Bayes' Nets

CSC 548, Artificial Intelligence II

Bayesian Networks

- A simple graphical notation for conditional independence assertions and hence for compact specification of full joint distributions

- Syntax:
  - a set of nodes, one per variable
  - a directed acyclic graph (link $\approx$ "directly influences")
  - a conditional distribution for each node given its parents: $P(X_i \mid Parents(X_i))$

- In the simplest case, conditional distribution represented as a conditional probability table (CPT) giving the distribution over X_i for each combination of parent values

Example

- Topology of network encodes conditional independence assertions:

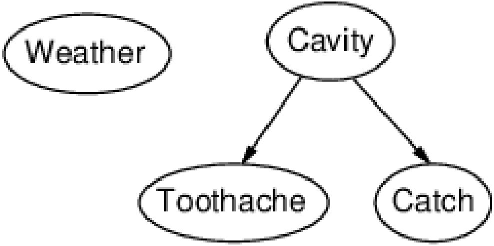

Figure 1: image

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

Example

- I am at work, neighbor John calls to say my alarm is ringing, but neighbor Mary does not call. Sometimes it is set off by minor earthquakes. Is there a burglar?
- Variables: $Burglar$, $Earthquake$, $Alarm$, $JonhCalls$, $MaryCalls$
- Network topology reflects "causal" knowledge
  - A burglar can set the alarm off
  - An earthquake can set the alarm off
  - The alarm causes Mary to call
  - The alarm causes John to call

Example Continued

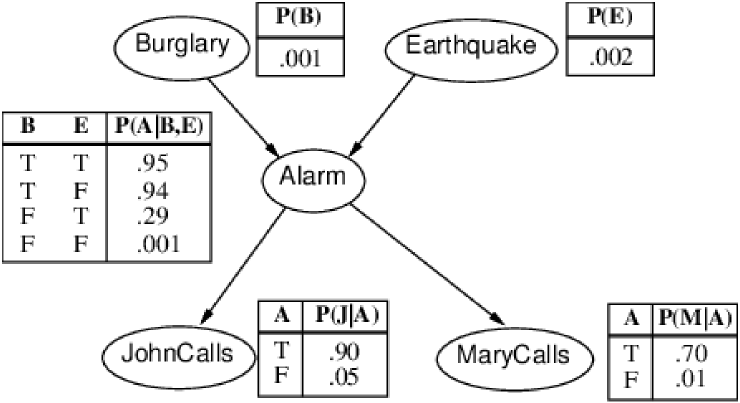

Figure 2: image

Compactness

- A CPT 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 $X_i = true$ (the number for $X_i = false$ is just $1 - p$)
- If each variable has no more than k parents, the complete network requires $\mathcal{O}(n \cdot 2^k)$ numbers
- That is, grows linearly with n, vs. $\mathcal{O}(2^n)$ for the full joint distribution
- For the burglary net, $1 + 1 + 4 + 2 + 2 = 10$ numbers (vs. $2^5 - 1 = 31$)

Global Semantics

- "Global" semantics defines the full joint distribution as the product of the local conditional distributions:
  $P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid parents(X_i))$
- Example, Burglary net:
  $P(j \wedge m \wedge a \wedge \neg b \wedge \neg e)$
  $= P(j \mid a)P(m \mid a)P(a \mid \neg b, \neg e)P(\neg b)P(\neg e)$
  $= 0.9 \times 0.7 \times 0.001 \times 0.999 \times 0.998$
  ≈ 0.00063

Local Semantics

- Local semantics: each node is conditionally independent of its nondescendants given its parent

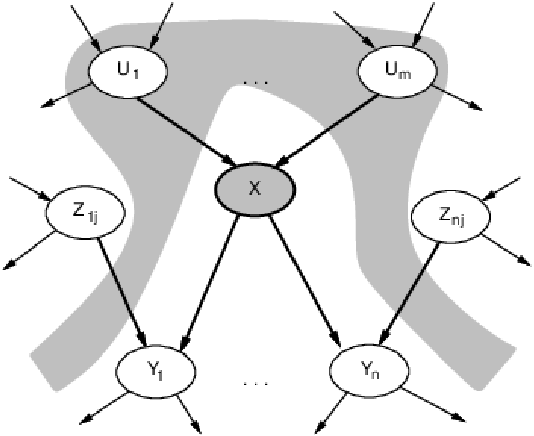

Figure 3: image

Markov Blanket

- Each node is conditionally independent of all others given its Markov blanket: parents + children + children's parents

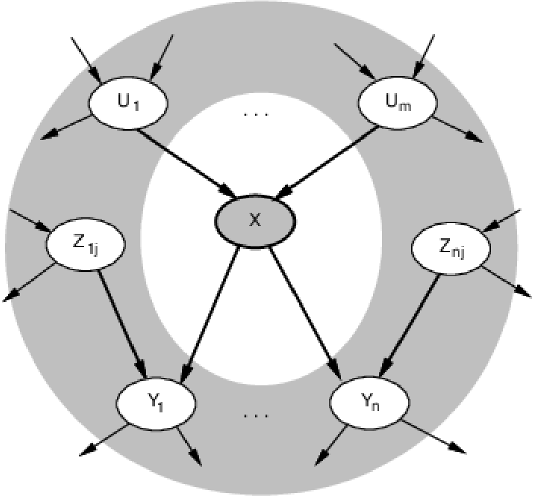

Figure 4: image

Constructing Bayesian Networks

- Need a method such that a series of locally testable assertions of conditional independence guarantees the required global semantics
- Choose an ordering of variables $X_1, \ldots, X_n$
- For $i = 1$ to n, add X_i to the network and select parents from $X_1, \ldots, X_{i-1}$ such that
  $P(X_i \mid Parents(X_i)) = P(X_i \mid X_1, \ldots, X_{i-1}$
- This choice of parents guarantees the global semantics:

$$\begin{aligned} P(X_1, \ldots, X_n) &= \prod_{i=1}^{n} P(X_i \mid X_1, \ldots, X_{i-1}) \\ &= \prod_{i=1}^{n} P(X_i \mid Parents(X_i)) \end{aligned}$$

Example

- Suppose we choose the ordering M, J, A, B, E

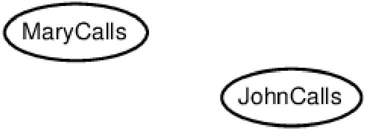

Figure 5: image

- $P(J \mid M) = P(J)$?

