DM825

Introduction to Machine Learning

Lecture 12
Bayesian Networks

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

Outline

1. Conditional Independence

2. Inference in BN
   - Exact inference by enumeration
   - Exact inference by variable elimination
   - Exact inference by message passing
   - Approximate inference by stochastic simulation
   - Approximate inference by Markov chain Monte Carlo

Factorization

BN encode local conditional independences

$$\Pr(X_i \mid X_{-i}) = \Pr(X_i \mid \mathrm{pa}(X_i))$$

Joint probability factorization (the global semantics simplifies to):

$$\begin{aligned}\Pr(X_1, \ldots, X_n) &= \prod_{i=1}^{n} \Pr(X_i \mid X_1, \ldots, X_{i-1}) \quad \text{(chain rule)} \\ &= \prod_{i=1}^{n} \Pr(X_i \mid \mathrm{pa}(X_i)) \quad \text{(by construction)}\end{aligned}$$

Important Rules

When working with Bayesian Networks, the following probability theory rules are worth remembering:

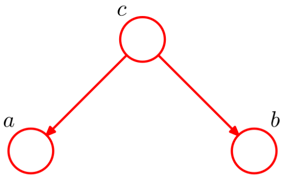
- Product rule
- Sum rule (marginalization)
- Bayes rule
- Factorization

Outline

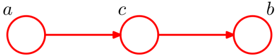
1. Conditional Independence

2. Inference in BN
   - Exact inference by enumeration
   - Exact inference by variable elimination
   - Exact inference by message passing
   - Approximate inference by stochastic simulation
   - Approximate inference by Markov chain Monte Carlo

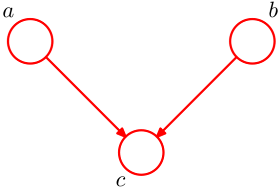
Three Examples



$$p(a,b,c) = p(a|c)p(b|c)p(c)$$
$$p(a,b) = \sum_c p(a,b,c) = \sum_c p(a|c)p(b|c)p(c)$$
$$p(a,b|c) = \frac{p(a,b,c)}{p(c)} = \frac{p(a|c)p(b|c)p(c)}{p(c)} = p(a|c)p(b|c)$$



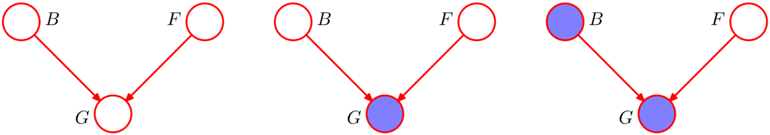
$$p(a,b,c) = p(a)p(c|a)p(b|c)$$
$$p(a,b) = \sum_c p(a,b,c) = \sum_c p(a)p(c|a)p(b|c)$$
$$p(a,b|c) = \frac{p(a,b,c)}{p(c)} = \frac{p(a)p(c|a)p(b|c)}{p(c)} = p(a|c)p(b|c)$$



$$p(a,b,c) = p(a)p(b)p(c|a,b)$$
$$p(a,b) = \sum_c p(a,b,c) = \sum_c p(a)p(b)p(c|a,b) = p(a)p(b)$$
$$p(a,b|c) = \frac{p(a,b,c)}{p(c)} = \frac{p(a)p(b)p(c|a,b)}{p(c)}$$

Example

$$
\begin{aligned}
p(B=1) &= 0.9 \\
p(F=1) &= 0.9
\end{aligned}
\qquad
\begin{aligned}
p(G=1|B=1,F=1) &= 0.8 \\
p(G=1|B=1,F=0) &= 0.2 \\
p(G=1|B=0,F=1) &= 0.2 \\
p(G=1|B=0,F=0) &= 0.1
\end{aligned}
$$



$p(F=0 \mid G=0) = ?$
$p(F=0 \mid G=0) \geq p(F=0)$
$p(F=0 \mid G=0, B=0) = ?$
$p(F=0 \mid G=0, B=0) \leq p(F=0 \mid G=0)$ (not conditional independent)
B explains away F

d-separation

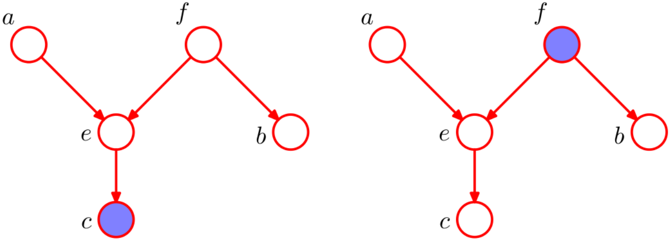
Definition (d-separation)

Two distinct variables A and B in a causal network are d-separated ("d" for "directed graph") if for all paths between A and B, there is an intermediate variable C (distinct from A and B) such that either

1. the connection is tail-to-tail or head-to-tail and C is instantiated or
2. the connection is head-to-head, and neither C nor any of C's descendants have received evidence.

If A and B are not d-separated, we call them d-connected.

If (1) then A indep. of B given C
If (2) then A indep. of B

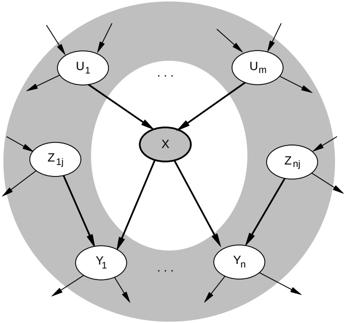

a
f
e
b
c
a
f
e
b
c

Markov Blanket

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

$$p(\mathbf{x}_i \mid \mathbf{x}_{j \neq i}) = \frac{p(\mathbf{x}_1, \ldots, \mathbf{x}_D)}{\sum_{\mathbf{x}_i} p(\mathbf{x}_1, \ldots, \mathbf{x}_D)}$$

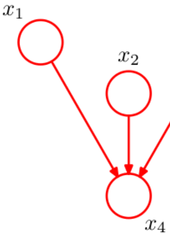
$$= \frac{\prod_k p(\mathbf{x}_k \mid \mathrm{pa}_k)}{\sum_{\mathbf{x}_i} \prod_k (\mathbf{x}_k \mid \mathrm{pa}_k)}$$

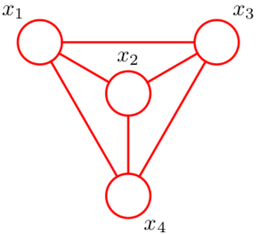


Algebra of Potentials

- General algebra of multiplication and marginalization on tables.
- For each outcome a variable has a corresponding state. States are mutually exclusive and exhaustive. The set of states associated with a variable A is denoted by $\mathrm{sp}(A) = (a_1, a_2, \ldots, a_n)$.
- Potential $\phi : sp(\mathcal{X}) \to \mathbb{R}$
- $\mathrm{dom}(\phi(A, B|C)) = \{A, B, C\}$ domain
- multiplication: $\phi_1\phi_2 : \mathrm{dom}(\phi_1\phi_2) = \mathrm{dom}(\phi_1) \cup \mathrm{dom}(\phi_2)$
- marginalization: $\sum_A \phi$ has domain $\mathrm{dom}(\phi) \setminus \{A\}$
- unit potential property: $\sum_A P(A \mid \mathcal{V}) = 1$
- projection for marginalization. Eg: if A and B are marginalized out of $\phi(A, B, C)$, we say ϕ is projected down to C

Moralization





Converting a directed graph into an undirected graph:
On the undirected:

$$p(\mathbf{x}) = \frac{1}{Z}\prod_C \psi_C(\mathbf{x}_c)$$

On the directed:

$$p(\mathbf{x}) = p(x_1)p(x_2)p(x_3)p(x_4 \mid x_1, x_2, x_3)$$

we introduce and edge for every arc and we marry parents

Outline

1. Conditional Independence

2. Inference in BN
    - Exact inference by enumeration
    - Exact inference by variable elimination
    - Exact inference by message passing
    - Approximate inference by stochastic simulation
    - Approximate inference by Markov chain Monte Carlo

Inference tasks

$\vec{e}$ assignment of values to some variables $\mathbf{E}$ (instantiation, evidence)

- Probability of Evidence $\Pr(\vec{e})$
  Example: probability that an individual will come out positive on both tests $\Pr(T1 = +\text{ve}, T2 = +\text{ve})$
  overall reliability of the system $\Pr(S = \text{avail})$
  related: node marginals query: probability $\Pr(x \mid e)$ for each X and for each of $x \in X$.

