Fundamentals of Artificial Intelligence – Bayesian Networks

Matthias Althoff

TU München

Winter semester 2023/24

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

- 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).
- 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; same 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:

$$Toothache \Rightarrow Cavity$$

This is not always true! Better:

$$Toothache \Rightarrow Cavity \lor GumProblem \lor Abscess \lor \ldots$$

... but we do not even know all causes! Maybe use a causal rule?

$$Cavity \Rightarrow Toothache$$

Not true either since not all cavities cause pain.

Problem with logics:

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

Degrees of Belief

- We are only convinced of rules and facts up to a certain degree.
- One option to express the **degree of belief** is to use **probabilities**.
- Example: 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.

Overview of Probabilistic Methods

This lecture focuses on static environments without actions.

	Static environment	**Dynamic environment**
Without actions	Bayesian networks (lecture 9)	Hidden Markov models (lecture 10)
With actions	Decision networks (lecture 11)	Markov decision processes (lecture 12)

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. $P(e_1 \cup e_2 \cup \ldots) = \sum_i P(e_i)$ 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

For convenience, we introduce a function

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

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

Example: Tossing a coin

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

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

Expectation

The expectation of a random variable is defined as

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

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**

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

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

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

Example: Throwing two dice

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

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

Conditional Probability

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

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

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

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

Bayes' Rule

Rearranging the conditional probability results in

$$P\big((X=x),(Y=y)\big) = P\big((X=x)|(Y=y)\big)P(Y=y) \tag{1}$$

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

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

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

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

which can be inserted into Bayes rule:

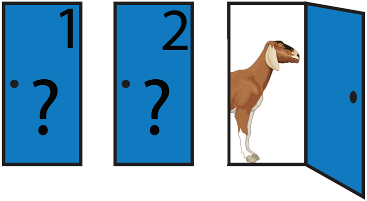
$$\boxed{P\big((Y=y)|(X=x)\big) = \frac{P\big((X=x)|(Y=y)\big)P(Y=y)}{\sum_{y\in\mathcal{D}_y} P\big((X=x)|(Y=y)\big)P(Y=y)}}$$

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*.

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?

Bayes' Rule: Monty Hall Problem (2)

$X = x_i \mathrel{\hat{=}}$ the prize is behind door i ($\mathcal{D}_x = \{1, 2, 3\}$).
$Y = y_j \mathrel{\hat{=}}$ the host has opened door j ($\mathcal{D}_y = \{1, 2, 3\}$).

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

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

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

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

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

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

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

$$\boxed{P\big((X=x),(Y=y)\big) = P(X=x)P(Y=y).}$$

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 'Karohemd'	$y_2 \hat{=}$ 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%

$$\frac{0.02}{0.08+0.02} = 0.2 = 20\%$$

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

- Are X and Y independent?

Conditional Independence

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

$$P\big((X=x)|(Y \cancel{=} y),(Z=z)\big) = P\big((X=x)|(Z=z)\big).$$

This is often written as

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

Notation Simplifications

- From now on, we abuse the notation for the sake of simplification.
- Instead of $P(Weather = sunny) = 0.6$, we write $P(sunny) = 0.6$. This requires that *sunny* is not used by another random variable.
- For Boolean random variables, we simplify

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

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

- Instead of listing all possible probabilities

$$\begin{aligned} P(Weather = sunny) &= 0.6 \\ P(Weather = rain) &= 0.1 \\ P(Weather = cloudy) &= 0.29 \\ P(Weather = snow) &= 0.01 \end{aligned}$$

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

Normalization

Normalization can be used for conditional probabilities of Boolean variables; denominator can be viewed as a **normalization constant** α:

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

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

$P(\alpha|\beta) + P(\neg\alpha|\beta)$
$= 1$

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

Bayesian Networks

- 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.

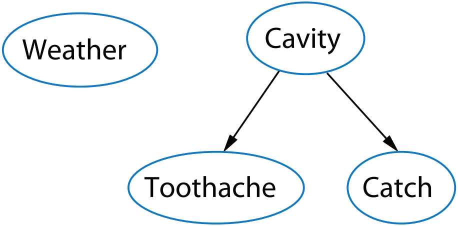
Bayesian network

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 $\mathrm{P}(X_i|Parents(X_i))$.

Example: Dentist

Topology of network encodes conditional independence assertions:



- *Weather* is independent of the other variables.
- *Toothache* and *Catch* are conditionally independent given *Cavity*.

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

Example: Burglar (1)

Considered scenario

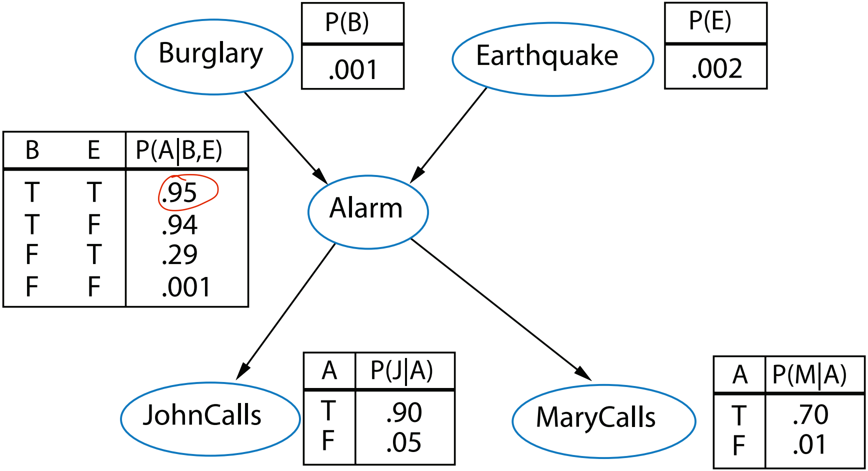
I'm at work, neighbor John calls to say my alarm is ringing, but neighbor Mary doesn't call. Sometimes it's set off by minor earthquakes. Is there a burglar?

