

Grundlagen der künstlichen Intelligenz – Bayesian Networks

Matthias Althoff

TU München

November 28, 2019

11/28

12/05

12/12

Organization

- 1 Acting under Uncertainty
- 2 Basics of Probability Theory
- 3 Bayesian Networks
- 4 Inference in Bayesian Networks
- 5 Approximate Inference in Bayesian Networks

The content is covered in the AI book by the section “Quantifying Uncertainty” and “Probabilistic Reasoning”.

Learning Outcomes

Exam-related

- You understand the concept of *probability space*, *random variable*, *expectation*, and conditional probability.
- You can apply Bayes' Rule.
- You can determine whether two random variables are independent or conditionally independent.
- You can create a Bayesian network.
- You can apply inference by enumeration and inference by variable elimination for a given Bayesian network.
- You can apply variable ordering and variable relevance to simplify inference computations.
- ~~✗~~ You can apply approximate Monte Carlo methods for inference using *direct sampling*, *rejection sampling*, and *likelihood weighting*.

Motivation

- In many cases, our knowledge about the world is incomplete (not enough information) or uncertain (sensors are unreliable).
- Often, laws governing the environment are not known or simply do not exist (as in an inherently non-deterministic world).
- Nevertheless, we have to act rationally given uncertain information.

Example

- Given is a plan A_{90} for getting to the airport on time when leaving 90 minutes before the flight.
- We can only infer “Plan A_{90} gets us to the airport on time, as long as the car does not break down, runs out of gas, is not involved in an accident, etc.”
- None of the conditions hold for sure. This also holds for A_{180} , A_{900} , etc.
- We need a measure to minimize the expected cost to the goal.

Another Example

Diagnosis/Expert System for dentists

Consider the following simple rule:

$$\textit{Toothache} \Rightarrow \textit{Cavity}$$

This is not always true! Better:

$$\textit{Toothache} \Rightarrow \textit{Cavity} \vee \textit{GumProblem} \vee \textit{Abscess} \vee \dots$$

... but we do not even know all causes! Maybe better expressed as a causal rule?

$$\textit{Cavity} \Rightarrow \textit{Toothache}$$

Not true either since not all cavities cause pain.

Problem with logics:

- **Laziness**: enumerating all causes is too much work or even impossible.
- **Theoretical ignorance**: the governing laws might be unknown (e.g., medicine).
- **Practical ignorance**: our world description is not sufficiently accurate.

Degrees of Belief

- We (or other agents) are only convinced of rules and facts up to a certain degree.
- One option to express the **degree of belief** is to use **probabilities**.
- The agent is convinced of a sensor reading to 0.9, i.e., the agent believes the reading will be correct 9 out of 10 times.
- Probabilities subsume the uncertainty caused by the lack of knowledge.

Sample Space and Event Space

Let us first recall some basics of probability theory:

Sample space

In probability theory, the set of possible outcomes is called the **sample space** Ω

Example: $\Omega = \{heads, tails\}$ for tossing a coin

We denote elements of Ω by $\omega \in \Omega$.

Event space

The **event space** \mathcal{F} is the powerset of Ω and contains all possible combinations of outcomes.

Example: $\mathcal{F} = \{\emptyset, \{heads\}, \{tails\}, \{heads, tails\}\}$ for tossing a coin

Probability Space

A probability space consists of

- a sample space Ω ,
- an event space \mathcal{F} ,
- a function P that assigns a probability to each event $e_i \in \mathcal{F}$, such that
 - ① $P(e_i) \geq 0$
 - ② $P(e_1 \cup e_2 \cup \dots) = \sum_i P(e_i)$ when events $e_i \in \mathcal{F}$ are mutually exclusive.
 - ③ $\sum_{\omega \in \Omega} P(\omega) = 1$

Example: Tossing a coin

$$P(\emptyset) = 0, P(heads) = 1 - P(tails), \sum_{\omega \in \Omega} P(\omega) = 1.$$

Random Variable

Since we wouldn't like to work with heads or tails, we will use random variables

For convenience, we introduce a function

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

from the sample space to some set \mathcal{D} . We call X a **random variable**.

Example: Tossing a coin

$$X(\omega) = \begin{cases} X(\omega) = 1, & \text{if } \omega = \text{heads}, \\ X(\omega) = 2, & \text{if } \omega = \text{tails}. \end{cases}$$

where $\mathcal{D} = \{1, 2\}$.

Expectation

The expectation of a random variable is defined as

$$\star E(X) = \sum_{x \in \mathcal{D}_X} x P(X = x),$$

where \mathcal{D}_X is the domain of X . The result is an average outcome over infinitely many experiments.

Example: Throwing a dice

$$\mathcal{D}_X = \{1, 2, 3, 4, 5, 6\}$$

$$E(X) = \sum_{x \in \mathcal{D}_X} x P(X = x) = \sum_{i=1}^6 i \frac{1}{6} = 3.5$$

Multidimensional Random Variable

The result of experiments often have to be described by several random variables. The **joint probability**

$$P((X = x), (Y = y))$$

refers to the event that $X = x$ and $Y = y$. **Marginalization:** Probabilities of single variables are obtained using the axiom $P(e_1 \cup e_2 \cup \dots) = \sum_i P(e_i)$ for mutually exclusive e_i :

prob of $X = x$
for all outcomes
of Y

$$P(X = x) = \sum_{y \in \mathcal{D}_Y} P((X = x), (Y = y))$$

Example: Throwing two dice

$$\mathcal{D}_X = \mathcal{D}_Y = \{1, 2, 3, 4, 5, 6\}$$

$$P(X = 3) = \sum_{y \in \mathcal{D}_Y} P((X = 3), (Y = y)) = \sum_{i=1}^6 \frac{1}{36} = \frac{1}{6}$$

Conditional Probability

The **conditional probability** that $X = x$ under the condition that it is known that $Y = y$ is written and defined as

$$P((X = x)|(Y = y)) = \frac{P((X = x), (Y = y))}{P(Y = y)}$$

→ Joint probability
→ Normalization term

Example: FC Bayern München (FCB); numbers are guessed

	$y_1 \hat{=}$ lives in Munich	$y_2 \hat{=}$ lives somewhere else in Germany
$x_1 \hat{=}$ fan of FCB	0.006	0.05
$x_2 \hat{=}$ not fan of FCB	0.007	0.937

- $P(Y = y_1) = P((X = x_1), (Y = y_1)) + P((X = x_2), (Y = y_1)) = 0.006 + 0.007 = 0.013.$
- $P((X = x_1)|(Y = y_1)) = \frac{P((X=x_1),(Y=y_1))}{P(Y=y_1)} = \frac{0.006}{0.013} \approx 0.46.$

Bayes' Rule

Swapping cause & effect

Rearranging the conditional probability results in

$$P((X = x), (Y = y)) = P((X = x)|(Y = y))P(Y = y) \quad (1)$$

and inserting (1) in $P((Y = y)|(X = x)) = \frac{P((X=x),(Y=y))}{P(X=x)}$ yields **Bayes' Rule**:

$$P((Y = y)|(X = x)) = \frac{P((X = x)|(Y = y))P(Y = y)}{P(X = x)}$$

Posterior Likelihood Prior
Hypothesis Data normalising Constant

Summation over all $y \in \mathcal{D}_y$ in (1) results in

$$\sum_{y \in \mathcal{D}_y} P((X = x), (Y = y)) = \sum_{y \in \mathcal{D}_y} P((X = x)|(Y = y))P(Y = y)$$

$$P(X = x) = \sum_{y \in \mathcal{D}_y} P((X = x)|(Y = y))P(Y = y)$$

which can be inserted into Bayes rule:

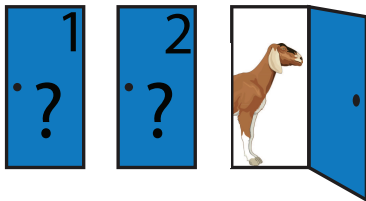
$$P((Y = y)|(X = x)) = \frac{P((X = x)|(Y = y))P(Y = y)}{\sum_{y \in \mathcal{D}_y} P((X = x)|(Y = y))P(Y = y)}$$

Bayes' Rule: Monty Hall Problem (1)

- The Monty Hall problem is based on the American TV game show *Let's Make a Deal* and is named after its host *Monty Hall*.
- The corresponding German TV show was *Geh aufs Ganze* with *Jörg Draeger*.