- Most Probable Explanation (MPE) $\arg\max_{\vec{q} \in \mathbf{Q}} \Pr(\vec{q} \mid \vec{e}), \mathbf{Q} = \overline{\mathbf{E}}$
  Example: find the most likely group, dissected by sex and condition, that will yield negative results for both tests ($\vec{e} = \{T_1 = \text{-ve}; T2 = \text{-ve}\}$ and $Q = \{S, C\}$)

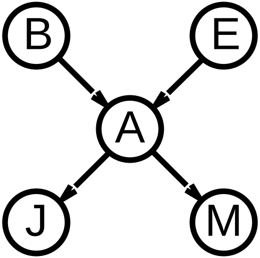
- Maximum a Posteriori Hypothesis (MAP) $\arg\max_{\vec{q} \in \mathbf{Q}} \Pr(\vec{q} \mid \vec{e}), \mathbf{Q} \subseteq \overline{\mathbf{E}}$
  Example: find most likely configuration of the two fans given that the system is unavailable ($\vec{e} = \{S = \text{unavail}\}, Q = \{F1, F2\}$).

Outline

Inference by enumeration

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

Simple query on the burglary network:

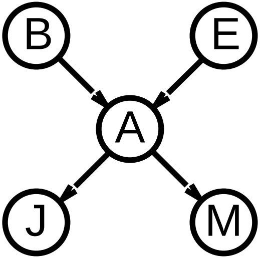


$$\begin{aligned}\Pr(B \mid j,m) &= \Pr(B,j,m)/P(j,m)\\ &= \alpha \Pr(B,j,m)\\ &= \alpha \textstyle\sum_e \sum_a \Pr(B,e,a,j,m)\end{aligned}$$

Inference by enumeration

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

Simple query on the burglary network:



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

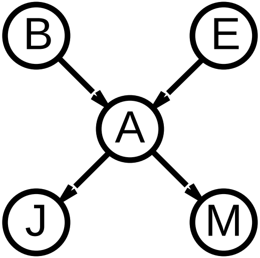
Rewrite full joint entries using product of CPT entries:

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

Inference by enumeration

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

Simple query on the burglary network:



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

Rewrite full joint entries using product of CPT entries:

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

Recursive depth-first enumeration: $O(n)$ space, $O(d^n)$ time

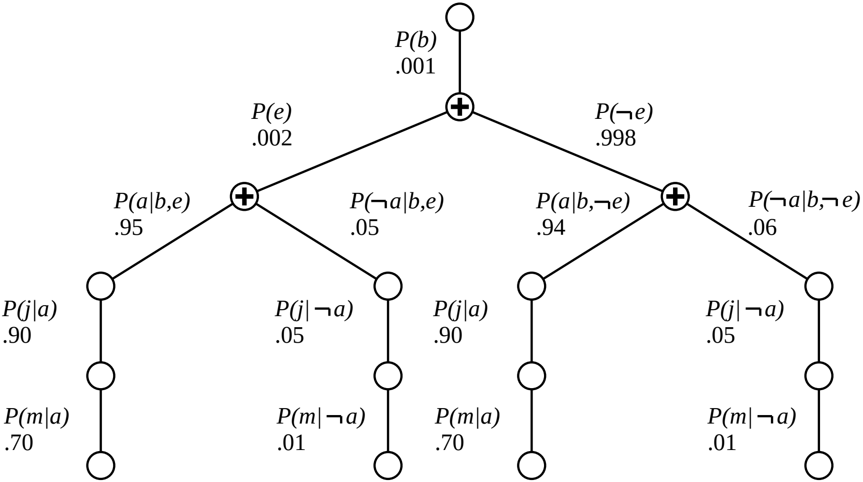
Enumeration algorithm

```
function Enumeration-Ask(X, e, bn) returns a distribution over X
   inputs: X,  the query variable
           e, observed values for variables E
           bn, a Bayesian network with variables {X} ∪ E ∪ Y

   Q(X) ← a distribution over X, initially empty
   for each value x_i of X do
       Q(x_i) ← Enumerate-All(bn.Vars, e ∪ {X = x_i})
   return Normalize(Q(X))
```

```
function Enumerate-All(vars, e) returns a real number
   if Empty?(vars) then return 1.0
   Y ← First(vars)
   if Y has value y in e
       then return P(y | parent(Y)) × Enumerate-All(Rest(vars), e)
   else return Σ_y P(y | parent(Y)) × Enumerate-All(Rest(vars), e ∪ {Y = y})
```

Evaluation tree



Enumeration is inefficient: repeated computation
e.g., computes $P(j \mid a)P(m \mid a)$ for each value of e

Outline

1. Conditional Independence

2. Inference in BN
    - Exact inference by enumeration
    - Exact inference by variable elimination
    - Exact inference by message passing
    - Approximate inference by stochastic simulation
    - Approximate inference by Markov chain Monte Carlo

Inference by variable elimination

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

$$
\begin{aligned}
&\Pr(B \mid j, m) \\
&= \alpha \underbrace{\Pr(B)}_{B} \sum_e \underbrace{P(e)}_{E} \sum_a \underbrace{\Pr(a \mid B, e)}_{A} \underbrace{P(j \mid a)}_{J} \underbrace{P(m \mid a)}_{M} \\
&= \alpha \Pr(B) \sum_e P(e) \sum_a \Pr(a \mid B, e) P(j \mid a) f_M(a) \\
&= \alpha \Pr(B) \sum_e P(e) \sum_a \Pr(a \mid B, e) f_J(a) f_M(a) \\
&= \alpha \Pr(B) \sum_e P(e) \sum_a f_A(a, b, e) f_J(a) f_M(a) \\
&= \alpha \Pr(B) \sum_e P(e) f_{\bar{A}JM}(b, e) \text{ (sum out } A) \\
&= \alpha \Pr(B) f_{\bar{E}\bar{A}JM}(b) \text{ (sum out } E) \\
&= \alpha f_B(b) \times f_{\bar{E}\bar{A}JM}(b)
\end{aligned}
$$

Variable elimination: Basic operations

Summing out a variable from a product of factors:

1. move any constant factors outside the summation:

2. add up submatrices in pointwise product of remaining factors:

Variable elimination: Basic operations

Summing out a variable from a product of factors:

1. move any constant factors outside the summation:

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

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

2. add up submatrices in pointwise product of remaining factors:

Variable elimination: Basic operations

Summing out a variable from a product of factors:

1. move any constant factors outside the summation:

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

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

2. add up submatrices in pointwise product of remaining factors:

   Eg: pointwise product of f_1 and f_2:

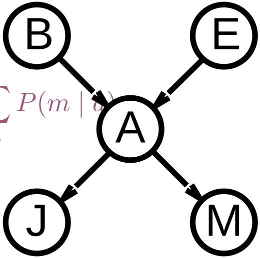
   $$f_1(x_1, \ldots, x_j, y_1, \ldots, y_k) \times f_2(y_1, \ldots, y_k, z_1, \ldots, z_l) \\ = f(x_1, \ldots, x_j, y_1, \ldots, y_k, z_1, \ldots, z_l)$$

   E.g., $f_1(a, b) \times f_2(b, c) = f(a, b, c)$

Irrelevant variables

Consider the query $P(JohnCalls \mid Burglary = true)$

$$P(J \mid b) = \alpha P(b) \sum_e P(e) \sum_a P(a \mid b, e) P(J \mid a) \sum_m P(m \mid a)$$



Sum over m is identically 1; M is **irrelevant** to the query

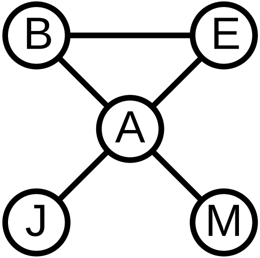
Irrelevant variables contd.

Defn: moral graph of DAG Bayes net: marry all parents and drop arrows
Defn: $\mathbf{A}$ is m-separated from $\mathbf{B}$ by $\mathbf{C}$ iff separated by $\mathbf{C}$ in the moral graph

Theorem

Y is irrelevant if m-separated from X by E

For $P(JohnCalls \mid Alarm = true)$, both $Burglary$ and $Earthquake$ are irrelevant



Complexity of exact inference

Singly connected networks (or polytrees):

– any two nodes are connected by at most one (undirected) path

– time and space cost (with variable elimination) are $O(d^k n)$, k number of parents

– hence time and space cost are linear in n and k bounded by a constant

Multiply connected networks:

– can reduce 3SAT to exact inference $\Longrightarrow$ NP-hard

– equivalent to **counting** 3SAT models $\Longrightarrow$ #P-complete

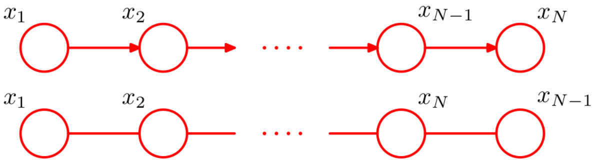
Outline

1. Conditional Independence

2. Inference in BN
   - Exact inference by enumeration
   - Exact inference by variable elimination
   - Exact inference by message passing
   - Approximate inference by stochastic simulation
   - Approximate inference by Markov chain Monte Carlo

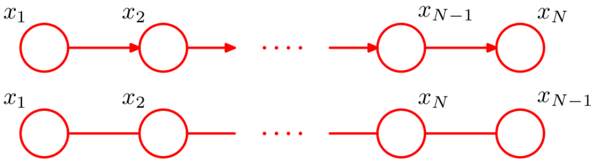
If we want the posteriror of each variable then even if poly tree $O(n)O(n)$
Join tree reduce the complexity to $O(n)$
Idea: join individual nodes such that the resulting network is a polytree

Chains



We want to infer maringal of x_j with no evidence

Chains



We want to infer maringal of x_j with no evidence

$$p(x_n) = \frac{1}{Z}$$

$$\underbrace{\left[\sum_{x_{n-1}} \psi_{n-1,n}(x_{n-1}, x_n) \cdots \left[\sum_{x_2} \psi_{2,3}(x_2, x_3) \left[\sum_{x_1} \psi_{1,2}(x_1, x_2)\right]\right] \cdots \right]}_{\mu_\alpha(x_n)}$$

$$\underbrace{\left[\sum_{x_{n+1}} \psi_{n,n+1}(x_n, x_{n+1}) \cdots \left[\sum_{x_N} \psi_{N-1,N}(x_{N-1}, x_N)\right] \cdots \right]}_{\mu_\beta(x_n)}. \qquad (8.52)$$

Outline

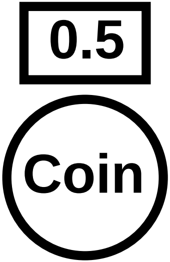
1. Conditional Independence

2. Inference in BN
   - Exact inference by enumeration
   - Exact inference by variable elimination
   - Exact inference by message passing
   - Approximate inference by stochastic simulation
   - Approximate inference by Markov chain Monte Carlo

Inference by stochastic simulation

Basic idea:

- Draw N samples from a sampling distribution S
- Compute an approximate posterior probability $\hat{P}$
- Show this converges to the true probability P



Outline:

– Sampling from an empty network
– Rejection sampling: reject samples disagreeing with evidence
– Likelihood weighting: use evidence to weight samples
– Markov chain Monte Carlo (MCMC): sample from a stochastic process whose stationary distribution is the true posterior

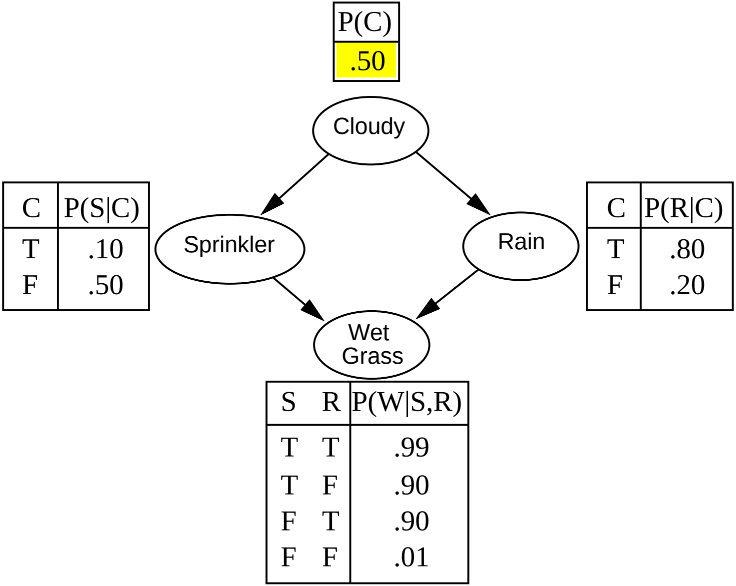
Sampling from an empty network

function Prior-Sample(*bn*) **returns** an event sampled from *bn*
inputs: *bn*, a belief network specifying joint distribution $\Pr(X_1, \ldots, X_n)$

$\mathbf{x} \leftarrow$ an event with n elements
for $i = 1$ **to** n **do**
$x_i \leftarrow$ a random sample from $\Pr(X_i \mid parents(X_i))$
given the values of $\mathrm{pa}(X_i)$ in $\mathbf{x}$
return $\mathbf{x}$

