# BELIEF NETWORKS

**Outline**

- Belief networks: syntax and semantics
- Inference in belief networks
- Multiply connected belief networks
- Other approaches to uncertain reasoning

Based on the textbook by S. Russell & P. Norvig:

Artificial Intelligence, A Modern Approach, Chapter 15

# BELIEF NETWORKS: SYNTAX

**Definition.** A belief network is a *directed acyclic graph* (DAG) $G = \langle \{X_1, \ldots, X_n\}, E \rangle$ where $E \subseteq \{X_1, \ldots, X_n\}^2$ and

1. nodes $X_1, \ldots, X_n$ are random variables,
2. an arc $\langle X, Y \rangle \in E$ of $G$ represents a direct influence relationship between the variables $X$ and $Y$, and
3. each node $X$ is assigned a completely specified probability distribution $\mathbf{P}(X|\mathrm{Parents}(X))$ where

$$\mathrm{Parents}(X) = \{Y \mid \langle Y, X \rangle \in E\}.$$

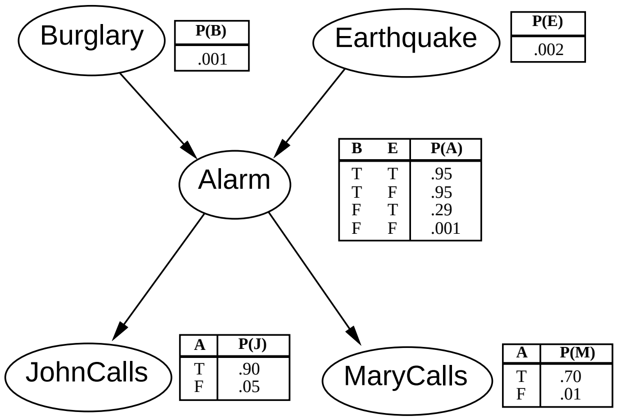
- Belief networks are also called *Bayesian networks*, *probabilistic networks*, *causal networks* or *knowledge maps*.
- A compact specification of the joint distribution $\mathbf{P}(X_1, \ldots, X_n)$.

**Example.** Consider a network based on five Boolean random variables:

1. $Burglary$ = "a burglar enters our home".
2. $Earthquake$ = "an earthquake occurs".
3. $Alarm$ = "our burglar alarm goes off".
   The alarm is fairly reliable at detecting a burglary, but may occasionally respond to minor earthquakes.
4. $JohnCals$ = "Our neighbor John calls and reports an alarm."
   He always calls when he hears the alarm, but sometimes confuses telephone ringing with the alarm.
5. $MaryCalls$ = "Our neighbor Mary calls and reports an alarm ".
   She likes loud music and sometimes misses the alarm altogether.

Shorthands $B$, $E$, $A$, $J$, and $M$ are also introduced for these variables.

- The relationships of the variables are given as a belief network.
- The probability distributions $\mathbf{P}(X \mid \mathrm{Parents}(X))$ associated with variables $X$ are given as *conditional probability tables* (CPTs).



| P(B) |
|---|
| .001 |

| P(E) |
|---|
| .002 |

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

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

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

## THE SEMANTICS OF BELIEF NETWORKS

- ➤ A belief network for the random variables $X_1, \ldots, X_n$ is a representation of the joint probability distribution $\mathbf{P}(X_1, \ldots, X_n)$.
- ➤ In the sequel, a shorthand $x_i$ is used for the atomic event $X_i = x_i$.
- ➤ Arrows encode conditional independence statements and therefore the probabilities of atomic events are determined by

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid \mathrm{Parents}(x_i))$$

where $\mathrm{Parents}(x_i)$ refers to the assignments of $X_j \in \mathrm{Parents}(X_i)$.

**Example.** Let us compute the probability of $J \wedge M \wedge A \wedge \neg B \wedge \neg E$:

$$\begin{aligned} & P(J \wedge M \wedge A \wedge \neg B \wedge \neg E) \\ = \; & P(J|A)P(M|A)P(A|\neg B \wedge \neg E)P(\neg E)P(\neg B) \\ = \; & 0.9 \times 0.7 \times 0.001 \times 0.998 \times 0.999 = 0.00063 \end{aligned}$$

## Conditional Independence Revisited

**Definition.** Let $P(\psi) > 0$. Sentences $\phi_1$ and $\phi_2$ are *conditionally independent given* $\psi \iff P(\phi_1 \wedge \phi_2 \mid \psi) = P(\phi_1 \mid \psi)P(\phi_2 \mid \psi)$.