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)



P(B)
.001

P(E)
.002

B	E	P(A|B,E)
T	T	.95
T	F	.94
F	T	.29
F	F	.001

A	P(J|A)
T	.90
F	.05

A	P(M|A)
T	.70
F	.01

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

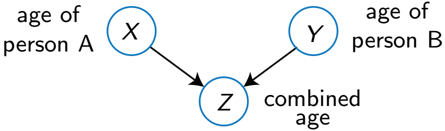
Compactness of Bayesian Networks

- 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 $X_i = \mathit{true}$ (the number for $X_i = \mathit{false}$ 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. $2^5 - 1 = 31$)

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

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

$\Leftrightarrow$ Variables X and Y share no common ancestry.

Conditional Independence

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

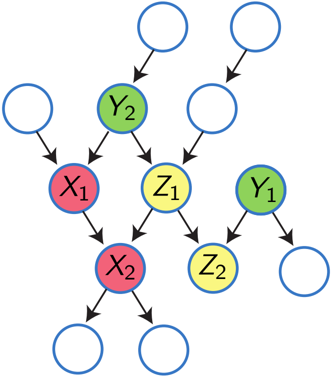
$\Leftrightarrow P(X|Y,Z) = P(X|Z)$ or $P(Y|X,Z) = P(Y|Z)$

$\Leftrightarrow$ every path from X to Y in an undirected Moral graph is blocked by Z (see next slide).

Slightly alternative presentation: David Barber (2020). Bayesian Reasoning and Machine Learning. Cambridge University Press (online available)

Determine Conditional Independence Graphically (1)

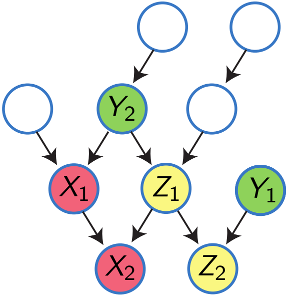
Is a set of nodes x conditionally independent of another set of nodes y given the set of evidences z (proof omitted)?



Determine Conditional Independence Graphically (2)

Is a set of nodes x conditionally independent of another set of nodes y given the set of evidences z (proof omitted)?

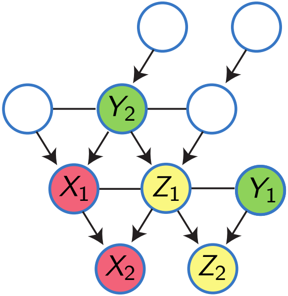
1. **Ancestral subgraph**: all nodes x, y, z, and their ancestors.



Determine Conditional Independence Graphically (3)

Is a set of nodes x conditionally independent of another set of nodes y given the set of evidences z (proof omitted)?

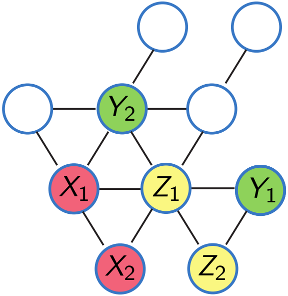
1. **Ancestral subgraph**: all nodes x, y, z, and their ancestors.
2. **Moral graph**: add links between any unlinked pair of nodes sharing a common child.



Determine Conditional Independence Graphically (4)

Is a set of nodes x conditionally independent of another set of nodes y given the set of evidences z (proof omitted)?

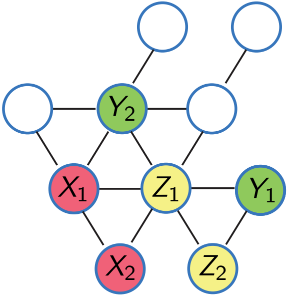
1. **Ancestral subgraph**: all nodes x, y, z, and their ancestors.
2. **Moral graph**: add links between any unlinked pair of nodes sharing a common child.
3. Replace all directed links by undirected links.



Determine Conditional Independence Graphically (5)

Is a set of nodes ⓧ conditionally independent of another set of nodes ⓨ given the set of evidences ⓩ (proof omitted)?

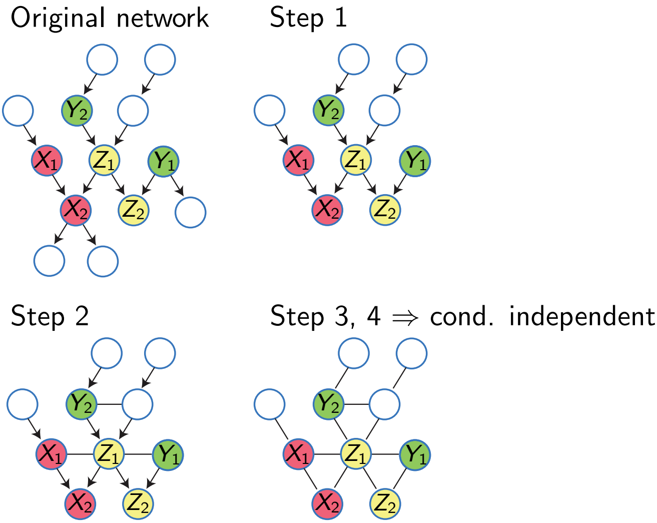
1. **Ancestral subgraph**: all nodes ⓧ, ⓨ, ⓩ, and their ancestors.
2. **Moral graph**: add links between any unlinked pair of nodes sharing a common child.
3. Replace all directed links by undirected links.
4. All paths between any ⓧ and ⓨ are blocked by ⓩ ⇒ conditional independence.



— unblocked path

⇒ not conditionally independent.

Conditional Independence: Examples (1)



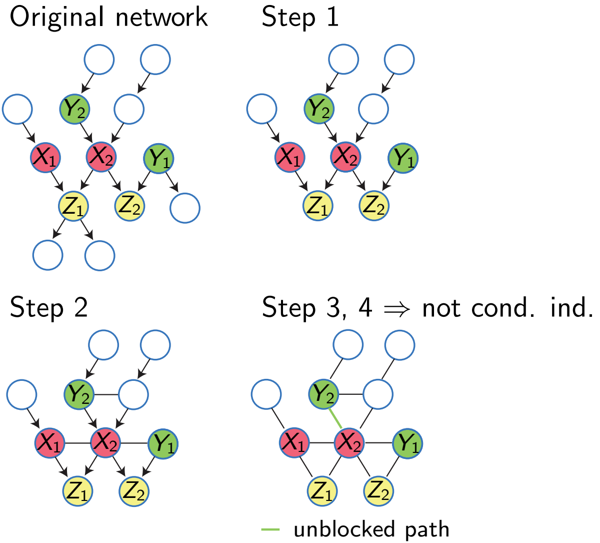
Original network

Step 1

Step 2

Step 3, 4 ⇒ cond. independent

Conditional Independence: Examples (2)



Original network

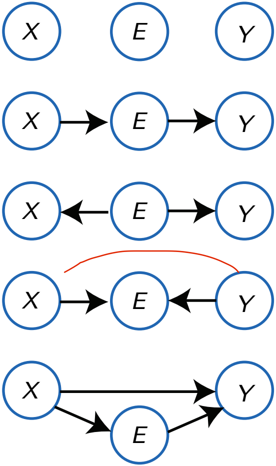
Step 1

Step 2

Step 3, 4 ⇒ not cond. ind.

