables x and y , there are two possible events $H \mid T$ for each (as in $x(H), x(T), y(H), y(T)$). Provided the probability of first event of x is given $p(x(H)) = 0.5$, and probability of the sequence of possible events is also given, ex: $p(y(T) \mid x(H)) = 0.9$ read: *the probability of y being T following x being H is 0.9*. The probability of the second event of x is the sum of the multiplication of the probability of the second event of x following the first event of x and the probability of the first event of y and the multiplication of the probability of the second event of x following the first event of y and the probability of the second event of y .

$$p(x(T)) = p(x(T) \mid y(H)) \times p(y(H)) + p(x(T) \mid y(T)) \times p(y(T)) \quad (1)$$

I'm so confused!

Bayes Rule

$$P(A \mid B) = \frac{P(B \mid A) \times P(A)}{P(B)} \quad (2)$$

$P(A | B)$ is Posterior: the probability of A given B, where A is the cause (variable we care about) and B is the evidence

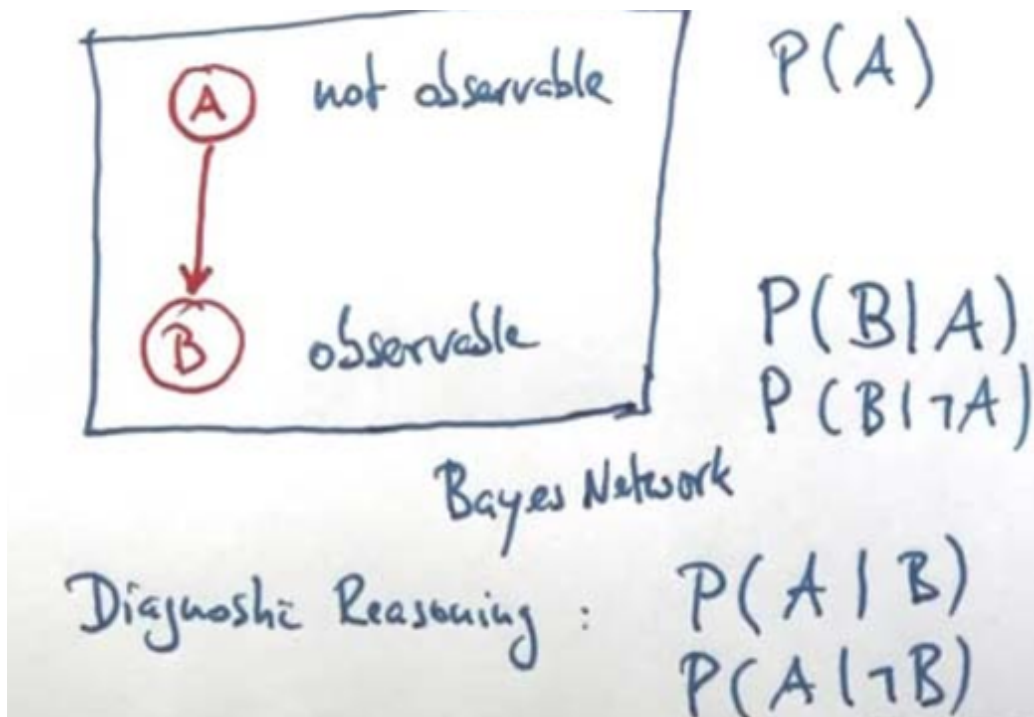
$P(B | A)$ is Likelihood: provided the A, the probability of B

$P(A)$ is Prior: probability of A

$P(B)$ is Marginal Likelihood: probability of B, also the Total Probability

Total Probability $P(B) = \sum_a P(B | A = a) \times P(A = a)$

Graphically,



You don't get it? [Go here](#)

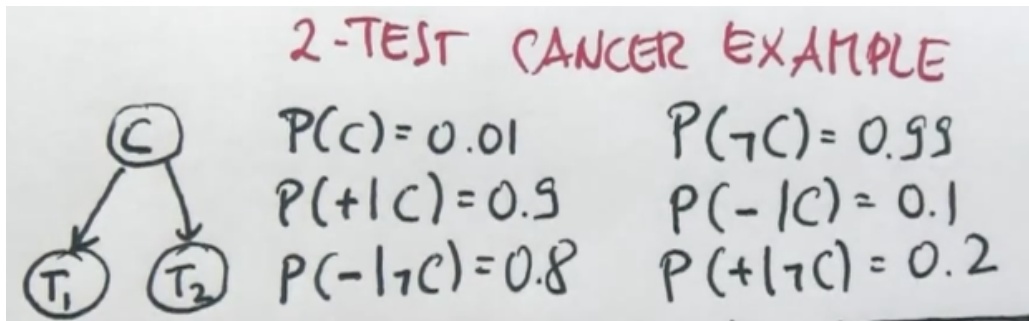
See that $P(B)$ in the denominator of Bayes' Rule? Yeah! Let's get rid of it cos why not?