**Proposition.** If $P(\psi) > 0$, $P(\phi_1 \wedge \psi) > 0$, and $P(\phi_2 \wedge \psi) > 0$, then $\phi_1$ and $\phi_2$ are conditionally independent given $\psi \iff$ $P(\phi_1 \mid \phi_2 \wedge \psi) = P(\phi_1 \mid \psi)$ and $P(\phi_2 \mid \phi_1 \wedge \psi) = P(\phi_2 \mid \psi)$ hold.

*Proof.* For the former equation, we note that

$$\begin{aligned} & P(\phi_1 \wedge \phi_2 \mid \psi) = P(\phi_1 \mid \psi)P(\phi_2 \mid \psi) \\ \iff \; & \tfrac{P(\phi_1 \wedge \phi_2 \wedge \psi)}{P(\psi)} = \tfrac{P(\phi_1 \wedge \psi)}{P(\psi)} \cdot \tfrac{P(\phi_2 \wedge \psi)}{P(\psi)} \\ \iff \; & P(\phi_1 \wedge \phi_2 \wedge \psi)P(\psi) = P(\phi_1 \wedge \psi)P(\phi_2 \wedge \psi) \\ \iff \; & P(\phi_1 \mid \phi_2 \wedge \psi) = \tfrac{P(\phi_1 \wedge \phi_2 \wedge \psi)}{P(\phi_2 \wedge \psi)} = \tfrac{P(\phi_1 \wedge \psi)}{P(\psi)} = P(\phi_1 \mid \psi). \end{aligned}$$

## A method for constructing belief networks

- ➤ In a belief network $G = \langle \{X_1, \ldots, X_n\}, E \rangle$, a node $X_j \neq X_i$ is a *predecessor* of $X_i \iff$ there are nodes $Y_1, \ldots, Y_m$ such that $Y_1 = X_j$, $Y_m = X_i$, and $\forall j \in \{1, \ldots, m-1\}$: $\langle Y_j, Y_{j+1} \rangle \in E$.
- ➤ Because $G$ is a DAG, we may assume that the nodes $X_1, \ldots, X_n$ are ordered so that the predecessors of $X_i$ are among $X_1, \ldots, X_{i-1}$. Thus also $\mathrm{Parents}(X_i) \subseteq \{X_1, \ldots, X_{i-1}\}$.
- ➤ By the definition of conditional probability, we have that

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

- ➤ A belief network is a correct representation if each variable $X$ is conditionally independent of its predecessors $Y$ given $\mathrm{Parents}(X)$.
- ➤ Under the assumptions on conditional independence and node ordering, it can be established that

$$\mathbf{P}(X_i \mid X_{i-1}, \ldots, X_1) = \mathbf{P}(X_i \mid \mathrm{Parents}(X_i)). \qquad (1)$$

- ➤ The choice of $\mathrm{Parents}(X)$ for a random variable $X$ affects how far conditional independence assumptions can be applied.
- ➤ $\mathrm{Parents}(X)$ should contain all variables that directly influence $X$.

**Example.** Only *Alarm* directly influences *MaryCalls*. Given *Alarm*, *MaryCals* is conditionally independent of *Earthquake* and *Burglary*:

$$\begin{aligned} & \mathbf{P}(\mathit{MaryCals} \mid \mathit{Alarm}, \mathit{Earthquake}, \mathit{Burglary}) \\ & = \mathbf{P}(\mathit{MaryCals} \mid \mathit{Alarm}). \end{aligned}$$

## Incremental Belief Network Construction

A belief network can be constructed as follows:

1. Choose variables $X_1, \ldots, X_n$ for describing the domain of interest.
2. Choose an ordering for the variables.
3. While there are unprocessed variables do the following:
   (a) Pick the next variable $X_i$ and add it as a node to the network.
   (b) Set $\mathrm{Parents}(X_i)$ to some minimal set of nodes already in the network so that conditional independence property (1) holds.
   (c) Define the conditional probability table for $X_i$.