Ancestor sampling

Example



P(C)
.50

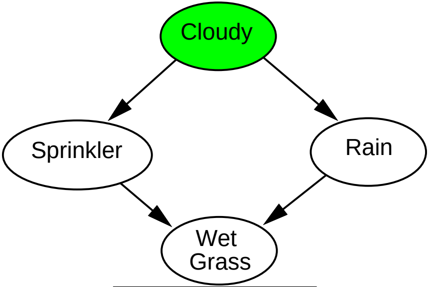
C	P(S|C)
T	.10
F	.50

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

Example

P(C)
.50

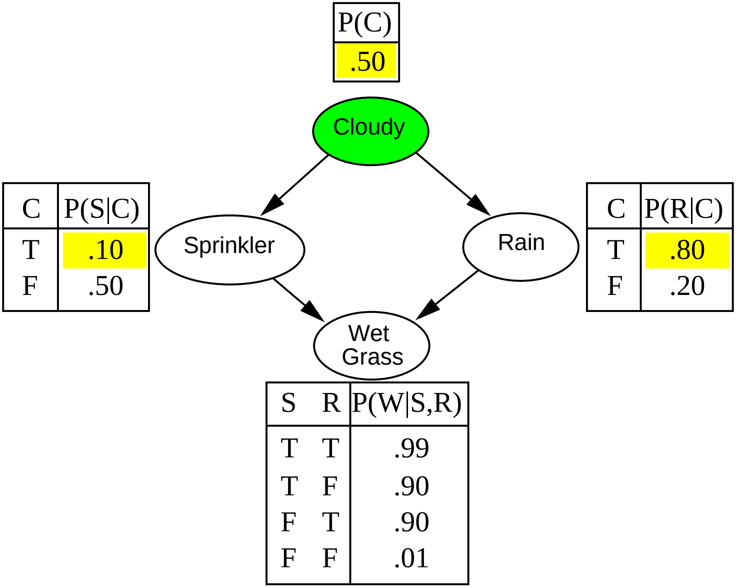


C	P(S|C)
T	.10
F	.50

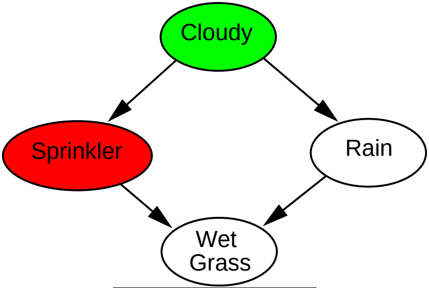
C	P(R|C)
T	.80
F	.20

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

Example


P(C)
.50
Cloudy
C
P(S|C)
T
.10
F
.50
Sprinkler
Rain
C
P(R|C)
T
.80
F
.20
Wet
Grass
S
R
P(W|S,R)
T T .99
T F .90
F T .90
F F .01

Example



P(C)
.50

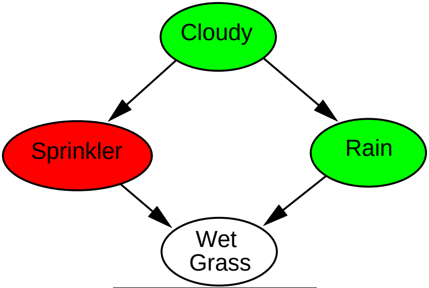
C	P(S|C)
T	.10
F	.50

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

Example

P(C)
.50



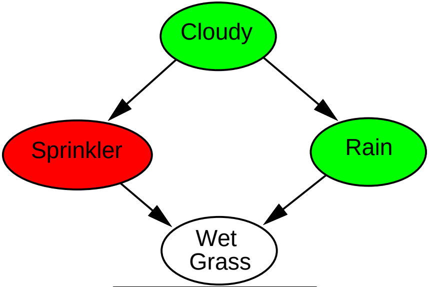
C	P(S|C)
T	.10
F	.50

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

Example

P(C)
.50

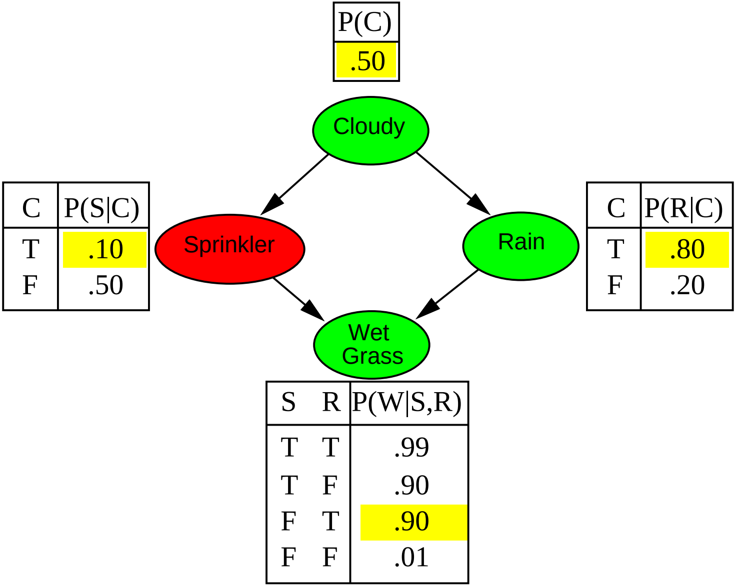


C	P(S|C)
T	.10
F	.50

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

Example



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

Sampling from an empty network contd.

Probability that PriorSample generates a particular event

$$S_{PS}(x_1 \dots x_n) = P(x_1 \dots x_n)$$

i.e., the true prior probability

Sampling from an empty network contd.

Probability that PriorSample generates a particular event

$$S_{PS}(x_1 \ldots x_n) = P(x_1 \ldots x_n)$$

i.e., the true prior probability

E.g., $S_{PS}(t, f, t, t) = 0.5 \times 0.9 \times 0.8 \times 0.9 = 0.324 \to P(t, f, t, t)$

Sampling from an empty network contd.

Probability that PriorSample generates a particular event

$$S_{PS}(x_1 \dots x_n) = P(x_1 \dots x_n)$$

i.e., the true prior probability

E.g., $S_{PS}(t,f,t,t) = 0.5 \times 0.9 \times 0.8 \times 0.9 = 0.324 \to P(t,f,t,t)$

Proof: Let $N_{PS}(x_1 \dots x_n)$ be the number of samples generated for event $x_1, \dots, x_n$. Then we have

$$\begin{aligned} \lim_{N\to\infty} \hat{P}(x_1,\dots,x_n) &= \lim_{N\to\infty} N_{PS}(x_1,\dots,x_n)/N \\ &= S_{PS}(x_1,\dots,x_n) \\ &= \prod_{i=1}^{n} P(x_i|parents(X_i)) = P(x_1 \dots x_n) \end{aligned}$$

Sampling from an empty network contd.

Probability that PriorSample generates a particular event

$$S_{PS}(x_1 \dots x_n) = P(x_1 \dots x_n)$$

i.e., the true prior probability

E.g., $S_{PS}(t,f,t,t) = 0.5 \times 0.9 \times 0.8 \times 0.9 = 0.324 \to P(t,f,t,t)$

Proof: Let $N_{PS}(x_1 \dots x_n)$ be the number of samples generated for event $x_1, \dots, x_n$. Then we have

$$\begin{aligned}
\lim_{N\to\infty} \hat{P}(x_1, \dots, x_n) &= \lim_{N\to\infty} N_{PS}(x_1, \dots, x_n)/N \\
&= S_{PS}(x_1, \dots, x_n) \\
&= \prod_{i=1}^{n} P(x_i | parents(X_i)) = P(x_1 \dots x_n)
\end{aligned}$$

$\leadsto$ That is, estimates derived from PriorSample are consistent
Shorthand: $\hat{P}(x_1, \dots, x_n) \approx P(x_1 \dots x_n)$

Rejection sampling

$\hat{\Pr}(X|\mathbf{e})$ estimated from samples agreeing with $\mathbf{e}$

Rejection sampling

$\hat{\Pr}(X|\mathbf{e})$ estimated from samples agreeing with $\mathbf{e}$

function Rejection-Sampling(X, $\mathbf{e}$, bn, N) **returns** an estimate of $P(X|\mathbf{e})$
local variables: $\mathbf{N}$, a vector of counts over X, initially zero

for $j = 1$ to N **do**
$\mathbf{x} \leftarrow$ Prior-Sample(bn)
if $\mathbf{x}$ is consistent with $\mathbf{e}$ **then**
$\mathbf{N}[x] \leftarrow \mathbf{N}[x]+1$ where x is the value of X in $\mathbf{x}$
return Normalize($\mathbf{N}[X]$)

Rejection sampling

$\hat{\Pr}(X|\mathbf{e})$ estimated from samples agreeing with $\mathbf{e}$

```
function Rejection-Sampling(X, e, bn, N) returns an estimate of P(X|e)
    local variables: N, a vector of counts over X, initially zero

    for j = 1 to N do
        x ← Prior-Sample(bn)
        if x is consistent with e then
            N[x] ← N[x]+1 where x is the value of X in x
    return Normalize(N[X])
```

E.g., estimate $\Pr(Rain|Sprinkler = true)$ using 100 samples
27 samples have $Sprinkler = true$
Of these, 8 have $Rain = true$ and 19 have $Rain = false$.