We know that $P(A | B) + P(\neg A | B) = 1$ and $P(B)$ serves as the

‘normalizer’ in the Bayes’ Rule. So, using psuedo probabilities, (P'), that are non-normalized: $P'(A | B) = P(B | A) \times P(A)$ and $P'(\neg A | B) = P(B | \neg A) \times P(\neg A)$, Bayes Rule is the product of the psuedo probability and the normalizer, η (pronounced ‘eta’) = $(P'(A | B) + P'(\neg A | B))^{-1}$:

$$P(A | B) = \eta P'(A | B) \quad (3)$$

Let’s use this new equation in an example:



Given the information in the graphic above, find $P(C | T_1 = + T_2 = -)$.

	P(C)	+	-	P'
C	0.01	0.9	0.1	0.0009
$\neg C$	0.99	0.2	0.8	0.1584
				$\eta = 0.1593^{-1}$

$$P(C | +, -) = \eta P'(C | +, -)$$

$$P'(C | +, -) = P(C | +) \times P(C | -)$$

$$P'(C | +, -) = (P(C | +) \times P(C)) \times (P(C | -) \times P(C))$$

$$P'(C | +, -) = P(C | +) \times P(C | -) \times P(C)$$

$$P'(C | +, -) = 0.9 \times 0.1 \times 0.01 = 0.0009$$

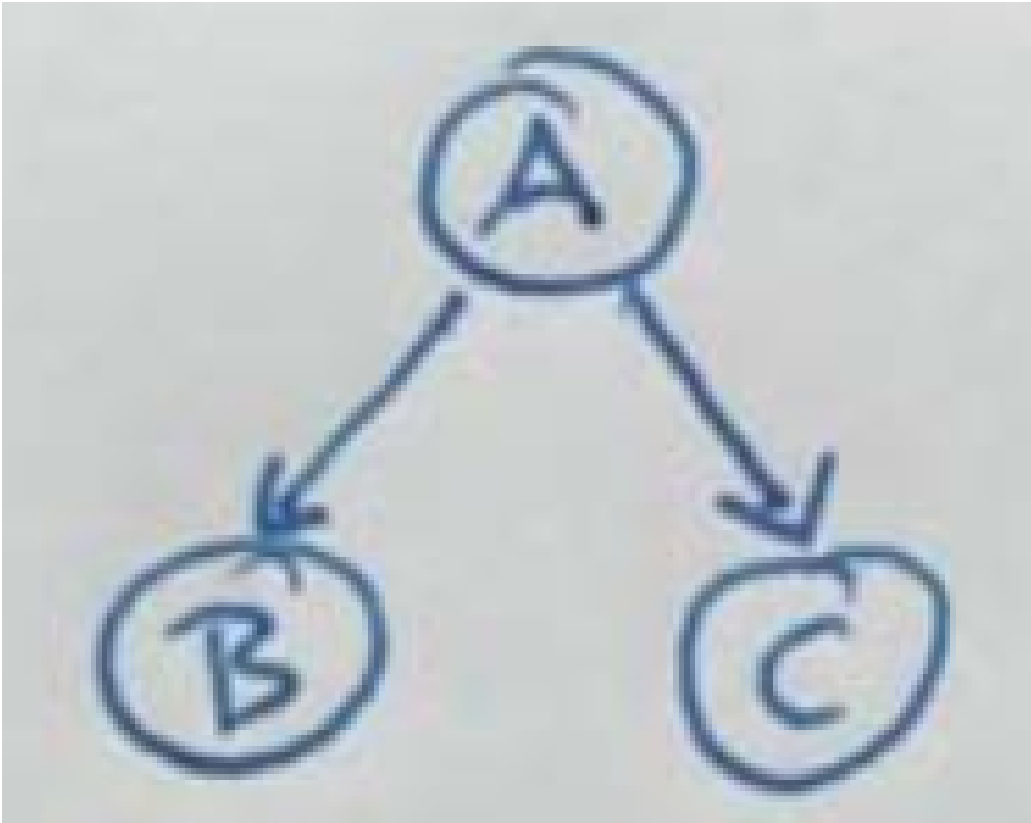
$$\eta = (P'(C | +, -) + P'(\neg C | +, -))^{-1}$$