☞ The resulting network is automatically acyclic and consequently the axioms of probability are also satisfied.

## On Compactness and Node Ordering

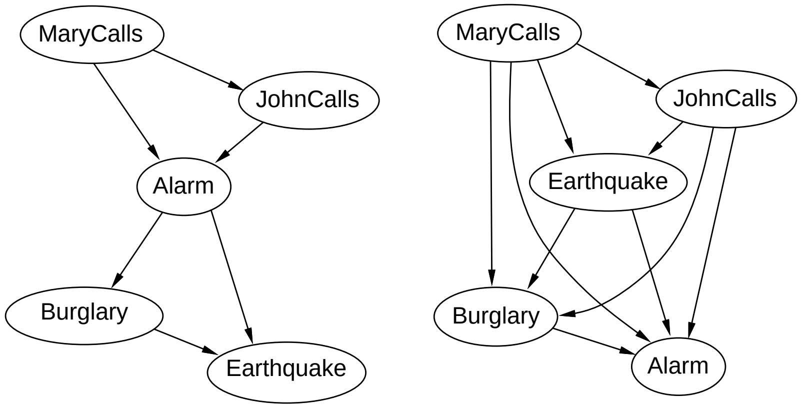
- ➤ A belief network can be a compact representation of the joint probability distribution (*locally structured* or *sparse* system).
- ➤ If each Boolean variable directly influences at most $k$ other, then only $n2^k$ probabilities have to be specified (instead of $2^n$).
  **Example.** When $n = 20$ and $k = 5$, we would have to specify $n2^k = 640$ and $2^n = 1048576$ probabilities, respectively.
- ➤ Number of arcs (accuracy of probabilities) *versus* cost of specifying extra information (extending CPTs).
- ➤ Choosing a good node ordering is a non-trivial task.
- ➤ Heuristics: the *root causes* of the domain should be added first, then the variables influenced by them, and so forth.

**Example.** Let us reconstruct the belief network for the alarm domain using a different node ordering:

$$MaryCalls,\ JohnCalls,\ Alarm,\ Burglary,\ Earthquake$$

1. As the first node, $MaryCalls$ gets no parents.
2. When $JohnCals$ is added, $MaryCals$ becomes a parent of $JohnCals$, as $P(JohnCalls \mid MaryCalls) \neq P(JohnCalls)$.
3. Similarly, $Alarm$ depends on both $MaryCalls$ and $JohnCals$.
4. Since $P(Burglary \mid Alarm, JohnCalls, MaryCalls) = P(Burglary \mid Alarm)$, the only parent of $Burglary$ is $Alarm$.
5. Nodes $Burglary$ and $Alarm$ become parents of $Earthquake$, as $P(Earthquake \mid Burglary, Alarm, JohnCalls, MaryCalls) = P(Earthquake \mid Burglary, Alarm)$.

- ➤ The resulting belief network is given below on the left-hand side:



- ➤ The one on the right-hand side is obtained with another ordering and it as complex as the full joint distribution!

## Representing Conditional Probability Tables

- Specifying conditional probability tables means often a lot of work.
- To ease this process, some canonical distributions have been proposed such as *deterministic* and *noisy* logical relationships.
- In the deterministic case, there is no uncertainty and the value of $X$ is obtained as a logical function from those of $\mathrm{Parents}(X)$.

**Example.** Define $\mathit{NorthAmerican} \leftrightarrow \mathit{Canadian} \lor \mathit{US} \lor \mathit{Mexican}$. This corresponds to specifying a CPT as follows:

| $\mathit{Canadian}$ | $\mathit{US}$ | $\mathit{Mexican}$ | $\mathit{NorthAmerican}$ |
|---|---|---|---|
| F | F | F | 0.0 |
| T | F | F | 1.0 |
| ⋮ | ⋮ | ⋮ | ⋮ |

## Noisy Logical Relationships

- Noisy logical relationships add some uncertainty to the scenario.
- A **noisy OR** relationship comprises the following principles:
  1. Each cause has an independent chance of causing the effect.
  2. All possible causes are listed.
  3. Whatever inhibits some cause from causing an effect is independent of whatever inhibits other causes from causing the effect. Inhibitors are summarized as **noise parameters**.
- A noisy OR relationship in which a variable depends on $k$ parents can be described using $k$ parameters.