Example

- Suppose we choose the ordering M, J, A, B, E

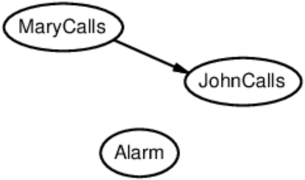

Figure 6: image

- $P(A \mid J, M) = P(A \mid J)$?
- $P(A \mid J, M) = P(A)$?

Example

- Suppose we choose the ordering M, J, A, B, E

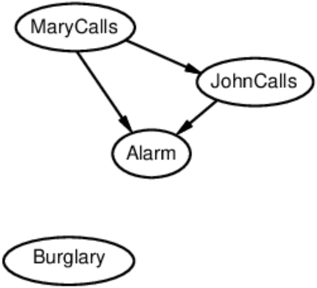

Figure 7: image

- $P(B \mid A, J, M) = P(B \mid A)$?
- $P(B \mid A, J, M) = P(B)$?

Example

- Suppose we choose the ordering M, J, A, B, E

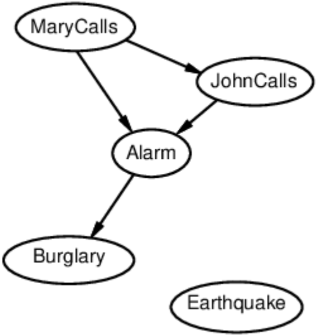

Figure 8: image

- $P(E \mid B, A, J, M) = P(E \mid A)$?
- $P(E \mid B, A, J, M) = P(E \mid A, B)$?

Example

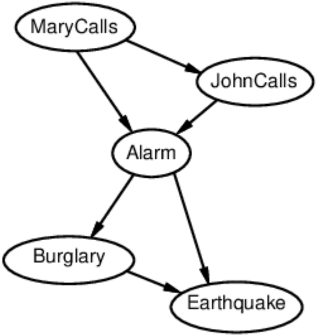

Figure 9: image

- Deciding conditional independence is hard in noncausal directions
- Assessing conditional probabilities is hard in noncausal directions

Example: Car Diagnosis

- Initial evidence: car will not start
- Testable variables (green), “broken, so fix it” variables (orange)
- Hidden variables (gray) ensure sparse structure, reduce parameters

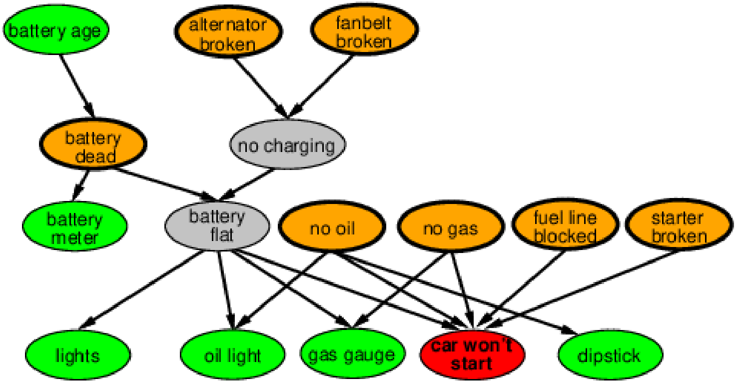

Figure 10: image

Example: Car Insurance

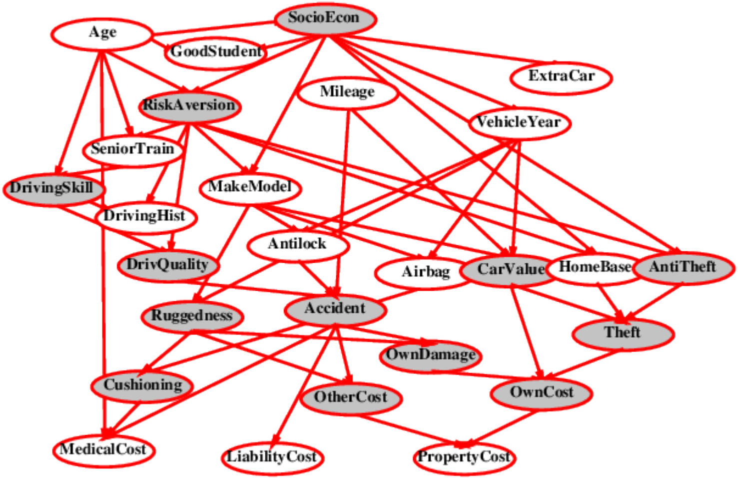

Figure 11: image

Compact Conditional Distributions

- CPT grows exponentially with number of parents
- CPT becomes infinite with continuous valued parent or child
- Solution: canonical distributions that are defined compactly
- Deterministic nodes are the simplest case:
  $X = f(Parents(X))$ for some function f
- Examples:
  - $NorthAmerican \leftrightarrow Canadian \vee US \vee Mexican$
  - $\frac{\partial Level}{\partial t} =$ inflow $+$ precipitation - outflow - evaporation

Compact Conditional Distributions

- Noisy-OR distributions model multiple noninteracting causes
  - Parents $U_1, \ldots, U_k$ include all causes
  - Independent failure probability q_i for each cause alone
    $P(X \mid U_1, \ldots, U_j, \neg U_{j+1}, \ldots, \neg U_k) = 1 - \prod_{i=1}^{j} q_i$

