Fundamentals of Artificial Intelligence and Knowledge Representation (Module 3)

Last update: 08 December 2023

Academic Year 2023 – 2024
Alma Mater Studiorum · University of Bologna

Contents

1 Introduction

1.1 Uncertainty

Uncertainty A task is uncertain if it has:

Uncertainty

- Partial observations
- Noisy or wrong information
- Uncertain outcomes of the actions
- Complex models

A purely logic approach leads to:

- Risks falsehood: unreasonable conclusion when applied in practice.
- Weak decisions: too many conditions required to make a conclusion.

1.1.1 Handling uncertainty

Default/non-monotonic logic Works on assumptions. An assumption can be contradicted by the evidence.

Default/non-monotonic logic

Rule-based systems with fudge factors Formulated as premise $\rightarrow_{\text{prob.}}$ effect. Have the following issues:

Rule-based systems with fudge factors

- Locality: how can the probability account all the evidence.
- Combination: chaining of unrelated concepts.

Probability Assign a probability given the available known evidence.

Probability

Note: fuzzy logic handles the degree of truth and not the uncertainty.

Decision theory Defined as:

Decision theory

$$\text{Decision theory} = \text{Utility theory} + \text{Probability theory}$$

where the utility theory depends on one's preferences.

2 Probability

Sample space

Sample space Set Ω of all possible worlds.

Event

Event Subset $A \subseteq \Omega$.

Sample point

Sample point/Possible world/Atomic event Element $\omega \in \Omega$.

Probability space

Probability space A probability space/model is a function $\mathcal{P}(\cdot) : \Omega \to [0, 1]$ assigned to a sample space such that:

- $0 \leq \mathcal{P}(\omega) \leq 1$
- $\sum_{\omega \in \Omega} \mathcal{P}(\omega) = 1$
- $\mathcal{P}(A) = \sum_{\omega \in A} \mathcal{P}(\omega)$

Random variable

Random variable A function from an event to some range (e.g. reals, booleans, . . .).

Probability distribution

Probability distribution For any random variable X:

$$\mathcal{P}(X = x_i) = \sum_{\omega \text{ s.t. } X(\omega)=x_i} \mathcal{P}(\omega)$$

Proposition

Proposition Event where a random variable has a certain value.

$$a = \{\omega \mid A(\omega) = \texttt{true}\}$$

$$\neg a = \{\omega \mid A(\omega) = \texttt{false}\}$$

$$(\texttt{Weather} = \texttt{rain}) = \{\omega \mid B(\omega) = \texttt{rain}\}$$

Prior probability

Prior probability Prior/unconditional probability of a proposition based on known evidence.

Probability distribution (all)

Probability distribution (all) Gives all the probabilities of a random variable.

$$\mathbf{P}(A) = \langle \mathcal{P}(A = a_1), \dots, \mathcal{P}(A = a_n) \rangle$$

Joint probability distribution

Joint probability distribution The joint probability distribution of a set of random variables gives the probability of all the different combinations of their atomic events.

Note: Every question on a domain can, in theory, be answered using the joint distribution. In practice, it is hard to apply.

Example. $\mathbf{P}(\texttt{Weather}, \texttt{Cavity}) =$

	`Weather=sunny`	`Weather=rain`	`Weather=cloudy`	`Weather=snow`
`Cavity=true`	0.144	0.02	0.016	0.02
`Cavity=false`	0.576	0.08	0.064	0.08

Probability density function

Probability density function The probability density function (PDF) of a random variable X is a function $p : \mathbb{R} \to \mathbb{R}$ such that:

$$\int_{\mathcal{T}_X} p(x)\, dx = 1$$

Uniform distribution

Uniform distribution

$$p(x) = \text{Unif}[a,b](x) = \begin{cases} \frac{1}{b-a} & a \leq x \leq b \\ 0 & \text{otherwise} \end{cases}$$

Gaussian (normal) distribution

Gaussian (normal) distribution

$$\mathcal{N}(\mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

$\mathcal{N}(0,1)$ is the standard Gaussian.

Conditional probability

Conditional probability Probability of a prior knowledge with new evidence:

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

The product rule gives an alternative formulation:

$$\mathcal{P}(a \wedge b) = \mathcal{P}(a|b)\mathcal{P}(b) = \mathcal{P}(b|a)\mathcal{P}(a)$$

Chain rule

Chain rule Successive application of the product rule:

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

Independence

Independence Two random variables A and B are independent ($A \perp B$) iff:

$$\mathbf{P}(A|B) = \mathbf{P}(A) \;\text{ or }\; \mathbf{P}(B|A) = \mathbf{P}(B) \;\text{ or }\; \mathbf{P}(A,B) = \mathbf{P}(A)\mathbf{P}(B)$$

Conditional independence

Conditional independence Two random variables A and B are conditionally independent iff:

$$\mathbf{P}(A \,|\, C, B) = \mathbf{P}(A \,|\, C)$$

2.1 Inference with full joint distributions

Given a joint distribution, the probability of any proposition ϕ can be computed as the sum of the atomic events where ϕ is true:

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

Example. Given the following joint distribution:

	toothache		¬toothache	
	catch	¬catch	catch	¬catch
cavity	0.108	0.012	0.072	0.008
¬cavity	0.016	0.064	0.144	0.576

We have that:

- $\mathcal{P}\left(\texttt{toothache}\right) = 0.108 + 0.012 + 0.016 + 0.064 = 0.2$
- $\mathcal{P}\left(\texttt{cavity} \vee \texttt{toothache}\right) = 0.108 + 0.012 + 0.072 + 0.008 + 0.016 + 0.064 = 0.28$
- $\mathcal{P}\left(\neg\texttt{cavity} \,|\, \texttt{toothache}\right) = \frac{\mathcal{P}(\neg\texttt{cavity} \wedge \texttt{toothache})}{\mathcal{P}(\texttt{toothache})} = \frac{0.016+0.064}{0.2} = 0.4$

Marginalization

Marginalization The probability that a random variable assumes a specific value is given by the sum off all the joint probabilities where that random variable assumes the given value.