  In contrast to this, $2^k$ entries are needed if a full CPT is specified.

**Example.** Let us consider a medical domain including the variables $\mathit{Fever}$ (a symptom), $\mathit{Cold}$, $\mathit{Flu}$, and $\mathit{Malaria}$ (diseases).

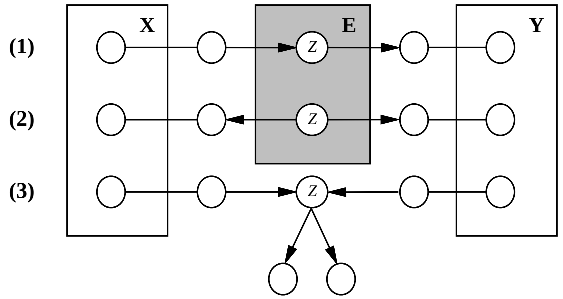
Using parameters $P(\neg\mathit{Fever} \mid \mathit{Cold}) = 0.6$, $P(\neg\mathit{Fever} \mid \mathit{Flu}) = 0.2$, and $P(\neg\mathit{Fever} \mid \mathit{Malaria}) = 0.1$, the following CPT is obtained:

| $\mathit{Cold}$ | $\mathit{Flu}$ | $\mathit{Malaria}$ | $P(\mathit{Fever})$ | $P(\neg\mathit{Fever})$ |
|---|---|---|---|---|
| $F$ | $F$ | $F$ | $0.0$ | $1.0$ |
| $F$ | $F$ | $T$ | $0.9$ | $0.1$ |
| $F$ | $T$ | $F$ | $0.8$ | $0.2$ |
| $F$ | $T$ | $T$ | $0.98$ | $0.02 = 0.2 \times 0.1$ |
| $T$ | $F$ | $F$ | $0.4$ | $0.6$ |
| $T$ | $F$ | $T$ | $0.94$ | $0.06 = 0.6 \times 0.1$ |
| $T$ | $T$ | $F$ | $0.88$ | $0.12 = 0.6 \times 0.2$ |
| $T$ | $T$ | $T$ | $0.988$ | $0.012 = 0.6 \times 0.2 \times 0.1$ |

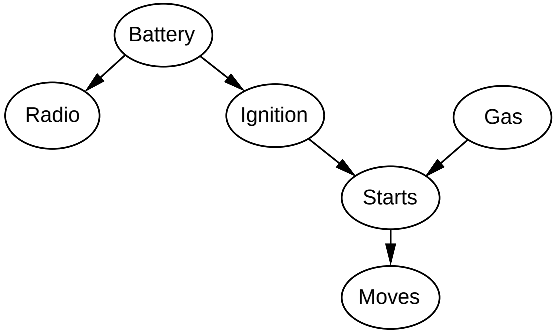
## Conditional Independence Relations

- Mutually independent sets of nodes can be distinguished using the notion of **direction-dependent separation** (or **d-separation**).

**Definition.** Let $X$, $Y$ and $E$ be sets of nodes/variables. Then $X$ and $Y$ are conditionally independent given $E$, if every undirected path from a node in $X$ to a node in $Y$ is d-separated by $E$.



**Example.** Let us have a look on the following belief network which describes some features of a car's electrical system and engine.



According to this model,

- $Gas$ and $Radio$ are independent given $Battery$, and
- $Gas$ and $Radio$ are dependent given $\neg Starts$.

## INFERENCE IN BELIEF NETWORKS

➤ The task is to compute $\mathbf{P}(Q_1, \ldots, Q_n \mid E_1 = e_1, \ldots, E_m = e_m)$ given **query variables** $Q_1, \ldots, Q_n$ and exact values $e_1, \ldots, e_m$ for some **evidence variables** $E_1, \ldots, E_m$.

**Examples.** Recalling the alarm example, how to evaluate queries such as $P(Burglary \mid JohnCalls, MaryCalls)$ and $P(Alarm \mid JohnCalls, Earthq^{uake})$?

➤ An agent gets values for evidence variables from its percepts and asks about the possible values of other variables so that it can decide what action to take (recall the decision theoretic design).