Cold	*Flu*	*Malaria*	$P(Fever)$	$P(\neg Fever)$
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	$0.2 = 0.2 \times 0.1$
T	F	F	0.4	**0.6**
T	F	T	0.4	$0.06 = 0.6 \times 0.1$
T	T	F	0.88	$0.12 = 0.6 \times 0.2$
T	T	T	0.988	$0.12 = 0.6 \times 0.2 \times 0.1$

- Number of parameters linear in number of parents

Hybrid (Discrete + Continuous) Networks

- Discrete (*Subsidy*? and *Buys*?); continuous (*Harvest* and *Cost*)

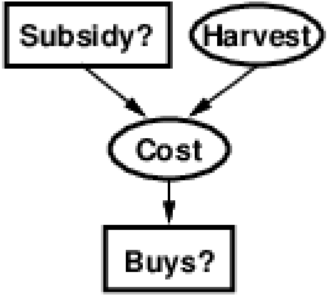

Figure 12: image

- Option 1: discretization – possibly large errors large CPTs
- Option 2: finitely parameterized canonical families
- Continuous variable, discrete + continuous parents (e.g. *Cost*)
- Discrete variable, continuous parents (e.g. *Buys*?)

Continuous Child Variables

- Need one conditional density function for child variable given continuous parents, for each possible assignment to discrete parents
- Most common is the linear Gaussian model, for example

$$
\begin{aligned}
& P(Cost = c \mid Harvest = h, Subsidy? = true) \\
& = \mathcal{N}(a_t h + b_t, \sigma_t)(c) \\
& = \frac{1}{\sigma_t \sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{c - (a_t h + b_t)}{\sigma_t}\right)^2\right)
\end{aligned}
$$

- Mean *Cost* varies linearly with *Harvest*, variance is fixed
- Linear variation is unreasonable over the full range but works if the likely range of *Harvest* is narrow

Continuous Child Variables

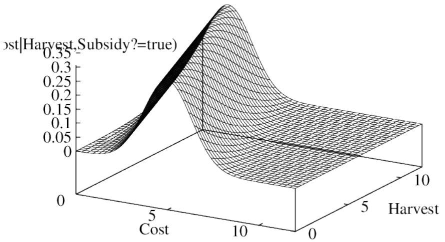

Figure 13: image

- All continuous network with linear Gaussian distributions $\rightarrow$ full joint distribution is a multivariate Gaussian
- Discrete + continuous linear Gaussian network is a conditional Gaussian network, that is, a multivariate Gaussian over all continuous variables for each combination of discrete values

Discrete Variable with Continuous Parents

- Probability of *Buys*? given *Cost* should be a "soft" threshold

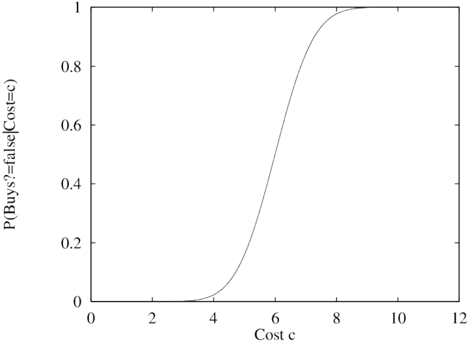

Figure 14: image

- Probit distribution uses integral of Gaussian

$$\Phi(x) = \int_{-\infty}^{x} \mathcal{N}(0,1)(x)dx$$

Why the Probit?

- It is sort of the right shape
- Can view as hard threshold whose location is subject to noise

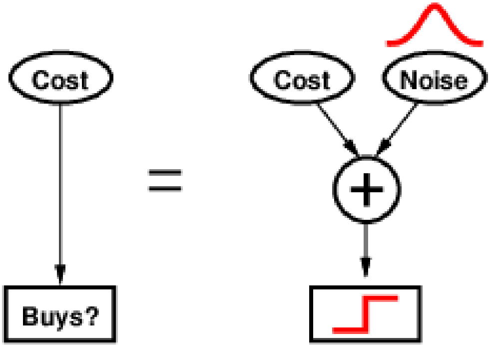

Figure 15: image

Discrete Variable Continued

- Sigmoid (or logit) distribution also used in neural networks:

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

- Sigmoid has a similar shape to probit, but much longer tails:

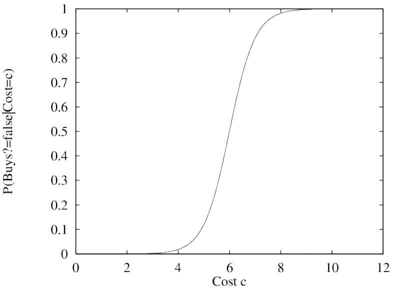

Figure 16: image

Inference Tasks

- Simple queries: compute posterior marginal $P(X_i \mid E = e)$
- Conjunctive queries:
  $P(X_i, X_j \mid E = e) = P(X_i \mid E = e)P(X_j \mid X_i, E = e)$
- Optimal decisions: decision networks include utility information; probabilistic inference required for
  $P(outcome \mid action, evidence)$
- 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

- Slightly intelligent way to sum out variables from the joint without actually constructing its explicit representation
- Simple query on the burglary network:

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

- Recursive depth-first search enumeration: $\mathcal{O}(n)$ space, $\mathcal{O}(d^n)$ time

Enumeration Algorithm

function $\text{ENUMERATION-ASK}(X, e, bn)$
  $Q(X) \leftarrow$ a distribution over X, initially empty
  for each value x_i of X **do**
    extend e with value x_i for X
    $Q(x_i) \leftarrow \text{ENUMERATE-ALL}(\text{VARS}[bn], e)$
  return $\text{NORMALIZE}(Q(X))$

function $\text{ENUMERATE-ALL}(vars, e)$
  if $\text{EMPTY?}(vars)$ **then**
    return 1.0
  $Y \leftarrow \text{FIRST}(vars)$
  if Y has value y in e **then**
    return $P(y \mid Pa(Y) \times \text{ENUMERATE-ALL}(\text{REST}(vars), e)$
  else
    return $\sum_y P(y \mid Pa(Y)) \times$
$\text{ENUMERATE-ALL}(\text{REST}(vars), e_y)$