$\hat{\Pr}(Rain|Sprinkler = true) =$ Normalize$(\langle 8, 19\rangle) = \langle 0.296, 0.704\rangle$
Similar to a basic real-world empirical estimation procedure

Analysis of rejection sampling

Rejection sampling returns consistent posterior estimates

Proof:

$\hat{\Pr}(X|\mathbf{e}) = \alpha \mathbf{N}_{PS}(X, \mathbf{e})$ (algorithm defn.)

$= \mathbf{N}_{PS}(X, \mathbf{e}) / N_{PS}(\mathbf{e})$ (normalized by $N_{PS}(\mathbf{e})$)

$\approx \Pr(X, \mathbf{e}) / P(\mathbf{e})$ (property of PriorSample)

$= \Pr(X|\mathbf{e})$ (defn. of conditional probability)

Analysis of rejection sampling

Rejection sampling returns consistent posterior estimates

Proof:

$\hat{\Pr}(X|\mathbf{e}) = \alpha \mathbf{N}_{PS}(X, \mathbf{e})$ (algorithm defn.)

$= \mathbf{N}_{PS}(X, \mathbf{e}) / N_{PS}(\mathbf{e})$ (normalized by $N_{PS}(\mathbf{e})$)

$\approx \Pr(X, \mathbf{e}) / P(\mathbf{e})$ (property of PriorSample)

$= \Pr(X|\mathbf{e})$ (defn. of conditional probability)

Problem: hopelessly expensive if $P(\mathbf{e})$ is small

$P(\mathbf{e})$ 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

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

function Likelihood-Weighting(X, $\mathbf{e}$, bn, N) **returns** an estimate of $P(X|\mathbf{e})$
  local variables: $\mathbf{W}$, a vector of weighted counts over X, initially zero

  for j = 1 to N **do**
    $\mathbf{x}$, $w \leftarrow$ Weighted-Sample(bn)
    $\mathbf{W}[x] \leftarrow \mathbf{W}[x] + w$ where x is the value of X in $\mathbf{x}$
  return Normalize($\mathbf{W}[X]$)

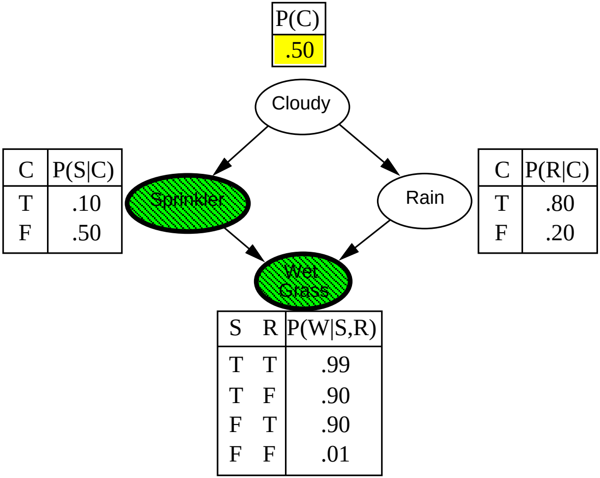
---

function Weighted-Sample(bn, $\mathbf{e}$) **returns** an event and a weight

  $\mathbf{x} \leftarrow$ an event with n elements; $w \leftarrow 1$
  for i = 1 **to** n **do**
    if X_i has a value x_i in $\mathbf{e}$
      then $w \leftarrow w \times\ P(X_i = x_i \mid parents(X_i))$
      else $x_i \leftarrow$ a random sample from $\Pr(X_i \mid parents(X_i))$
  return $\mathbf{x}$, w

Likelihood weighting example

$P(Rain|Sprinkler = true, WetGrass = true)$



P(C)
.50

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

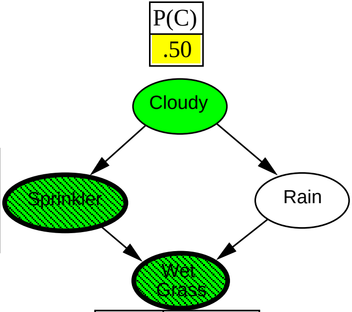
C	P(R|C)
T	.80
F	.20

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

$w = 1.0$

Likelihood weighting example

$P(Rain|Sprinkler = true, WetGrass = true)$



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

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

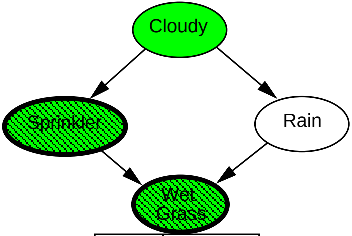
S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

$w = 1.0$

Likelihood weighting example

$P(Rain|Sprinkler = true, WetGrass = true)$

P(C)
.50



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

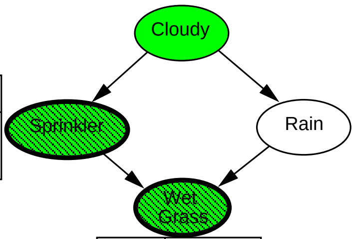
C	P(R|C)
T	.80
F	.20

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

$w = 1.0$

Likelihood weighting example

$P(Rain|Sprinkler = true, WetGrass = true)$



P(C)
.50

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

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

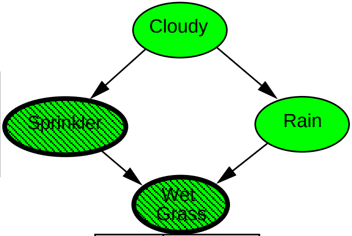
S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