$$\eta = (0.0009 + 0.1584)^{-1} = 6.2775$$

$$\text{Then } P(C \mid +, -) = 6.2775 \times 0.0009 = 0.0056$$

Conditional Independence

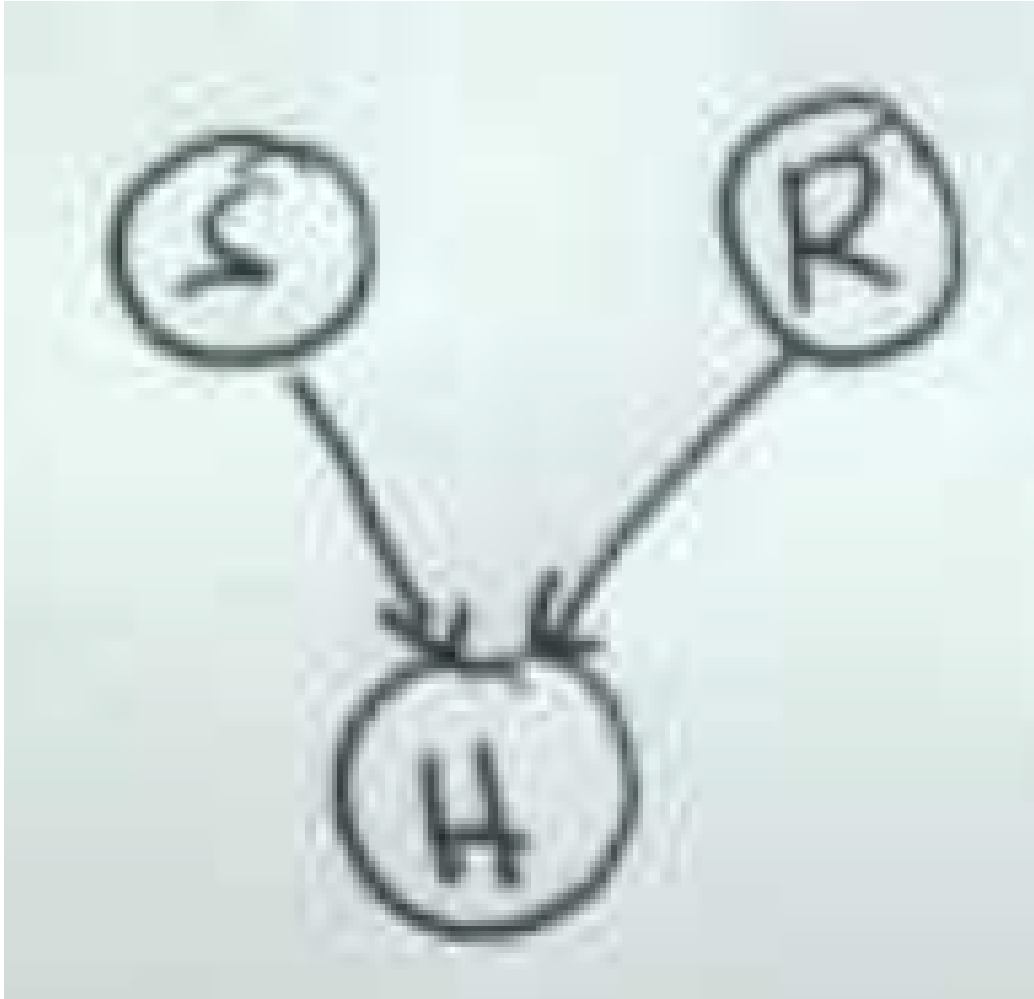
In Bayes' network, given three causes **A**, **B** and **C**,



if A is a known variable i.e. the value of A is certain, then B is **conditionally independent** of C, $(B \perp C \mid A)$. For example, the probability of C given A and B, provided that the value of A is known, is equivalent to the probability of C given A:

$$P(C \mid A, B) = P(C \mid A) \tag{4}$$

Explaining Away



Given an Effect that could be caused by two or more Causes, if one of the Causes is known to have lead to the Effect then it is less likely that the other Causes lead to the Effect. For example, say a sunny day or a raise makes Sebastian happy; if Sebastian happens to be happy on a sunny day then it is less likely that a raise is the reason for Sebastian's happiness.