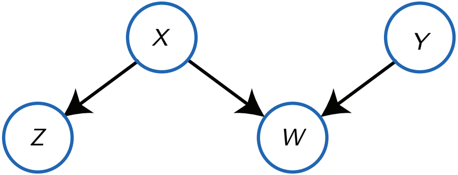
— unblocked path

Conditional Independence: Examples (3)



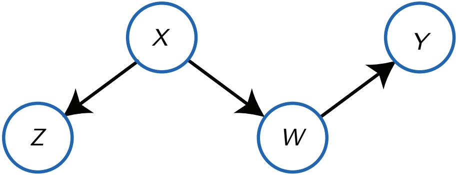
X ind. of Y?	yes
X cond. ind. of Y given E?	yes
X ind. of Y?	no
X cond. ind. of Y given E?	yes
X ind. of Y?	no
X cond. ind. of Y given E?	yes
X ind. of Y?	yes
X cond. ind. of Y given E?	no
X ind. of Y?	no
X cond. ind. of Y given E?	no

Conditional Independence: Examples (4)



$P(X,Y) = P(X)P(Y)$ yes

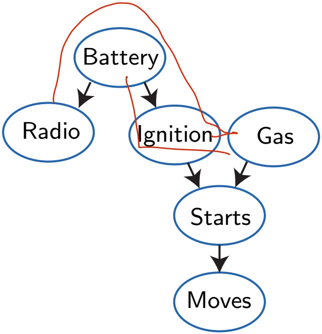
$P(X|Y,Z) = P(X|Z)$ yes



$P(X,Y) = P(X)P(Y)$ no

$P(X|Y,Z) = P(X|Z)$ no

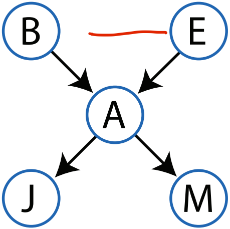
Conditional Independence: Examples (5)



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

Burglar scenario:



True or false?

- B and J are independent. no
- B is conditionally independent of J given A. yes
- B is conditionally independent of J given E. no
- B is conditionally independent of E given A. no
- M is conditionally independent of J given A. yes

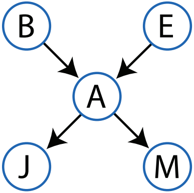
Semantics

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

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

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 *parents*() returns the values of the parents, while *Parents*() returns the random variables of the parents.

Chain Rule

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

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

yields the chain rule

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

Comparison with the previous slide results in the requirement

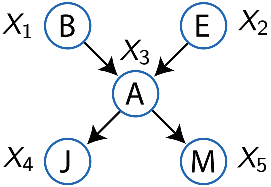
$$P(x_i | x_{i-1}, \ldots, x_1) = P(x_i | parents(X_i))$$

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

Example

We use the burglar example and choose

$$\begin{aligned} X_1 &= B, \\ X_2 &= E, \\ X_3 &= A, \\ X_4 &= J, \\ X_5 &= M. \end{aligned}$$



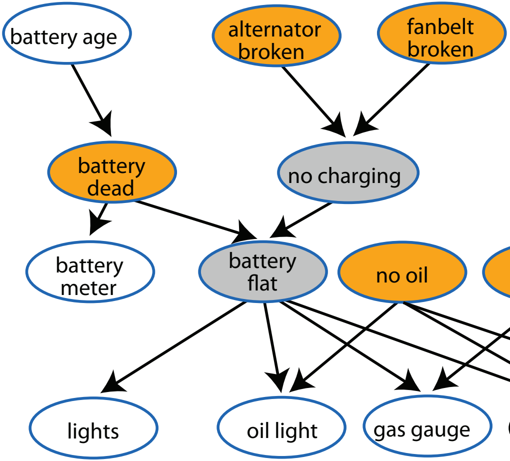
Due to the conditional independence we have that

$$\begin{aligned} P(x_1) &= P(x_1), \\ P(x_2|x_1) &= P(x_2) = P(x_2|parents(X_2)), \\ P(x_3|x_2, x_1) &= P(x_3|parents(X_3)), \\ P(x_4|x_3, x_2, x_1) &= P(x_4|x_3) = P(x_4|parents(X_4)), \\ P(x_5|x_4, x_3, x_2, x_1) &= P(x_5|x_3) = P(x_5|parents(X_5)). \end{aligned}$$

Example: Car Diagnosis

Initial evidence: car won't start

Hidden variables (gray) ensure sparse structure, reduce parameters.