Example. Given the joint distribution:

	`Weather=sunny`	`Weather=rain`	`Weather=cloudy`	`Weather=snow`
`Cavity=true`	0.144	0.02	0.016	0.02
`Cavity=false`	0.576	0.08	0.064	0.08

We have that $\mathcal{P}\left(\texttt{Weather} = \texttt{sunny}\right) = 0.144 + 0.576$

Conditioning

Conditioning Adding a condition to a probability (reduction and renormalization).

Normalization

Normalization Given a conditional probability distribution $\mathbf{P}(A|B)$, it can be formulated as:

$$\mathbf{P}(A|B) = \alpha\mathbf{P}(A, B)$$

where α is a normalization constant. In fact, fixed the evidence B, the denominator to compute the conditional probability is the same for each probability.

Example. Given the joint distribution:

	`toothache`		`¬toothache`	
	`catch`	`¬catch`	`catch`	`¬catch`
`cavity`	0.108	0.012	0.072	0.008
`¬cavity`	0.016	0.064	0.144	0.576

We have that:

$$\mathbf{P}(\texttt{Cavity}|\texttt{toothache}) = \langle \frac{\mathcal{P}\left(\texttt{cavity}, \texttt{toothache}, \texttt{catch}\right)}{\mathcal{P}\left(\texttt{toothache}\right)}, \frac{\mathcal{P}\left(\neg\texttt{cavity}, \texttt{toothache}, \neg\texttt{catch}\right)}{\mathcal{P}\left(\texttt{toothache}\right)} \rangle$$

Probability query

Probability query Given a set of query variables $\boldsymbol{Y}$, the evidence variables $\mathbf{e}$ and the other hidden variables $\boldsymbol{H}$, the probability of the query can be computed as:

$$\mathbf{P}(\boldsymbol{Y}|\boldsymbol{E} = \mathbf{e}) = \alpha\mathbf{P}(\boldsymbol{Y}, \boldsymbol{E} = \mathbf{e}) = \alpha \sum_{\mathbf{h}} \mathbf{P}(\boldsymbol{Y}, \boldsymbol{E} = \mathbf{e}, \boldsymbol{H} = \mathbf{h})$$

The problem of this approach is that it has exponential time and space complexity that makes it not applicable in practice.

To reduce the size of the variables, conditional independence can be exploited.

Example. Knowing that $\mathbf{P} \models (\texttt{Catch} \perp \texttt{Toothache}|\texttt{Cavity})$, we can compute the distribution $\mathbf{P}(\texttt{Toothache}, \texttt{Catch}, \texttt{Cavity})$ as follows:

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

$\mathbf{P}(\texttt{Toothache}, \texttt{Catch}, \texttt{Cavity})$ has 7 independent values that grows exponentially ($2 \cdot 2 \cdot 2 = 8$ values, but one of them can be omitted as a probability always sums up to 1).

$\mathbf{P}(\texttt{Toothache} \mid \texttt{Cavity})\mathbf{P}(\texttt{Catch} \mid \texttt{Cavity})\mathbf{P}(\texttt{Cavity})$ has 5 independent values that grows linearly ($4 + 4 + 2 = 10$, but a value of $\mathbf{P}(\texttt{Cavity})$ can be omitted. The conditional probabilities require two tables (one for each prior) each with 2 values, but for each table a value can be omitted, therefore requiring 2 independent values per conditional probability instead of 4).

3 Bayesian networks

3.1 Bayes' rule

Bayes' rule

Bayes' rule

$$\mathcal{P}(a \mid b) = \frac{\mathcal{P}(b \mid a)\mathcal{P}(a)}{\mathcal{P}(b)}$$

Bayes' rule and conditional independence Given the random variables $\texttt{Cause}$ and $\texttt{Effect}_1, \dots, \texttt{Effect}_n$, with $\texttt{Effect}_i$ independent from each other, we can compute $\mathbf{P}(\texttt{Cause}, \texttt{Effect}_1, \dots, \texttt{Effect}_n)$ as follows:

$$\mathbf{P}(\texttt{Cause}, \texttt{Effect}_1, \dots, \texttt{Effect}_n) = \left(\prod_i \mathbf{P}(\texttt{Effect}_i \mid \texttt{Cause}) \right) \mathbf{P}(\texttt{Cause})$$

The number of parameters is linear.

Example. Knowing that $\mathbf{P} \models (\texttt{Catch} \perp \texttt{Toothache} | \texttt{Cavity})$:

$$\begin{aligned} &\mathbf{P}(\texttt{Cavity} \mid \texttt{toothache} \land \texttt{catch}) \\ &\quad = \alpha \mathbf{P}(\texttt{toothache} \land \texttt{catch} \mid \texttt{Cavity}) \mathbf{P}(\texttt{Cavity}) \\ &\quad = \alpha \mathbf{P}(\texttt{toothache} \mid \texttt{Cavity}) \mathbf{P}(\texttt{catch} \mid \texttt{Cavity}) \mathbf{P}(\texttt{Cavity}) \end{aligned}$$

3.2 Bayesian network reasoning

Bayesian network

Bayesian network Graph for conditional independence assertions and a compact specification of full joint distributions.

- Directed acyclic graph.
- Nodes represent variables.
- The conditional distribution of a node is given by its parents

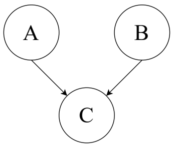
  $$\mathbf{P}(X_i \mid \texttt{parents}(X_i))$$

  In other words, if there is an edge from A to B, then A (cause) influences B (effect).

Conditional probability table (CPT)

Conditional probability table (CPT) In the case of boolean variables, the conditional distribution of a node can be represented using a table by considering all the combinations of the parents.

Example. Given the boolean variables A, B and C, with C depending on A and B, we have that:



A	B	$\mathcal{P}(c \vert A, B)$	$\mathcal{P}(\neg c \vert A, B)$
a	b	α	$1 - \alpha$
¬a	b	β	$1 - \beta$
a	¬b	γ	$1 - \gamma$
¬a	¬b	δ	$1 - \delta$

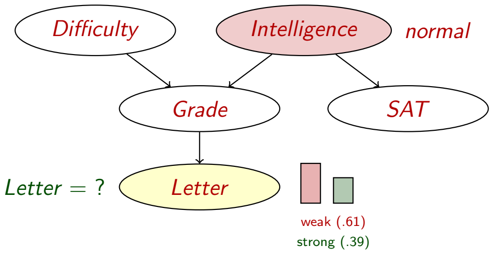
Reasoning patterns Given a Bayesian network, the following reasoning patterns can be used:

Reasoning patterns

Causal To make a prediction. From the cause, derive the effect.