$w = 1.0 \times 0.1$

Likelihood weighting example

$P(Rain|Sprinkler = true, WetGrass = true)$

P(C)
.50



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

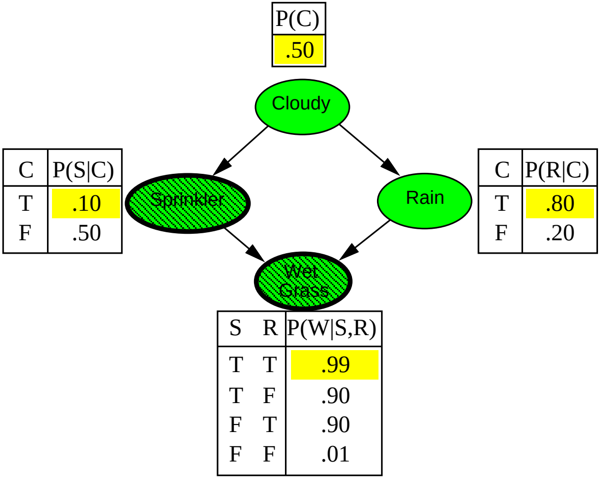
C	P(R|C)
T	.80
F	.20

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

$w = 1.0 \times 0.1$

Likelihood weighting example

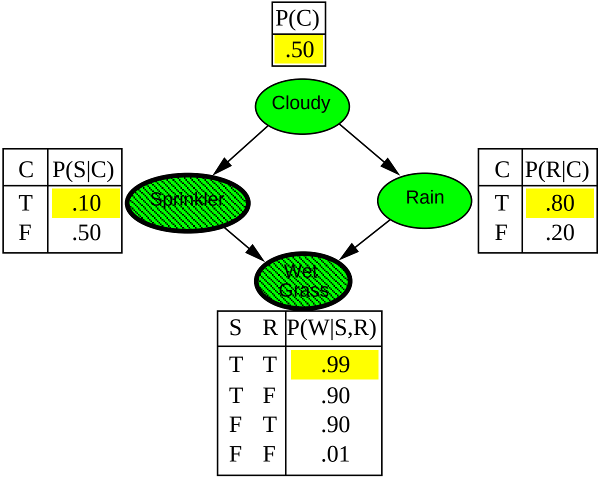
$P(Rain|Sprinkler = true, WetGrass = true)$



$w = 1.0 \ \times 0.1$

Likelihood weighting example

$P(Rain|Sprinkler = true, WetGrass = true)$



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

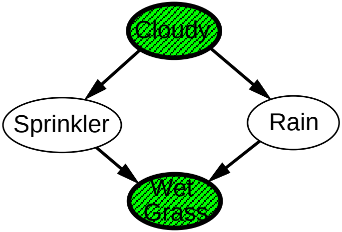
$w = 1.0 \ \times 0.1 \ \times 0.99 = 0.099$

Likelihood weighting analysis

Likelihood weighting returns consistent estimates

Sampling probability for WeightedSample is

$$S_{WS}(\mathbf{z},\mathbf{e}) = \prod_{i=1}^{l} P(z_i|parents(Z_i))$$



(pays attention to evidence in **ancestors** only)
$\leadsto$somewhere "in between" prior and posterior distribution

Weight for a given sample $\mathbf{z},\mathbf{e}$ is

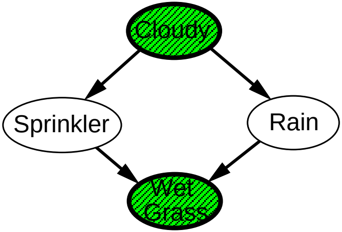
$$w(\mathbf{z},\mathbf{e}) = \prod_{i=1}^{m} P(e_i|parents(E_i))$$

Likelihood weighting analysis

Likelihood weighting returns consistent estimates

Sampling probability for WeightedSample is

$$S_{WS}(\mathbf{z}, \mathbf{e}) = \prod_{i=1}^{l} P(z_i | parents(Z_i))$$



(pays attention to evidence in **ancestors** only)
$\rightsquigarrow$somewhere "in between" prior and posterior distribution

Weight for a given sample $\mathbf{z}, \mathbf{e}$ is

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

Weighted sampling probability is

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

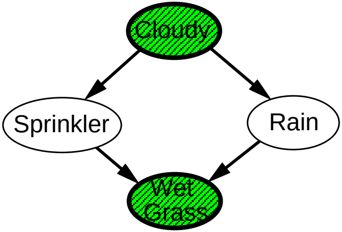
Likelihood weighting analysis

Likelihood weighting returns consistent estimates

Sampling probability for WeightedSample is

$$S_{WS}(\mathbf{z},\mathbf{e}) = \prod_{i=1}^{l} P(z_i|parents(Z_i))$$

(pays attention to evidence in **ancestors** only)
$\rightsquigarrow$somewhere "in between" prior and posterior distribution



Weight for a given sample $\mathbf{z},\mathbf{e}$ is

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

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

Weighted sampling probability is

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

Summary

Approximate inference by LW:

– LW does poorly when there is lots of (late-in-the-order) evidence
– LW generally insensitive to topology
– Convergence can be very slow with probabilities close to 1 or 0
– Can handle arbitrary combinations of discrete and continuous variables

Outline

Approximate inference using MCMC

"State" of network = current assignment to all variables.
Generate next state by sampling one variable given Markov blanket
Sample each variable in turn, keeping evidence fixed

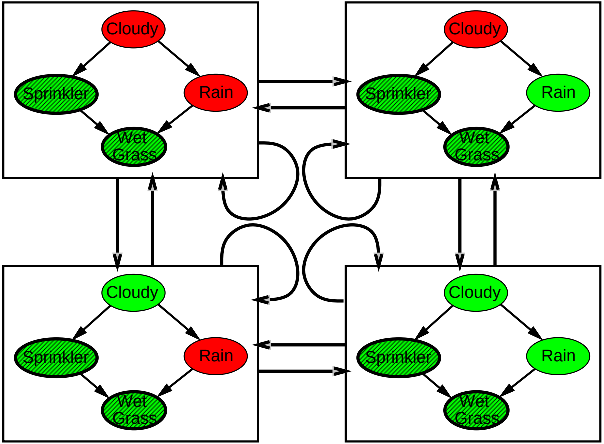
function MCMC-Ask(X, $\mathbf{e}$, bn, N) **returns** an estimate of $P(X|\mathbf{e})$
  local variables: $\mathbf{N}[X]$, a vector of counts over X, initially zero
    $\mathbf{Z}$, nonevidence variables in bn, hidden + query
    $\mathbf{x}$, current state of the network, initially copied from $\mathbf{e}$

  initialize $\mathbf{x}$ with random values for the variables in $\mathbf{Z}$
  for j = 1 to N **do**
    $\mathbf{N}[x] \leftarrow \mathbf{N}[x] + 1$ where x is the value of X in $\mathbf{x}$
    for each Z_i in $\mathbf{Z}$ **do**
      sample the value of Z_i in $\mathbf{x}$ from $\Pr(Z_i|mb(Z_i))$
        given the values of $MB(Z_i)$ in $\mathbf{x}$
  return Normalize($\mathbf{N}[X]$)

Can also choose a variable to sample at random each time

The Markov chain

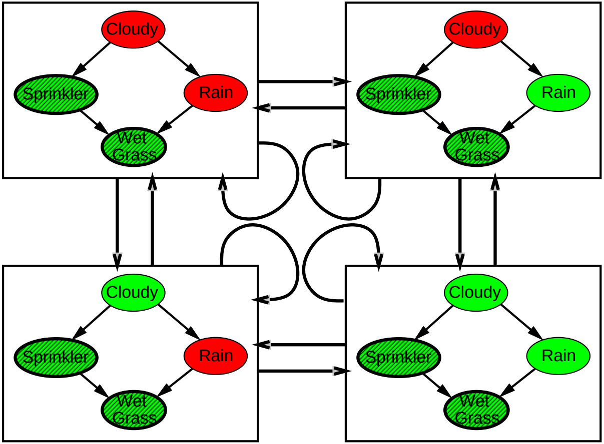
With $Sprinkler = true, WetGrass = true$, there are four states:



Wander about for a while, average what you see

The Markov chain

With $Sprinkler = true, WetGrass = true$, there are four states:



Wander about for a while, average what you see

Probabilistic finite state machine

MCMC example contd.

Estimate $\Pr(Rain|Sprinkler = true, WetGrass = true)$

MCMC example contd.

Estimate $\Pr(Rain|Sprinkler = true, WetGrass = true)$

Sample $Cloudy$ or $Rain$ given its Markov blanket, repeat.