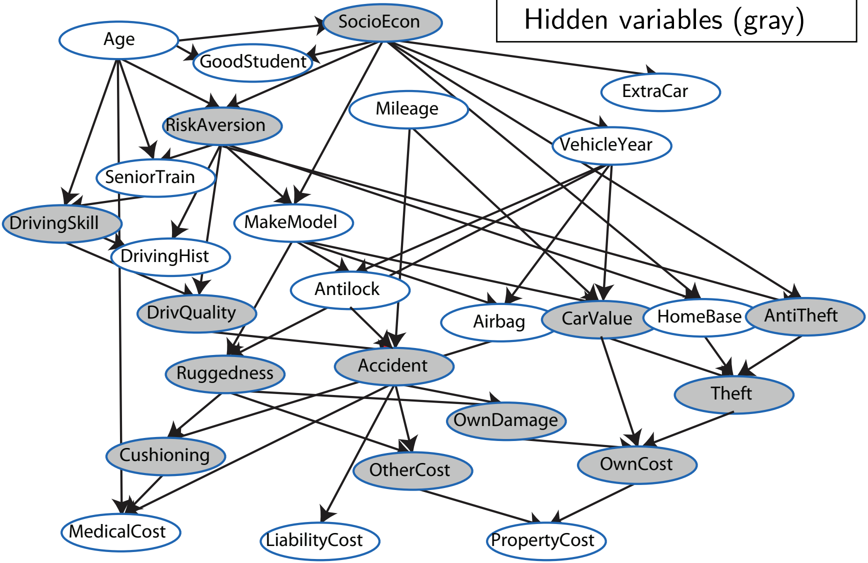
Testable variables (white)

"fix it" variables (orange)

Hidden variables (gray)

Example: Car Insurance

Testable variables (white)
Hidden variables (gray)



Typical Inference Tasks

- **Simple queries**: compute probabilities given some evidence, e.g., $P((NoGas = true)|(Gauge = empty), (Lights = on), (Starts = false))$
- **Conjunctive queries**: $\boxed{P(X_j, X_i|E) = P(X_j|X_i, E)P(X_i|E)}$ ✓
- **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?

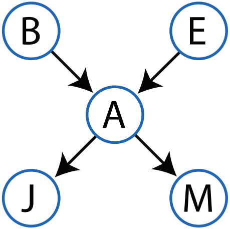
Inference by Enumeration (bayesian_networks.ipynb)

Sum out variables from the joint probability without actually constructing its explicit representation using

$$P(x_1, \dots, x_n) = \prod_{i=1}^{n} P(x_i | parents(X_i)). \qquad (2)$$

Burglary example:

$$\begin{aligned}
&\mathrm{P}(B|j,m)\\
&= \mathrm{P}(B,j,m)/P(j,m)\\
&= \alpha \mathrm{P}(B,j,m)\\
&= \alpha \sum_e \sum_a \mathrm{P}(B,e,a,j,m)\\
&\overset{(2)}{=} \alpha \sum_e \sum_a \mathrm{P}(B)P(e)\mathrm{P}(a|B,e)P(j|a)P(m|a)\\
&= \alpha \mathrm{P}(B) \sum_e P(e) \sum_a \mathrm{P}(a|B,e)P(j|a)P(m|a)
\end{aligned}$$

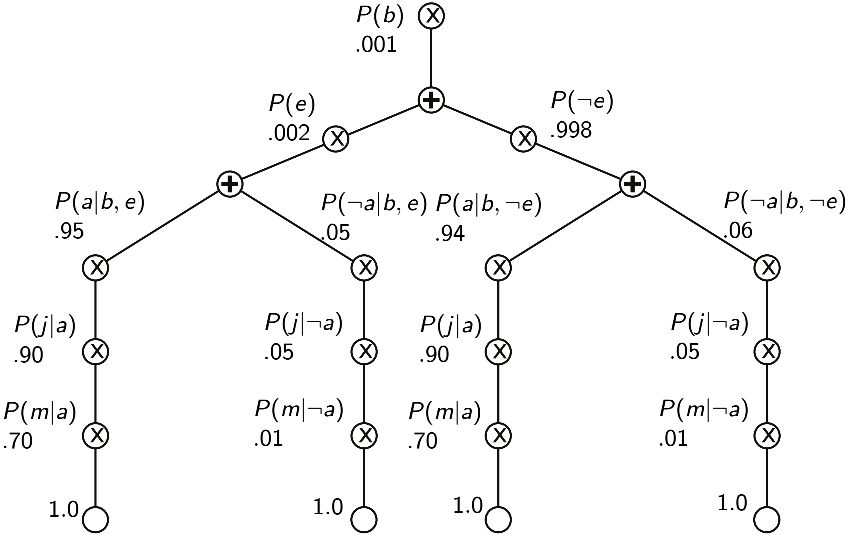


Recursive Enumeration Algorithm

```
function EnumerateAll (vars, e) returns a real number
inputs: vars, all variables in the BN
        e, observed values for variables E
if Empty(vars) then return 1.0
V ← First(vars)
if V is an evidence variable with value v in e then
    return $P(v \mid parents(V)) \cdot \texttt{EnumerateAll}(\texttt{Rest}(vars), \mathbf{e})$
else
    return $\sum_v P(v \mid parents(V)) \cdot \texttt{EnumerateAll}(\texttt{Rest}(vars), \mathbf{e}_v)$
        where $\mathbf{e}_v$ is $\mathbf{e}$ extended with $V = v$
```

Recursive depth-first enumeration: $\mathcal{O}(n)$ space, $\mathcal{O}(2^n)$ time (n: nr of variables).

Evaluation Tree



Repeated computation: e.g., $P(j|a)P(m|a)$.

Inference by Variable Elimination (jupyter bayesian_networks.ipynb)

Variable elimination: carry out summations right-to-left, storing intermediate results (**factors** f_i) to avoid re-computation. We evaluate

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

For example, the factors $\mathrm{f}_4(A)$ and $\mathrm{f}_5(A)$ 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:

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

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

Operations on Factors: Pointwise Product

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

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

Example:

A	B	$\mathrm{f}_1(A,B)$	B	C	$\mathrm{f}_2(B,C)$	A	B	C	$\mathrm{f}_3(A,B,C)$
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 Variables

Any factor that does not depend on the variable to be summed out should be moved outside the summation, e.g.,

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

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

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

Inference by Variable Elimination: Example

Using the × operator for **pointwise products**, we have

$$P(B|j,m) = \alpha f_1(B) \times \sum_e f_2(E) \times \underbrace{\sum_a f_3(A,B,E) \times f_4(A) \times f_5(A)}_{=f_6(B,E)} .$$

f_7

We sum out variables from right to left:

- First, we sum out A from f_3, f_4, f_5, which gives us the 2 × 2 factor

$$f_6(B,E) = \sum_a f_3(A,B,E) \times f_4(A) \times f_5(A)$$
$$= \big(f_3(a,B,E) \times f_4(a) \times f_5(a)\big) + \big(f_3(\neg a,B,E) \times f_4(\neg a) \times f_5(\neg a)\big).$$

- Next, we sum out E from the product of f_2 and f_6:

$$f_7(B) = \sum_e f_2(E) \times f_6(B,E)$$
$$= \big(f_2(e) \times f_6(B,e)\big) + \big(f_2(\neg e) \times f_6(B,\neg e)\big).$$

This leaves $P(B|j,m) = \alpha f_1(B) \times f_7(B)$.