Causal reasoning

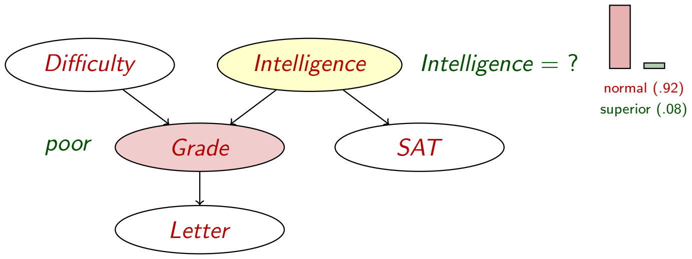
Example. Knowing `Intelligence`, it is possible to make a prediction of `Letter`.



Evidential To find an explanation. From the effect, derive the cause.

Evidential reasoning

Example. Knowing `Grade`, it is possible to explain it by estimating `Intelligence`.

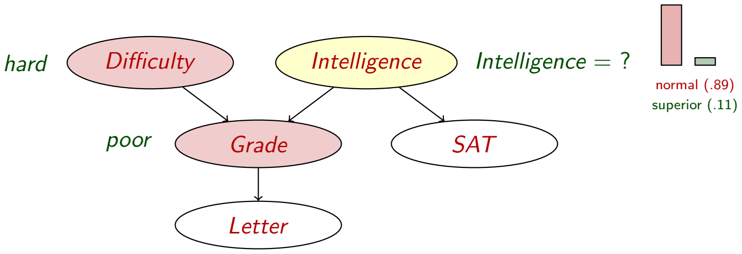


Explain away Observation obtained "passing through" other observations.

Explain away reasoning

Example. Knowing `Difficulty` and `Grade`, it is possible to estimate `Intelligence`.

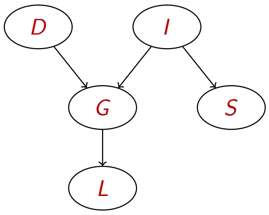
Note that if `Grade` was not known, `Difficulty` and `Intelligence` would have been independent.



Independence Intuitively, an effect is independent from a cause, if there is another cause in the middle whose value is already known.

Bayesian network independence

Example.



$$\mathbf{P} \models (\mathtt{L} \perp \mathtt{D}, \mathtt{I}, \mathtt{S} \mid \mathtt{G})$$

$$\mathbf{P} \models (\mathtt{S} \perp \mathtt{L} \mid \mathtt{G})$$

$$\mathbf{P} \models (\mathtt{S} \perp \mathtt{D}) \text{ but } \mathbf{P} \models (\mathtt{S} \not\perp \mathtt{D} \mid \mathtt{G}) \text{ (explain away)}$$

V-structure Effect with two causes. If the effect is not in the evidence, the causes are independent.

V-structure

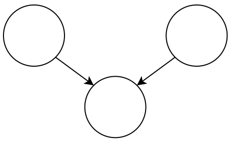


Figure 3.1: V-structure

Active two-edge trail The trail $X \leftrightharpoons Z \leftrightharpoons Y$ is active either if:

Active two-edge trail

- X, Z, Y is a v-structure $X \rightarrow Z \leftarrow Y$ and Z or one of its children is in the evidence.
- Z is not in the evidence.

In other words, influence can flow from X to Y passing by Z.

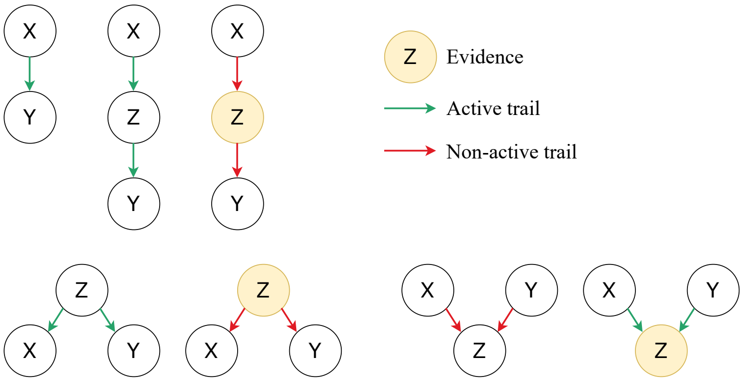


Figure 3.2: Example of active and non-active two-edge trails

Active trail A trail $X_1 \leftrightharpoons \cdots \leftrightharpoons X_n$ is active iff each two-edge trail $X_{i-1} \leftrightharpoons X_i \leftrightharpoons X_{i+1}$ along the trail is active.

Active trail

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

D-separation

Theorem 3.2.1. Two d-separated nodes are independent. In other words, two nodes are independent if there are no active trails between them.

Independence algorithm

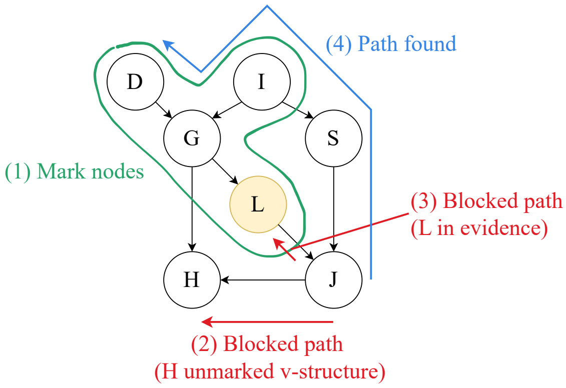
Blocked node A node is blocked if it blocks the flow. This happens if one and only one of the following conditions are met:

- The node is in the middle of an unmarked v-structure.
- The node is in the evidence.

To determine if $X \perp Y$ given the evidence $\mathbf{Z}$:

1. Traverse the graph bottom-up marking all nodes in $\mathbf{Z}$ or having a child in $\mathbf{Z}$.
2. Find a path from X to Y that does not pass through a blocked node.
3. If Y is not reachable from X, then X and Y are independent. Otherwise X and Y are dependent.