Monty Hall problem

Suppose you're on a game show, and you're given the choice of three doors: Behind one door is a car; behind the others, goats. You pick a door, say No. 1, and the host, who knows what's behind the doors, opens another door, say No. 3, which has a goat. He then says to you, "Do you want to pick door No. 2?" Is it to your advantage to switch your choice?



Tweedback Question

Should the candidate change the door?

yes!

Bayes' Rule: Monty Hall Problem (2)

$X = x_i \hat{=}$ the prize is behind door i ($\mathcal{D}_x = \{1, 2, 3\}$).

$Y = y_j \hat{=}$ the host has opened door j ($\mathcal{D}_y = \{1, 2, 3\}$).

$$P(X = x_1) = P(X = x_2) = P(X = x_3) = \frac{1}{3}$$

the host has to
choose bet door
2 or 3 only

$$P(Y = y_3 | X = x_1) = \frac{1}{2} \quad (\text{reminder: we pick door 1})$$

while the host opened door 3

$$P(Y = y_3 | X = x_2) = 1$$

$$P(Y = y_3 | X = x_3) = 0$$

We would like to know $P(X = x_2 | Y = y_3)$ using Bayes' rule:

$$\begin{aligned} P(X = x_2 | Y = y_3) &= \frac{P(Y = y_3 | X = x_2)P(X = x_2)}{\sum_{x \in \mathcal{D}_x} P(Y = y_3 | X = x)P(X = x)} \\ &= \frac{1 \cdot \frac{1}{3}}{\frac{1}{2} \cdot \frac{1}{3} + 1 \cdot \frac{1}{3} + 0 \cdot \frac{1}{3}} = \frac{2}{3} \end{aligned}$$

Changing the door doubles the chance of winning from $\frac{1}{3}$ to $\frac{2}{3}$.

Independence of Random Variables

From the definition of conditional probability follows that

$$P((X = x), (Y = y)) = P((X = x)|(Y = y))P(Y = y).$$

Assuming that $P((X = x)|(Y = y)) = P(X = x)$, i.e., X is independent from Y :

$$P((X = x), (Y = y)) = P(X = x)P(Y = y)$$

for stochastic independent variables.

Example: FC Bayern München (FCB); numbers are guessed

	$y_1 \hat{=}$ lives in Munich	$y_2 \hat{=}$ lives somewhere else in Germany
$x_1 \hat{=}$ fan of FCB	0.006	0.05
$x_2 \hat{=}$ not fan of FCB	0.007	0.937

- $P(Y = y_1) = 0.013$ ("lives in Munich");
- $P(X = x_1) = 0.056$ ("fan of FCB");
- $P((X = x_1), (Y = y_1)) = 0.006 \neq 0.013 \cdot 0.056 \approx 7 \cdot 10^{-4} \rightarrow$ not independent.

Tweedback Questions

- What is the conditional probability $P(Y = y_2 | X = x_1)$ given

	$y_1 \hat{=}$ has a 'Karoheemd'	$y_2 \hat{=}$ not y_1
$x_1 \hat{=}$ studies mech. eng.	0.08	0.02
$x_2 \hat{=}$ not x_1	0.1	0.8

- A 2%
- B 8%
- ☒ C 20%

$$\therefore P(Y=y_2, X=x_1) = 0.02$$

$$\therefore P(X=x_1) = 0.08 + 0.02 = 0.1$$

$$\therefore P(Y=y_2 | X=x_1) = \frac{P(Y=y_2, X=x_1)}{P(X=x_1)} = 0.2$$

- Are X and Y independent?

X & Y are dependent

Conditional Independence

The random variable X is conditionally independent of Y given Z if

$$\star P((X = x)|(Y = y), (Z = z)) = P((X = x)|(Z = z)).$$

This is often written as

$$\begin{aligned} \star & P((X = x), (Y = y)|(Z = z)) \\ &= \frac{P((X = x), (Y = y), (Z = z))}{P(Z = z)} \\ &= \frac{P((X = x)|(Y = y), (Z = z))P((Y = y)|(Z = z))P(Z = z)}{P(Z = z)} \\ &= P((X = x)|(Y = y), (Z = z))P((Y = y)|(Z = z)). \end{aligned}$$

Notation Simplifications

- From now on, we are abusing the notation for the sake of simplification.
- Instead of $P(\textit{Weather} = \textit{sunny}) = 0.6$, we write $P(\textit{sunny}) = 0.6$. This requires that *sunny* is not used by another random variable.
- For Boolean random variables, such as $\textit{Cavity} \in \{\textit{true}, \textit{false}\}$, $\textit{Toothache} \in \{\textit{true}, \textit{false}\}$, $\textit{Teen} \in \{\textit{true}, \textit{false}\}$, we simplify

$$P((\textit{Cavity} = \textit{true}) | (\textit{Toothache} = \textit{false}), (\textit{Teen} = \textit{true}))$$

to $P(\textit{cavity} | \neg \textit{toothache}, \textit{teen})$

- Instead of listing all possible probabilities

$$P(\textit{Weather} = \textit{sunny}) = 0.6$$

$$P(\textit{Weather} = \textit{rain}) = 0.1$$

$$P(\textit{Weather} = \textit{cloudy}) = 0.29$$

$$P(\textit{Weather} = \textit{snow}) = 0.01$$

we write $\mathbf{P}(\textit{Weather}) = [0.6, 0.1, 0.29, 0.01]$ for a defined ordering of the domain of *Weather* (*sunny*, *rain*, *cloudy*, *snow*).

Normalization

For Boolean random variables (uppercase), we can use normalization to compute conditional probabilities.

	toothache		\neg toothache	
	catch	\neg catch	catch	\neg catch
cavity	.108	.012	.072	.008
\neg cavity	.016	.064	.144	.576

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

$$\begin{aligned}
 * \quad & \mathbf{P}(\text{Cavity} | \text{toothache}) = \alpha \mathbf{P}(\text{Cavity}, \text{toothache}) \\
 & = \alpha [\mathbf{P}(\text{Cavity}, \text{toothache}, \text{catch}) + \mathbf{P}(\text{Cavity}, \text{toothache}, \neg \text{catch})] \\
 & = \alpha [\langle \underline{0.108}, 0.016 \rangle + \langle \underline{0.012}, 0.064 \rangle] \\
 & = \alpha \langle \underline{0.12}, 0.08 \rangle = \langle 0.6, 0.4 \rangle
 \end{aligned}$$

normalizing so that both numbers
add up to one

* General idea: compute distribution on query variable by fixing **evidence variables** (here: *Toothache*) and summing over **hidden variables** (here: *Catch*).

Bayesian Networks

represent joint distribution, independence
& conditional probability

- A full joint probability distribution can answer any question about a domain, but can become intractably large as the number of variables grows.
- Independence and conditional independence can greatly reduce the amount of information required to construct the joint probability.
- **Bayesian networks** are used to represent dependencies among variables.

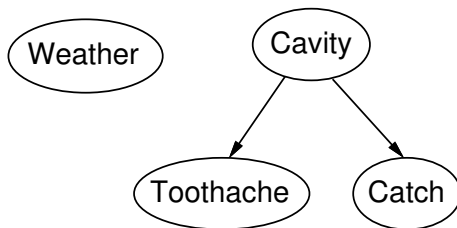
Bayesian network

A Bayesian network is a directed acyclic graph, where

- each node corresponds to a random variable,
- arrows between nodes start at parents,
- each node N_i has a conditional probability distribution $\mathbf{P}(X_i | \text{Parents}(X_i))$.

Example: Dentist

Topology of network encodes conditional independence assertions:



- *Weather* is independent of the other variables. *children of the same node*
- ~~✗~~ *Toothache* and *Catch* are conditionally independent given *Cavity*.

(*Catch*: The dentist's nasty steel probe catches in one's tooth)

Example: Burglar (1)

* BN tables are done for true values only.