Variable Ordering

Let us change the ordering of the computation from

$$P(B|j,m) = \alpha f_1(B) \times \sum_e f_2(E) \times \sum_a f_3(A,B,E) \times f_4(A) \times f_5(A)$$

to $P(B|j,m) = \alpha f_1(B) \times \sum_e f_2(E) \times \sum_a f_4(A) \times f_5(A) \times f_3(A,B,E)$.

- $f_6(B,E) = \sum_a f_3(A,B,E) \times f_4(A) \times f_5(A)$: 4 additions and 10 multiplications.
- $f_6(B,E) = \sum_a f_4(A) \times f_5(A) \times f_3(A,B,E)$: 4 additions and 16 multiplications.

	first order	**second order**
additions	6	6
multiplications	16	22

Heuristics: Eliminate whichever variable minimizes the size of the next factor to be constructed.

Comparing the Number of Operations

Without factors (slide 49)

The figure on slide 49 shows 3 additions and 15 multiplications. Since 2 trees have to be computed for normalization, we have to double those numbers.

With factors (slide 53)

- $\mathrm{f}_6(B, E) = \sum_a \mathrm{f}_3(A, B, E) \times \mathrm{f}_4(A) \times \mathrm{f}_5(A)$: 4 additions and 10 multiplications.
- $\mathrm{f}_7(B) = \sum_e \mathrm{f}_2(E) \times \mathrm{f}_6(B, E)$: 2 additions and 4 multiplications.
- The final calculation $\mathrm{f}_1(B) \times \mathrm{f}_7(B)$ requires 2 multiplications.

	without factors	**with factors**
additions	6	6
multiplications	30	16

The improvements become more significant for larger Bayesian networks.

Variable Relevance

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

$$\mathrm{P}(J|b) = \alpha P(b) \sum_{e} P(e) \sum_{a} P(a|b,e) \mathrm{P}(J|a) \sum_{m} P(m|a).$$

- 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.

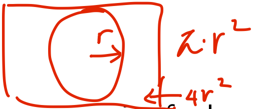
Complexity of Probabilistic Inference

Not relevant for the exam

- 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

π



ratio $\frac{\pi}{4}$

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.

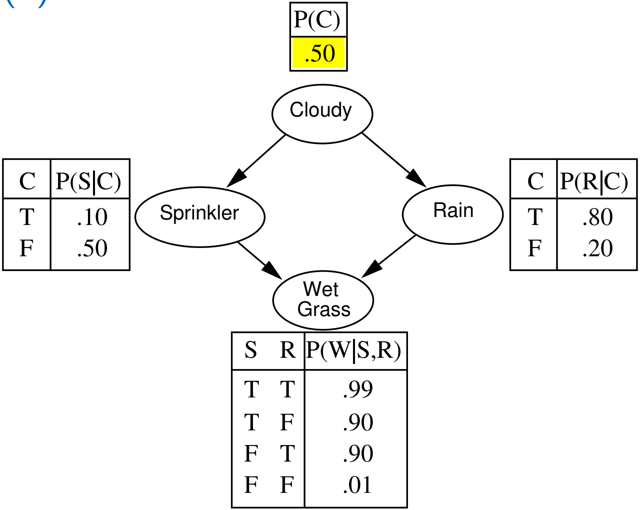
There exist two main directions:

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

Approximate Inference: Monte Carlo Simulation

direct sampling

Example (1)



P(C)
.50

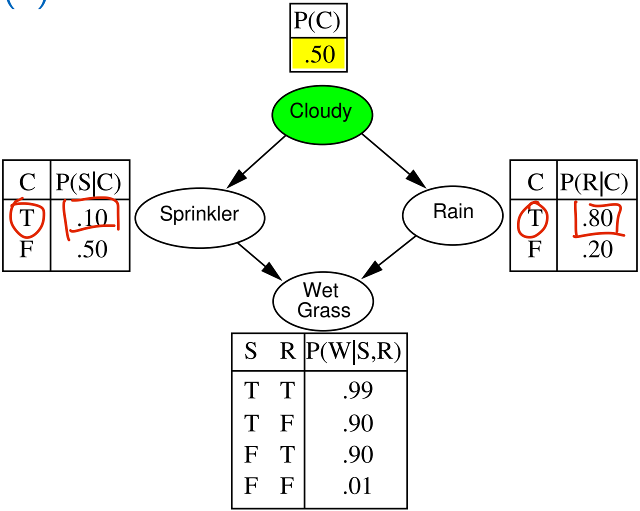
C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

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

Example (2)



P(C)
.50

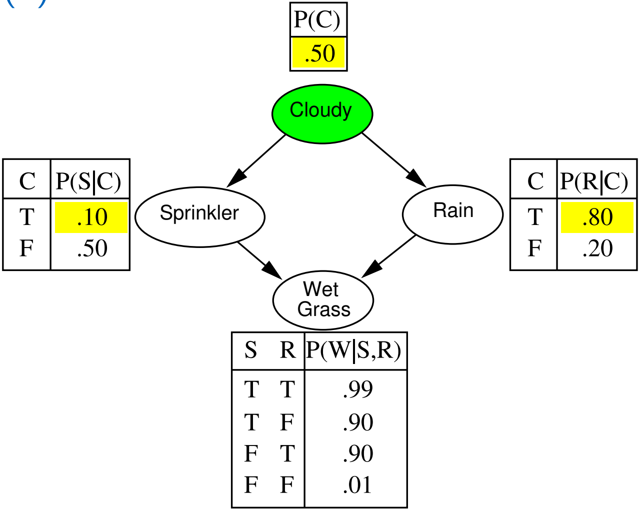
C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Sample from P(*Cloudy*) is *true*.