Example. To determine if $J \perp D$:



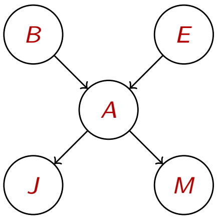
As a path has been found, $J \not\perp D$.

Global semantics Given a Bayesian network, the full joint distribution can be defined as the product of the local conditional distributions:

Global semantics

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

Example. Given the following Bayesian network:



$$\mathcal{P}\left(j \wedge m \wedge a \wedge \neg b \wedge \neg e\right)$$
$$= \mathcal{P}\left(\neg b\right)\mathcal{P}\left(\neg e\right)\mathcal{P}\left(a \,|\, \neg b, \neg e\right)\mathcal{P}\left(j \,|\, a\right)\mathcal{P}\left(m \,|\, a\right)$$

Local semantics Each node is conditionally independent of its non-descendants given its parents.

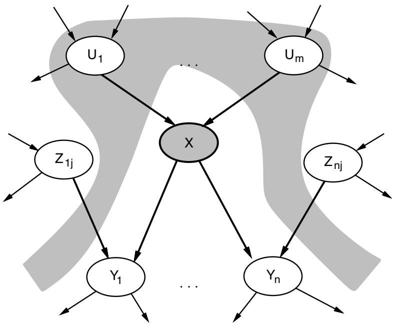


Figure 3.3: Local independence

Theorem 3.2.2. Local semantics $\iff$ Global semantics

Markov blanket Each node is conditionally independent of all the other nodes if its Markov blanket (parents, children, children's parents) is in the evidence.

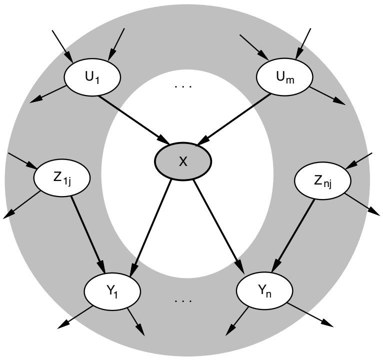


Figure 3.4: Markov blanket

3.3 Building Bayesian networks

3.3.1 Algorithm

The following algorithm can be used to construct a Bayesian network of n random variables:

1. Choose an ordering of the variables $X_1, \dots, X_n$.
2. For $i = 1, \dots, n$:
   - Add X_i to the network.
   - Select the parents of X_i from $X_1, \dots, X_{i-1}$ such that:

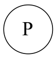
$$\mathbf{P}(X_i \,|\, \texttt{parents}(X_i)) = \mathbf{P}(X_i \,|\, X_1, \dots, X_{i-1})$$

By construction, this algorithm guarantees the global semantics.

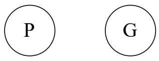
Example (Monty Hall)**.** The variables are:

- G: the choice of the guest.
- H: the choice of the host.
- P: the position of the prize.

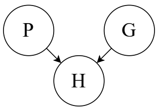
Note that $P \perp G$. Let the order be fixed as follows: P, G, H.



(a) First interaction



(b) Second interaction (note that $P \perp G$)



(c) Third interaction

The nodes of the resulting network can be classified as:

Initial evidence The initial observation.

Testable variables Variables that can be verified.

Operable variables Variables that can be changed by intervening on them.

Hidden variables Variables that ”compress” more variables to reduce the parameters.

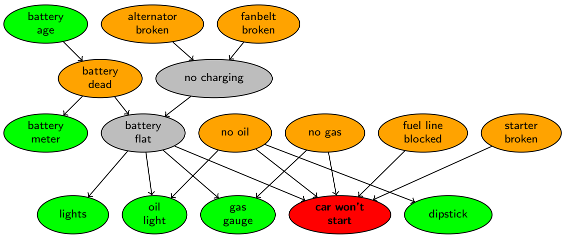
Example.

Initial evidence Red.

Testable variables Green.

Operable variables Orange.

Hidden variables Gray.



3.3.2 Structure learning

Structure learning

Learn the network from the available data.

Constraint-based Independence tests to identify the constraints of the edges.

Score-based Define a score to evaluate the network.

3.4 Causal networks

When building a Bayesian network, a correct ordering of the nodes that respects the causality allows to obtain more compact networks.

Structural equation

Structural equation Given a variable X_i with values x_i, its structural equation is a function f_i such that it represents all its possible values:

$$x_i = f_i(\text{other variables}, U_i)$$

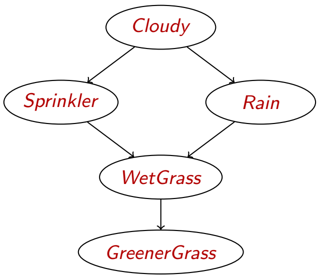
U_i represents unmodeled variables or error terms.

Causal network

Causal network Restricted class of Bayesian networks that only allows causally compatible ordering.

An edge exists between $X_j \rightarrow X_i$ iff X_j is an argument of the structural equation f_i of X_i.

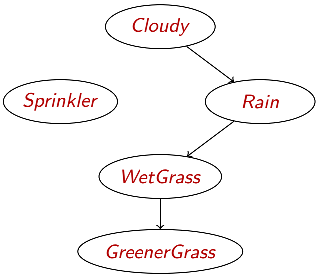
Example.



The structural equations are:

$$\begin{aligned}
\texttt{cloudy} &= f_C(U_C) \\
\texttt{sprinkler} &= f_S(\texttt{Cloudy}, U_S) \\
\texttt{rain} &= f_R(\texttt{Cloudy}, U_R) \\
\texttt{wet\_grass} &= f_W(\texttt{Sprinkler}, \texttt{Rain}, U_W) \\
\texttt{greener\_grass} &= f_G(\texttt{WetGrass}, U_G)
\end{aligned}$$

If the sprinkler is disabled, the network becomes:



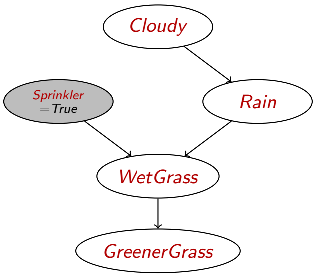
The structural equations become:

$$\begin{aligned}
\texttt{cloudy} &= f_C(U_C) \\
\texttt{sprinkler} &= f_S(U_S) \\
\texttt{rain} &= f_R(\texttt{Cloudy}, U_R) \\
\texttt{wet\_grass} &= f_W(\texttt{Rain}, U_W) \\
\texttt{greener\_grass} &= f_G(\texttt{WetGrass}, U_G)
\end{aligned}$$

do-operator The do-operator allows to represent manual interventions on the network. The operation $\texttt{do}(X_i = x_i)$ makes the structural equation of X_i constant (i.e. $f_i = x_i$, without arguments, so there won't be inward edges to X_i).