Considered scenario

I'm at work, neighbor John calls to say my alarm is ringing, but neighbor Mary doesn't call. Sometimes it's set off by minor earthquakes. Is there a burglar?

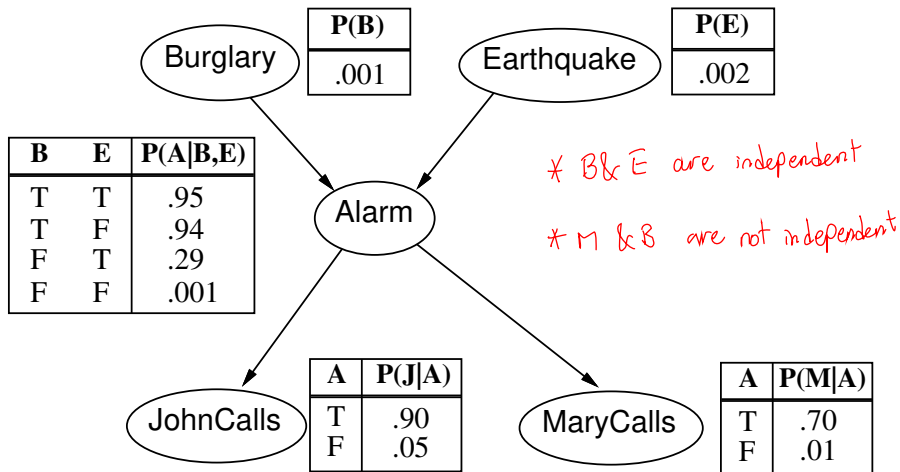


Variables: *Burglar*, *Earthquake*, *Alarm*, *JohnCalls*, *MaryCalls*

Network topology reflects “causal” knowledge:

- A burglar can set the alarm off.
- An earthquake can set the alarm off.
- The alarm can cause Mary to call.
- The alarm can cause John to call.

Example: Burglar (2)



Each row has to sum up to 1. We omit the second probability. E.g.
 $P(\neg j|a) = 1 - P(j|a) = 0.1$.

Compactness of Bayesian Networks

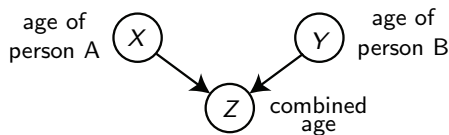
- * Each conditional probability table (see previous slide) for Boolean X_i with k Boolean parents has 2^k rows for the combinations of parent values.
- Each row requires one number p for $X_i = \text{true}$ (the number for $X_i = \text{false}$ is just $1 - p$).
- * If each variable has no more than k parents, the complete network requires $\mathcal{O}(n \cdot 2^k)$ numbers for n variables.
- Thus, the space requirement grows linearly with n vs. $\mathcal{O}(2^n)$ for the full joint distribution.
- Burglary scenario: $1 + 1 + 4 + 2 + 2 = 10$ numbers (vs. $2^5 - 1 = 31$)

Important

Tweedback Question

If X and Y are independent, are they also independent given any variables?
 I.e., if $P(X|Y) = P(X)$, can we conclude that $P(X|Y, Z) = P(X|Z)$?

No. Here is a counter example:



$$P(X|Y) = P(X)$$

$$P(X|Y, Z) \neq P(X|Z)$$

Whether two variables are conditionally independent is not obvious from the Bayesian network. We present how to infer conditional independence subsequently.

Determine Conditional Independence in Bayesian Networks

Independence

Variables X and Y are independent

$$\Leftrightarrow P(X, Y) = P(X)P(Y) \text{ or } P(X|Y) = P(X) \text{ or } P(Y|X) = P(Y)$$

\Leftrightarrow Variables X and Y share no common ancestry.

Conditional Independence

Important

Variables X and Y are conditionally independent given a set of evidences E

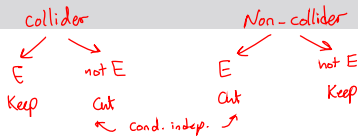
$$\Leftrightarrow P(X|Y, E) = P(X|E) \text{ or } P(Y|X, E) = P(Y|E)$$

\Leftrightarrow every path U from X to Y is blocked, i.e., there is a node w on U such that either

- w is a collider (see next slide) and neither w nor any of its descendants is in E , or
- w is not a collider on U and w is in E .

Further reading: David Barber (2012). Bayesian Reasoning and Machine Learning. Cambridge University Press (online available)

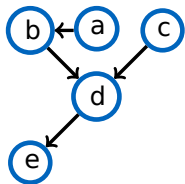
Colliders in Bayesian Networks



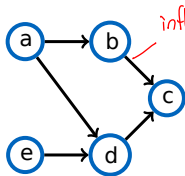
Definition

is always irrelevant
of nodes directions

Given a path R , a **collider** is a node c on R with neighbours a and b on R such that $a \rightarrow c \leftarrow b$.



d is a collider along the path $a - b - d - c$, but not along the path $a - b - d - e$.



influences

Tweedback:

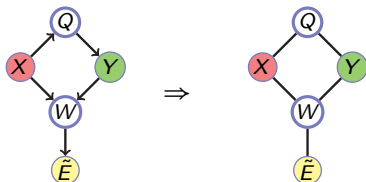
- Is d a collider along the path $a - d - e$? Yes
- Is d a collider along the path $a - b - c - d - e$? No

If d is a collider, $e \rightarrow d$
 $c \rightarrow d$

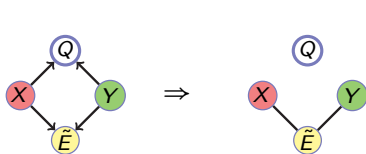
Determine Conditional Independence Graphically (1)



If \tilde{E} is a collider, keep undirected KEEP
 connections between the neighbours of the collider.



If \tilde{E} is a descendant of a collider, this KEEP
 could induce dependence, so we retain the links.



If there is a collider ^Q not in the set of evidences E (upper path), we cut the CUT
 links to the collider variable.

Determine Conditional Independence Graphically (2)

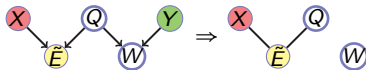


If there is a non-collider \tilde{E} in the set of evidences E (bottom path), we cut its links, which cannot induce dependence between X and Y .



In this case, neither path contributes to dependence and thus X and Y are conditionally independent given E .

X & Y are independent



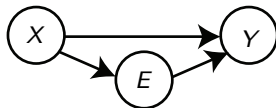
While \tilde{E} is a collider in the set of evidences E , W is a collider that is not in the set of evidences. Thus, there is no path between X and Y , and hence X and Y are independent given E .

* If there is no undirected path between X and Y after all manipulations, X and Y are independent given E .

→ Cut

Determine Conditional Independence Graphically:

Examples (1)



NO common
ancestry

X ind of Y?

X ind. of Y given E?

X ind. of Y?

X ind. of Y given E?

X ind. of Y?

X ind. of Y given E?

X ind. of Y?

X ind. of Y given E?

X ind. of Y?

X ind. of Y given E?

yes

yes

no

yes

no

yes

yes

no

no

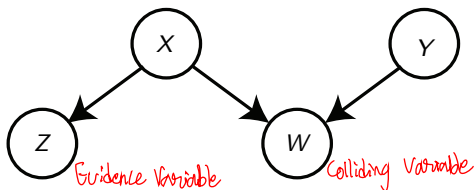
no

(NB) there need not be a causal relationship between X & Y, and they don't have to occur simultaneously.

(NB) There is no direction in independence! X indep of Y is the same as Y indep of X

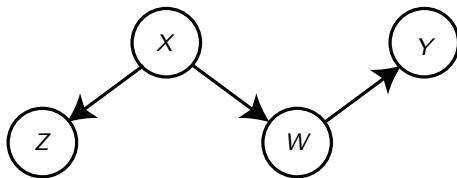
Determine Conditional Independence Graphically:

Examples (2)



$$P(X, Y) = P(X)P(Y) \quad \text{yes}$$

$$P(X|Y, Z) = P(X|Z) \quad \text{yes}$$



because X is the ancestor of Y

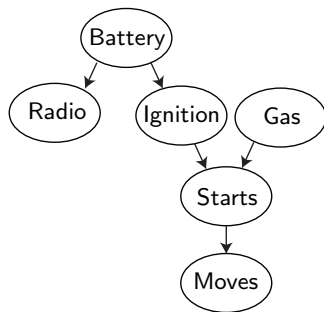
$$P(X, Y) = P(X)P(Y) \quad \text{no}$$

$$P(X|Y, Z) = P(X|Z) \quad \text{no}$$

Determine Conditional Independence Graphically:

Examples (3)

(NB) Check for paths bet X & Y first!