MCMC example contd.

Estimate $\Pr(Rain|Sprinkler = true, WetGrass = true)$

Sample $Cloudy$ or $Rain$ given its Markov blanket, repeat.
Count number of times $Rain$ is true and false in the samples.

MCMC example contd.

Estimate $\Pr(Rain|Sprinkler = true, WetGrass = true)$

Sample $Cloudy$ or $Rain$ given its Markov blanket, repeat.
Count number of times $Rain$ is true and false in the samples.

E.g., visit 100 states
31 have $Rain = true$, 69 have $Rain = false$

$\hat{\Pr}(Rain|Sprinkler = true, WetGrass = true) = \text{Normalize}(\langle 31, 69\rangle) = \langle 0.31, 0.$

MCMC example contd.

Estimate $\Pr(Rain|Sprinkler = true, WetGrass = true)$

Sample $Cloudy$ or $Rain$ given its Markov blanket, repeat.
Count number of times $Rain$ is true and false in the samples.

E.g., visit 100 states
31 have $Rain = true$, 69 have $Rain = false$

$\hat{\Pr}(Rain|Sprinkler = true, WetGrass = true) = \text{Normalize}(\langle 31, 69 \rangle) = \langle 0.31, 0.$

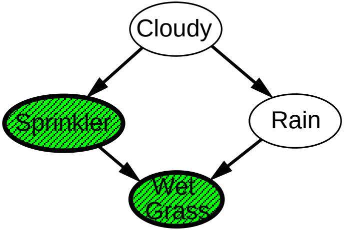
Theorem

The Markov Chain approaches a 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 $Rain$

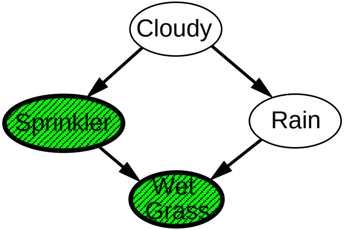
Markov blanket of $Rain$ is
$Cloudy$, $Sprinkler$, and $WetGrass$



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:

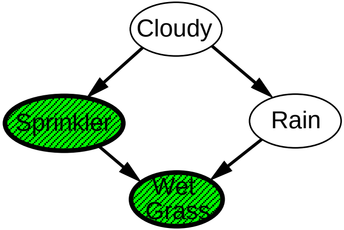
$$P(x_i'|mb(X_i)) = P(x_i'|parents(X_i)) \prod_{Z_j \in Children(X_i)} P(z_j|parents(Z_j))$$

Easily implemented in message-passing parallel systems

Markov blanket sampling

Markov blanket of $Cloudy$ is
$Sprinkler$ and $Rain$

Markov blanket of $Rain$ is
$Cloudy$, $Sprinkler$, and $WetGrass$



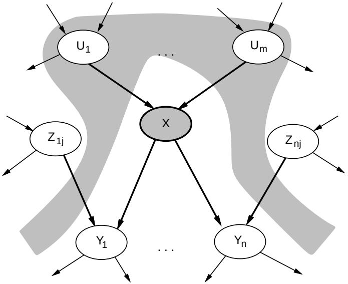
Probability given the Markov blanket is calculated as follows:

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

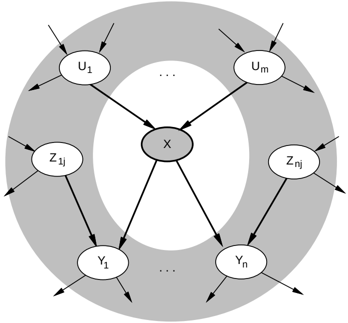
Easily implemented in message-passing parallel systems
Main computational problems:
1) Difficult to tell if convergence has been achieved
2) Can be wasteful if Markov blanket is large:
$P(X_i|mb(X_i))$ won't change much (law of large numbers)

Local semantics and Markov Blanket

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



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



MCMC analysis: Outline

- Transition probability $q(\mathbf{x} \to \mathbf{x}')$

MCMC analysis: Outline

- Transition probability $q(\mathbf{x} \to \mathbf{x}')$
- Occupancy probability $\pi_t(\mathbf{x})$ at time t

MCMC analysis: Outline

- Transition probability $q(\mathbf{x} \to \mathbf{x}')$
- Occupancy probability $\pi_t(\mathbf{x})$ at time t
- Equilibrium condition on π_t defines stationary distribution $\pi(\mathbf{x})$
  Note: stationary distribution depends on choice of $q(\mathbf{x} \to \mathbf{x}')$

MCMC analysis: Outline

- Transition probability $q(\mathbf{x} \to \mathbf{x}')$
- Occupancy probability $\pi_t(\mathbf{x})$ at time t
- Equilibrium condition on π_t defines stationary distribution $\pi(\mathbf{x})$
  Note: stationary distribution depends on choice of $q(\mathbf{x} \to \mathbf{x}')$
- Pairwise detailed balance on states guarantees equilibrium

MCMC analysis: Outline

- Transition probability $q(\mathbf{x} \rightarrow \mathbf{x}')$
- Occupancy probability $\pi_t(\mathbf{x})$ at time t
- Equilibrium condition on π_t defines stationary distribution $\pi(\mathbf{x})$
  Note: stationary distribution depends on choice of $q(\mathbf{x} \rightarrow \mathbf{x}')$
- Pairwise detailed balance on states guarantees equilibrium
- Gibbs sampling transition probability:
  sample each variable given current values of all others
  $\Longrightarrow$ detailed balance with the true posterior

MCMC analysis: Outline

- Transition probability $q(\mathbf{x} \rightarrow \mathbf{x}')$
- Occupancy probability $\pi_t(\mathbf{x})$ at time t
- Equilibrium condition on π_t defines stationary distribution $\pi(\mathbf{x})$
  Note: stationary distribution depends on choice of $q(\mathbf{x} \rightarrow \mathbf{x}')$
- Pairwise detailed balance on states guarantees equilibrium
- Gibbs sampling transition probability:
  sample each variable given current values of all others
  $\Longrightarrow$ detailed balance with the true posterior
- For Bayesian networks, Gibbs sampling reduces to sampling conditioned on each variable's Markov blanket

Stationary distribution

- $\pi_t(\mathbf{x})$ = probability in state $\mathbf{x}$ at time t
  $\pi_{t+1}(\mathbf{x}')$ = probability in state $\mathbf{x}'$ at time $t+1$

Stationary distribution

- $\pi_t(\mathbf{x})$ = probability in state $\mathbf{x}$ at time t
  $\pi_{t+1}(\mathbf{x}')$ = probability in state $\mathbf{x}'$ at time $t+1$

- π_{t+1} in terms of π_t and $q(\mathbf{x} \to \mathbf{x}')$

$$\pi_{t+1}(\mathbf{x}') = \sum\nolimits_{\mathbf{x}} \pi_t(\mathbf{x}) q(\mathbf{x} \to \mathbf{x}')$$

Stationary distribution

- $\pi_t(\mathbf{x})$ = probability in state $\mathbf{x}$ at time t
  $\pi_{t+1}(\mathbf{x}')$ = probability in state $\mathbf{x}'$ at time $t+1$

- π_{t+1} in terms of π_t and $q(\mathbf{x} \rightarrow \mathbf{x}')$

$$\pi_{t+1}(\mathbf{x}') = \sum_{\mathbf{x}} \pi_t(\mathbf{x}) q(\mathbf{x} \rightarrow \mathbf{x}')$$