➤ We need a procedure BELIEF-NET-TELL for adding evidence to the network and a function BELIEF-NET-ASK for computing the posterior probability distribution.

## On the Nature of Probabilistic Inferences

➤ **Causalinferences** (from causes to effects):

$P(JohnCals \mid Burglary) = 0.9 \times 0.95 + 0.05 \times 0.05 = 0.8575$ and
$P(MaryCalls \mid Burglary) = 0.7 \times 0.95 + 0.01 \times 0.05 = 0.6655$.

☞ John and Mary call quite reliably in case of a burglary.

➤ **Diagnostic inferences** (from effects to causes):

$P(Burglary \mid JohnCalls) = \frac{P(JohnCalls|Burglary)P(Burglary)}{P(JohnCalls)} \approx 0.0164$

where $P(JohnCalls) =$

$$P(JohnCals \mid Alarm)P(Alarm) + P(JohnCals \mid \neg Alarm)P(\neg Alarm) \approx 0.0521$$

and $P(\neg Alarm) = 1 - P(Alarm) \approx 0.99747$
(see the next slide how to compute $P(Alarm) \approx 0.00253$).

➤ **Intercausalinferences** (between causes/common effect):

$P(Burglary \mid Alarm) = \frac{P(Alarm|Burglary)P(Burglary)}{P(Alarm)} \approx 0.376$

where $P(Alarm \mid Burglary) = 0.95$, $P(Burglary) = 0.001$, and

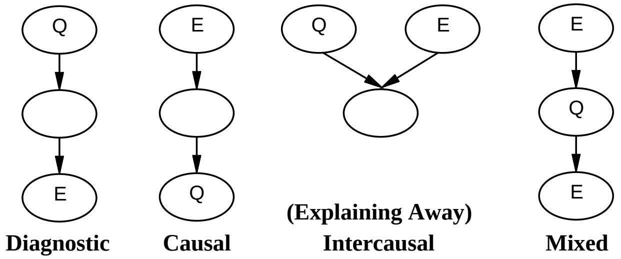
$$\begin{aligned} P(Alarm) &= 0.95 \times 0.001 \times 0.002 + \\ &\quad 0.95 \times 0.001 \times 0.998 + \\ &\quad 0.29 \times 0.999 \times 0.002 + \\ &\quad 0.001 \times 0.999 \times 0.998 \\ &\approx 0.0025264. \end{aligned}$$

On the other hand, $P(Bu^{rglary}|Alarm \wedge Earthq^{uake}) \approx 0.003$

☞ An earthquake **explains away** the possibility of a burglary.

- **Mixed inferences** (combining two or more of the above):
  $P(Alarm|JohnCalls \wedge \neg Earthquake) \approx 0.030$ and
  $P(Burglary|JohnCalls \wedge \neg Earthquake) \approx 0.017$.

- The four reasoning modes can be illustrated as follows:



- One might perform **sensitivity analysis** to understand which aspects of the model have the greatest impact on the probabilities of the query variables.

## An Algorithm for Query Answering

- A **polytree** is a **singly connected** graph: there is at most one undirected path between any two nodes.

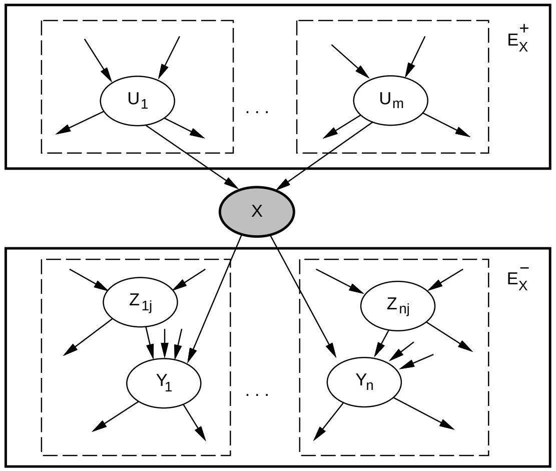
- If a belief network forms a polytree, the probability distribution $\mathbf{P}(X \mid E)$ can be computed very efficiently (in **linear time**).

- The set of evidence variables $E$ is partitioned w.r.t. $X$:
  - The **causal support** $E_X^+$ for $X$: evidence variables in $E$ that are connected to $X$ through its parents.
  - The **evidential support** $E_X^-$ for $X$: evidence variables in $E$ that are connected to $X$ through its children.