Radio and Ignition, given Battery? **yes**

Radio and Starts, given Ignition? **yes**

Gas and Radio, given Battery? **yes**

Gas and Radio, given Starts? **no**

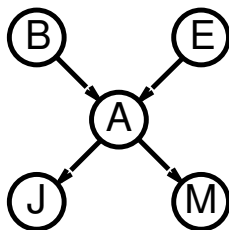
Gas and Radio, given nil? **yes**

Gas and Battery, given Moves? **no**

Further Tweedback Questions

Exam Question

Burglar scenario:



- $J \perp\!\!\!\perp M$ are not independent;
A is their common ancestry.
- $B \perp\!\!\!\perp E$ are independent.

E is not on path of B & J,
A is . A is not evidence variable
& non collider, hence we will
keep connection.

True or false?

- B and J are independent. **No**
- B is conditionally independent of J given A. **Yes**
- ~~B~~ B is conditionally independent of J given E. **No**
- J is conditionally independent of B given E. **No**
- M is conditionally independent of J given A. **Yes**

Semantics

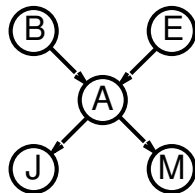
Inference

The semantics is defined by the full joint distribution as the product of the local conditional distributions:

$$\star P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i | \text{parents}(X_i))$$

e.g., $P(j, m, a, \neg b, \neg e)$

$$\begin{aligned} &= P(j|a)P(m|a)P(a|\neg b, \neg e)P(\neg b)P(\neg e) \\ &= 0.9 \cdot 0.7 \cdot 0.001 \cdot 0.999 \cdot 0.998 \\ &\approx 0.00063 \end{aligned}$$



Note that $\text{parents}()$ returns the values of the parents, while $\text{Parents}()$ returns the nodes of the parents.

Chain Rule

The semantics is a direct consequence of the chain rule. Repeated application of the product rule

$$P(x_1, \dots, x_n) = P(x_n | x_{n-1}, \dots, x_1) P(x_{n-1}, \dots, x_1)$$

yields the chain rule

$$\begin{aligned} \star P(x_1, \dots, x_n) &= P(x_n | x_{n-1}, \dots, x_1) P(x_{n-1} | x_{n-2}, \dots, x_1) \cdots P(x_2 | x_1) P(x_1) \\ &= \prod_{i=1}^n P(x_i | x_{i-1}, \dots, x_1). \end{aligned}$$

Comparison with the previous slide results in the requirement

$$P(x_i | x_{i-1}, \dots, x_1) = P(x_i | \text{parents}(X_i))$$

parents shall
have smaller
indices

provided that $\text{Parents}(X_i) \subseteq \{X_{i-1}, \dots, X_1\}$. \star This last condition is satisfied by numbering the nodes in a way that is consistent with the partial order in the Bayesian network, which is always possible (acyclic graph).

Example

We use the burglar example and choose

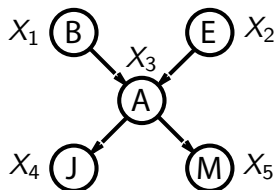
$$X_1 = B,$$

$$X_2 = E,$$

$$X_3 = A,$$

$$X_4 = J,$$

$$X_5 = M.$$



Due to the conditional independence we have that

$$P(x_1) = P(x_1),$$

$$P(x_2|x_1) = P(x_2) = P(x_2|\text{parents}(X_2)),$$

$$P(x_3|x_2, x_1) = P(x_3|\text{parents}(X_3)),$$

$$P(x_4|x_3, x_2, x_1) = P(x_4|x_3) = P(x_4|\text{parents}(X_4)),$$

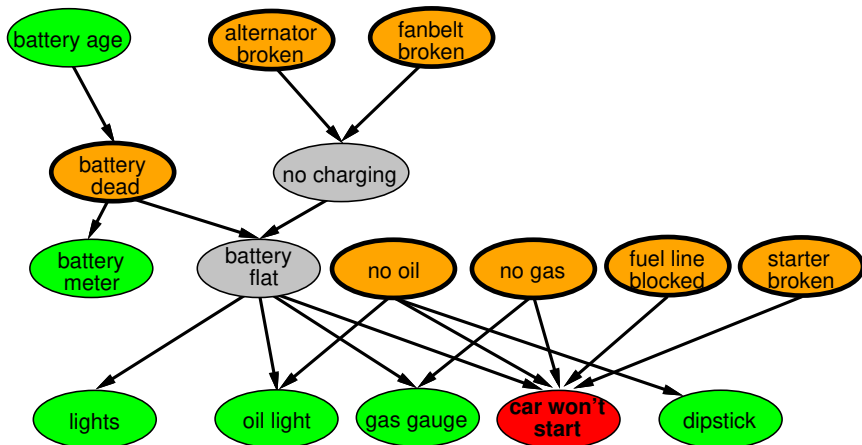
$$P(x_5|x_4, x_3, x_2, x_1) = P(x_5|x_3) = P(x_5|\text{parents}(X_5)).$$

Example: Car Diagnosis

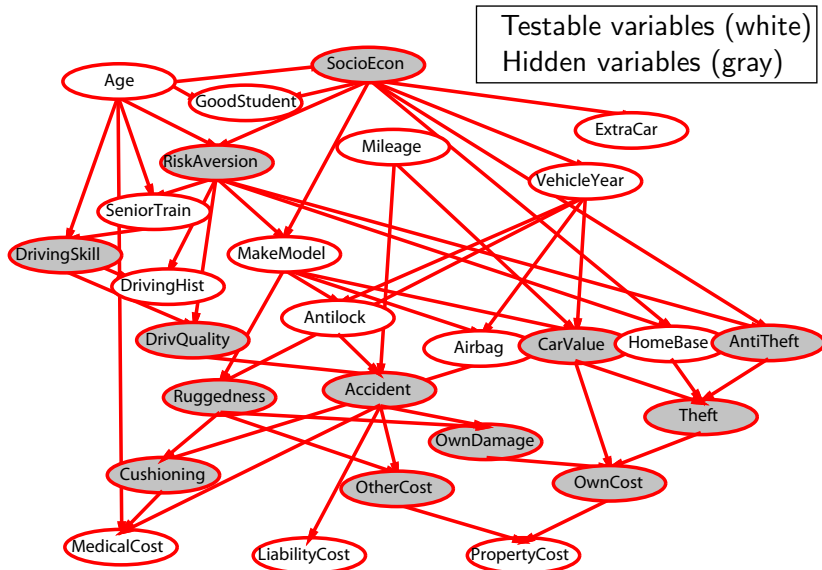
Initial evidence: car won't start

Testable variables (green), "broken, so fix it" variables (orange)

Hidden variables (gray) ensure sparse structure, reduce parameters



Example: Car Insurance



Typical Inference Tasks

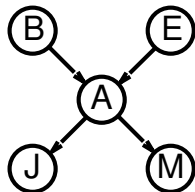
- **Simple queries:** compute probabilities given some evidence, e.g.,
 $P((NoGas = true) | (Gauge = empty), (Lights = on), (Starts = false))$
- ✗ **Conjunctive queries:** $P(X_j, X_i | E) = P(X_j | X_i, E)P(X_i | E)$
- **Optimal decisions:** decision networks include utility information; probabilistic inference required for $P(outcome | action, evidence)$
- **Value of information:** which evidence to seek next?
- **Sensitivity analysis:** which probability values are most critical?
- **Explanation:** why do I need a new starter motor?

Inference by Enumeration (bayesian_networks.ipynb)

Slightly intelligent way to sum out variables from the joint probability without actually constructing its explicit representation.

