



Module 3 notes - by Juan Guillermo Jaramillo Saa



Probability basics + Inference with Joint Distribution

Probability basics

[Sample space and events](#)

[Probability space](#)

[Random variable](#)

[Probability distribution](#)

[Propositions and Propositional Logic Model](#)

[Syntax for propositions](#)

[Prior probability](#)

[Probability distribution](#)

[Joint Probability Distribution](#)

[Conditional probability](#)

Inference using Full Joint Distributions

[Inference by enumeration](#)

[Common terminology for operations on CPDs.](#)

Probability basics

Agents may need to handle uncertainty due to:

- partial observability
- nondeterminism
- a combination of both

Sample space and events

The set of all possible worlds is called **sample space**, denoted Ω .

Any subset $A \subseteq \Omega$ is an event.

Any element $\omega \in \Omega$ is called a sample point/atomic event/sample world.

Example:

$$\begin{aligned}\Omega_{\text{die}} &= \{1, 2, 3, 4, 5, 6\} = \text{outcomes of die roll} \\ A &= \{1, 2, 3\} = \text{outcomes of die roll less than 4}\end{aligned}$$

Probability space

A probability space or probability model is a sample space with an assignment $P(\omega)$ for every $\omega \in \Omega$ s.t.:

- $0 \leq P(\omega) \leq 1$
- $\sum_{\omega} P(\omega) = 1$

Accordingly:

$$P(A) = \sum_{\omega \in A} P(\omega)$$

Random variable

A random variable is a function from sample space to some range (call it \mathbf{D}), e.g., the reals or the Booleans.

$$X : \Omega \rightarrow \mathbf{D}$$

Example (with $\Omega = \Omega_{\text{die}} = \{i\}_{i=1, \dots, 6} = \mathbf{D} = \mathbb{B}$):

$$\begin{aligned} \text{Odd}(1) &= \text{true} \\ \text{Odd} : \Omega_{\text{die}} &\rightarrow \mathbb{B} = \{0, 1\} \\ \Omega_{\text{die}} &\text{ is a random variable} \end{aligned}$$

Probability distribution

For each X discrete random variable (i.e. a r.v. which can take only discrete values), P induces a probability distribution:

$$P(X = x_i) = \sum_{\omega: X(\omega) = x_i} P(\omega)$$

Example:

$$\begin{aligned} P(\text{Odd} = \text{true}) &= P(1) + P(3) + P(5) \\ &= \frac{1}{6} + \frac{1}{6} + \frac{1}{6} \\ &= \frac{1}{2} \end{aligned}$$

Propositions and Propositional Logic Model

In AI, questions are **logical statements**, not sets.

Think of a proposition as the event where the proposition is true.

An example, given Boolean r.v. A and B :

- event a = the set of sample points where $A(\omega) = \text{true}$

$$a = \{\omega \in \Omega \mid A(\omega) = \text{true}\}$$

- event $\neg a$ = the set of sample points where $A(\omega) = \text{false}$

$$\neg a = \{\omega \in \Omega \mid A(\omega) = \text{false}\}$$

- event $a \wedge b$ = the set of sample points where $A(\omega) = \text{true} \wedge B(\omega) = \text{true}$

So each proposition corresponds to the **collection of worlds** where that proposition holds.

So with Boolean variables, each sample point = propositional logic model (i.e. an exhaustive assignment of all the variables).

e.g. $A = \text{true}$, $B = \text{false}$, or $a \wedge \neg b$

Take a proposition such as $A \vee B$.

- Look at all atomic events (truth assignments) where $A \vee B$ is true:

$$A = \text{true}, B = \text{false}$$

$$A = \text{false}, B = \text{true}$$

$$A = \text{true}, B = \text{true}$$

Each one of these is an atomic event, call them $\omega_1, \omega_2, \omega_3$.

Then logically one could write:

$$A \vee B \equiv \omega_1 \vee \omega_2 \vee \omega_3$$

Syntax for propositions

- Boolean random variables (two possible outcomes)

example: *Cavity* (do I have cavity?)

Cavity = true is a proposition, also written cavity

- Discrete random variables (finite, or infinite)

example: *Weather* which could be one of $\langle \text{sunny}, \text{rain}, \text{cloudy}, \text{snow} \rangle$

Weather = rain is a proposition

Values must be exhaustive and mutually exclusive

- Continuous random variables (bounded on interval or not)

Temp which lies in \mathbb{R} domain.

Temp < 22.0 is a proposition

- Arbitrary Boolean combinations of basic propositions

Cavity = true \vee *Tooache = true* is a proposition, also written cavity \vee tooache

So remember:

- Capital letter to refer to the specific variable
- Small letter to refer to Boolean proposition

Prior probability

Prior or **unconditional probabilities** of propositions correspond to belief prior to arrival of any (new) evidence.

Probability distribution

A probability distribution gives values for all possible assignments (each possible world).

e.g. $P(\text{Weather}) = \langle 0.72, 0.1, 0.08, 0.1 \rangle$ (normalized, sums to 1)

which recalling what we've seen is obtained in the following way, e.g. $P(\text{Weather} = \text{sunny})$:

$$P(\text{Weather} = \text{sunny}) = \sum_{\omega \in \Omega: \text{Weather}(\omega) = \text{sunny}} P(\omega) = 0.72$$

Joint Probability Distribution

The joint probability distribution over a set of random variables **gives the probability of every atomic event on those random variables** (i.e. every sample point)

e.g. $\mathbf{P}(\text{Weather}, \text{Cavity})$ = a 4×2 matrix of values:

<i>Weather =</i>	<i>sunny</i>	<i>rain</i>	<i>cloudy</i>	<i>snow</i>
<i>Cavity = true</i>	0.144	0.02	0.016	0.02
<i>Cavity = false</i>	0.576	0.08	0.064	0.08

Every question about a domain can be answered by the joint distribution because is a sum of sample points.

Conditional probability

With respect to prior probabilities $P(X)$, conditional or posterior probabilities $P(X|Evidence)$ represent a more informed distribution in the light of the (new) Evidence

Example:

e.g., $P(\text{cavity}|\text{toothache}) = 0.8$:
i.e., **given that toothache is all I know**
NOT "if *toothache* then 80% chance of *cavity*"

The definition of conditional probability two events a and b is the following:

$$P(a|b) = \frac{P(a, b)}{P(b)}$$

Given that b had happened the new sample space becomes this one, that's why $P(b)$ is in the denominator.

The product rule gives an alternative definition:

$$P(a, b) = P(a|b) \cdot P(b)$$

The **Chain Rule** is derived by successive application of product rule:

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

Inference using Full Joint Distributions

Inference is the task of using a given Probabilistic Model of a system to answer certain questions regarding that system.

In other words, it answers the following question:

How can we derive information based on probabilities we have?

A probability query $P(Y|e)$ defines the posterior joint distribution of a set of **query variables** Y given specific values e for some **evidence variables**.