- The distribution $\mathbf{P}(X \mid E)$ is obtained by normalization:

$$\mathbf{P}(X \mid E) = \alpha \mathbf{P}(E_X^- \mid X)\mathbf{P}(X \mid E_X^+).$$

- The supports $E_X^+$ and $E_X^-$ for $X$ can be illustrated as follows (all the boxes are disjoint and have no links connecting them):



## MULTIPLY CONNECTED NETWORKS

- A belief network is **multiply connected** if there are at least two variables $X$ and $Y$ connected by more than one path, i.e., $X$ and $Y$ are interconnected by several causal mechanisms.

- Algorithms for polytree networks can be used as subroutines in algorithms for general (multiply connected) networks.

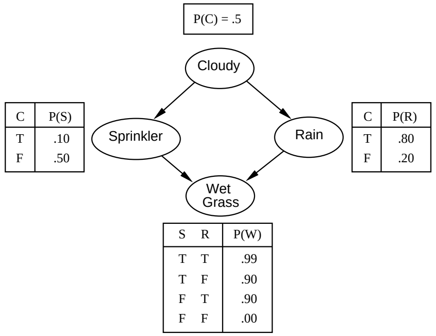
- Different methods exist for multiply connected networks:
  - Clustering methods
  - Cutset conditioning methods
  - Stochastic simulation methods

- In the general case, exact inference in belief networks is NP-hard.

# Clustering Methods

- ➤ Multiply connected belief networks are transformed into polytrees by combining some nodes into **meganodes**.

**Example.** Consider clustering the nodes *Sprinkler* and *Rain* in the following multiply connected network:



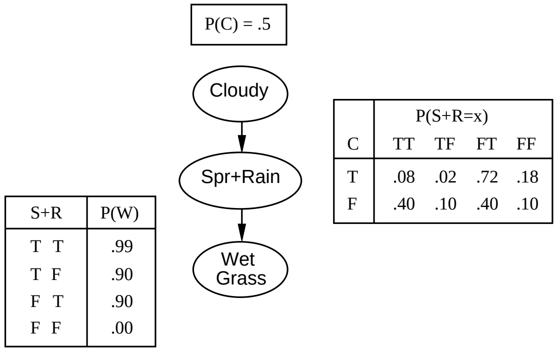
P(C) = .5

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

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

| S | R | P(W) |
|---|---|---|
| T | T | .99 |
| T | F | .90 |
| F | T | .90 |
| F | F | .00 |

- ➤ The following polytree network is obtained:



P(C) = .5

| S+R | P(W) |
|---|---|
| T T | .99 |
| T F | .90 |
| F T | .90 |
| F F | .00 |

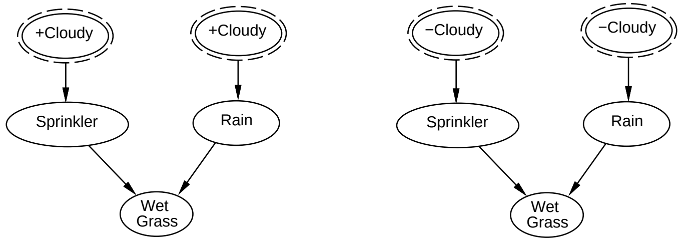
|  | P(S+R=x) |  |  |  |
|---|---|---|---|---|
| C | TT | TF | FT | FF |
| T | .08 | .02 | .72 | .18 |
| F | .40 | .10 | .40 | .10 |

- ➤ Linear time algorithms can be used for query answering, but the size of the network grows exponentially in the worst case.
- ➤ Typically, there are several ways to compose meganodes and it is non-trivial to choose the best way to perform clustering.

# Cutset Conditioning Methods

- ➤ In **cutset conditioning**, the network is decomposed into several simpler polytrees by instantiating variables to definite values.
- ➤ The probability $P(X \mid E)$ is computed as a weighted average over the probabilities computed using each polytree in turn.

