Stochastic inference in Bayesian networks, Markov chain Monte Carlo methods

AI: Stochastic inference in BNs

Outline

- ♢ Types of inference in (causal) BNs
- ♢ Hardness of exact inference in general BNs
- ♢ Approximate inference by stochastic simulation
- ♢ Approximate inference by Markov chain Monte Carlo

Inference tasks

Simple queries: compute posterior marginal $\mathbf{P}(X_i|\mathbf{E}=\mathbf{e})$
e.g., $P(NoGas|Gauge=empty, Lights=on, Starts=false)$

Conjunctive queries: $\mathbf{P}(X_i, X_j|\mathbf{E}=\mathbf{e}) = \mathbf{P}(X_i|\mathbf{E}=\mathbf{e})\mathbf{P}(X_j|X_i, \mathbf{E}=\mathbf{e})$

Optimal decisions: decision networks include utility information;
probabilistic inference required for $P(outcome|action, evidence)$

Value of information: which evidence to seek next?

Sensitivity analysis: which probability values are most critical?

Explanation: why do I need a new starter motor?

Causal inference: what is the effect of an intervention?

Counterfactual inference: what would have been the effect of a hypothetical/imagery past intervention&observation?

Inference by enumeration: principle

Let $\mathbf{X}$ be all the variables. Typically, we want the posterior joint distribution of the query variables $\mathbf{Y}$ given specific values $\mathbf{e}$ for the evidence variables $\mathbf{E}$.

Let the hidden variables be $\mathbf{H} = \mathbf{X} - \mathbf{Y} - \mathbf{E}$.

Then the required summation of joint entries is done by summing out the hidden variables:

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

The terms in the summation are joint entries!

Obvious problems:

1) Worst-case time complexity $O(d^n)$ where d is the largest arity
2) Space complexity $O(d^n)$ to store the joint distribution
3) How to find the numbers for $O(d^n)$ entries???

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 exact inference $O(d^k n)$

Multiply connected networks:

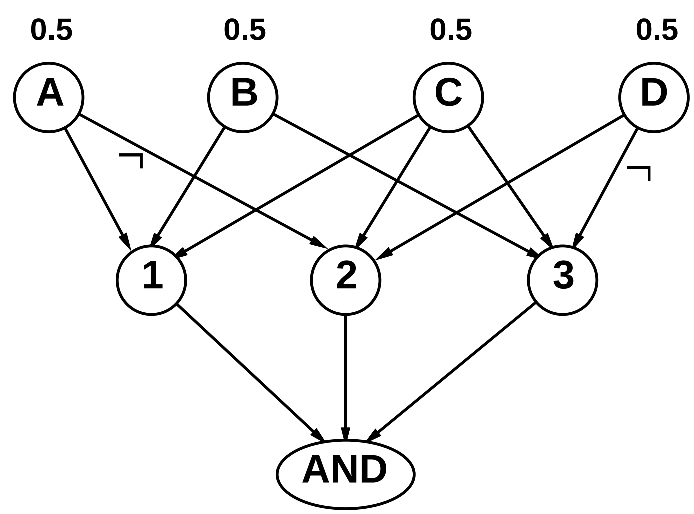
– can reduce 3SAT to exact inference: 0<p(AND)? $\Rightarrow$ NP-hard

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

1. A v B v C

2. C v D v ¬A

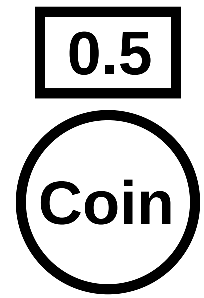
3. B v C v ¬D



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 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