Simple query on the burglary network:

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

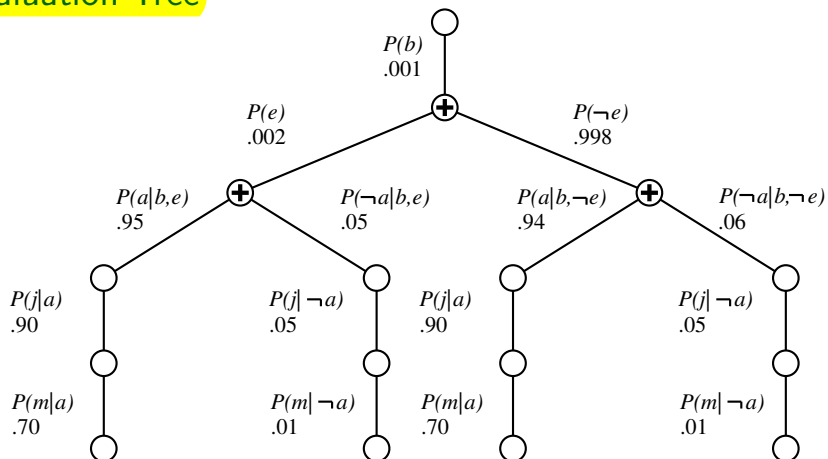


Rewrite full joint entries using $P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i | \text{parents}(X_i))$:

$$\begin{aligned}
 \mathbf{P}(B|j, m) &= \alpha \sum_e \sum_a \mathbf{P}(B, e, a, j, m) \\
 &= \alpha \sum_e \sum_a \mathbf{P}(B) P(e) \mathbf{P}(a|B, e) P(j|a) P(m|a) \\
 &= \alpha \mathbf{P}(B) \sum_{\textcolor{violet}{e}} P(\textcolor{violet}{e}) \sum_{\textcolor{teal}{a}} \mathbf{P}(\textcolor{teal}{a}|B, e) P(j|\textcolor{teal}{a}) P(m|\textcolor{teal}{a})
 \end{aligned}$$


The next slide visualizes the computation procedure.

Evaluation Tree



- Recursive depth-first enumeration: $\mathcal{O}(n)$ space, $\mathcal{O}(2^n)$ time (n : nr of Var.).
- Enumeration is inefficient: repeated computation, e.g., $P(j|a)P(m|a)$ is computed for each value of e .

Inference by Variable Elimination

( bayesian_networks.ipynb)

Variable elimination: carry out summations right-to-left, storing intermediate results (**factors** \mathbf{f}_i) to avoid re-computation. We evaluate

$$\mathbf{P}(B|j, m) = \alpha \underbrace{\mathbf{P}(B)}_{\mathbf{f}_1(B)} \sum_e \underbrace{P(e)}_{\mathbf{f}_2(E)} \sum_a \underbrace{\mathbf{P}(a|B, e)}_{\mathbf{f}_3(A, B, E)} \underbrace{P(j|a)}_{\mathbf{f}_4(A)} \underbrace{P(m|a)}_{\mathbf{f}_5(A)}.$$

intermediate variable

For example, the factors $\mathbf{f}_4(A)$ and $\mathbf{f}_5(A)$ corresponding to $P(j|a)$ and $P(m|a)$ depend just on A since J and M are fixed by the query. They are two-element vectors: given

$$\mathbf{f}_4(A) = \begin{bmatrix} P(j|a) \\ P(j|\neg a) \end{bmatrix} = \begin{bmatrix} 0.90 \\ 0.05 \end{bmatrix}, \quad \mathbf{f}_5(A) = \begin{bmatrix} P(m|a) \\ P(m|\neg a) \end{bmatrix} = \begin{bmatrix} 0.70 \\ 0.01 \end{bmatrix}.$$

$\mathbf{f}_3(A, B, E)$ is a $2 \times 2 \times 2$ matrix, which cannot be easily displayed on a slide.

Operations on Factors: Pointwise Product¹

Suppose, two factors have variables Y_1, \dots, Y_k in common:

$$\begin{aligned} \star \quad & \mathbf{f}(X_1, \dots, X_j, Y_1, \dots, Y_k, Z_1, \dots, Z_l) \\ &= \mathbf{f}_1(X_1, \dots, X_j, Y_1, \dots, Y_k) \times \mathbf{f}_2(Y_1, \dots, Y_k, Z_1, \dots, Z_l). \end{aligned}$$

Example:

A	B	$\mathbf{f}_1(A, B)$	B	C	$\mathbf{f}_2(B, C)$	A	B	C	$\mathbf{f}_3(A, B, C)$
T	T	0.3	T	T	0.2	T	T	T	$0.3 \cdot 0.2 = 0.06$
T	F	0.7	T	F	0.8	T	T	F	$0.3 \cdot 0.8 = 0.24$
F	T	0.9	F	T	0.6	T	F	T	$0.7 \cdot 0.6 = 0.42$
F	F	0.1	F	F	0.4	T	F	F	$0.7 \cdot 0.4 = 0.28$
						F	T	T	$0.9 \cdot 0.2 = 0.18$
						F	T	F	$0.9 \cdot 0.8 = 0.72$
						F	F	T	$0.1 \cdot 0.6 = 0.06$
						F	F	F	$0.1 \cdot 0.4 = 0.04$

Operations on Factors: Summing Out Variables²

Summing out variables is the process of removing variables by summation of probabilities of all possible values. Any factor that does not depend on the variable to be summed out should be moved outside the summation, e.g.,

$$\begin{aligned} \times \quad f_6(A, B) &= \sum_e f_2(E) \times f_3(A, B, E) \times f_4(A) \times f_5(A) \\ &= f_4(A) \times f_5(A) \times \sum_e f_2(E) \times f_3(A, B, E). \end{aligned}$$

The additions are performed as for matrices, e.g.,

$$\begin{aligned} \sum_a f_3(A, B, C) &= f_3(a, B, C) + f_3(\neg a, B, C) \\ &= \begin{bmatrix} 0.06 & 0.24 \\ 0.42 & 0.28 \end{bmatrix} + \begin{bmatrix} 0.18 & 0.72 \\ 0.06 & 0.04 \end{bmatrix} = \begin{bmatrix} 0.24 & 0.96 \\ 0.48 & 0.32 \end{bmatrix}. \end{aligned}$$

Inference by Variable Elimination: Example

Using the \times operator for **pointwise products** we have

$$\mathbf{P}(B|j, m) = \alpha \mathbf{f}_1(B) \times \sum_e \mathbf{f}_2(E) \times \sum_a \mathbf{f}_3(A, B, E) \times \mathbf{f}_4(A) \times \mathbf{f}_5(A).$$

We **sum out variables** from right to left:

- First, we sum out A from \mathbf{f}_3 , \mathbf{f}_4 , \mathbf{f}_5 , which gives us the 2×2 factor $\mathbf{f}_6(B, E)$:

$$\begin{aligned} \mathbf{f}_6(B, E) &= \sum_a \mathbf{f}_3(A, B, E) \times \mathbf{f}_4(A) \times \mathbf{f}_5(A) \\ &= (\mathbf{f}_3(a, B, E) \times \mathbf{f}_4(a) \times \mathbf{f}_5(a)) + (\mathbf{f}_3(\neg a, B, E) \times \mathbf{f}_4(\neg a) \times \mathbf{f}_5(\neg a)). \end{aligned}$$

Now we have $\mathbf{P}(B|j, m) = \alpha \mathbf{f}_1(B) \times \sum_e \mathbf{f}_2(E) \times \mathbf{f}_6(B, E)$.

- Next, we sum out E from the product of \mathbf{f}_2 and \mathbf{f}_6 :

$$\begin{aligned} \mathbf{f}_7(B) &= \sum_e \mathbf{f}_2(E) \times \mathbf{f}_6(B, E) \\ &= (\mathbf{f}_2(e) \times \mathbf{f}_6(B, e)) + (\mathbf{f}_2(\neg e) \times \mathbf{f}_6(B, \neg e)). \end{aligned}$$

This leaves $\mathbf{P}(B|j, m) = \alpha \mathbf{f}_1(B) \times \mathbf{f}_7(B)$.