Example (3)



P(C)
.50

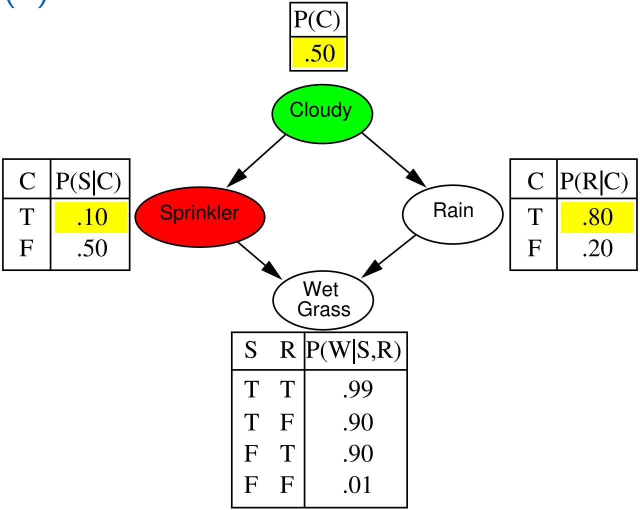
C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

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

Example (4)



P(C)
.50

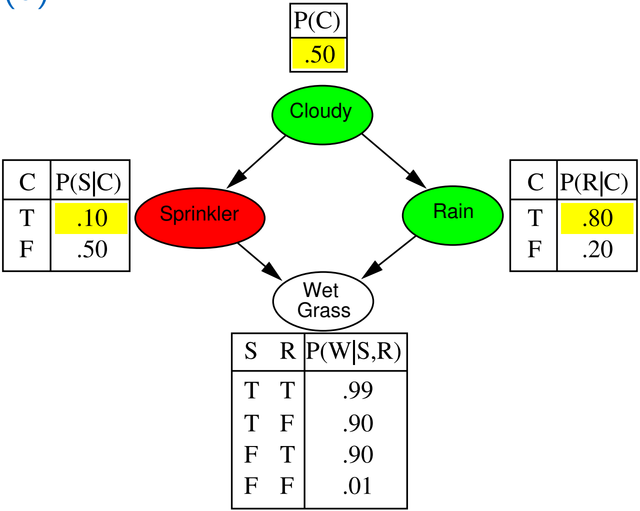
C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Sample from P($Sprinkler | Cloudy = true$) is *false*.

Example (5)



P(C)
.50

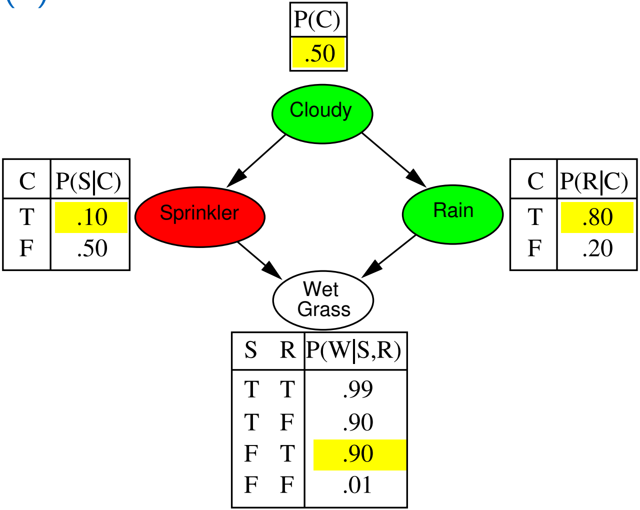
C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Sample from $P(Rain|Cloudy = true)$ is *true*.

Example (6)



P(C)
.50

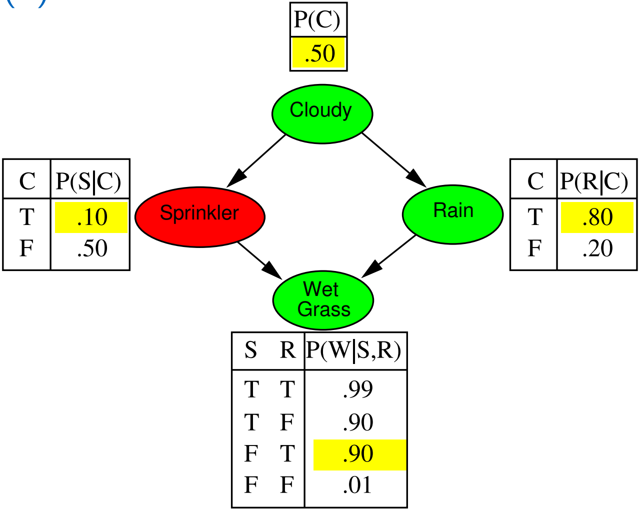
C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

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

Example (7)



P(C)
.50

C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Sample from $P(WetGrass|Sprinkler = false, Rain = true)$ is *true*.

Direct Sampling: 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

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

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

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

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

$$\hat{P}(x_1, \ldots, x_n) = N_{PS}(x_1, \ldots, x_n)/N \approx P(x_1 \ldots x_n). \quad (3)$$

Approximate Inference: Monte Carlo Simulation

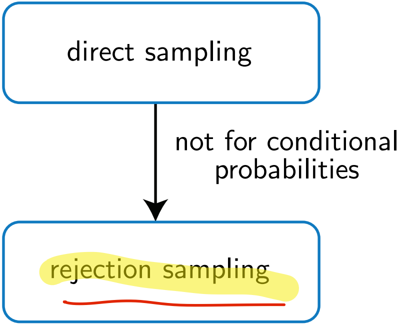
direct sampling

Approximate Inference: Monte Carlo Simulation

direct sampling

not for conditional probabilities

Approximate Inference: Monte Carlo Simulation



direct sampling

not for conditional probabilities

rejection sampling

Rejection Sampling(1) (jupyter 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 $\mathrm{P}(X|\mathrm{e})$.
- We sample as before and reject all samples that do not match the evidence e.

Example

Estimate $\mathrm{P}(Rain|Sprinkler = true)$ using 100 samples.
27 samples have $Sprinkler = true$.
Of these, 8 have $Rain = true$ and 19 have $Rain = false$.