A first very basic way to compute any query considered as a proposition ϕ , having the full joint distribution is the following:

$$P(\phi) = \sum_{\omega: \omega \models \phi} P(\omega)$$

In other words, sum all the atomic events in which ϕ is true.

Apart from Y and e variables, there's another set of variables H (**hidden variables**), which are variables of the system that are **not part of the query** (i.e. are neither query nor evidence variables).

In principle one could compute ANY query in the following way:

$$\begin{aligned} & P(Y|e) \\ &= \frac{P(Y, E=e)}{P(E=e)} \\ &= \alpha P(Y, E=e) \\ &= \alpha \sum_{h \in H} P(Y, E=e, H=h) \end{aligned}$$

General idea: compute distribution on query variable by fixing evidence variables and summing over hidden variables

This is possible because of the law of total probabilities, which states, given B_i partition for sample space S and A arbitrary event:

$$P(A) = \sum_n P(A, B_n)$$

Given a Full Joint Distribution like the following:

	<i>toothache</i>		<i>¬toothache</i>	
	<i>catch</i>	<i>¬catch</i>	<i>catch</i>	<i>¬catch</i>
<i>cavity</i>	0.108	0.012	0.072	0.008
<i>¬cavity</i>	0.016	0.064	0.144	0.576

We'll try to make basic inferences from it.

Inference by enumeration

Enumeration Inference is an inference algorithm that can compute any query via *sums-of-products* of conditional probabilities from the CPTs (later on CPTs will be introduced)

- $\phi = \text{toothache}$

$$P(\phi) = 0.108 + 0.012 + 0.016 + 0.064 = 0.2$$

- $\phi = \text{toothache} \vee \text{cavity}$

$$P(\phi) = 0.2 + 0.072 + 0.008 = 0.28$$

- What we want conditional probabilities?

$$P(\neg \text{cavity} | \text{toothache}) = \frac{P(\neg \text{cavity}, \text{toothache})}{P(\text{toothache})}$$

Denominator can be viewed as a **normalization constant** α :

$$P(\neg \text{cavity} | \text{toothache}) = \alpha P(\neg \text{cavity}, \text{toothache})$$

- What if we are interested in a **conditional probability distribution** of a particular variable?

$$\begin{aligned} P(\text{Cavity} | \text{toothache}) &= \alpha P(\text{Cavity}, \text{toothache}) \\ &= \alpha [P(\text{Cavity}, \text{toothache}, \text{catch}) + P(\text{Cavity}, \text{toothache}, \neg \text{catch})] \\ &= \alpha [0.108 + 0.016 + 0.012 + 0.064] \\ &= \alpha \cdot 0.2 \end{aligned}$$

Note how all the possible values of the hidden variable *Catch* are used in the joint distribution.

Common terminology for operations on CPDs.

<i>Weather</i> =	<i>sunny</i>	<i>rain</i>	<i>cloudy</i>	<i>snow</i>
<i>Cavity</i> = <i>true</i>	0.144	0.02	0.016	0.02
<i>Cavity</i> = <i>false</i>	0.576	0.08	0.064	0.08

- **Marginalization** or **Summing Out**

$$\text{e.g. } P(\text{Weather} = \text{sunny}) = \sum_{\text{cavity}} P(\text{sunny}, \text{cavity})$$

- **Conditioning**

$$\text{Condition on } \text{Weather} = \text{sunny}: P(\text{Cavity} | \text{Weather} = \text{sunny})$$



Independence and Bayesian Networks

Independence

[Overview: Why independence matters to us](#)

[Conditional Independence](#)

[Bayes Rule](#)

Bayesian Network Representation

[Factorization of Bayesian Networks](#)

[Understanding the number of parameters](#)

[Reasoning Patterns](#)

[Types of probabilistic influence](#)

[Direct separation and probabilistic independence](#)

[D-separation: practical algorithm \(1/2\)](#)

[D-separation: practical algorithm \(2/2\)](#)

[D-separation: algorithm execution](#)

Semantics of a Bayesian Network

[Bayesian Networks: Global semantics](#)

[Bayesian Networks: Local semantics](#)

[Markov Blanket](#)

Independence

Given a sample space Ω and two events A, B over Ω :

- A and B are independent denoted $P \models (A \perp B)$ if and only if:

$$P(A|B) = P(A) \quad \vee \quad P(B|A) = P(B)$$

In such case, their conjunction has the following form:

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

Overview: Why independence matters to us

Let's assume we have a system described by k Boolean variables X_i .

Suppose we want to inference queries out of the system.

The only way we've seen so far to inference so far is by means of the Full Joint Distribution:

$$\begin{aligned} P(X_1, \dots, X_k) \\ &= P(X_1 | X_2, \dots, X_k) P(X_2, \dots, X_k) \\ &= P(X_1 | X_2, \dots, X_k) P(X_2 | X_3 \dots X_k) P(X_3 \dots X_k) \\ &= \dots \end{aligned}$$

To fully specify this joint distribution for k Boolean variables, we must assign a probability to every possible atomic event. Since there are 2^k possible combinations of values this requires $2^k - 1$ independent parameters.

Independence allows to simplify the terms in the chain rule.

If a variable X_i is conditionally independent of its predecessors given a small set of "parents," we can replace the complex term $P(X_i | X_{i+1} \dots X_k)$ with a much simpler local probability:

$$P(X_i | \text{Parents}(X_i))$$

In the extreme case, if all variables were **absolutely independent**, we would only need $k - 1$ **parameters**, instead of $2^k - 1$, to represent that joint distribution:

$$\begin{aligned} P(X_1, \dots, X_k) \\ &= P(X_1) P(X_2) \dots P(X_k) \\ &= \Theta_1 \Theta_2 \dots \Theta_k \end{aligned}$$

Conditional Independence

Absolute (marginal) dependence are powerful but rare. More commonly, conditional independence is encountered.

Given the following random variables X, Y, Z :

- **X is independent of Y given Z if:**

$$\begin{aligned} P(X | Y, Z) &= P(X | Z) \\ &\equiv \\ P(Y | X, Z) &= P(Y | Z) \end{aligned}$$

Notation: $P \models (X \perp Y | Z)$

EXAMPLE: Let's assume our system is described by the *Toothache*, *Cavity*, *Catch* Boolean random variables. So to fully specify the FJD $2^3 - 1$ parameters are needed.