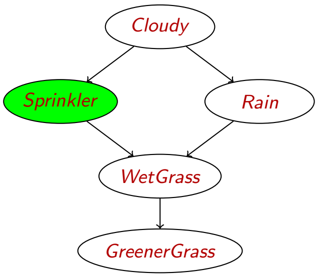
do-operator

Example.



By applying $\texttt{do}(\texttt{Sprinkler} = \texttt{true})$, the structural equations become:

$$\begin{aligned}
\texttt{cloudy} &= f_C(U_C)\\
\texttt{sprinkler} &= \texttt{true}\\
\texttt{rain} &= f_R(\texttt{Cloudy}, U_R)\\
\texttt{wet\_grass} &= f_W(\texttt{Sprinkler}, \texttt{Rain}, U_W)\\
\texttt{greener\_grass} &= f_G(\texttt{WetGrass}, U_G)
\end{aligned}$$



Note that Bayesian networks are not capable of modelling manual interventions. In fact, intervening and observing a variable are different concepts:

$$\mathcal{P}\left(\texttt{WetGrass} \mid \texttt{do}(\texttt{Sprinkler} = \texttt{true})\right)$$

$$\neq$$

$$\mathcal{P}\left(\texttt{WetGrass} \mid \texttt{Sprinkler} = \texttt{true}\right)$$

3.5 Compact conditional distributions

Use canonical distributions (standard patterns) to reduce the number of variables in a conditional probability table.

3.5.1 Noisy-OR

Noisy-OR distributions model a network of non-interacting causes with a common effect. A node X has k parents $U_1, \dots, U_k$ and possibly a leak node U_L to capture unmodeled concepts.

Noisy-OR

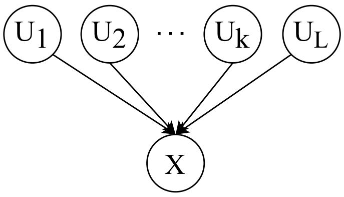


Figure 3.6: Example of noisy-OR network

Each node U_i has a failure (inhibition) probability q_i:

$$q_i = \mathcal{P}\left(\neg x \mid u_i, \neg u_j \text{ for } j \neq i\right)$$

The CPT can be built by computing the probabilities as:

$$\mathcal{P}\left(\neg x \mid \texttt{Parents}(X)\right) = \prod_{j\colon U_j = \texttt{true}} q_j$$

In other words:

$$\mathcal{P}(\neg x \mid u_1, \dots, u_n) = \mathcal{P}(\neg x \mid u_1) \cdot \mathcal{P}(\neg x \mid u_2) \cdot \dots \cdot \mathcal{P}(\neg x \mid u_n)$$

Because only the failure probabilities are required, the number of parameters is linear in the number of parents.

Example. We have as causes `Cold`, `Flu` and `Malaria` and as effect `Fever`. For simplicity there are no leak nodes. The failure probabilities are:

$$\begin{aligned} q_{\texttt{cold}} &= \mathcal{P}(\neg\texttt{fever} \mid \texttt{cold}, \neg\texttt{flu}, \neg\texttt{malaria}) = 0.6 \\ q_{\texttt{flu}} &= \mathcal{P}(\neg\texttt{fever} \mid \neg\texttt{cold}, \texttt{flu}, \neg\texttt{malaria}) = 0.2 \\ q_{\texttt{malaria}} &= \mathcal{P}(\neg\texttt{fever} \mid \neg\texttt{cold}, \neg\texttt{flu}, \texttt{malaria}) = 0.1 \end{aligned}$$

Known the failure probabilities, the entire CPT can be computed:

`Cold`	`Flu`	`Malaria`	$\mathcal{P}(\neg\texttt{fever})$	$1 - \mathcal{P}(\neg\texttt{fever})$
F	F	F	0.0	1.0
F	F	T	$q_{\texttt{malaria}} =$ 0.1	0.9
F	T	F	$q_{\texttt{flu}} =$ 0.2	0.8
F	T	T	$q_{\texttt{flu}} \cdot q_{\texttt{malaria}} =$ 0.02	0.98
T	F	F	$q_{\texttt{cold}} =$ 0.6	0.4
T	F	T	$q_{\texttt{cold}} \cdot q_{\texttt{malaria}} =$ 0.06	0.94
T	T	F	$q_{\texttt{cold}} \cdot q_{\texttt{flu}} =$ 0.12	0.88
T	T	T	$q_{\texttt{cold}} \cdot q_{\texttt{flu}} \cdot q_{\texttt{malaria}} =$ 0.012	0.988

3.5.2 Hybrid Bayesian networks

Hybrid Bayesian networks

Network with discrete and continuous random variables. Continuous variables must be converted into a finite representation. Possible approaches are:

Discretization

Discretization Values are divided into a fixed set of intervals. This approach may introduce large errors and large CPTs.

Finitely parametrized canonical families

Finitely parametrized canonical families There are two cases to handle using this approach:

Continuous child Given the continuous variables X and C and a discrete (boolean, for simplicity) variable D, we want to compute the distribution $\mathbf{P}(X \mid C, D)$.

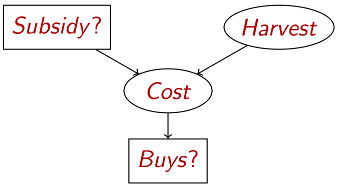
The discrete parent is handled by enumeration, by computing the probability over the domain of D.

Linear Gaussian

For the continuous parent, an arbitrarily chosen distribution over the values of X is used. A common choice is the **linear Gaussian** whose mean is a linear combination of the values of the parents and the variance is fixed.

A network with all continuous linear Gaussian distributions has the property of having a multivariate Gaussian distribution as joint distribution. Moreover, if a continuous variable has some discrete parents, it defines a conditional Gaussian distribution where, fixed the values of the discrete variables, the distribution over the continuous variable is a multivariate Gaussian.