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

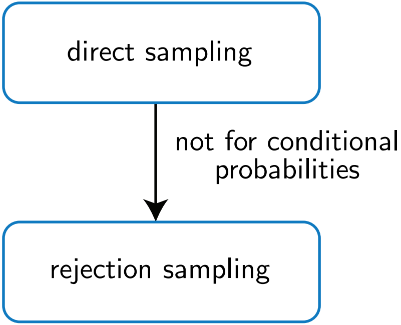
Rejection Sampling (2)

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

$$\begin{aligned}
\hat{\mathrm{P}}(X|\mathrm{e}) &= \alpha \mathrm{N}_{PS}(X,\mathrm{e}) && \text{(algorithm defn.)}\\
&= \mathrm{N}_{PS}(X,\mathrm{e})/N_{PS}(\mathrm{e}) && \text{(normalized by } N_{PS}(\mathrm{e}))\\
&\approx \mathrm{P}(X,\mathrm{e})/P(\mathrm{e}) && \text{(property of eq. (3))}\\
&= \underline{\mathrm{P}(X|\mathrm{e})} && \text{(defn. of conditional probability)}
\end{aligned}$$

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

Approximate Inference: Monte Carlo Simulation

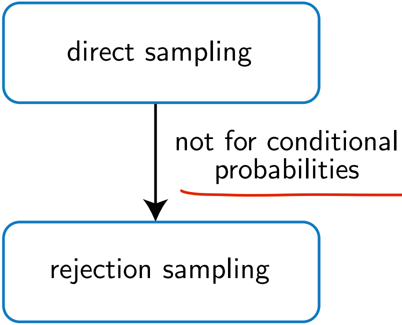


direct sampling

not for conditional probabilities

rejection sampling

Approximate Inference: Monte Carlo Simulation



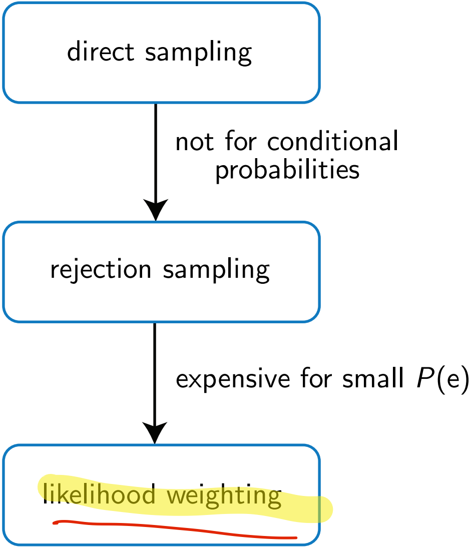
direct sampling

not for conditional probabilities

rejection sampling

expensive for small $P(\mathrm{e})$

Approximate Inference: Monte Carlo Simulation



direct sampling

not for conditional probabilities

rejection sampling

expensive for small $P(\mathrm{e})$

likelihood weighting

Likelihood Weighting (bayesian_networks.ipynb)

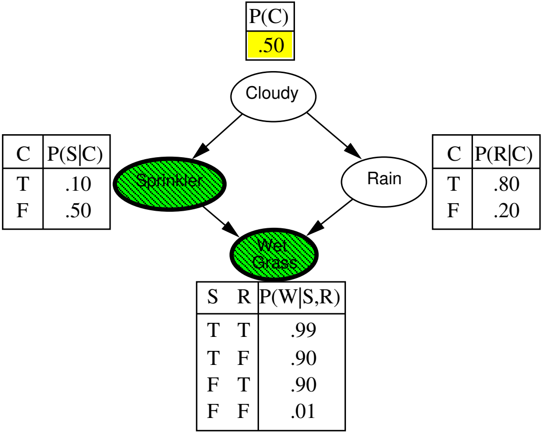
- **Likelihood weighting** avoids the inefficiency of rejection sampling by only generating events including the evidence e.
- Likelihood weighting is a particular instance of **importance sampling**, which is widely used in Monte Carlo simulations.

Basic idea

- Fix the values of the evidence variables 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: $\mathrm{P}(Rain|Sprinkler = true, WetGrass = true)$.



P(C)
.50

C	P(S|C)
T	.10
F	.50

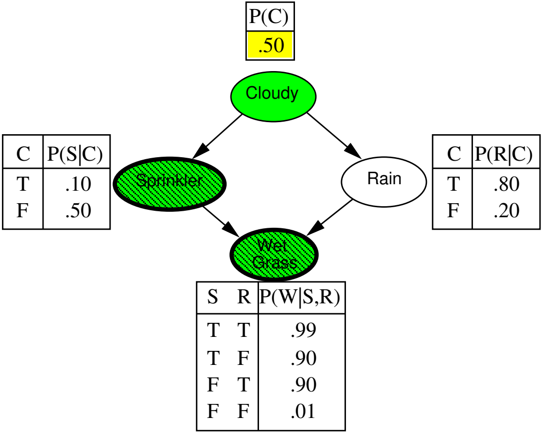
C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

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

Likelihood Weighting: Example (2)

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



P(C)
.50

C	P(S|C)
T	.10
F	.50

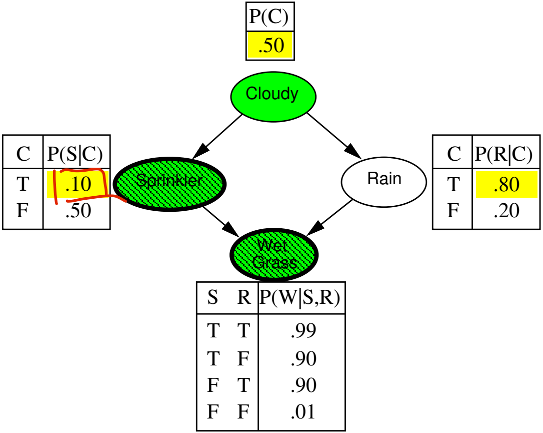
C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

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

Likelihood Weighting: Example (3)

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



P(C)
.50

C	P(S|C)
T	.10
F	.50

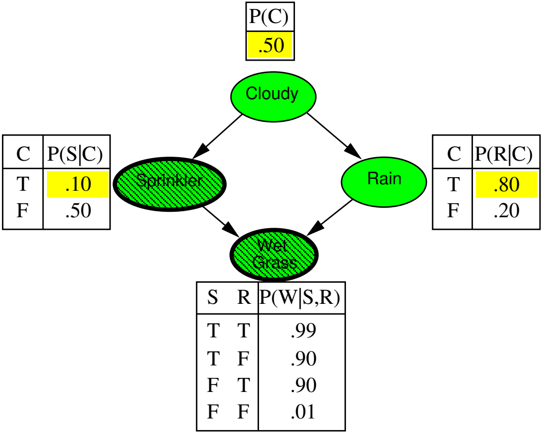
C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

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

Likelihood Weighting: Example (4)

Query: $P(Rain|Sprinkler = true, WetGrass = true)$.