But if we observe:

| If I have a cavity, the probability that the probe catches in it doesn't depend on whether I have a toothache

In other words:

$$\begin{aligned} P(\text{Catch}|\text{Toothache}, \text{Cavity}) &= P(\text{Catch}|\text{Cavity}) \\ &\equiv \\ P(\text{Toothache}|\text{Catch}, \text{Cavity}) &= P(\text{Toothache}|\text{Cavity}) \end{aligned}$$

Then the FJD using the chain rule becomes:

$$\begin{aligned} \mathbf{P}(\text{Toothache}, \text{Catch}, \text{Cavity}) \\ &= \mathbf{P}(\text{Toothache}|\text{Catch}, \text{Cavity})\mathbf{P}(\text{Catch}|\text{Cavity})\mathbf{P}(\text{Cavity}) \\ &= \mathbf{P}(\text{Toothache}|\text{Cavity})\mathbf{P}(\text{Catch}|\text{Cavity})\mathbf{P}(\text{Cavity}) \end{aligned}$$

So $2 + 2 + 1 = 5$ independent parameters are needed.

Bayes Rule

From product rule (recall $P(A \wedge B) = P(A|B)P(B) = P(B|A)P(A)$ holds)

The Bayes Rule states the following:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Which rewrite in another, more familiar way, is:

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

Useful to revert the question from cause to effect.

EXAMPLE: Say 1 individual in 50,000 suffers from meningitis, 1% from a stiff neck, and 70% of the times meningitis causes a stiff neck. What is the probability that an individual with a stiff neck has meningitis?

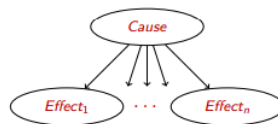
$$\begin{aligned}
 P(M) &= \frac{1}{50000} = 0.00002 \\
 P(S) &= 0.01 \\
 P(S|M) &= 0.70 \\
 P(M|S) &= ? \\
 P(M|S) &= \frac{P(S|M)P(M)}{P(S)} = \frac{0.000014}{0.01} = 0.0014
 \end{aligned}$$

So in 0.14 percent of the times, stiff neck is cause for meningitis.

Another useful application of the Bayes Rule is when we take a look at how it fits with conditional independence:

$$\begin{aligned}
 &P(\text{Cavity} \mid \text{toothache} \wedge \text{catch}) \\
 &= \alpha P(\text{toothache} \wedge \text{catch} \mid \text{Cavity}) P(\text{Cavity}) \\
 &= \alpha P(\text{toothache} \mid \text{Cavity}) P(\text{catch} \mid \text{Cavity}) P(\text{Cavity})
 \end{aligned}$$

Bayes' Rule and conditional independence



This is an example of a naive Bayes model:

$$P(\text{Cause}, \text{Effect}_1, \dots, \text{Effect}_n) = P(\text{Cause}) \prod_i P(\text{Effect}_i \mid \text{Cause})$$

Total number of parameters is linear in n

Bayesian Network Representation

Simply said, a Bayesian Network is a probabilistic graphical model made to encode independence relationships between variables in the reasoning system and is composed of:

- **Structure:** A Directed Acyclic Graph (DAG) in which nodes are Variables and edges point from causes (parents) to their direct effects (children):

$$[Cause] \rightarrow [Effect]$$

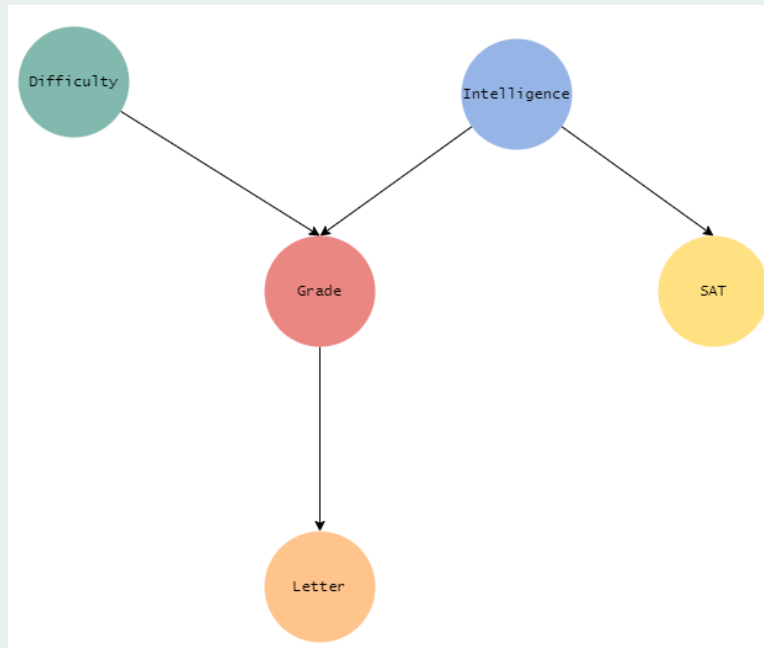
- **Probabilistic Semantics:** conditional probability distributions (or tables, in the case of discrete variables, known as CPTs) associated with each node.

Factorization of Bayesian Networks

Given a graph G and a probability distribution P , P is said to factorize over G if P can be expressed as a product of each variable's conditional probability on their parents:

$$P(X_1, \dots, X_n) = \prod_{i=1}^n P(X_i | Pa(X_i))$$

EXAMPLE:



Let's break down the network:

- Difficulty D :
 - $P(D)$
- Intelligence I :
 - $P(I)$
- Grade G :
 - $P(G|I, D)$
- SAT S :
 - $P(S|I)$
- Letter L :
 - $P(L|G)$

We can now take the product of all of these, which gives us the joint probability:

$$P(D, I, G, S, L) = P(D)P(I)P(G|D, I)P(S|I)P(L|G)$$

Understanding the number of parameters

Let's assume we have the following system:

An industry has designed a bot which sells products door-to-door but must decide which houses to visit to maximize its sales.

If the bot sees the lights of a house off, it should skip that house

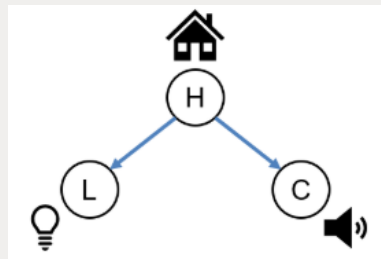
If the bot don't hear noise from the house, it should skip the house.

So we can introduce three variables to describe the system:

- H : whether or not someone is home
- L : whether or not lights are on
- C : whether or not there's noise detected in the house

So the FJD has $2^3 = 8$ params to remember.

A possible Bayesian Network Representation could be:



Let's add also the CPTS:

H		P(H)
F		0.4
T		0.6

H	L	P(L H)
F	F	0.75
F	T	0.25
T	F	0.17
T	T	0.83

H	C	P(C H)
F	F	0.9
F	T	0.1
T	F	0.2
T	T	0.8

The FJD has 8 rows to remember. Here we have 10!