Example. Let `Subsidy` and `Buys` be discrete variables and `Harvest` and `Cost` be continuous variables.



To compute $\mathbf{P}(\texttt{Cost} \mid \texttt{Harvest}, \texttt{Subsidy})$, we split the probabilities over the values of the discrete variable `Subsidy` and use a linear Gaussian for `Harvest`. We therefore have that:

$$\mathcal{P}\left(\texttt{C} = \texttt{c} \mid \texttt{Harvest} = \texttt{h}, \texttt{Subsidy} = \texttt{true}\right) = \mathcal{N}(a_t h + b_t, \sigma_t)(c)$$
$$\mathcal{P}\left(\texttt{C} = \texttt{c} \mid \texttt{Harvest} = \texttt{h}, \texttt{Subsidy} = \texttt{false}\right) = \mathcal{N}(a_f h + b_f, \sigma_f)(c)$$

where a_t, b_t, σ_t, a_f, b_f and σ_t are parameters.

Discrete child with continuous parents Given the continuous variable C and a discrete variable X, the probability of X given C in obtained by using a threshold function. For instance, probit or sigmoid distributions can be used.

3.5.3 Other methods

Dynamic Bayesian network Useful to model the evolution through time. A template variable X_i is instantiated as $X_i^{(t)}$ at each time step.

Dynamic Bayesian network

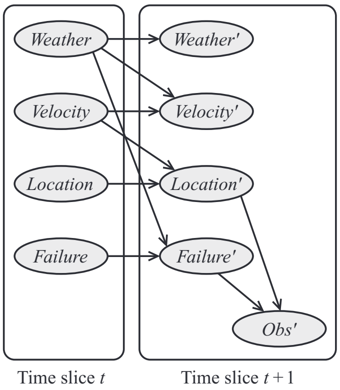


Figure 3.7: Example of dynamic Bayesian network

Density estimation Parameters of the conditional distribution can be learned using:

Density estimation

Bayesian learning calculate the probability of each hypothesis.

Approximations using the maximum-a-posteriori and maximum-likelihood hypothesis.

Expectation-maximization algorithm.

Undirected graphical models Markov networks are an alternative to probabilistic graphical models (as Bayesian networks). Markov networks are undirected graphs with factors (instead of probabilities) and are able to naturally capture independence relations.

Undirected graphical models

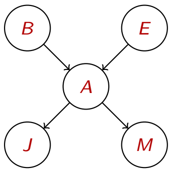
4 Exact inference

4.1 Inference by enumeration

Inference by enumeration

Method to sum out a joint probability without explicitly representing it by using CPT entries.
Enumeration follows a depth-first exploration and has a space complexity of $O(n)$ and time complexity of $O(d^n)$. It must be noted that some probabilities appear multiple times but require to be recomputed because of the definition of the algorithm.

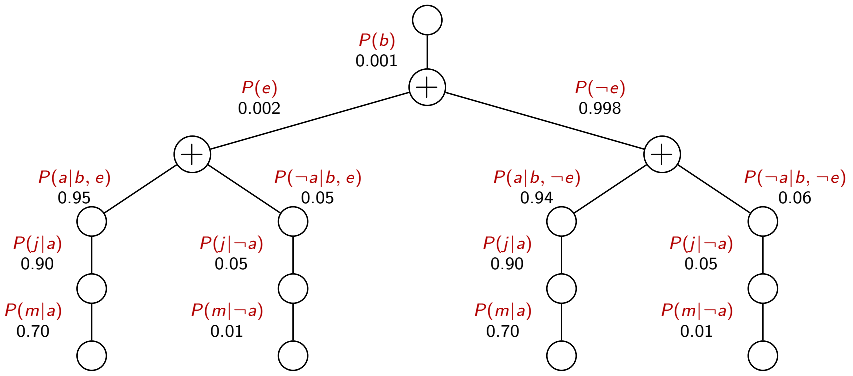
Example (Burglary)**.** Given the Bayesian network:



We want to compute $\mathbf{P}(B \mid j, m)$:

$$
\begin{aligned}
\mathbf{P}(B \mid j, m) &= \frac{\mathbf{P}(B, j, m)}{\mathcal{P}(j, m)} \\
&= \alpha \mathbf{P}(B, j, m) \\
&= \alpha \sum_{e} \sum_{a} \mathbf{P}(B, j, m, e, a) \\
&= \alpha \sum_{e} \sum_{a} \mathbf{P}(B) \mathcal{P}(e) \mathbf{P}(a \mid B, e) \mathcal{P}(j \mid a) \mathcal{P}(m \mid a) \\
&= \alpha \mathbf{P}(B) \sum_{e} \mathcal{P}(e) \sum_{a} \mathbf{P}(a \mid B, e) \mathcal{P}(j \mid a) \mathcal{P}(m \mid a)
\end{aligned}
$$

This can be represented using a tree:



4.2 Inference by variable elimination

Inference by variable elimination

Method that carries out summations right-to-left and stores intermediate results (called factors).

Pointwise product of factors $f(X, Y) \times g(Y, Z) = p(X, Y, Z)$

X	Y	$f(X,Y)$
0	0	1
0	1	3
1	0	2
1	1	1

Y	Z	$g(Y,Z)$
0	0	4
0	1	3
1	0	1
1	1	2

X	Y	Z	$f(X,Y) \times g(Y,Z)$
0	0	0	$1 \cdot 4$
0	0	1	$1 \cdot 3$
0	1	0	$3 \cdot 1$
0	1	1	$3 \cdot 2$
1	0	0	$2 \cdot 4$
1	0	1	$2 \cdot 3$
1	1	0	$1 \cdot 1$
1	1	1	$1 \cdot 2$

Figure 4.1: Example of pointwise product

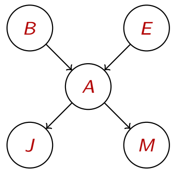
Summing out To sum out a variable X from a product of factors:

1. Move constant factors outside (i.e. factors that do not depend on X).
2. Compute the pointwise product of the remaining terms.