Let's do an example,

$P(S) = 0.7$	$P(H S, R) = 1$
$P(R) = 0.01$	$P(H \neg S, R) = 0.9$
$P(R S)$	$P(H S, \neg R) = 0.7$
$= \boxed{0.01}$	$P(H \neg S, \neg R) = 0.1$

Given the above information, what is the probability that of a raise given that Sebastian is happy and it's sunny $P(R | H, S)$?

Using Bayes' Rule

$$P(R | H, S) = \frac{P(H | R, S) \times P(R | S)}{P(H | S)} \quad (5)$$

Using Conditional Independence

$$P(R | H, S) = \frac{P(H | R, S) \times P(R)}{P(H | S)} \quad (6)$$

Using Total Probability

$$P(R | H, S) = \frac{P(H | R, S) \times P(R)}{P(H | R, S) \times P(R) + P(H | \neg R, S) \times P(\neg R)} \quad (7)$$

$$P(R | H, S) = \frac{1 \times 0.01}{1 \times 0.01 + 0.7 \times 0.99} = 0.0142 \quad (8)$$

Now, let's do the opposite. Find the probability of a raise given that Sebastian is happy $P(R | H)$.

First, we need to compute $P(H)$:

$$\begin{aligned}
 P(H) &= P(H \mid S, R) \times P(S \mid R) \\
 &\quad + P(H \mid \neg S, R) \times P(\neg S \mid R) \\
 &\quad + P(H \mid S, \neg R) \times P(S \mid \neg R) \\
 &\quad + P(H \mid \neg S, \neg R) \times P(\neg S \mid \neg R) \\
 &= 0.5245
 \end{aligned}$$

Using Bayes' Rule

$$P(R \mid H) = \frac{P(H \mid R) \times P(R)}{P(H)} \quad (9)$$

Using Total Probability

$$\begin{aligned}
 P(H \mid R) &= P(H \mid R, S) \times P(S) + P(H \mid R, \neg S) \times P(\neg S) \\
 &= 0.97
 \end{aligned}$$

$$P(R \mid H) = \frac{0.97 \times 0.01}{0.5245} = 0.0185$$

Let's observe the situation where Sebastian is happy and it's not sunny.
What's the probability of a raise $P(R \mid H, \neg S)$?

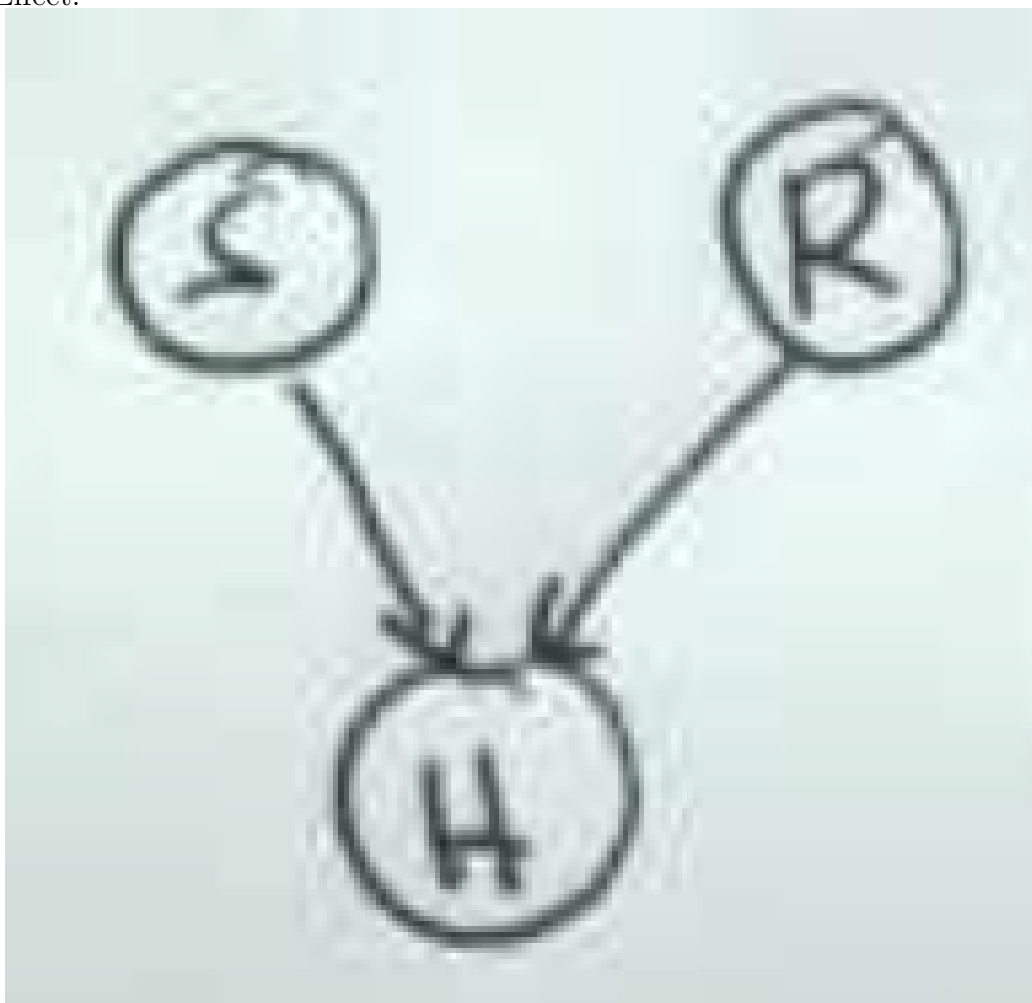
$$\begin{aligned}
 P(R \mid H, \neg S) &= \frac{P(H \mid R, \neg S) \times P(R, \neg S)}{P(H, \neg S)} \\
 &= \frac{0.9 \times 0.01}{1 \times 0.01 + 0.7 \times 0.99} \\
 &= 0.0833
 \end{aligned}$$