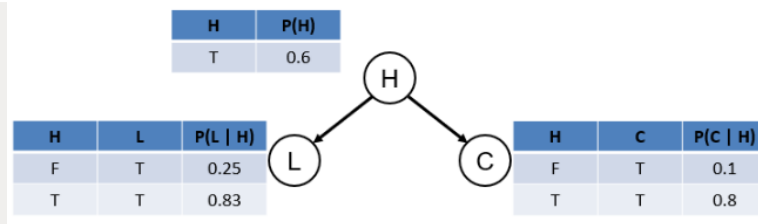
If we observe the following highlights, we can see how we could reduce the number of rows that we must remember:

H		P(H)
F		0.4
T		0.6

H	L	P(L H)
F	F	0.75
F	T	0.25
T	F	0.17
T	T	0.83

H	C	P(C H)
F	F	0.9
F	T	0.1
T	F	0.2
T	T	0.8

They all sum to 1!



So with factored CPTs just 5 rows are needed.

In general the number of parameters needed for each variable X_i is:

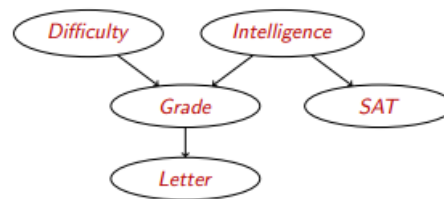
$$\#_{X_i} = (\text{arity of } X_i - 1) \cdot \text{number of parents combinations}$$

If the variable has no parents, then is multiplied by 1. Hence, for L this is:

$$2 = 1 \cdot 2$$

Reasoning Patterns

We can now try to understand how we can use different methods of reasoning in Bayesian Networks. Let us consider the example above.



Three types of reasoning can be done on Bayesian Networks:

- **Casual:** When we want to predict the "downstream" effect of various factors
- **Evidential:** When we reason from effects to causes
- **Intercasual:** Reasoning between causes of a common effect

What can we say about the independences that hold in the network?

$$P \models (L \perp I, D, S | G)$$

$$P \models (S \perp D, G, L | I)$$

$$P \models (G \perp S | I)$$

$$P \models (D \perp I, S)$$

Types of probabilistic influence

In order to determine independence between two arbitrary variables in a Bayesian Network, we should study the types of influence between them. We first have to distinguish between *active* and *inactive* trails:

- **active trail:** a path is active if it implies dependence
- **inactive trail:** if all the paths that connect two variables are not active, then they are **d-separated**.

Suppose we have X, Y, Z random variables.

- $X \rightarrow Y$: **direct cause** **active**
- $X \rightarrow Y$: **direct effect** **active**
- $X \rightarrow Z \rightarrow Y$: **casual trail** active if Z is not observed
- $X \leftarrow Z \leftarrow Y$: **evidential trail** active if Z is not observed
- $X \leftarrow Z \rightarrow Y$: **common cause** active if Z is not observed (also known as **v-structure**)
- $X \rightarrow Z \leftarrow Y$: **common effect** active if and only if Z is observed or one of Z 's descendants is observed

Direct separation and probabilistic independence

Definition (d-separation):

Two sets of nodes \mathbf{X}, \mathbf{Y} are d-separated given \mathbf{Z} if there is no active trail between any $X \in \mathbf{X}$ and $Y \in \mathbf{Y}$ given \mathbf{Z}

D-separation: practical algorithm (1/2)

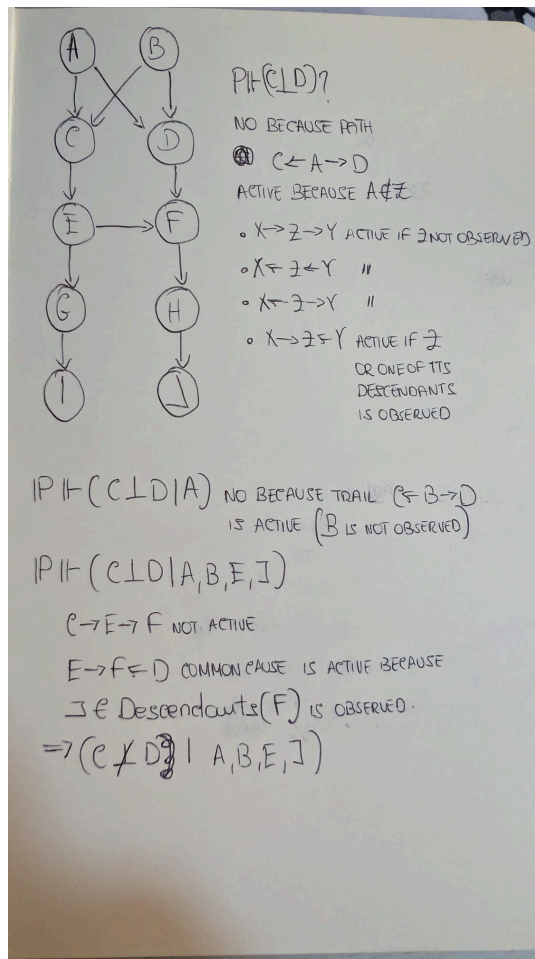
1. Draw the Ancestral graph — i.e., draw a reduced version of the network, consisting only of the variables mentioned and all of their ancestors.
2. Moralize the graph by connecting the parents. For each pair of variables with a common child, draw an undirected edge between them.
3. Disorient the graph, i.e., replace the directed edges with undirected ones.
4. Delete the givens and their edges — for example, in the question "Are X and Y independent given Z ?", then Z and all its edges must be deleted.
5. Read the graph — if the variables are disconnected in this graph, they are guaranteed to be independent. If the variables are connected in this graph, they are not guaranteed to be independent.

D-separation: practical algorithm (2/2)

1. traverse the graph bottom-up marking all nodes in Z or having descendants in given Z
2. traverse the graph from X to Y , stopping if we get to a blocked node
3. if we can't reach Y , then X and Y are independent

A node is blocked if either the middle of an unmarked v-structure, or in Z (not both)

D-separation: algorithm execution



Semantics of a Bayesian Network

The **semantics of a Bayesian Network** define the rules by which probabilistic queries can be made from the network.

Bayesian Networks: Global semantics

Global semantics tell us that the full joint distribution is the product of the local conditional distributions. We already knew this but this property formalize it.

For defining this product, a linear ordering of the nodes of the network has to be given:

$$\begin{aligned}
 P(X_1, \dots, X_n) \\
 &= \prod_{i=1}^n P(X_i | Pa(X_i)) \\
 &= \{\text{Product of all CPTs in the BN}\}
 \end{aligned}$$

Bayesian Networks: Local semantics

Local semantics, instead, tell us that each node is conditionally independent of its nondescendants given its parents

$$P(X_1 | X_2, \dots, X_n) = P(X_1 | Pa(X_1))$$

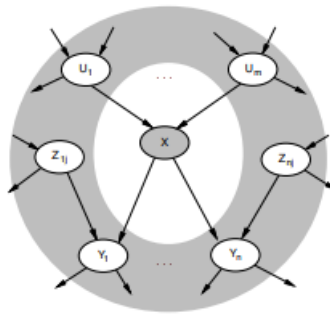
with $Pa(X_1) \subseteq \{X_2, \dots, X_n\}$

An additional theorem tell us that local semantics \iff global semantics.