Evaluation Tree

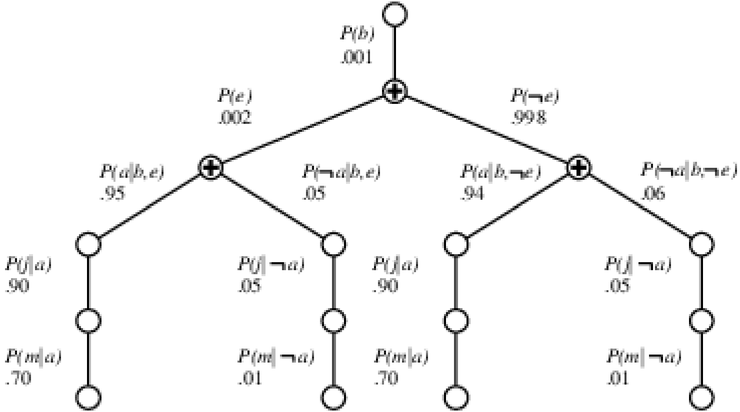

Figure 17: image

- Enumeration is inefficient: repeated computation, for example, $P(j \mid a)P(m \mid a)$ is computed for each value of e

Inference by Variable Elimination

- Variable elimination: carry out summations right-to-left, storing intermediate results (factors) to avoid recomputation

$$
\begin{aligned}
&P(B \mid j, m) \\
&= \alpha P(B) \sum_e P(e) \sum_a P(a \mid B, e) P(j \mid a) P(m \mid a) \\
&= \alpha P(B) \sum_e P(e) \sum_a P(a \mid B, e) P(j \mid a) f_M(a) \\
&= \alpha P(B) \sum_e P(e) \sum_a P(a \mid B, e) f_J(a) f_M(a) \\
&= \alpha P(B) \sum_e P(e) \sum_a f_A(a, b, e) f_J(a) f_M(a) \\
&= \alpha P(B) \sum_e P(e) f_{\bar{A}JM}(b, e) \\
&= \alpha P(B) f_{\bar{E}\bar{A}JM}(b) \\
&= \alpha f_B(b) f_{\bar{E}\bar{A}JM}(b)
\end{aligned}
$$

Variable Elimination: Basic Operations

- Summing out a variable from a product of factors: move any constant factors outside the summation and add up submatrices in pointwise product of remaining factors

$$\begin{aligned}&\sum_x f_1 \times \cdots \times f_k \\ &= f_1 \times \cdots \times f_i \sum_x f_{i+1} \times \cdots \times f_k \\ &= f_1 \times \cdots f_i \times f_{\bar{X}}\end{aligned}$$

  assuming $f_1, \ldots, f_i$ do not depend on X

- Pointwise product of factors f_1 and f_2:
  $f_1(x_1, \ldots, x_j, y_1, \ldots, y_k) \times f_2(y_1, \ldots, y_k, z_1, \ldots, z_l) =$
  $f(x_1, \ldots, x_j, y_1, \ldots, y_k, z_1, \ldots, z_l)$
- Example: $f_1(a, b) \times f_2(b, c) = f(a, b, c)$

Variable Elimination Algorithm

```
function ELIMINATION-ASK(X, e, bn)
    factors ← []
    vars ← REVERSE(VARS([bn])
    for each var in vars do
        factors ← [ MAKE-FACTOR(var, e) | factors]
        if var is a hidden variable then
            factors ← SUM-OUT(vars, factors)
    return NORMALIZE(POINTWISE-PRODUCT(factors))
```

Irrelevant Variables

- Consider the query $P(JohnCalls \mid Burglary = true)$
  $P(J \mid b) =$
  $\alpha P(b) \sum_e P(e) \sum_a P(a \mid b, e) P(J \mid a) \sum_m P(m \mid a)$
- The sum over m is identically 1; M is irrelevant to the query
- Theorem: Y is irrelevant unless $Y \in Ancestors(\{X\} \cup E)$
- Here, $X = JohnCalls, E = \{Burglary\}$, and $Ancestors(\{X\} \cup E) = \{Alarm, Earthquake\}$ so $MaryCalls$ is irrelevant

Irrelevant Variables

- Definition: moral graph of Bayes net – marry all parents and drop arrows
- Definition: A is *m-separated* from B by C iff separated by C in the moral graph
- Theorem: Y is irrelevant if m-separated from X by E
- Example: For $P(JohnCalls \mid Alarm = true)$, both *Burglary* and *Earthquake* are irrelevant

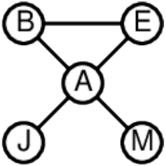

Figure 18: image

Complexity of Exact Inference

- Singly connected networks (or polytrees):
  - any two nodes are connected by at most one (undirected) path
  - time and space cost of variable elimination are $\mathcal{O}(d^k n)$
- Multiply connected networks:
  - can reduce to 3SAT to exact inference $\Rightarrow$ NP-hard
  - equivalent to counting 3SAT models $\Rightarrow$ P-complete

Inference by Stochastic Simulation

- Basic idea:
  1. Draw N samples from a sampling distribution S
  2. Compute an approximate posterior probability $\hat{P}$
  3. Show this converges to the true probability P
- Outline
  - Sampling from an empty network
  - Rejection sampling: reject samples that disagree with the evidence
  - Likelihood weighting (LW): use evidence to weight samples
  - Markov chain Monte Carlo (MCMC): sample from a stochastic process whose stationary distribution is the true posterior

Sampling from an Empty Network

function PRIOR-SAMPLE(*bn*)
  $x \leftarrow$ an event with n elements
  for $i = 1$ to n **do**
    $x_i \leftarrow$ a random sample from $P(X_i \mid \mathit{parents}(X_i))$
      given the values of $\mathit{Parents}(X_i)$ in x