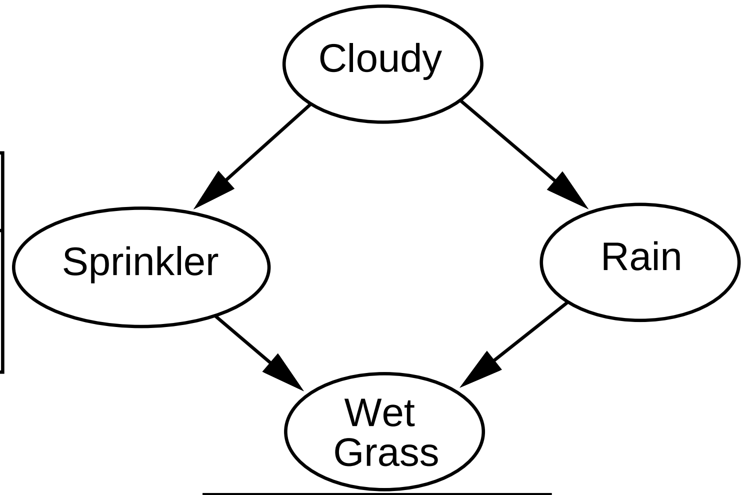
Sampling from an empty network

function PRIOR-SAMPLE(bn) **returns** an event sampled from bn
  inputs: bn, a belief network specifying joint distribution $\mathbf{P}(X_1, \ldots, X_n)$