Markov Blanket

Each node is conditionally independent of all others given its **Markov blanket**:

parents + children + children's parents



Probability given the Markov blanket is calculated as follows:

$$P(x_i' | mb(X_i)) = \alpha P(x_i' | \text{parents}(X_i)) \prod_{Z_j \in \text{Children}(X_i)} P(z_j | \text{parents}(Z_j))$$



Building Bayesian Networks

[Constructing Bayesian Networks](#)

[Model Sources](#)

[Casual Networks](#)

[Do-operator](#)

[Representing conditional distributions](#)

[Noisy-OR](#)

[Continuous Variables in Bayesian Networks](#)

[Finding CPTs](#)

Constructing Bayesian Networks

In order to construct a Bayesian network one needs a method such that a series of locally testable assertions of conditional independence guarantees the required global semantics.

1. Choose an ordering of variables X_1, \dots, X_n .
This ordering is usually chosen by following a causality intuition.
2. For i to n :
 - Add X_i to the network
 - Select parents from X_1, \dots, X_{i-1} such that:

$$P(X_i | Pa(X_i)) = P(X_i | X_1, \dots, X_{i-1})$$

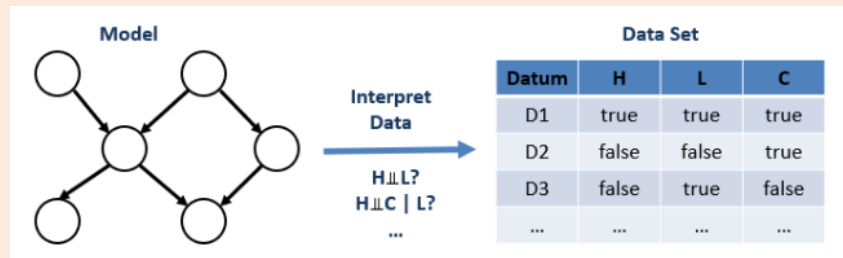
The choice of parents guarantees the global semantics:

$$\begin{aligned} P(X_1, \dots, X_n) &= \prod_{i=1}^n P(X_i | X_1, \dots, X_{i-1}) && \text{(chain rule)} \\ &= \prod_{i=1}^n P(X_i | Pa(X_i)) && \text{(by construction)} \end{aligned}$$

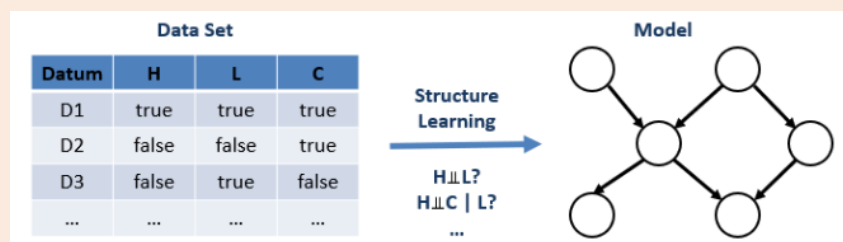
Model Sources

There are two main approaches for creating and then using Bayesian Networks:

- **Top-down:** start with a programmer-curated model and then interpret witnessed data through the model



- **Bottom-up:** start with data and attempt to learn the model based on witnessed independence relationships'



Casual Networks

In principle, any ordering of nodes permits a consistent construction of the network.

A **Casual Bayesian Network** is a Bayesian Network wherein the structure ALSO encodes **casual relations**, such that for two variables $X \rightarrow Y$, we say X is a "direct cause" of Y , meaning y responds to the manipulation of X but not vice versa:

$$\begin{aligned}
 Pa(Y) &\rightarrow Y \\
 \Rightarrow \\
 Y &= f_Y(Pa(Y)) \quad \text{called structural equation}
 \end{aligned}$$

So in principle:

- Causal networks are build to represent **causal asymmetries**
- The arrow directionality is important and it respond to the question:

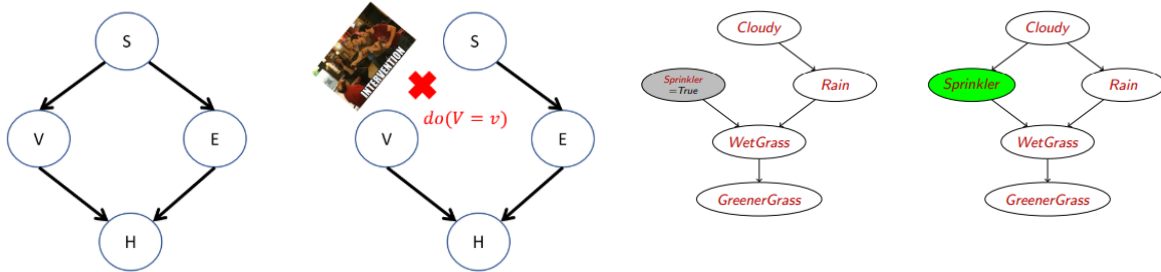
| **Which responds to which?**

Do-operator

The $do(\cdot)$ operator is a tool used in casual networks important for representing interventions and predicting their observable consequences.

Interventions force variables to attain some value.

Structurally speaking, the effect of an intervention $do(X = x)$ creates a new Bayesian Network which consists of the structure of the original with all inbound edges to X removed:



For instance one query in which we can see how the $do(\cdot)$ operator works is:

- $P(H = 1 | V = 1) = ?$
- $P(H = 1 | do(V = 1)) = ?$

The difference is:

- In the first one we are asking what is the health of people who happen to vape
- In the second one we are asking what would be the health of the population if I **force** everyone to vape

So the $do(\cdot)$ operator allows to respond to "What-if" questions

Practically speaking the difference is:

- In the first case we need to calculate $P(V = 1)$
- In the second case we assume $P(V = 1) = 1$ and every $P(V = 1 | \dots) = 1$

Representing conditional distributions

Noisy-OR

The number N of independent entries in the CPT grows exponentially with the number of parents.

In large networks this could be a problem. One solution is to use instead of free distributions, canonical parameterized that are defined compactly.

Noisy-OR distributions model multiple noninteracting causes. $Pa(X)$ should be all discrete random variables.

HOW TO: Given X how can we model $P(X | Pa(X))$ using Noisy-OR?