Example

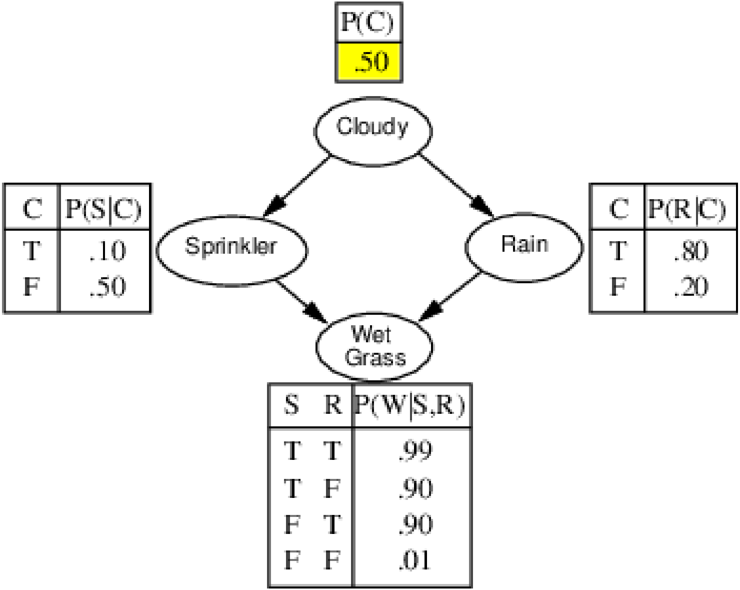

Figure 19: image

Example

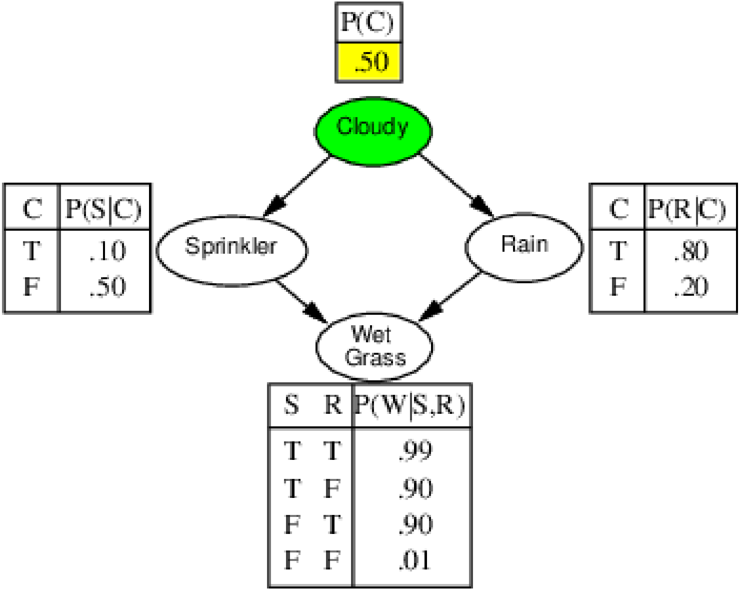

Figure 20: image

Example

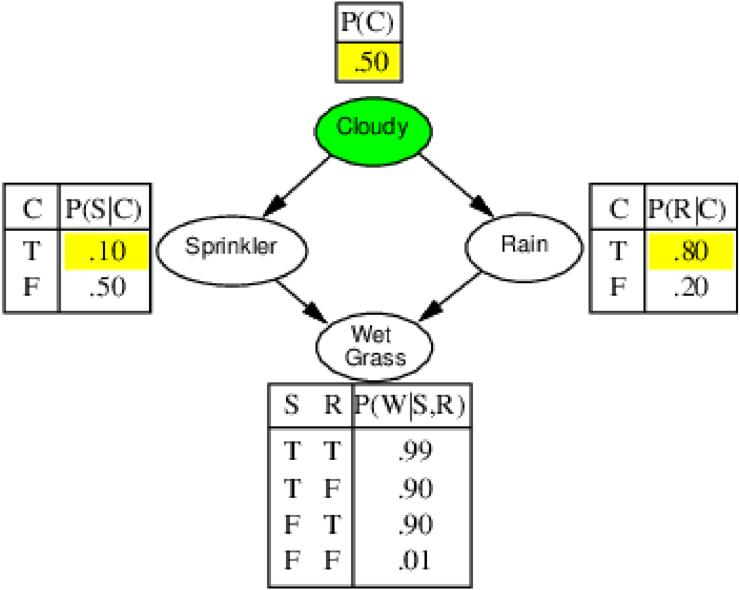

Figure 21: image

Example

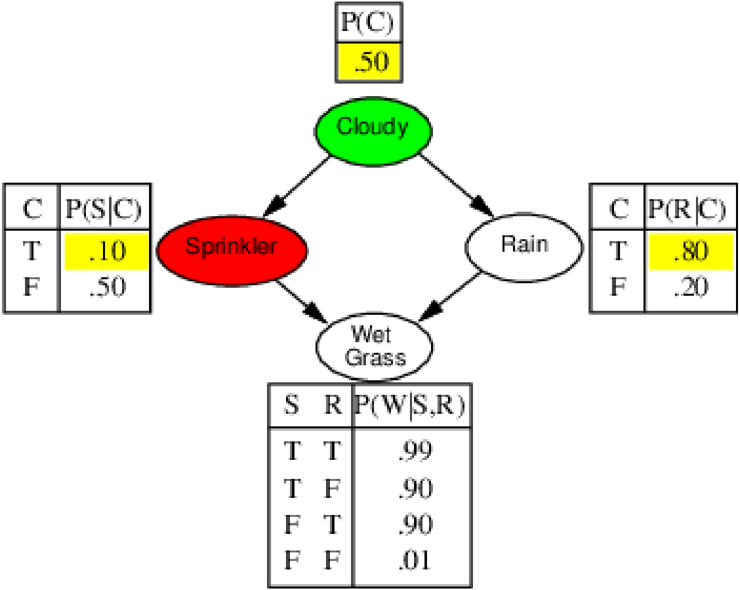

Figure 22: image

Example

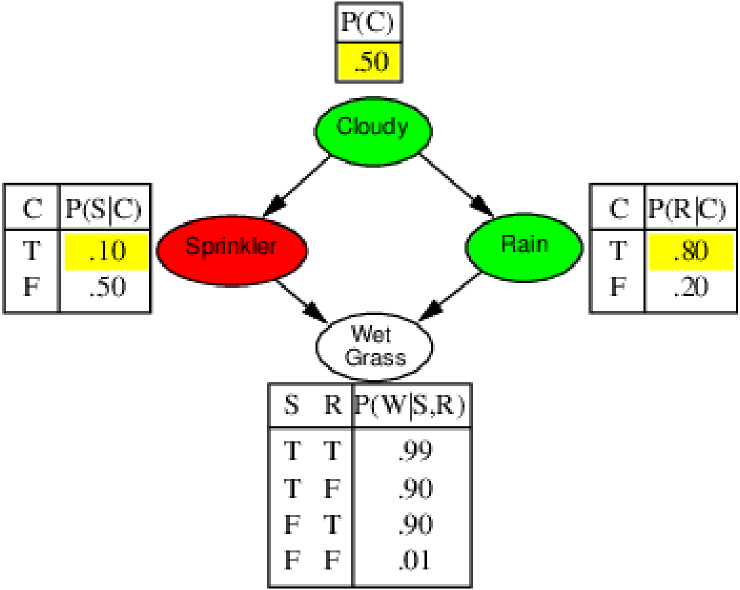

Figure 23: image

Example

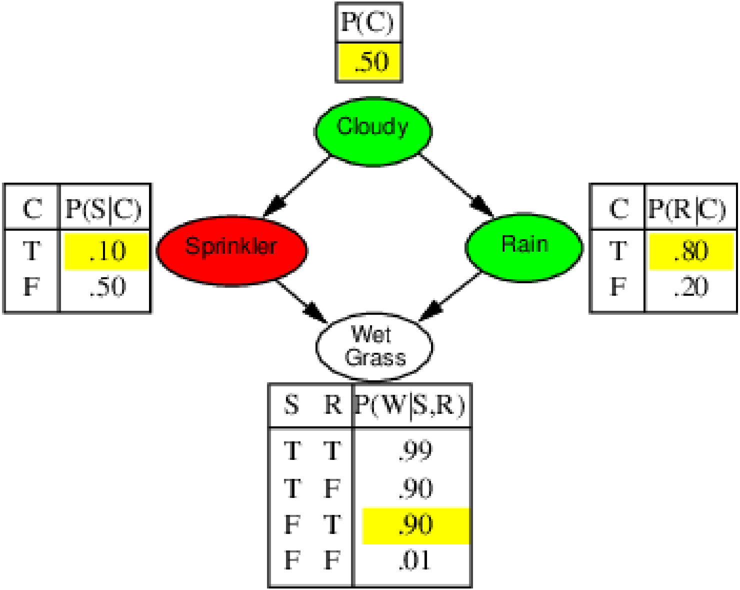

Figure 24: image

Example

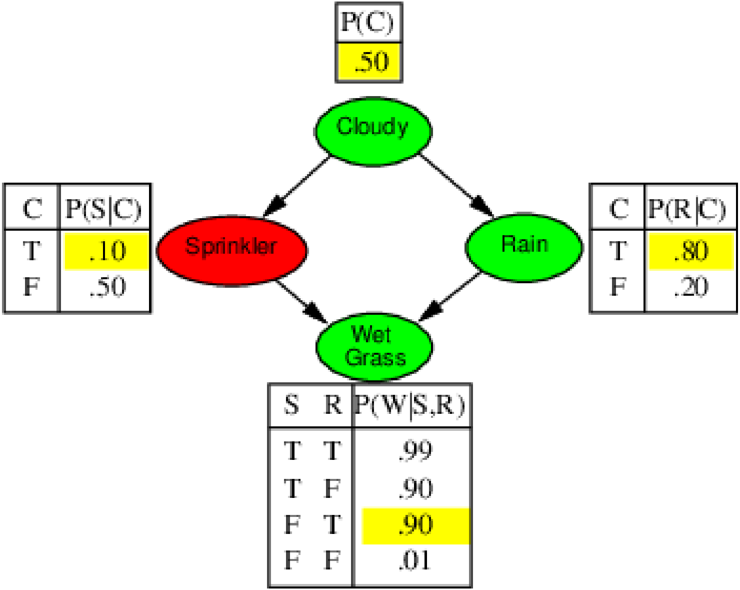

Figure 25: image

Sampling from an Empty Network Continued

- Probability the Prior-Sample generates a particular event

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

  that is, the true prior probability

- Let $N_{PS}(x_i, \ldots, x_n)$ be the number of samples generated for event $x_1, \ldots, x_n$, then we have

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

- Shorthand: $\hat{P}(x_1, \ldots, x_n) \approx P(x_1, \ldots, x_n)$

Rejection Sampling

- $\hat{P}(X \mid e)$ estimated from samples agreeing with e
- Example: estimate $P(Rain \mid Sprinkler = true)$ using 100 samples: 27 samples have $Sprinkler = true$ and of these 8 have $Rain = true$ and 19 have $Rain = false$.
- $\hat{P}(Rain \mid Sprinkler = true) = \textsc{Normalize}(\langle 8, 19 \rangle) = \langle 0.296, 0.704 \rangle$

Analysis of Rejection Sampling

$$
\begin{aligned}
\hat{P}(X \mid e) &= \alpha N_{PS}(X, e) \\
&= \frac{N_{PS}(X, e)}{N_{PS}(e)} \\
&\approx \frac{P(X, e)}{P(e)} \\
&= P(X \mid e)
\end{aligned}
$$