- Stationary distribution: $\pi_t = \pi_{t+1} = \pi$

$$\pi(\mathbf{x}') = \sum_{\mathbf{x}} \pi(\mathbf{x}) q(\mathbf{x} \rightarrow \mathbf{x}') \qquad \text{for all } \mathbf{x}'$$

Stationary distribution

- $\pi_t(\mathbf{x})$ = probability in state $\mathbf{x}$ at time t
  $\pi_{t+1}(\mathbf{x}')$ = probability in state $\mathbf{x}'$ at time $t+1$

- π_{t+1} in terms of π_t and $q(\mathbf{x} \to \mathbf{x}')$

$$\pi_{t+1}(\mathbf{x}') = \sum_{\mathbf{x}} \pi_t(\mathbf{x}) q(\mathbf{x} \to \mathbf{x}')$$

- Stationary distribution: $\pi_t = \pi_{t+1} = \pi$

$$\pi(\mathbf{x}') = \sum_{\mathbf{x}} \pi(\mathbf{x}) q(\mathbf{x} \to \mathbf{x}') \qquad \text{for all } \mathbf{x}'$$

- If π exists, it is unique (specific to $q(\mathbf{x} \to \mathbf{x}')$)

Stationary distribution

- $\pi_t(\mathbf{x})$ = probability in state $\mathbf{x}$ at time t
  $\pi_{t+1}(\mathbf{x}')$ = probability in state $\mathbf{x}'$ at time $t+1$

- π_{t+1} in terms of π_t and $q(\mathbf{x} \to \mathbf{x}')$

$$\pi_{t+1}(\mathbf{x}') = \sum_{\mathbf{x}} \pi_t(\mathbf{x}) q(\mathbf{x} \to \mathbf{x}')$$

- Stationary distribution: $\pi_t = \pi_{t+1} = \pi$

$$\pi(\mathbf{x}') = \sum_{\mathbf{x}} \pi(\mathbf{x}) q(\mathbf{x} \to \mathbf{x}') \qquad \text{for all } \mathbf{x}'$$

- If π exists, it is unique (specific to $q(\mathbf{x} \to \mathbf{x}')$)

- In equilibrium, expected "outflow" = expected "inflow"

Detailed balance

- "Outflow" = "inflow" for each pair of states:

$$\pi(\mathbf{x})q(\mathbf{x} \to \mathbf{x}') = \pi(\mathbf{x}')q(\mathbf{x}' \to \mathbf{x}) \qquad \text{for all } \mathbf{x},\ \mathbf{x}'$$

Detailed balance

- "Outflow" = "inflow" for each pair of states:

$$\pi(\mathbf{x})q(\mathbf{x} \to \mathbf{x}') = \pi(\mathbf{x}')q(\mathbf{x}' \to \mathbf{x}) \qquad \text{for all } \mathbf{x},\ \mathbf{x}'$$

- Detailed balance $\implies$ stationarity:

$$\begin{aligned}\sum\nolimits_{\mathbf{x}} \pi(\mathbf{x})q(\mathbf{x} \to \mathbf{x}') &= \sum\nolimits_{\mathbf{x}} \pi(\mathbf{x}')q(\mathbf{x}' \to \mathbf{x}) \\ &= \pi(\mathbf{x}') \sum\nolimits_{\mathbf{x}} q(\mathbf{x}' \to \mathbf{x}) \\ &= \pi(\mathbf{x}')\end{aligned}$$

Detailed balance

- "Outflow" = "inflow" for each pair of states:

$$\pi(\mathbf{x})q(\mathbf{x} \to \mathbf{x}') = \pi(\mathbf{x}')q(\mathbf{x}' \to \mathbf{x}) \qquad \text{for all } \mathbf{x},\ \mathbf{x}'$$

- Detailed balance $\implies$ stationarity:

$$\begin{aligned} \sum_{\mathbf{x}} \pi(\mathbf{x})q(\mathbf{x} \to \mathbf{x}') &= \sum_{\mathbf{x}} \pi(\mathbf{x}')q(\mathbf{x}' \to \mathbf{x}) \\ &= \pi(\mathbf{x}') \sum_{\mathbf{x}} q(\mathbf{x}' \to \mathbf{x}) \\ &= \pi(\mathbf{x}') \end{aligned}$$

- MCMC algorithms typically constructed by designing a transition probability q that is in detailed balance with desired π

Gibbs sampling

- Sample each variable in turn, given **all other variables**

Gibbs sampling

- Sample each variable in turn, given **all other variables**
- Sampling X_i, let $\bar{\mathbf{X}}_i$ be all other nonevidence variables

Gibbs sampling

- Sample each variable in turn, given **all other variables**
- Sampling X_i, let $\bar{\mathbf{X}}_i$ be all other nonevidence variables
- Current values are x_i and $\bar{\mathbf{x}}_i$; $\mathbf{e}$ is fixed

Gibbs sampling

- Sample each variable in turn, given **all other variables**
- Sampling X_i, let $\bar{\mathbf{X}}_i$ be all other nonevidence variables
- Current values are x_i and $\bar{\mathbf{x}}_i$; $\mathbf{e}$ is fixed
- Transition probability is given by

$$q(\mathbf{x} \rightarrow \mathbf{x}') = q(x_i, \bar{\mathbf{x}}_i \rightarrow x'_i, \bar{\mathbf{x}}_i) = P(x'_i | \bar{\mathbf{x}}_i, \mathbf{e})$$

Gibbs sampling

- Sample each variable in turn, given **all other variables**
- Sampling X_i, let $\bar{\mathbf{X}}_i$ be all other nonevidence variables
- Current values are x_i and $\bar{\mathbf{x}}_i$; $\mathbf{e}$ is fixed
- Transition probability is given by

$$q(\mathbf{x} \to \mathbf{x}') = q(x_i, \bar{\mathbf{x}}_i \to x_i', \bar{\mathbf{x}}_i) = P(x_i' | \bar{\mathbf{x}}_i, \mathbf{e})$$

- This gives detailed balance with true posterior $P(\mathbf{x}|\mathbf{e})$:

$$\begin{aligned}
\pi(\mathbf{x})q(\mathbf{x} \to \mathbf{x}') &= P(\mathbf{x}|\mathbf{e})P(x_i'|\bar{\mathbf{x}}_i, \mathbf{e}) = P(x_i, \bar{\mathbf{x}}_i|\mathbf{e})P(x_i'|\bar{\mathbf{x}}_i, \mathbf{e}) \\
&= P(x_i|\bar{\mathbf{x}}_i, \mathbf{e})P(\bar{\mathbf{x}}_i|\mathbf{e})P(x_i'|\bar{\mathbf{x}}_i, \mathbf{e}) \quad \text{(chain rule)} \\
&= P(x_i|\bar{\mathbf{x}}_i, \mathbf{e})P(x_i', \bar{\mathbf{x}}_i|\mathbf{e}) \quad \text{(chain rule backwards)} \\
&= q(\mathbf{x}' \to \mathbf{x})\pi(\mathbf{x}') = \pi(\mathbf{x}')q(\mathbf{x}' \to \mathbf{x})
\end{aligned}$$

Summary

Exact inference by variable elimination:
- polytime on polytrees, NP-hard on general graphs
- space = time, very sensitive to topology

Approximate inference by LW, MCMC:

– PriorSampling and RejectionSampling unusable as evidence grow
- LW does poorly when there is lots of (late-in-the-order) evidence
- LW, MCMC generally insensitive to topology
- Convergence can be very slow with probabilities close to 1 or 0
- Can handle arbitrary combinations of discrete and continuous variables