Example.

$$\begin{aligned}\sum_X f_1 \times \cdots \times f_k &= f_1 \times \cdots \times f_i \sum_X f_{i+1} \times \cdots \times f_k \\ &= f_1 \times \cdots \times f_i \times f_X\end{aligned}$$

Example (Burglary)**.** Given the Bayesian network:



We want to compute $\mathbf{P}(B \mid j, m) = \alpha \mathbf{P}(B) \sum_e \mathcal{P}(e) \sum_a \mathbf{P}(a \mid B, e) \mathcal{P}(j \mid a) \mathcal{P}(m \mid a)$.
We first work on the summation on A. We introduce as factors the entries of the CPT:

$$\mathbf{P}(B \mid j, m) = \alpha \mathbf{P}(B) \sum_e \mathcal{P}(e) \sum_a f_A(a, b, e) f_J(a) f_M(a)$$

Note that j and m are not parameters of the factors f_J and f_M because they are already given. We then sum out on A:

$$\mathbf{P}(B \mid j, m) = \alpha \mathbf{P}(B) \sum_e \mathcal{P}(e) f_{AJM}(b, e)$$

Now, we repeat the same process and sum out E:

$$\mathbf{P}(B \mid j, m) = \alpha \mathbf{P}(B) f_{EAJM}(b)$$

At last, we factor $\mathbf{P}(B)$:

$$\mathbf{P}(B \mid j, m) = \alpha f_B(b) f_{EAJM}(b)$$

4.2.1 Irrelevant variables

A variable X is irrelevant if summing over it results in a probability of 1.

Irrelevant variables

Theorem 4.2.1. Given a query X, the evidence $\boldsymbol{E}$ and a variable Y:

$$Y \notin (\texttt{Ancestors(}\{X\}\texttt{)} \cup \texttt{Ancestors(}\boldsymbol{E}\texttt{)}) \rightarrow Y \text{ is irrelevant}$$

Theorem 4.2.2. Given a query X, the evidence $\boldsymbol{E}$ and a variable Y:

$$Y \text{ d-separated from } X \text{ by } \boldsymbol{E} \rightarrow Y \text{ is irrelevant}$$

4.2.2 Complexity

Singly connected networks Network where any two nodes are connected with at most one undirected path. Time and space complexity is $O(d^k n)$.

Multiply connected networks The problem is NP-hard.

4.3 Clustering algorithm

Method that joins individual nodes to form clusters. Allows to estimate the posterior probabilities for n variables with complexity $O(n)$.

Clustering algorithm

5 Approximate inference

Stochastic simulation Class of methods that draw N samples from the distribution and estimate an approximate posterior $\hat{\mathcal{P}}$.

Stochastic simulation

δ-stochastic absolute approximation Given $\delta \in]0, 0.5[$ and $\varepsilon \in]0, 0.5[$, a δ-stochastic absolute approximation has error:

$$\left| \mathcal{P}(X|\boldsymbol{E}) - \hat{\mathcal{P}}(X|\boldsymbol{E}) \right| \leq \varepsilon$$

Moreover, the method might fail (with greater error) with probability δ.

δ-stochastic relative approximation Given $\delta \in]0, 0.5[$ and $\varepsilon \in]0, 0.5[$, a δ-stochastic relative approximation has error:

$$\frac{\left| \mathcal{P}(X|\boldsymbol{E}) - \hat{\mathcal{P}}(X|\boldsymbol{E}) \right|}{\mathcal{P}(X|\boldsymbol{E})} \leq \varepsilon$$

Moreover, the method might fail (with greater error) with probability δ.

Theorem 5.0.1. Approximate inference is NP-hard for any $\delta, \epsilon < 0.5$.

Consistency A sampling method is consistent if:

Consistency

$$\lim_{N \to \infty} \hat{\mathcal{P}}(x) = \mathcal{P}(x)$$

5.1 Sampling from an empty network

Sampling from an empty network

Sample each variable in topological order (i.e. from parents to children).
The probability $\mathcal{S}$ of sampling a specific event $x_1, \dots, x_n$ is given by the probability of the single events knowing their parents:

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

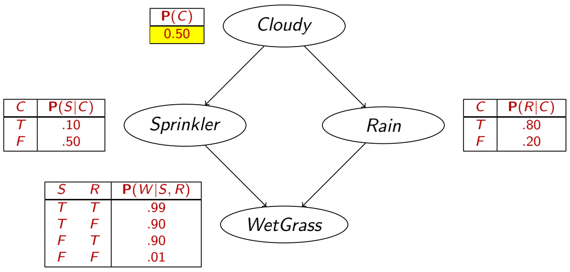
Theorem 5.1.1. Sampling from an empty network is consistent.

Proof. Let N be the number of samples and $\mathcal{N}(x_1, \dots, x_n)$ the number of times the event $x_1, \dots, x_n$ has been sampled.

$$\begin{aligned} \lim_{N \to \infty} \hat{\mathcal{P}}(x_1, \dots, x_n) &= \lim_{N \to \infty} \frac{\mathcal{N}(x_1, \dots, x_n)}{N} \\ &= \mathcal{S}(x_1, \dots, x_n) = \mathcal{P}(x_1, \dots, x_n) \end{aligned}$$

□

Example. Given the following Bayesian network:



A possible sampling order is `Cloudy`, `Sprinkler`, `Rain`, `WetGrass`.
Assuming that a random generator gives the sequence of probabilities $(0.4, 0.8, 0.1, 0.5)$, the sample will be:

$$\langle \mathcal{P}(C), \mathcal{P}(S|C), \mathcal{P}(R|C), \mathcal{P}(W|S, R) \rangle$$

$$\langle C = \texttt{false}, \mathcal{P}(S|C = \texttt{false}), \mathcal{P}(R|C = \texttt{false}), \mathcal{P}(W|S, R) \rangle$$

$$\langle C = \texttt{false}, S = \texttt{false}, R = \texttt{true}, \mathcal{P}(W|S = \texttt{false}, R = \texttt{true}) \rangle$$

$$\langle C = \texttt{false}, S = \texttt{false}, R = \texttt{true}, W = \texttt{true} \rangle$$

Note that the adopted convention is the following: if r it the probability given by a random generator and $\mathcal{P}(X) = p$, $X = \texttt{true}$ if $r \leq p$.