- Hence rejection sampling returns consistent posterior estimates
- Problem: hopelessly expensive if $P(e)$ is small
- $P(e)$ drops off exponentially with number of evidence variables

Likelihood Weighting

- Idea: fix evidence variables, sample only nonevidence variables, and weight each sample by the likelihood it accords the evidence

Likelihood Weighting Example

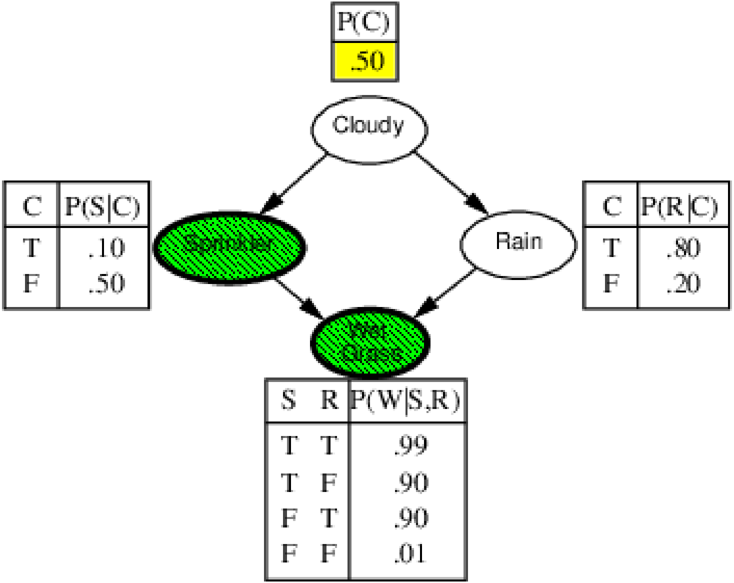

Figure 26: image

Likelihood Weighting Example

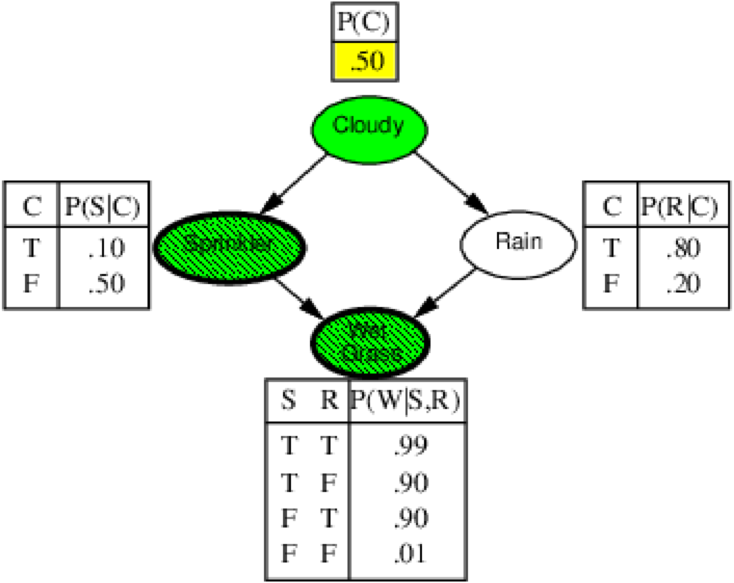

Figure 27: image

Likelihood Weighting Example

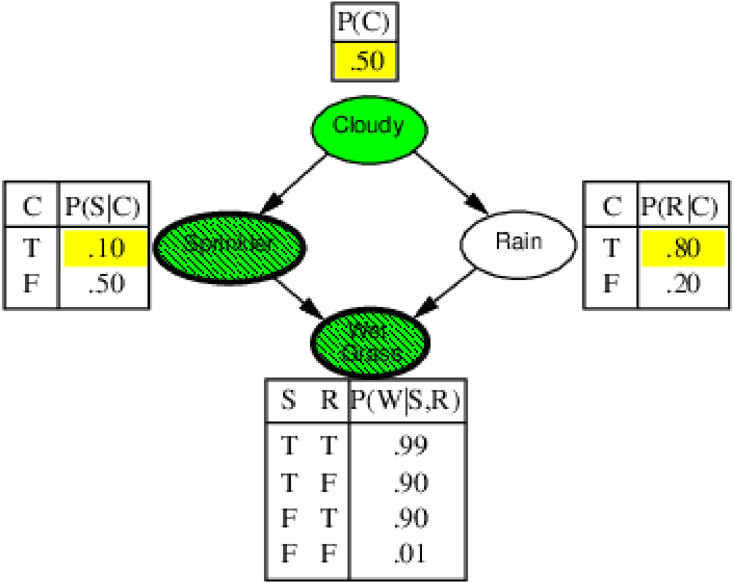

Figure 28: image

Likelihood Weighting Example

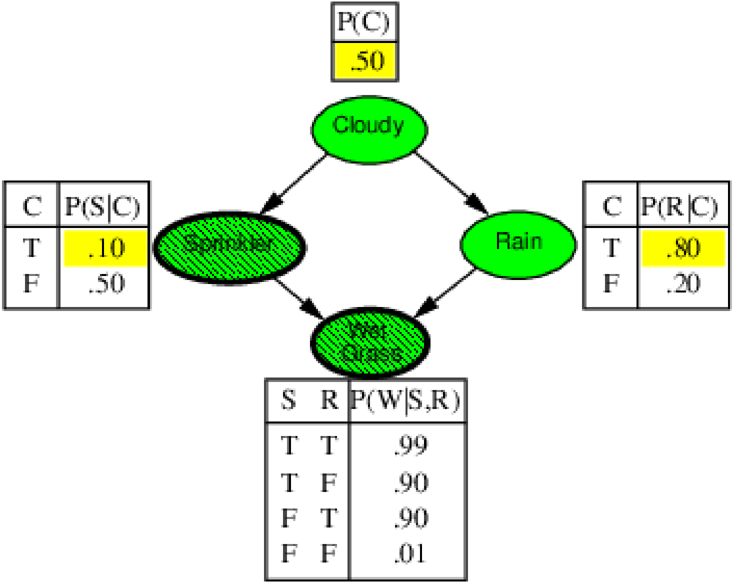

Figure 29: image

Likelihood Weighting Example

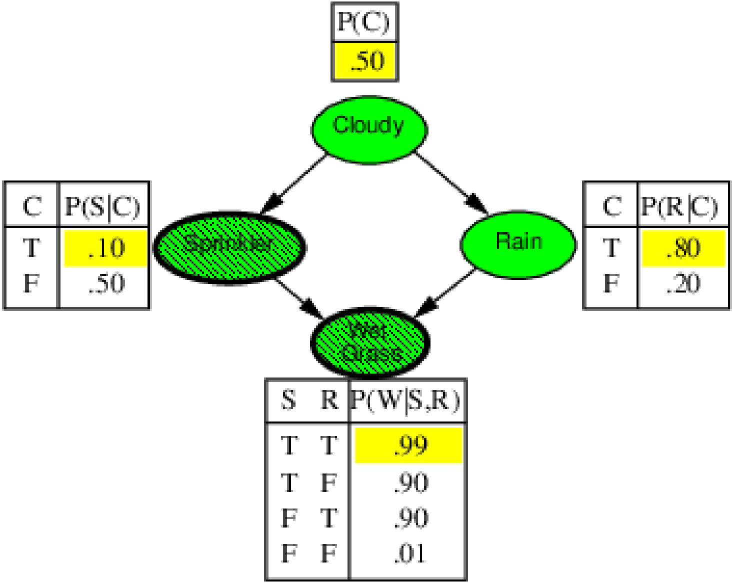

Figure 30: image

Likelihood Weighting Analysis

- Sampling probability for WEIGHTED-SAMPLE is
  $S_{WS}(z, e) = \prod_{i=1} lP(z_i \mid parents(Z_i))$
- Note: pays attention to evidence in *ancestors* only –
  somewhere "in between" prior and posterior distribution
- Weight for a given sample z, e is
  $w(z, e) = \prod_{i=1}^{m} P(e_i \mid parents(E_i))$
- Weighted sampling probability is

$$\begin{aligned} & S_{WS}(z, e) w(z, e) \\ & \quad = \prod_{i=1}^{l} P(z_i \mid parents(Z_i)) \prod_{i=1} mP(e_i \mid parents(E_i)) \\ & \quad = P(z, e) \text{ (by standard global semantics of network)} \end{aligned}$$

- Hence likelihood weighting returns consistent estimates but
  performance still degrades with many evidence variables
  because few samples have nearly all the total weight

Approximate Inference Using MCMC

- "State" of network is current assignment to all variables
- Generate next state by sampling one variable given Markov blanket

The Markov Chain

- With $\mathit{Sprinkler} = \mathit{true}, \mathit{WetGrass} = \mathit{true}$, there are four states:

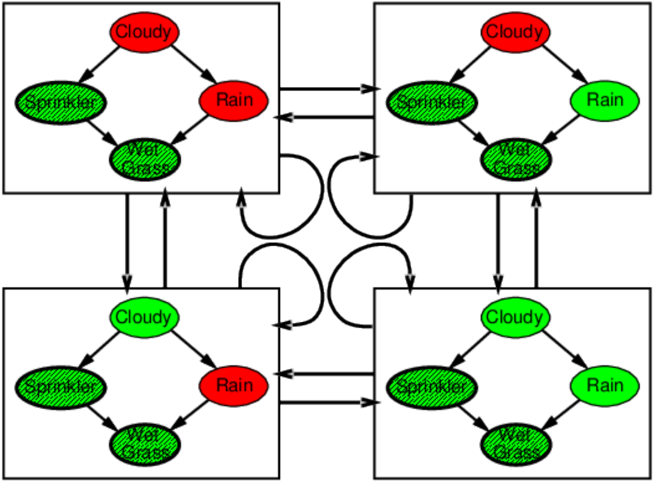

Figure 31: image

MCMC Example Continued

- Estimate $P(\mathit{Rain} \mid \mathit{Sprinkler} = \mathit{true}, \mathit{WetGrass} = \mathit{true})$
- Sample *Cloudy* or *Rain* given its Markov blanket, repeat; count the number of times *Rain* is true and false in the samples.
- For example, visit 100 states: 31 have $\mathit{Rain} = \mathit{true}$, 69 have $\mathit{Rain} = \mathit{false}$
  $\hat{P}(\mathit{Rain} \mid \mathit{Sprinkler} = \mathit{true}, \mathit{WetGrass} = \mathit{true}) =$
  $\textsc{Normalize}(\langle 31, 69 \rangle) = \langle 0.31, 0.69 \rangle$
- Theorem: chain approaches stationary distribution: long-run fraction of time spent in each state is exactly proportional to its posterior probability.

Markov Blanket Sampling

- Markov blanket of *Cloudy* is *Sprinkler* and *Rain*
- Markov blanket of *Rain* is *Cloudy*, *Sprinkler*, and *WetGrass*
- Probability given the Markov blanket is calculated as follows:

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

- Easily implemented in message-passing parallel systems
- Main computational problems:
  1. Difficult to tell if convergence has been achieved
  2. Can be wasteful if Markov blanket is large: $P(X_i \mid mb(X_i))$ will not change much (law of large numbers)

Summary

- Bayes nets provide a natural representation for (causally induced) conditional independence
- Topology $+$ CPTs $=$ compact representation for joint distribution
- Generally easy for (non)experts to construct
- Canonical distributions (e.g. noisy-OR) $=$ compact representation of CPTs
- Continuous variables $\rightarrow$ parameterized distributions
- Exact inference by variable elimination: NP-hard in general
- Approximate inference methods: likelihood weighting and Markov chain Monte-Carlo sampling