Comparing the Number of Operations

Without factors

The figure on slide 43 shows 3 additions and 15 multiplications. Since 2 trees have to be computed for normalization, we have to double those numbers.

With factors

- The factor $\mathbf{f}_6(B, E) = \sum_a \mathbf{f}_3(A, B, E) \times \mathbf{f}_4(A) \times \mathbf{f}_5(A)$ on slide 47 requires 4 additions and 16 multiplications.
- The factor $\mathbf{f}_7(B) = \sum_e \mathbf{f}_2(E) \times \mathbf{f}_6(B, E)$ on slide 47 requires 2 additions and 4 multiplications.
- The final calculation $\mathbf{f}_1(B) \times \mathbf{f}_7(B)$ requires 2 multiplications.

	without factors	with factors
additions	6	6
multiplications	30	22

The improvements become more significant the larger the Bayesian network is.

Variable Ordering

Let us change the ordering of the computation from

$$\mathbf{P}(B|j, m) = \alpha \mathbf{f}_1(B) \times \sum_e \mathbf{f}_2(E) \times \sum_a \mathbf{f}_3(A, B, E) \times \mathbf{f}_4(A) \times \mathbf{f}_5(A)$$

$$\text{to } \mathbf{P}(B|j, m) = \alpha \mathbf{f}_1(B) \times \sum_a \mathbf{f}_4(A) \times \mathbf{f}_5(A) \times \sum_e \mathbf{f}_2(E) \times \mathbf{f}_3(A, B, E).$$

- $\mathbf{f}_6(A, B) = \sum_e \mathbf{f}_2(E) \times \mathbf{f}_3(A, B, E)$ requires 4 additions and 8 multiplications.
- $\mathbf{f}_7(B) = \sum_a \mathbf{f}_4(A) \times \mathbf{f}_5(A) \times \mathbf{f}_6(A, B)$ req. 2 additions and 8 multiplications.
- Finally, we have 2 multiplications for $\mathbf{f}_1(B) \times \mathbf{f}_7(B)$.

	old order	new order
additions	6	6
multiplications	22	18

Heuristics: Eliminate whichever variable minimizes the size of the next factor to be constructed (heuristic would not be able to choose in our example between $\mathbf{f}_6(B, E)$ and $\mathbf{f}_6(A, B)$).

Variable Relevance

Let us consider the query $\mathbf{P}(\text{JohnCalls} | \text{Burglary} = \text{true})$. The corresponding nested summation is

$$\mathbf{P}(J|b) = \alpha P(b) \sum_e P(e) \sum_a P(a|b, e) \mathbf{P}(J|a) \sum_m P(m|a).$$

- We notice that $\sum_m P(m|a) = 1$ by definition.
- Hence, the variable M is irrelevant for this query.
- ✗ In general, we can remove any leaf node that is not a query variable or an evidence variable (i.e., observed variables with probability 1).
- After removing a leaf, some of the newly obtained leaf nodes may be irrelevant, too.

Complexity of Probabilistic Inference

* It can be shown that inference in Bayesian networks is as hard as computing the number of satisfying assignments of a propositional logic formula.

→ Probabilistic inference is NP hard.

* There are many similarities with constraint satisfaction problems (CSPs):

- When the corresponding undirected graph of the Bayesian network is a tree, the complexity is only linear in the number of nodes as for CSPs.
- The variable elimination algorithm can be generalized to solve CSPs as well as Bayesian networks.

Monte Carlo Simulation

Important

Exact inference is computationally too expensive for large Bayesian networks, requiring us to approximate probabilities using Monte Carlo simulation.

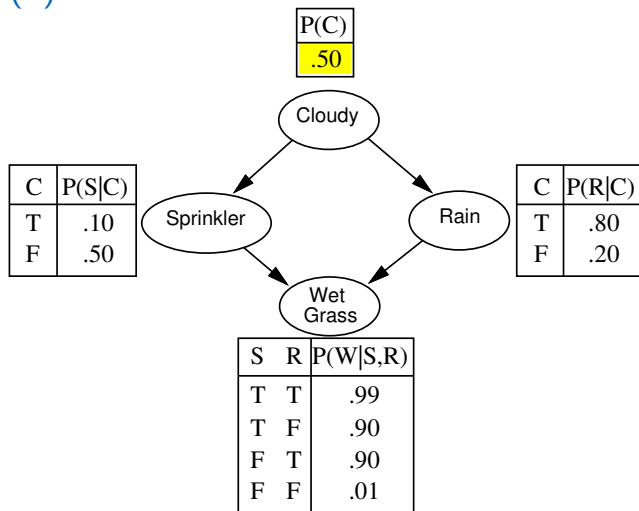
Basic procedure

- ① Create samples according to given probability distributions.
- ② Deterministic simulation.
- ③ Aggregation of individual simulations to obtain expected values of probability distributions.

There exist two main directions:

- Direct sampling methods (see next slides)
- Markov chain Monte Carlo methods (see AI book Sec. 5.2. “Inference by Markov chain simulation”)

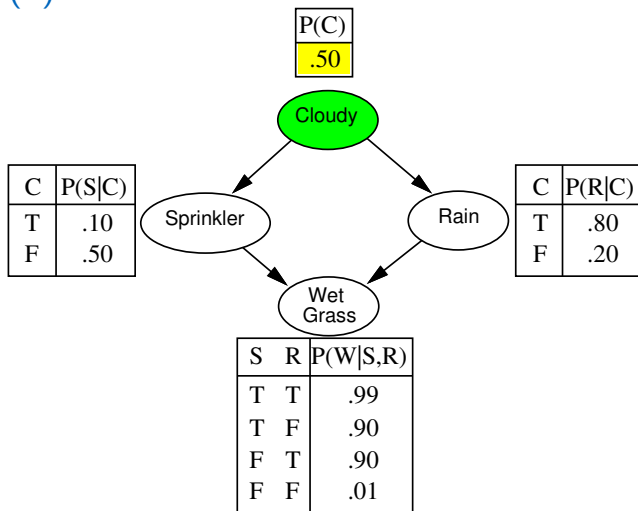
Example (1)



Sample from $\mathbf{P}(\text{Cloudy}) = \langle 0.5, 0.5 \rangle$.

< True, False >

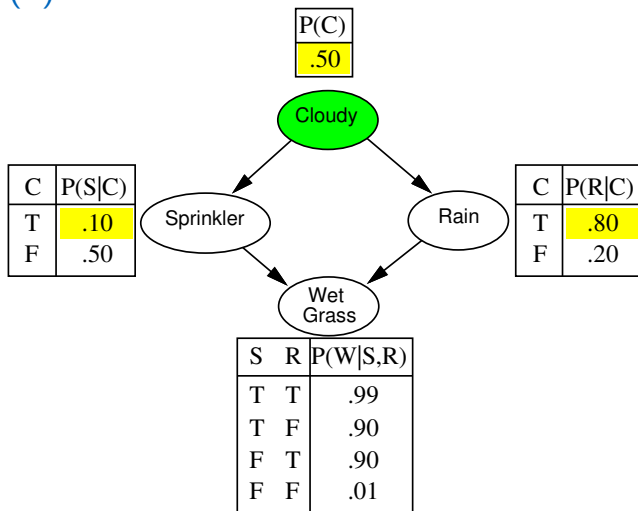
Example (2)



Sample from $\mathbf{P}(\text{Cloudy})$ is *true*.

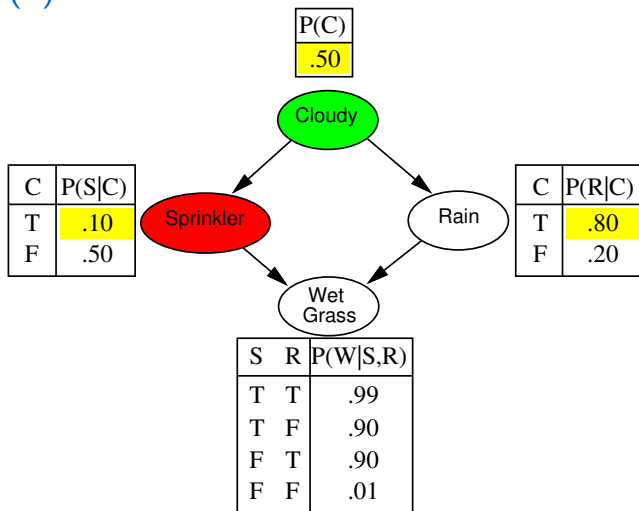
It might have been false as well

Example (3)



Sample from $\mathbf{P}(\text{Sprinkler} | \text{Cloudy} = \text{true}) = \langle 0.1, 0.9 \rangle$ and $\mathbf{P}(\text{Rain} | \text{Cloudy} = \text{true}) = \langle 0.8, 0.2 \rangle$.