1. All possible causes U_i for an event X are listed
2. Negated causes $\neg U_i$ do not have any influence on X
3. Independent failure probability q_i for each cause:

$$P(X|U_1, \dots, U_k) = 1 - \prod_{i=1}^k q_i$$

equivalently

$$P(\neg X|U_1, \dots, U_k) = \prod_{i=1}^k q_i$$

<i>Cold</i>	<i>Flu</i>	<i>Malaria</i>	<i>P(Fever)</i>	<i>P(¬Fever)</i>
F	F	F	0.0	1.0
F	F	T	0.9	0.1
F	T	F	0.8	0.2
F	T	T	0.98	$0.02 = 0.2 \times 0.1$
T	F	F	0.4	0.6
T	F	T	0.94	$0.06 = 0.6 \times 0.1$
T	T	F	0.88	$0.12 = 0.6 \times 0.2$
T	T	T	0.988	$0.012 = 0.6 \times 0.2 \times 0.1$

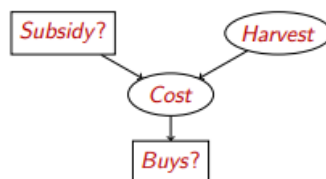
$\mu(X|U_1 \dots U_j, \neg U_{j+1} \dots \neg U_k) = 1 - \prod_{i=1}^j q_i$

Cold	Flu	Malaria	$\mu(\text{Fever})$	$\mu(\neg \text{Fever})$
F	F	F	0	1
F	F	T	0.9	0.1
F	T	F	0.8	0.2
F	T	T	0.98	$0.02 = 0.2 \times 0.1$
T	F	F	0.4	0.6
T	F	T	0.94	$0.06 = 0.6 \times 0.1$
T	T	F	0.88	$0.12 = 0.6 \times 0.2$
T	T	T	0.988	$0.012 = 0.6 \times 0.2 \times 0.1$

Continuous Variables in Bayesian Networks

So far, we have considered discrete values for random variables.

Imagine we have a random variable which is supposed to represent some type of continuous value (e.g. Cost).



There is actually nothing in the definition of what is a Bayesian Network that tells us that we can only use discrete variables, we simply need the given Conditional Probability Distributions.

Then the CPTs for *Buys* (discrete) will have to deal with all possible values of its parent (infinite).

- **Solution 1:** Discretize *Cost* values (possible large errors and still large CPTs)

- **Solution 2:** Instead of using free distributions, use parameterized canonical families

- Case 1

Continuous variable with discrete + continuous parents (e.g. *Cost*)

One approach is to model the distribution of the child as a Gaussian with parameters (the mean μ) that depends on both discrete and continuous parents:

$$P(\text{Cost} = c | \text{Harvest} = h, \text{Subsidy?} = \text{true}) \\ = \mathcal{N}(a_t h + b_t, \sigma_t)(c) = \frac{1}{\sigma_t \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{c - (a_t h + b_t)}{\sigma_t} \right)^2}$$

In case of a non-Boolean discrete parent, each combination of discrete variables is associated with a Gaussian with parameters that depends on the continuous parents. This results in a multivariate Gaussian associated on the child node.

- Case 2

Discrete variable with continuous parents (e.g. *Buys*)

Here the variable can be modeled by means of a sigmoid distribution once again parameterized on the value of the continuous parent

$$P(\text{Buys?} = \text{true} | \text{Cost} = c) = \frac{1}{1 + \exp(-2 \frac{c + \mu}{\sigma})}$$

Finding CPTs

Conditional distributions of the nodes in the BN seen so far have always been given to us.

But in real-world CPTs need to be found. How?

- Many methods for learning distributions from data. Idea is:
 - Data are evidence
 - Hypotheses are probabilistic theories about the domain
- Bayesian learning calculates the probability of each hypothesis, given the data
- Approximations: MAP hypothesis and maximum-likelihood hypothesis
- EM algorithm for learning with hidden variables



Exact Inference

Exact Inference

Inference by enumeration

Inference by variable elimination

Defining factors

Sum Out

Multiply

Practical Example

Irrelevant variables

Complexity of Exact Inference

Exact Inference

Inference by enumeration

Enumeration Inference is an inference algorithm that can compute any query via *sums-of-products* of conditional probabilities from the CPTs.

Enumeration begins by categorizing BN variables in either:

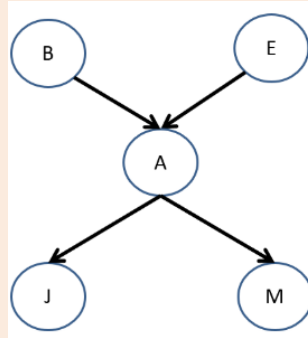
- Query Variable
- Evidence Variables
- Hidden Variables

Once we've identified which variables in the network belong to which category, the objective is to compute the query as follows:

$$P(Q|e) = \frac{P(Q, e)}{P(e)} = \frac{\sum_{h \in H} P(Q, e, h)}{P(e)}$$

Example:

Given the following BN:



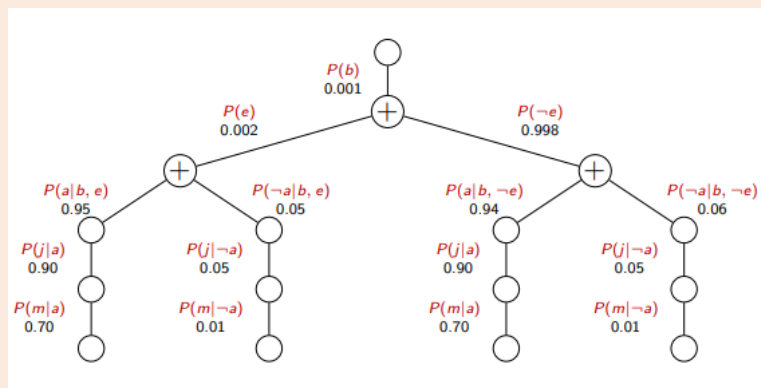
Let's say we are interested in computing $P(B|J = j, M = m)$.

Enumeration proceed as follows:

$$\begin{aligned}
 P(B|J = j, M = m) &= \alpha \sum_e \sum_a P(B, a, e, j, m) \\
 &= \alpha \sum_e \sum_a P(B)P(e)P(a|B, e)P(j|a)P(m|a) \\
 &= \alpha \cdot P(B) \sum_e P(e) \sum_a P(a|B, e)P(j|a)P(m|a)
 \end{aligned}$$

Problem: We can notice how the second summation ($\sum_a \dots$) will be calculated for multiple values of E in the outer sum ($\sum_e \dots$).

We can see it further with the evaluation tree:



Note how $P(j|a)$, $P(m|a)$, $P(j|\neg a)$, $P(m|\neg a)$ get calculated multiple times.

Inference by variable elimination

The variable elimination algorithm uses dynamic programming (storing intermediate calculations).

The general idea of VE is:

1. Define factors

2. Perform operations on these factors (sum out, multiply)

Defining factors

A factor is an abstract concept. It can represent a joint probability or a conditional probability.

For each node in the Bayesian network, we need to take the conditional probability distribution and convert it into a factor

Sum Out

To eliminate a hidden variable, we need to find all the factors containing the variable, multiply the factors together, and sum out the variable from the product.

Suppose that we have a factor f with variables X_1 up to X_j .

We want to sum out X_1 . X_1 's domain contains k values, v_1 up to v_k .