P(C)
.50

C	P(S|C)
T	.10
F	.50

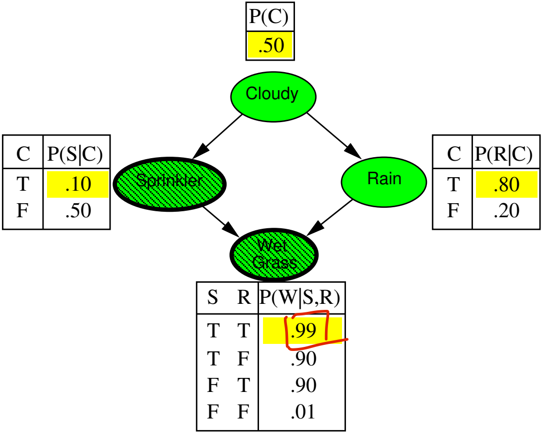
C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

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

Likelihood Weighting: Example (5)

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



P(C)
.50

C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

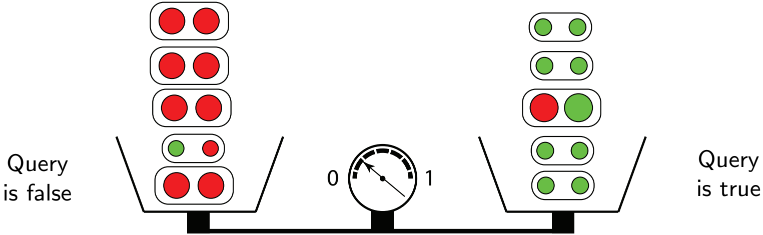
S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

WetGrass is an evidence variable with value *true*. Therefore, we set

$w \leftarrow w \cdot P(WetGrass = true | Sprinkler = true, Rain = true) = 0.1 \cdot 0.99 = 0.099.$

Likelihood Weighting: Example (6)

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

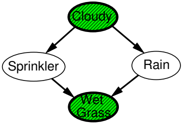


Events		**Weight = Probability of Evidence**
Cloudy	Rain	$P(Sprinkler = T/F) \cdot P(WetGrass = T/F)$
🟢	🟢	$0.1 \cdot 0.99 = 0.099$
🟢	🔴	$0.1 \cdot 0.9 = 0.09$
🔴	🟢	$0.5 \cdot 0.99 = 0.495$
🔴	🔴	$0.5 \cdot 0.9 = 0.45$

Likelihood Weighting Analysis

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

$$S_{WS}(\mathrm{z}, \mathrm{e}) = \underline{\prod_{i=1}^{l} P(z_i|parents(Z_i))}. \tag{4}$$



The likelihood weight w corrects the difference between the actual and the sampling distributions:

$$w(\mathrm{z}, \mathrm{e}) = \underline{\prod_{i=1}^{m} P(e_i|parents(E_i))}. \tag{5}$$

After multiplying (4) and (5) we obtain

$$\underline{S_{WS}(\mathrm{z}, \mathrm{e})w(\mathrm{z}, \mathrm{e})} = \prod_{i=1}^{l} P(z_i|parents(Z_i)) \prod_{i=1}^{m} P(e_i|parents(E_i)) = \underline{P(\mathrm{z}, \mathrm{e})} \tag{6}$$

since the two products cover all the variables in the network
→ apply $P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|parents(X_i))$ from slide 41.

Consistency of Likelihood Weighting

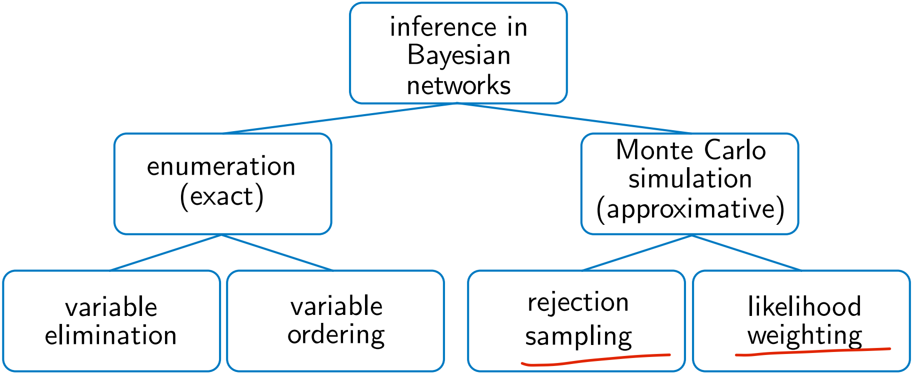
Not relevant for the exam

Now we can show that the likelihood weighting estimates are consistent. The weighted sampling probability is (y: values of hidden variables)

$$\begin{aligned}
\hat{P}(x|\mathrm{e}) &= \alpha \sum_{\mathrm{y}} N_{WS}(x, \mathrm{y}, \mathrm{e}) w(x, \mathrm{y}, \mathrm{e}) \quad \text{from likelihood weighting} \\
&\approx \alpha' \sum_{\mathrm{y}} S_{WS}(x, \mathrm{y}, \mathrm{e}) w(x, \mathrm{y}, \mathrm{e}) \quad \text{from large } N \\
&= \alpha' \sum_{\mathrm{y}} P(x, \mathrm{y}, \mathrm{e}) \quad \text{see eq. (6)} \\
&= \alpha' P(x, \mathrm{e}) = P(x|\mathrm{e}).
\end{aligned}$$

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

Overview of Probabilistic Inference Methods



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.