5.2 Rejection sampling

Rejection sampling

Given a known evidence $\boldsymbol{E}$, rejection sampling works as sampling from an empty network but removes any sample that does no agree with the evidence.
Obviously if $\mathcal{P}(\boldsymbol{E})$ is low, the majority of the samples will be discarded and more iterations are required to reach the desired number of samples.

Theorem 5.2.1. Rejection sampling is consistent.

Proof. Let $\mathcal{N}(\boldsymbol{X})$ be the number of times the event $\boldsymbol{X}$ has been sampled.

$$\begin{aligned} \hat{\mathcal{P}}(\boldsymbol{X}|\boldsymbol{E}) &= \frac{\mathcal{N}(\boldsymbol{X}, \boldsymbol{E})}{\mathcal{N}(\boldsymbol{E})} \\ &\approx \frac{\mathcal{P}(\boldsymbol{X}, \boldsymbol{E})}{\mathcal{P}(\boldsymbol{E})} = \mathcal{P}(\boldsymbol{X}|\boldsymbol{E}) \end{aligned}$$

The approximation derives from the consistency of sampling from an empty network. □

5.3 Likelihood weighting

Likelihood weighting

Given a known evidence $\boldsymbol{E}$, likelihood weighting samples non-evidence variables and weights each sample by the likelihood of the evidence.
The probability $\mathcal{S}$ of sampling a specific event $\boldsymbol{Z}$ and evidence $\boldsymbol{E}$ is given by the probability of the single events in $\boldsymbol{Z}$ knowing their parents:

$$\mathcal{S}(\boldsymbol{Z}, \boldsymbol{E}) = \prod_{z_i \in \boldsymbol{Z}} \mathcal{P}(z_i | \texttt{parents}(z_i))$$

The weight of a sample $(\boldsymbol{Z}, \boldsymbol{E})$ is given by the probability of the single events in $\boldsymbol{E}$ knowing their parents:

$$\mathrm{w}(\boldsymbol{Z}, \boldsymbol{E}) = \prod_{e_i \in \boldsymbol{E}} \mathcal{P}(e_i | \texttt{parents}(e_i))$$

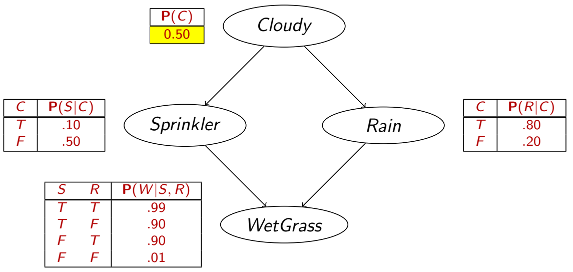
Theorem 5.3.1. Likelihood weighting is consistent.

Proof. The weighted sampling probability is given by:

$$\begin{aligned}\mathcal{S}(\boldsymbol{Z}, \boldsymbol{E}) \cdot \mathrm{w}(\boldsymbol{Z}, \boldsymbol{E}) &= \prod_{z_i \in \boldsymbol{Z}} \mathcal{P}\left(z_i | \texttt{parents}(z_i)\right) \cdot \prod_{e_i \in E} \mathcal{P}\left(e_i | \texttt{parents}(e_i)\right) \\ &= \mathcal{P}\left(\boldsymbol{Z}, \boldsymbol{E}\right)\end{aligned}$$

This is a consequence of the global semantics of Bayesian networks. □

Example. Given the following Bayesian network:



P(*C*)
0.50

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

Knowing that $S = \texttt{true}$ and $W = \texttt{false}$, we sample in the order: `Cloudy`, `Rain`. Assuming that a random generator gives the sequence of probabilities $(0.4, 0.1)$, the sample will be:

$$\langle \mathcal{P}\left(C\right), S = \texttt{true}, \mathcal{P}\left(R|C\right), W = \texttt{false} \rangle$$

$$\langle C = \texttt{true}, S = \texttt{true}, \mathcal{P}\left(R|C = \texttt{true}\right), W = \texttt{false} \rangle$$

$$\langle C = \texttt{true}, S = \texttt{true}, R = \texttt{true}, W = \texttt{false} \rangle$$

The weight associated to the sample is given by the probability of the evidence:

$$\begin{aligned}\mathrm{w} &= \mathcal{P}\left(S = \texttt{true} | C = \texttt{true}\right) \cdot \mathcal{P}\left(W = \texttt{false} | S = \texttt{true}, R = \texttt{true}\right) \\ &= 0.1 \cdot (1 - 0.99) = 0.001\end{aligned}$$

5.4 Markov chain Monte Carlo

Markov chain Monte Carlo

Sampling on a Markov chain where states contain an assignment to all variables. Adjacent states of the Markov chain differ by only one variable. Therefore, the probability of an edge connecting two states is given by the probability of the updated variable known its Markov blanket:

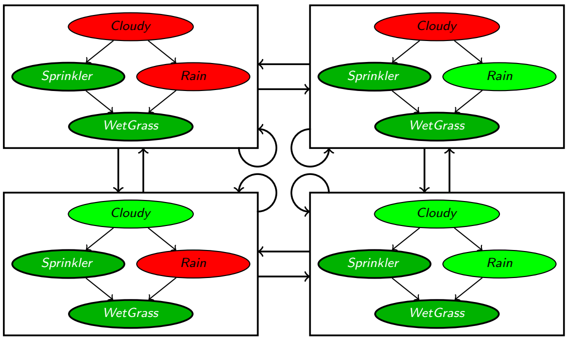
$$\mathcal{P}\left(x_i | \texttt{markov\_blanket}(X_i)\right) = \mathcal{P}\left(x_i | \texttt{parents}(X_i)\right) \cdot \prod_{Z_j \in \texttt{children}(x_i)} \mathcal{P}\left(z_j | \texttt{parents}(Z_j)\right)$$

Theorem 5.4.1. Markov chain Monte Carlo is consistent.
Note: nevertheless, it is difficult to tell if convergence has been achieved.

Proof. Consequence of the fact that a long-run on a Markov chain converges to the posterior probability of the states. □

Compiled network A naive implementation of Markov chain Monte Carlo requires to repeatedly compute the probabilities with the Markov blanket. A solution is to compile the network into a model-specific inference code.

Example. Given the evidence $S =$ `true` and $W =$ `true`, the structure of the Markov chain can be defined as follows:



`<end of course>`