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

Example

P(C)
.50



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

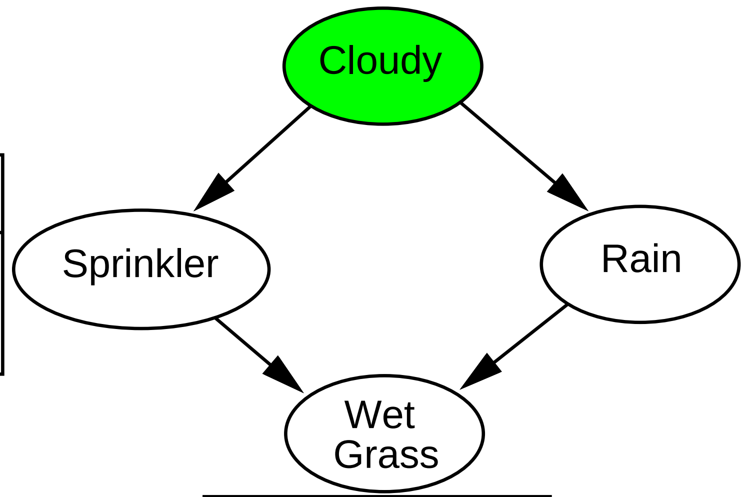
C	P(R|C)
T	.80
F	.20

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

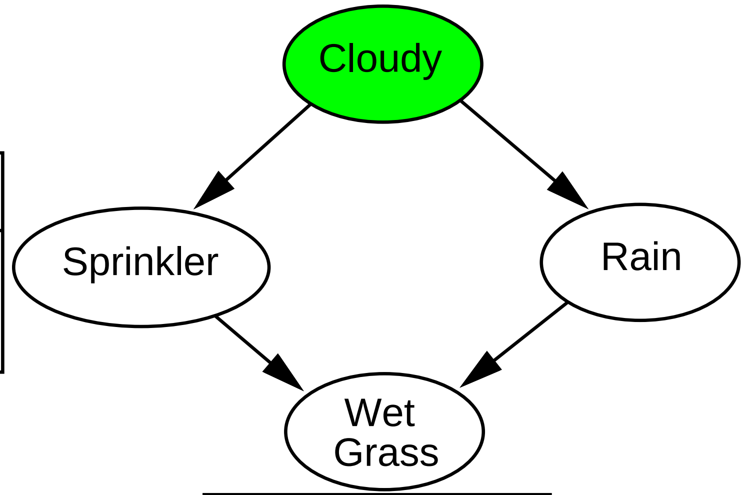


C	P(R|C)
T	.80
F	.20

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

Example

P(C)
.50



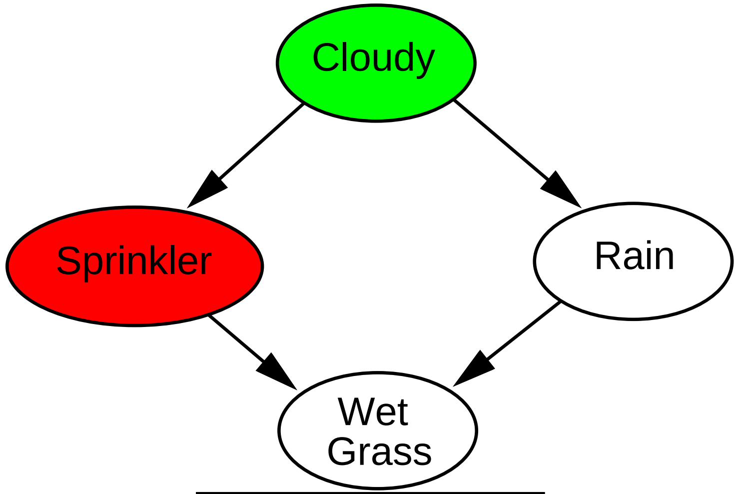
C	P(S|C)
T	.10
F	.50

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



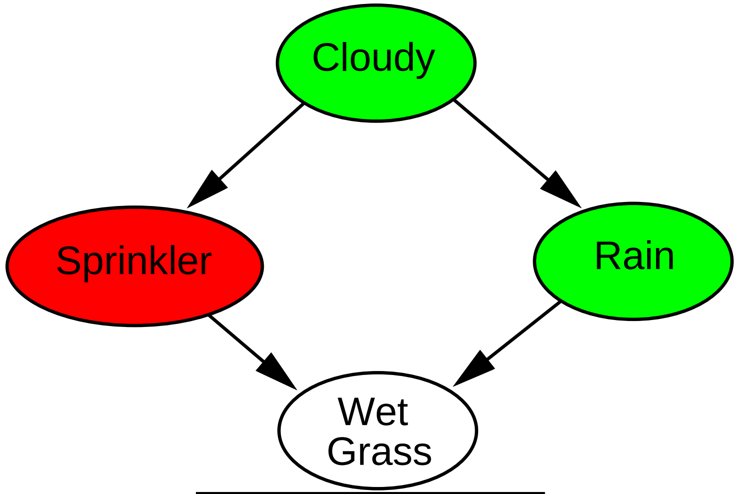
C	P(S|C)
T	.10
F	.50

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

Example

P(C)
.50



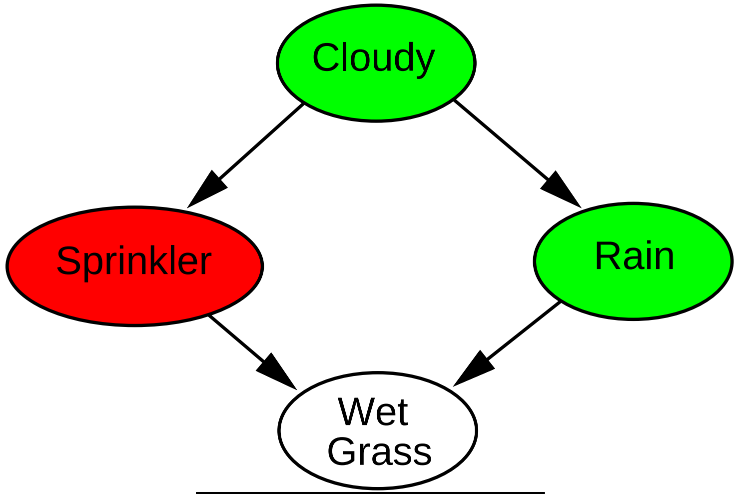
C	P(S|C)
T	.10
F	.50

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



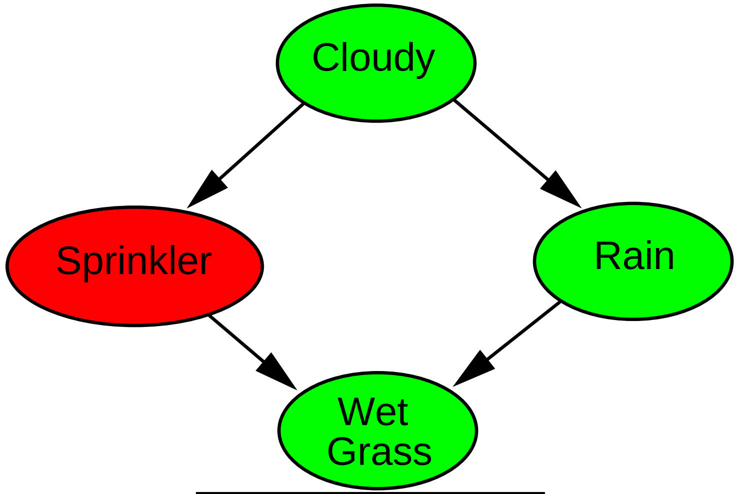