**Example.** The instantiation of *Cloudy* yields two polytrees:



# Stochastic Simulation Methods

**Logic sampling:**

- ➤ Logic sampling is based on a stochastic simulation of the world described by a belief network.
- ➤ Starting from the root nodes, atomic events are randomly generated by selecting definite values for random variables.
- ➤ The value for a random variable $X$ is chosen according to the conditional probability table associated with $X$.
- ➤ A distribution $\mathbf{P}(X \mid E) = \frac{\mathbf{P}(X \wedge E)}{\mathbf{P}(E)}$ of interest is estimated by counting the frequencies with which events occur.
- ➤ Logic sampling is not very useful if $E$ occurs very rarely.

**Example.** E.g., $P(\mathit{WetGrass} \mid \mathit{Sprinkler} \wedge \mathit{Rain})$ converges slowly.

**Likelihood weighting:**

- ➤ Likelihood weighting is similar to logic sampling, but the values of evidence variables $E$ are not randomly chosen.
- ➤ The CPT associated with $E$ is consulted to to see how likely the value of $E = e$ is given the values of preceding nodes $X_1, \ldots, X_n$.
- ➤ In this way, the conditional probability $P(e \mid x_1, \ldots, x_n)$ is interpreted as a likelihood weight for that particular run.
- ➤ An estimate of $P(X = x \mid E = e)$ is obtained as a weighted proportion of runs with $X = x$ among the runs accumulated so far.
- ➤ Likelihood weighting converges much faster than logic sampling.
- ➤ Getting accurate probabilities for unlikely events is still a problem.

**Example.** Let us estimate the conditional probability $P(\mathit{WetGrass} \mid \mathit{Rain})$ by likelihood weighting.

The values of variables are chosen randomly as follows:

1. $P(\mathit{Cloudy}) = 0.5$: $\mathit{Cloudy} := \mathit{False}$ is chosen.
2. $P(\mathit{Sprinkler} \mid \neg\mathit{Cloudy}) = 0.5$: $\mathit{Sprinkler} := \mathit{True}$ is chosen.
3. $\mathit{Rain}$ is an evidence variable that has been set to $\mathit{True}$: $P(\mathit{Rain} \mid \neg\mathit{Cloudy}) = 0.2$.
4. $P(WetGrass|Sprinkler \wedge \mathit{Rain}) = 0.99$: $WetGrass := True$ is chosen.

☞ We have completed a run saying that $\mathit{WetGrass} = \mathit{True}$ given $\mathit{Rain} = \mathit{True}$ with a likelihood weight $0.2$.

## Knowledge Engineering for Uncertainty

- ➤ Decide which aspects of the system are modeled.
- ➤ Decide on a vocabulary of random variables.
- ➤ Encode general knowledge about dependencies among variables:
  - – Qualitative dependency information as links between variables
  - – Quantitative dependency information as probabilities (frequencies or experts' subjective estimates)
- ➤ Encode a description of the specific problem instance.
- ➤ Pose queries to the inference procedure and get answers.

**Example.** PATHFINDER is a diagnostic expert system for lymph-node diseases. When compared with real physicians, PATHFINDER IV made a successful diagnosis for 89% out of 53 patients being diagnosed.

## OTHER APPROACHES TO UNCERTAINTY

- ➤ Early expert systems were based on strict logical reasoning.
- ➤ Probabilistic techniques were dominating in the second generation, but these techniques suffered from the exponential blow-up of the joint probability distribution w.r.t. the number of variables.
- ➤ Consequently, many alternatives to probabilities were pursued:
  1. Default reasoning
  2. Rules with certainty factors
  3. Dempster-Shafer theory
  4. Fuzzy logic

## Default Reasoning

- ➤ Reasoning by default means inferring something in the absence of any information to the contrary.
- ➤ Provides a compact way to encode exceptions to general principles.
- ➤ A qualitative approach to handle uncertainty.
- ➤ Default reasoning *violates* the **monotonicity** property of classical logic: if $\Sigma_1 \models \phi$ and $\Sigma_1 \subseteq \Sigma_2$, then $\Sigma_2 \models \phi$.
- ➤ Several formalizations of *non-monotonic reasoning* have been proposed: **default logic** [Reiter, 1980], **circumscription** [McCarthy, 1980], **autoepistemic logic** [Moore, 1983], . . .
- ➤ Implementation techniques have substantially improved during 90s.

- ➤ Logic programs with *negation as failure to prove* from an important subclass of non-monotonic theories.