The new factor will depend only on X_2 to X_j variables and the rows will be calculated summing out the values in the original factor for each value of X_1 for each combination of the other variables.

Example:

We f_1 factor:

$f_1(X, Y, Z):$	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

Which rows should be in the new factor if we want to sum out Y ?

We can start by filling the value combinations of X and Z .

Then, for each combination we go in f_1 and add the rows containing the combination (since Y is binary two rows are found, so add the two values).

$f_2(X, Z):$	X	Z	val
	t	t	0.57
	t	f	0.43
	f	t	0.54
	f	f	0.46

Multiply

To eliminate a hidden variable, we need to find all the factors containing the variable, multiply the factors together, and sum out the variable from the product. Multiply is the first part of this step.

If we multiply f_1 and f_2 together, we will produce a new factor which contains all the variables that appear in either factor.

Example:

We have two factors f_1 and f_2 .

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

Let's multiply them together:

The new set of variables should be the union of the two sets. It should contain all three variables X , Y , and Z

Suppose we want to fill in the value for $X = 1, Y = 1, Z = 0$.

- We need the $f_1(1, 1)$ and $f_2(1, 0)$
- Multiply them together (0.07) and put the row in the new factor:

$$f_3(1, 1, 0) = f_1(1, 1) \cdot f_2(1, 0)$$

The new factor $f_3 = f_1 \times f_2$ is given below:

$f_1 \times f_2$:

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

Practical Example

$P(E)=0.1$

$P(B)=0.3$

$E \rightarrow A \rightarrow W$

$B \rightarrow A$

E	B	A
f	f	0.1
f	t	0.7
t	f	0.2
t	t	0.8

$A \rightarrow W$

A	W
t	0.8
f	0.4

$$P(B|ta) \propto \sum_e \sum_w P(B, ta, e, w)$$

\uparrow

W IRRELEVANT, $W \notin \text{Ancestors}(\{B, A\}) = \{E\}$

$$= \alpha \cdot \sum_e P(B, ta, e)$$

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

$$= \alpha \cdot \underbrace{P(B)}_{f_3} \sum_e \underbrace{P(e)}_{f_2} \underbrace{P(A|B, e)}_{f_1}$$

$$= \alpha \cdot f_3(B) \sum_e f_2(e) f_1(ta, B, e)$$

$f_2(e) \cdot f_1(ta, B, e) = f_2(e) \cdot f_5(B, e)$

$= f_4(B, e)$

NOTE HOW $f_2(e) = P(E)$

$f_1(ta, B, e) = P(A|B, e)$

BUT WE HAVE $P(A|B, e)$ FROM EPT

SO

E	B	$f_5(B, e)$
f	f	0.9
f	t	0.3
t	f	0.8
t	t	0.2

E	B	$f_4(B, E)$
f	f	$0.9 \cdot 0.9 = 0.81$
f	t	$0.3 \cdot 0.9 = 0.27$
t	f	$0.8 \cdot 0.1 = 0.08$
t	t	$0.2 \cdot 0.1 = 0.02$

Now sum out E; obtaining $f_5(B)$

B $f_5(B)$

f $0.81 + 0.08 = 0.89$

t $0.27 + 0.02 = 0.29$

Now multiply $f_3(B) \cdot f_5(B) = f_6(B)$

B $f_6(B)$

f $0.89 \cdot 0.7 = 0.623$

t $0.29 \cdot 0.3 = 0.087$

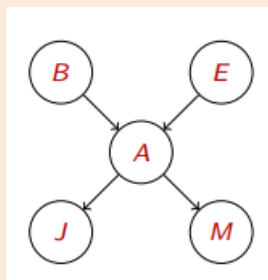
$\Rightarrow P(B|ta) = \left\langle \frac{0.623}{0.71}, \frac{0.087}{0.71} \right\rangle = \langle 0.87, 0.13 \rangle$

Irrelevant variables

Two important theorems define irrelevancy when solving inference queries:

Theorem 1: Ancestors

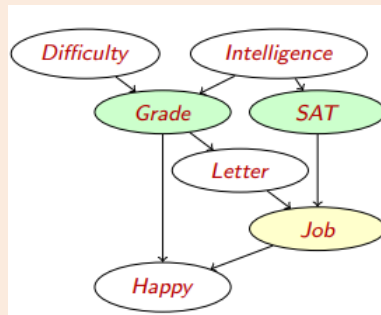
Y is irrelevant unless $Y \in \text{Ancestors}(X \cup E)$



Here, $X = \text{JohnCalls}$, $E = \{\text{Burglary}\}$ and $\text{Ancestors}(\{\text{JohnCalls}, \text{Burglary}\}) = \{\text{Alarm}, \text{Earthquake}\}$ so MaryCalls is irrelevant

Theorem 2: D-separation

Y is irrelevant if d-separated from X by \mathbf{E}



For $P(Job|Grade, SAT)$, not only *Happy* is irrelevant (outside of ancestral graph) but also *Difficulty* and *Intelligence* are (d-separated) even though they are members of $Ancestors(\{Job, Grade, SAT\})$

Complexity of Exact Inference

Theorem: BN with Polytrees

It turns out that Bayesian Networks that have a **polytree / singly connected structure**, i.e. for whom there is at most one undirected path between any node in the network, have complexity that is linear in the size of the network $O(d^k \cdot n)$



Approximate inference

Approximate Inference

[Sampling basics](#)

[Sampling in a Bayesian Network](#)

[Prior Sampling](#)

[Rejection Sampling](#)

[Likelihood weighting](#)

[Gibbs Sampling](#)

Approximate Inference

Sampling basics

The process of sampling from some probability distribution is the ability to generate “fake” data point that is consistent with the underlying distribution / training set / model.

RECALL:

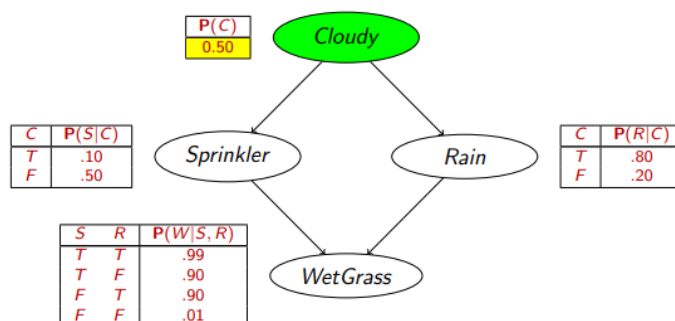
The **Law of Large Numbers** (briefly summarized) states that as the sample size trends towards infinity, the likelihood of some event converges to its true likelihood.