C	P(S|C)
T	.10
F	.50

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

Rejection sampling

$\hat{\mathbf{P}}(X|\mathbf{e})$ 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 $\mathbf{P}(Rain|Sprinkler = true)$ using 100 samples

27 samples have $Sprinkler = true$

Of these, 8 have $Rain = true$ and 19 have $Rain = false$.

$$\hat{\mathbf{P}}(Rain|Sprinkler = true) = \text{NORMALIZE}(\langle 8, 19\rangle) = \langle 0.296, 0.704\rangle$$

Analysis of rejection sampling

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

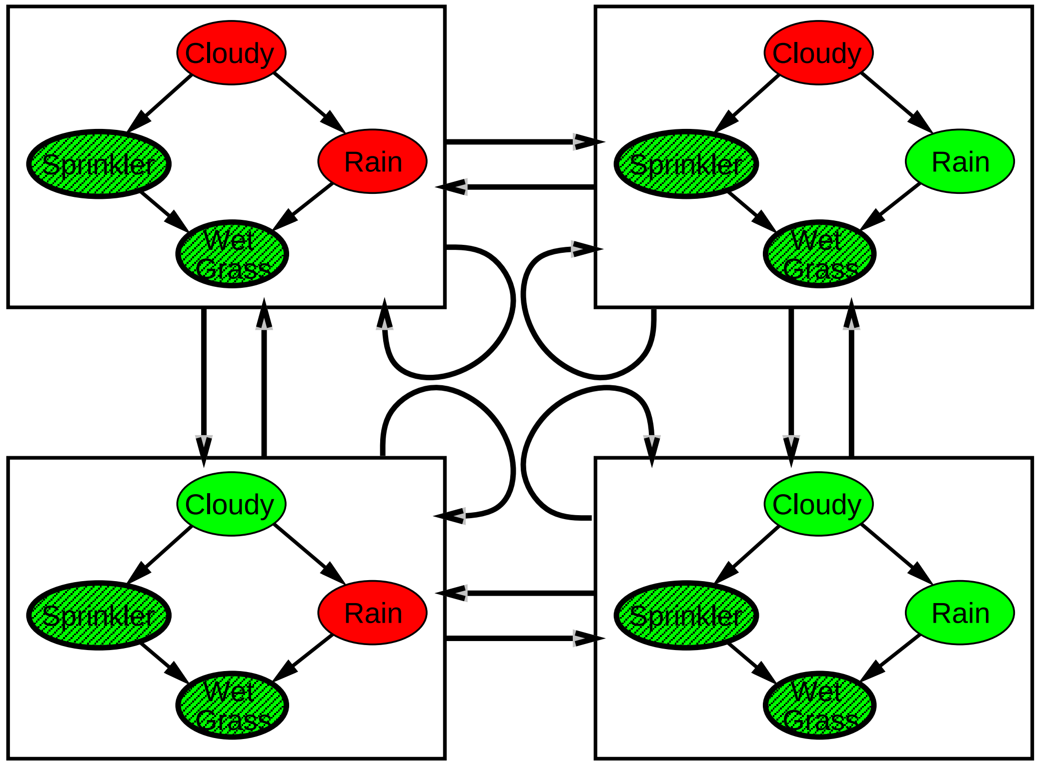
Hence rejection sampling returns consistent posterior estimates

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

$P(\mathbf{e})$ drops off exponentially with number of evidence variables!