Bringing all three situations together:

$$\begin{aligned}
 P(R \mid H, S) &= 0.0142 \\
 P(R \mid H) &= 0.0185 \\
 P(R \mid H, \neg S) &= 0.0833
 \end{aligned}$$

Conditional Dependence

The previous section introduced a situation where two Causes lead to one Effect.

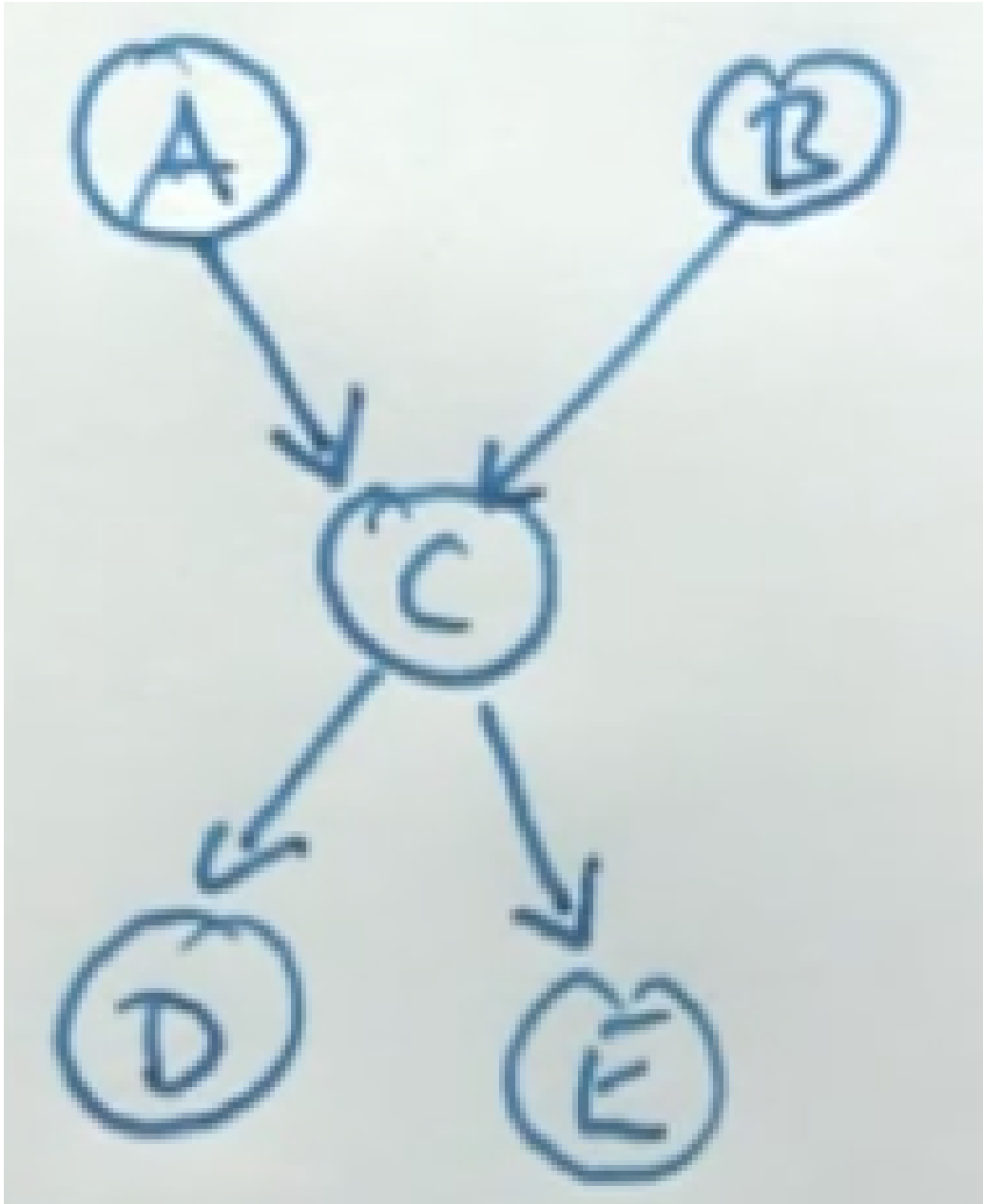


Given that H is unknown, then S and R are independent. They do not affect each other, which is why $P(R | S)$ is the same as $P(R)$. However, if H is known then S and R are not independent anymore but **conditionally dependent**.

Note the difference between **independence**, **conditional independence**,

and **conditional dependence**.

General Bayes Net



With Bayes' network, finding the probability of all nodes in a network is significantly better than without using Bayes' network. The probability of

all nodes $P(A, B, C, D, E)$ is as follows:

$$P(A, B, C, D, E) = P(A) \times P(B) \times P(C \mid A, B) \times P(D \mid C) \times P(E \mid C) \quad (10)$$


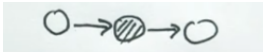
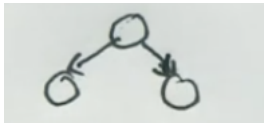
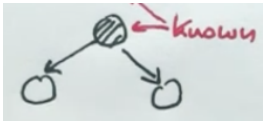
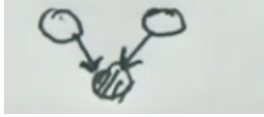
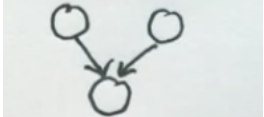
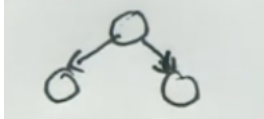