**Example.** Let us describe the applicability of actions using rules:

$$\begin{aligned}\{\quad & \mathit{doable}(A) \leftarrow \mathit{preconds}(A) \land \text{not } \mathit{exceptional}(A), \\ & \mathit{exceptional}(A) \leftarrow \text{not } \mathit{deterministic}(A), \\ & \mathit{exceptional}(A) \leftarrow \mathit{delayed}(A)\ \}\end{aligned}$$

- ➤ The semantics of "not $\phi$" is different from classical negation $\neg\phi$.
- ➤ The conclusion $\mathit{doable}(\mathit{run})$ can be drawn by the rules above given the premises $\mathit{preconds}(\mathit{run})$ and $\mathit{deterministic}(\mathit{run})$.
- ➤ Such a conclusion is no longer possible if $\mathit{delayed}(\mathit{run})$ is introduced as an additional premise.
- ➤ Dropping the premise $\mathit{deterministic}(A)$ has the same effect.

## Logical Rules and Certainty Factors

- ➤ Reasoning systems based on classical logic have important properties that are lacked by their probabilistic counterparts:
  1. **Locality**: a rule can be used for making inferences without worrying about the other rules in the system.
  2. **Detachment:** if a sentence $\phi$ is proven to be valid, it can be detached from its justification (proof), as it universally true.
  3. **Truth-functionality:** the truth values of complex sentences can be computed from the truth values of their components.
- ➤ Unfortunately, problems arise with truth-functionality and chained inferences, if logical rules are equipped with certainty factors.

**Example.** For instance, $\mathit{Sprinkler} \mapsto \mathit{WetGrass}$ and $\mathit{WetGrass} \mapsto \mathit{Rain}$ tend to imply $\mathit{Sprinkler} \mapsto \mathit{Rain}$.

## Dempster-Shafer theory

- ➤ Dempster-Shafer theory has been designed to deal with the distinction between **uncertainty** and **ignorance**.
- ➤ The *belief function* $\mathit{Bel}(X)$ gives the probability that the evidence obtained so far supports $X$.

**Example.** Consider flipping a coin under the following circumstances:

1. If the coin is doubted to be unfair (nothing can be assumed about its behavior), then $\mathit{Bel}(\mathit{Heads}) = 0$ and $\mathit{Bel}(\neg\mathit{Heads}) = 0$.
2. If the coin is fair with a certainty of 0.9, then we have $\mathit{Bel}(\mathit{Heads}) = 0.5 \times 0.9 = 0.45$ and $\mathit{Bel}(\neg\mathit{Heads}) = 0.45$

☞ We obtain *probability intervals* $[0, 1]$ and $[0.45, 0.55]$ for $\mathit{Heads}$.

## Fuzzy Logic

- **Fuzzy set theory** is about specifying how well an object satisfies a vague description rather than uncertainty.

  **Example.** For instance, a statement like “Mika Myllylä is tall” can be assigned a truth value between 0 and 1 (even if it is known how tall he is).

- The *fuzzy truth* of complex sentences is defined truth-functionally:

$$T(\phi \wedge \psi) = \min(T(\phi), T(\psi)),$$
$$T(\phi \vee \psi) = \max(T(\phi), T(\psi)), \text{ and}$$
$$T(\neg A) = 1 - T(A).$$

- Despite of semantic difficulties, fuzzy logic has been very successful in commercial applications involving *automated control*.

## SUMMARY

- **Conditionalindependence** information can be used for structuring and simplifying knowledge about an uncertain domain.
- **Belief networks** provide a natural way to represent conditional independence information.
- A belief network is a complete (and often also very compact) representation of the joint probability distribution.
- Belief networks support various reasoning models: causal, diagnostic, mixed, intercausal, . . .
- Efficient algorithms exist for belief networks that are topologically *polytrees*, but reasoning with belief networks is NP-hard in general.
- Probabilities can be estimated by **stochastic simulation**.

## QUESTIONS

- Build a belief network for the soccer domain.
  1. Choose appropriate variables for the description of the domain.
  2. Choose an ordering for the variables.
  3. Construct the actual belief network by (i) analyzing dependencies among variables and (ii) defining CPTs for each variable.
- Make both causal and diagnostic inferences using the network.