The Markov chain

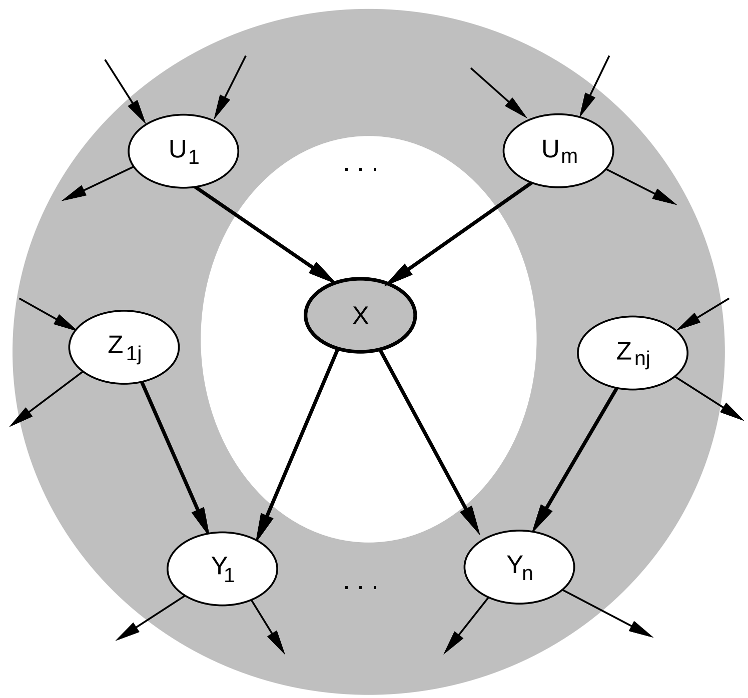
With $Sprinkler = true, WetGrass = true$, there are four states:



Wander about for a while, average what you see

Markov blanket

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



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
        $\mathbf{Z}$, the nonevidence variables in bn
        $\mathbf{x}$, the current state of the network, initially copied from $\mathbf{e}$

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

Can also choose a variable to sample at random each time

MCMC example contd.

Estimate $\mathbf{P}(Rain|Sprinkler = true, WetGrass = true)$

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

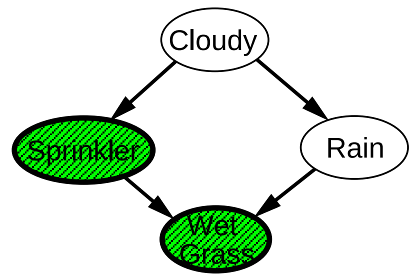
E.g., visit 100 states
31 have $Rain = true$, 69 have $Rain = false$

$\hat{\mathbf{P}}(Rain|Sprinkler = true, WetGrass = true)$
$= \textsc{Normalize}(\langle 31, 69 \rangle) = \langle 0.31, 0.69 \rangle$

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:

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

Easily implemented in message-passing parallel systems, brains

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)

The MCMC age

- ♢ Hardware!
- ♢ Bayesian model averaging

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

Gibbs sampling transition probability:

sample each variable given current values of all others

$\Rightarrow$ 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 $t+1$

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

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

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

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

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

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 $\Rightarrow$ stationarity:

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

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**

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

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

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

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

Performance of approximation algorithms

Absolute approximation: $|P(X|\mathbf{e}) - \hat{P}(X|\mathbf{e})| \leq \epsilon$

Relative approximation: $\frac{|P(X|\mathbf{e}) - \hat{P}(X|\mathbf{e})|}{P(X|\mathbf{e})} \leq \epsilon$

Relative $\Rightarrow$ absolute since $0 \leq P \leq 1$ (may be $O(2^{-n})$)

Randomized algorithms may fail with probability at most δ

Polytime approximation: $\text{poly}(n, \epsilon^{-1}, \log \delta^{-1})$

Theorem (Dagum and Luby, 1993): both absolute and relative approximation for either deterministic or randomized algorithms are NP-hard for any $\epsilon, \delta < 0.5$

(Absolute approximation polytime with no evidence—Chernoff bounds)

Summary

Exact inference:
– polytime on polytrees (NBNs,HMMs), NP-hard on general graphs
– space $=$ time, very sensitive to topology

Approximate inference:
– Convergence can be very slow with probabilities close to 1 or 0
– Can handle arbitrary combinations of discrete and continuous variables