The number of values of the probability of each node in the network is determined by the number of “incoming arcs” the node has. For the graphics above, A and B have zero incoming arcs, C has two incoming arcs and D and E have one incoming arcs.

$$\begin{aligned} \text{the \# of } P(X) &= 2^k \text{ where } k \text{ is \# of incoming arcs} \\ \text{for } A, P_n(A) &= 2^0 \\ &= 1 : (P(A)) \\ \text{for } C, P_n(C) &= 2^2 \\ &= 4 : \\ &\quad (P(C \mid A, B), \\ &\quad P(C \mid A, \neg B), \\ &\quad P(C \mid \neg A, B), \\ &\quad P(C \mid \neg A, \neg B)) \end{aligned}$$

D Separation a.k.a Reachability

D Separation can be understood with the concepts of **Active Triplets** and **Inactive Triplets** where both concepts apply some rules to a sequence of three variables in a network. Active Triplets renders variables dependent, on the other hand, Inactive Triplets renders variables independent.

$A = \text{first node}, B = \text{middle node}, C = \text{last node}, D = \text{extended node}$

Active Triplets		Inactive Triplets	
	A and C are dependent when B is unknown .		A and C are independent when B is known .
	A and C are dependent when B is unknown .		A and C are independent when B is known .
	A and C are dependent when B is known .		A and C are independent when B is unknown .
	A, B and C are dependent when D is known .		