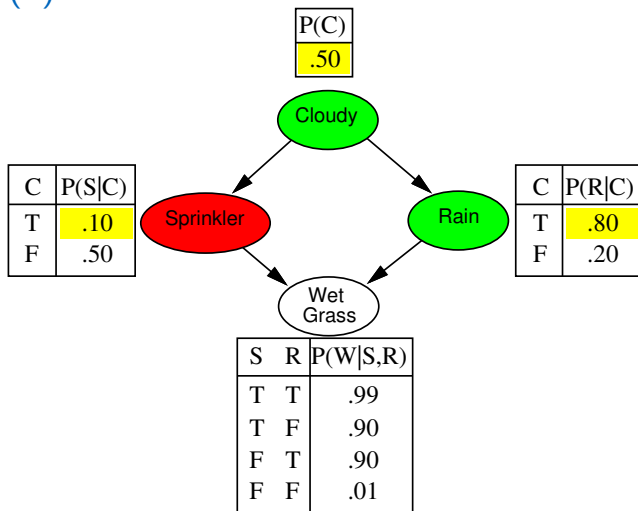
Example (4)



0.9

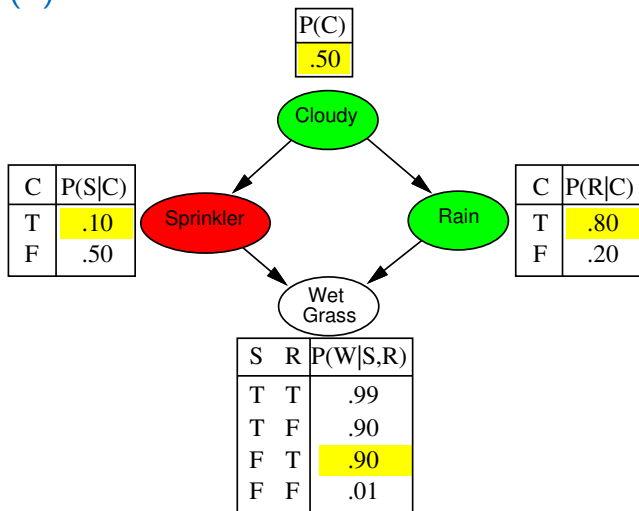
Sample from $\mathbf{P}(\text{Sprinkler} | \text{Cloudy} = \text{true})$ is *false*.

Example (5)



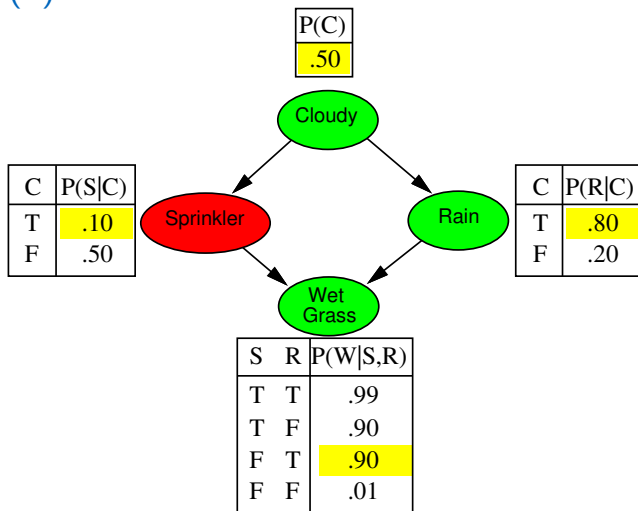
Sample from $\mathbf{P}(Rain|Cloudy = true)$ is *true*. 0.8

Example (6)




Sample from $\mathbf{P}(WetGrass|Sprinkler = false, Rain = true) = \langle 0.9, 0.1 \rangle$.

Example (7)



Sample from $\mathbf{P}(WetGrass|Sprinkler = false, Rain = true)$ is *true*. 0.9

Direct Sampling: Basic Principle

( bayesian_networks.ipynb)

Let $S_{PS}()$ be the probability that a specific event is generated by sampling. We can infer from the sampling procedure that

$$S_{PS}(x_1 \dots x_n) = \prod_{i=1}^n P(x_i | \text{parents}(X_i)) = P(x_1 \dots x_n),$$

e.g., $S_{PS}(c, \neg s, r, w) = 0.5 \cdot 0.9 \cdot 0.8 \cdot 0.9 = 0.324 = P(c, \neg s, r, w)$.

Let $N_{PS}(x_1 \dots x_n)$ be the number of samples generated for event x_1, \dots, x_n and $\hat{P}()$ return the estimated probability, then we have

$$\begin{aligned} \lim_{N \rightarrow \infty} \hat{P}(x_1, \dots, x_n) &= \lim_{N \rightarrow \infty} N_{PS}(x_1, \dots, x_n) / N \\ &= S_{PS}(x_1, \dots, x_n) \\ &= P(x_1 \dots x_n) \end{aligned}$$

We say that the estimates are **consistent** so that

$$\hat{P}(x_1, \dots, x_n) = N_{PS}(x_1, \dots, x_n) / N \approx P(x_1 \dots x_n). \quad (2)$$

Rejection Sampling(1) (bayesian_networks.ipynb)

- **Rejection sampling** is used to produce samples from a hard-to-sample distribution given an easy-to-sample distribution.
- We consider the simple form of determining conditional probabilities $\mathbf{P}(X|\mathbf{e})$. *X: query, e: evidence*
- We use `PriorSample` as before and reject all samples that do not match the evidence e.

Example

Estimate $\mathbf{P}(\text{Rain} | \text{Sprinkler} = \text{true})$ using 100 samples.

27 samples have *Sprinkler = true*.

Of these, 8 have *Rain = true* and 19 have *Rain = false*.

$$\hat{\mathbf{P}}(\text{Rain} | \text{Sprinkler} = \text{true}) = \text{Normalize}(\langle 8, 19 \rangle) = \langle 0.296, 0.704 \rangle$$

Rejection Sampling (2)

Let $\hat{\mathbf{P}}(X|\mathbf{e})$ be the estimated distribution that rejection sampling returns:

$$\begin{aligned}\hat{\mathbf{P}}(X|\mathbf{e}) &= \alpha \mathbf{N}_{PS}(X, \mathbf{e}) && \text{(algorithm defn.)} \\ &= \mathbf{N}_{PS}(X, \mathbf{e}) / N_{PS}(\mathbf{e}) && \text{(normalized by } N_{PS}(\mathbf{e})\text{)} \\ &\approx \mathbf{P}(X, \mathbf{e}) / P(\mathbf{e}) && \text{(property of PriorSample, eq. (2))} \\ &= \mathbf{P}(X|\mathbf{e}) && \text{(defn. of conditional probability)}\end{aligned}$$

- Thus rejection sampling returns consistent posterior estimates.
- **Problem:** hopelessly expensive if $P(\mathbf{e})$ is small.
- $P(\mathbf{e})$ drops off exponentially with number of evidence variables!

* Sometimes setting the problem as we want, is the soln

Likelihood Weighting (bayesian_networks.ipynb)

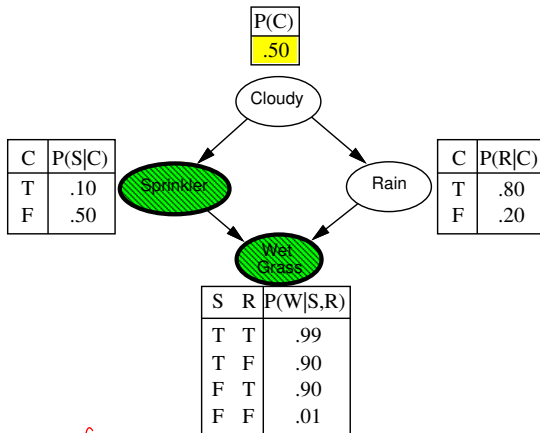
- **Likelihood weighting** avoids the inefficiency of rejection sampling by only generating events including the evidence \mathbf{e} .
- Likelihood weighting is a particular instance of **importance sampling**, which is widely used in Monte Carlo simulations.