Sampling in a Bayesian Network

Sampling in a BN comes with the following challenges:

- There are multiple Random Variables
- Some of them are independent/conditionally independent

Given the following BN:



Suppose we have generated the samples, ignoring how we got them:

Sample	C	S	R	W
0	1	0	1	1
1	1	1	1	1
2	0	1	1	0
3	1	0	1	1
4	0	0	0	1

Examining the samples generated how would we estimate $P(W = 1)$?

$$\hat{P}(W = 1) = \frac{4}{5} = 0.80$$

How about estimating $P(C = 1|W = 1)$?

$$\hat{P}(C = 1|W = 1) = \frac{3}{4} = 0.75$$

How to **simulate sampling**?

C	P(C)	
Red	0.5	<div> <div>Red Range: [0,0.5]</div> <div>Blue: [0.5,0.7]</div> <div>Green [0.7,1]</div> </div>
Blu	0.2	
Grn	0.3	
		Example RNG values: 0.32 => Red 0.67 => Blue

The CHANCE that we sample each value from the distribution is proportionate to how likely it is in the table.

Prior Sampling

Question: What would be a reasonable way to **generate samples** from a BN?

Prior sampling ensures via a **topological sort** that we never sample a child node before we know the value of its parents in the sample

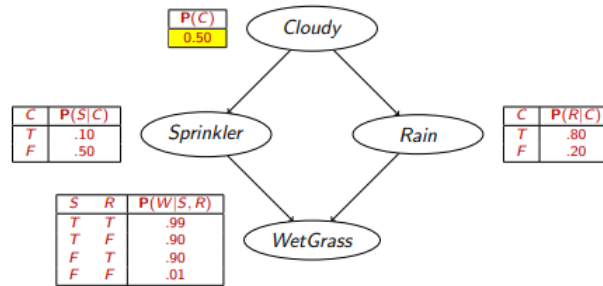
In our example, there are a couple of viable topological sorts:

- C, S, R, W (Let's use this one)
- C, R, S, W

Let's say we have the following random sequence:

0.1, 0.7, 0.6, 0.7

And the following BN:



1. We know $C = true$ with $P(C) = 0.5$ so:

$$\begin{aligned}
 & [0, 0.5] \Rightarrow true \\
 & \quad \wedge \\
 & (0.5] \Rightarrow false \\
 & \text{so having 0.1 sample will have } C = true
 \end{aligned}$$

2. Now we can restrict the CPTs to case in which $C = true$.

Turn for S , we know:

$$\begin{aligned}
 & [0, 0.1) \Rightarrow S = true \\
 & \quad \wedge \\
 & [0.1, 1] \Rightarrow S = false \\
 & \text{so having 0.7 sample will have } S = false
 \end{aligned}$$

3. Similarly done previously for R

4. Same for W

Finally we have sample $C = true, S = false, R = true, W = true$ with the random values specified.

What are the problems for Prior Sampling for estimating posterior queries like $P(C|S = 0)$?
 We may generate many samples for which $S = 1$, due to topological order.

Rejection Sampling

Rejection Sampling is an enhancement for Prior Sampling when estimating posterior queries.

The idea is simple:

Any sample inconsistent with evidence is immediately ignored / thrown out.

EXAMPLE:

Estimate $P(Rain|Sprinkler = true)$ using 100 samples.

- 27 samples have $Sprinkler = true$
- Of these 8 have $Rain = true$ and the others have $Rain = false$

$$\hat{P}(Rain|Sprinkler = true) = \langle \frac{8}{27}, \frac{19}{27} \rangle$$

What is the problem with Rejection Sampling?

It's inefficient if $P(e)$ is small

Likelihood weighting

This is another way to deal with posterior queries.

The idea is:

Fix evidence variables, sample only nonevidence variables, and weight each sample by the likelihood it accords the evidence.

Two main components when estimating a distribution $P(Q|e)$:

- Weight for a given sample (z, e) is given by:

$$w = \prod_{i=1}^m P(e_i | \text{parents}(E_i))$$

- Weighted sampling probability is:

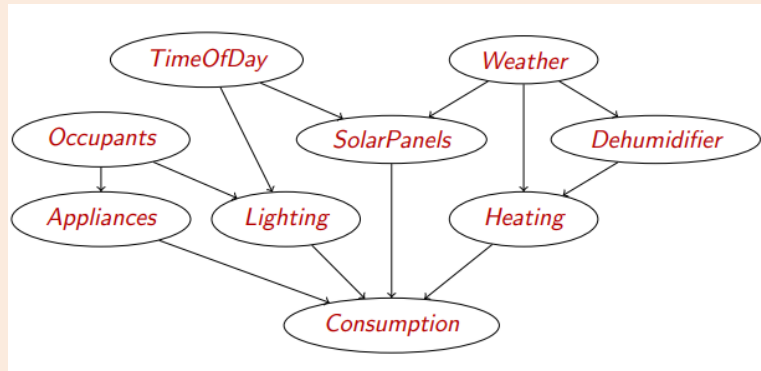
$$\hat{P}(Q|e) = w \cdot S_{WS}(z, e)$$

Example:

Estimate $P(H|S = \text{high}, D = \text{low})$, given that:

$$P(H|S = \text{high}, D = \text{low}) \\ = P(T)P(W)P(S = \text{high}|T, W)P(D = \text{low}|W)P(H|W, D = \text{low})$$

On the following BN:



So topological ordering could be:

- T, W, H
- W, T, H

Note how S and D are not included because they are fixed and their CPTs will be used instead as factors to find the *weight* of the sample.

A possible CPT for $P(S=\text{high}|t,w)$ is:

t, w	$P(S=\text{high} t, w)$
c, m	0.8 (c)
c, a	0.7 (d)
c, e	0.01 (e)
o, m	0.4 (f)
o, a	0.2 (g)
o, e	0.001 (h)

A possible CPT for $P(D=\text{low}|w)$ is:

w	$P(D=\text{low} w)$
c	0.8 (j)
o	0.4 (k)

Using the sequence .11, .93, .28, .53, .05, I need to draw a sample for T , W , and $H|W, D=\text{low}$, and assign it the correct weight.

- T has a uniform distribution, so we have $.11 \in [0, .33) \rightarrow T = \text{morning}$
- W has a uniform distribution, so we have $.93 \in (.5, 1) \rightarrow W = \text{overcast}$
- $P(H|W=\text{overcast}, D=\text{low}) = \langle b, 1-b \rangle = \langle .2, .8 \rangle$, so we have $.28 \in [.2, 1) \rightarrow H = \text{low}$
- $P(S=\text{high}|T=\text{morning}, W=\text{overcast}) = f = 0.4$
- $P(D=\text{low}|W=\text{overcast}) = k = 0.4$

Therefore, we have a sample with $H=\text{low}$ and weight = $0.4 \times 0.4 = 0.16$

Gibbs Sampling

Gibbs Sampling is a method for efficiently sampling posterior queries $P(Q|e)$ consisting of the following steps:

1. Initialize all variables to some random assignment, except for the evidence e which remains fixed to its value.
2. Sample a single variable at a time conditioned on the rest (again keeping e fixed).
3. Repeat Step 2 many times.