Basic idea

- Fix the values of the evidence variables \mathbf{E} and sample only the non-evidence variables.
- Since not all events are equal, we have to weight each event with the likelihood of its occurrence.

Likelihood Weighting: Example (1)

* Query: $P(\text{Rain} | \text{Sprinkler} = \text{true}, \text{WetGrass} = \text{true})$.



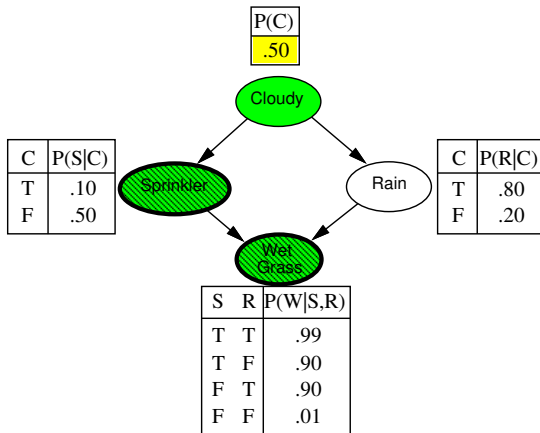
Correction factor

* w is updated for evidence var only

Initially the weight is set to $w = 1.0$.

Likelihood Weighting: Example (2)

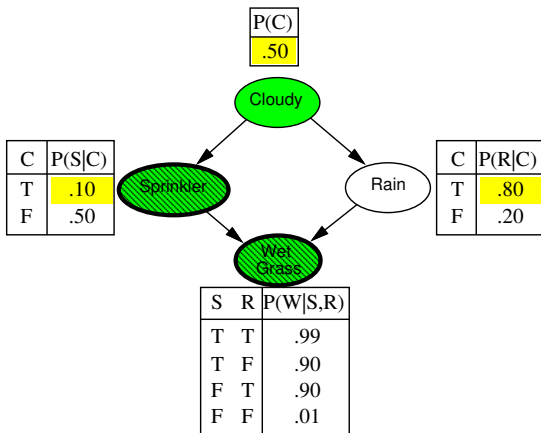
Query: $P(\text{Rain} | \text{Sprinkler} = \text{true}, \text{WetGrass} = \text{true})$.



Sample from *Cloudy*, suppose it returns *true*.

Likelihood Weighting: Example (3)

Query: $P(\text{Rain} | \text{Sprinkler} = \text{true}, \text{WetGrass} = \text{true})$.

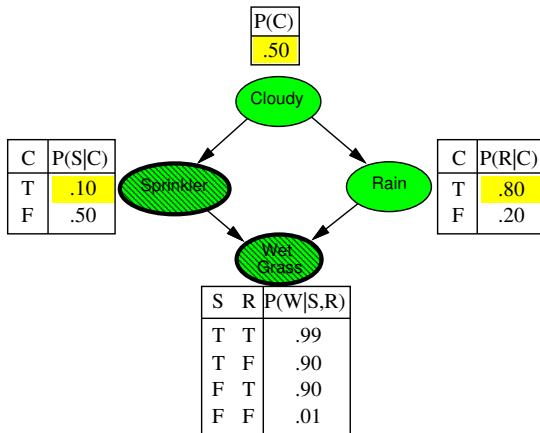


Updating w for
evidence var

Sprinkler is an evidence variable with value *true*. Therefore, we set
 $w \leftarrow w \cdot P(\text{Sprinkler} = \text{true} | \text{Cloudy} = \text{true}) = 0.1$.

Likelihood Weighting: Example (4)

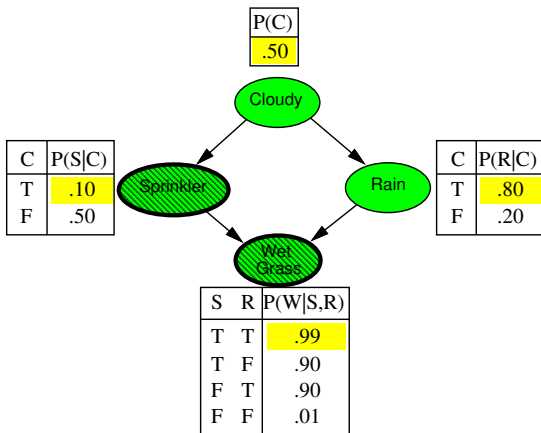
Query: $P(\text{Rain} | \text{Sprinkler} = \text{true}, \text{WetGrass} = \text{true})$.



Rain is not an evidence variable. Sample from *Rain*, suppose it returns *true*.

Likelihood Weighting: Example (5)

Query: $P(\text{Rain} | \text{Sprinkler} = \text{true}, \text{WetGrass} = \text{true})$.



WetGrass is an evidence variable with value *true*. Therefore, we set $w \leftarrow w \cdot P(\text{WetGrass} = \text{true} | \text{Sprinkler} = \text{true}, \text{Rain} = \text{true}) = 0.1 \cdot 0.99 = 0.099$.

Likelihood Weighting Analysis (1)

We denote the non-evidence variables (including the query variable X) by \mathbf{Z} . The sampling probability is

$$S_{WS}(\mathbf{z}, \mathbf{e}) = \prod_{i=1}^I P(z_i | \text{parents}(Z_i)). \quad (3)$$

Note: $\text{Parents}(Z_i)$ can include non-evidence and evidence variables.



The likelihood weight w makes up for the difference between the actual and the sampling distributions:

$$w(\mathbf{z}, \mathbf{e}) = \prod_{i=1}^m P(e_i | \text{parents}(E_i)). \quad (4)$$

After multiplying (3) and (4) we obtain

$$S_{WS}(\mathbf{z}, \mathbf{e}) w(\mathbf{z}, \mathbf{e}) = \prod_{i=1}^I P(z_i | \text{parents}(Z_i)) \prod_{i=1}^m P(e_i | \text{parents}(E_i)) = P(\mathbf{z}, \mathbf{e}) \quad (5)$$

since the two products cover all the variables in the network
 \rightarrow apply $P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i | \text{parents}(X_i))$ from slide 36.

Likelihood Weighting Analysis (2)

Now we can show that the likelihood weighting estimates are consistent. The weighted sampling probability is $\underbrace{(\mathbf{y})}_{\text{values of hidden variables}}$

$$\begin{aligned}
 \hat{P}(x|\mathbf{e}) &= \alpha \sum_{\mathbf{y}} N_{WS}(x, \mathbf{y}, \mathbf{e}) w(x, \mathbf{y}, \mathbf{e}) \quad \text{from likelihood weighting} \\
 &\approx \alpha' \sum_{\mathbf{y}} S_{WS}(x, \mathbf{y}, \mathbf{e}) w(x, \mathbf{y}, \mathbf{e}) \quad \text{from large } N \\
 &= \alpha' \sum_{\mathbf{y}} P(x, \mathbf{y}, \mathbf{e}) \quad \text{see eq. (5)} \\
 &= \alpha' P(x, \mathbf{e}) = P(x|\mathbf{e}). \quad \text{True prob}
 \end{aligned}$$

Performance still degrades with many evidence variables because a few samples have nearly all the total weight.

Summary

- Probabilities express the agent's inability to reach a definite decision regarding the truth of a sentence. Probabilities summarize the agent's belief relative to the evidence.
- * Bayesian networks provide a concise way to represent **conditional independence** from which we can infer the joint probability.
- * **Inference in Bayesian networks** means computing the probability distribution of a set of query variables, given a set of evidence variables.
- Exact inference can be computationally expensive for large networks. Approximate techniques based on Monte Carlo simulation provide a trade-off between accuracy and efficiency.