Prior Sampling

\[ P(C) \]
\[
\begin{array}{c|c}
  c & 0.5 \\
  \neg c & 0.5 \\
\end{array}
\]

\[ P(S \mid C) \]
\[
\begin{array}{c|c|c}
  c & s & 0.1 \\
  c & \neg s & 0.9 \\
  \neg c & s & 0.5 \\
  \neg c & \neg s & 0.5 \\
\end{array}
\]

\[ P(W \mid S,R) \]
\[
\begin{array}{c|c|c|c}
  s & r & w & 0.99 \\
  s & \neg r & w & 0.01 \\
  \neg s & r & w & 0.90 \\
  \neg s & \neg r & w & 0.10 \\
\end{array}
\]

\[ P(R \mid C) \]
\[
\begin{array}{c|c|c}
  c & r & 0.8 \\
  c & \neg r & 0.2 \\
  \neg c & r & 0.2 \\
  \neg c & \neg r & 0.8 \\
\end{array}
\]

Samples:
- c, \neg s, r, w
- \neg c, s, \neg r, w
- ...

Prior Sampling

- For i=1, 2, …, n (in topological order)
  - Sample $X_i$ from $P(X_i | \text{parents}(X_i))$
  - Return $(x_1, x_2, \ldots, x_n)$
Prior Sampling

- This process generates samples with probability:
  \[ S_{ps}(x_1, \ldots, x_n) = \prod_i P(x_i | \text{parents}(X_i)) = P(x_1, \ldots, x_n) \]
  …i.e. the BN’s joint probability

- Let the number of samples of an event be \( N_{ps}(x_1, \ldots, x_n) \)
- Estimate from \( N \) samples is \( Q_N(x_1, \ldots, x_n) = N_{ps}(x_1, \ldots, x_n)/N \)
- Then \( \lim_{N \to \infty} Q_N(x_1, \ldots, x_n) = \lim_{N \to \infty} N_{ps}(x_1, \ldots, x_n)/N \)
  \[ = S_{ps}(x_1, \ldots, x_n) \]
  \[ = P(x_1, \ldots, x_n) \]
- I.e., the sampling procedure is consistent
We’ll get a bunch of samples from the BN:

- c, ¬s, r, w
- c, s, r, w
- ¬c, s, r, ¬w
- c, ¬s, r, w
- ¬c, ¬s, ¬r, w

If we want to know $P(W)$

- We have counts $<w:4, ¬w:1>$
- Normalize to get $P(W) = <w:0.8, ¬w:0.2>$
- This will get closer to the true distribution with more samples
Rejection Sampling
Rejection Sampling

A simple application of prior sampling for estimating conditional probabilities

- Let’s say we want $P(C| r, w) = \alpha P(C, r, w)$
- For these counts, samples with $\neg r$ or $\neg w$ are not relevant
- So count the $C$ outcomes for samples with $r, w$ and reject all other samples

This is called rejection sampling

- It is also consistent for conditional probabilities (i.e., correct in the limit)
Likelihood Weighting
**Likelihood Weighting**

**Problem with rejection sampling:**
- If evidence is unlikely, rejects lots of samples
- Evidence not exploited as you sample
- Consider $P(\text{Shape} | \text{Color}=\text{blue})$

**Solution:** weight each sample by probability of evidence variables given parents

**Idea: fix evidence variables, sample the rest**
- Problem: sample distribution not consistent!
- Solution: weight each sample by probability of evidence variables given parents

### Examples
- Pyramid, green
- Pyramid, red
- Sphere, blue
- Cube, red
- Sphere, green
Likelihood Weighting

\[
P(C) = \begin{cases} 
  0.5 & \text{if } c \\
  0.5 & \text{if } \neg c 
\end{cases}
\]

\[
P(S \mid C) = \begin{cases} 
  0.1 & \text{if } s, c \\
  0.9 & \text{if } \neg s, c \\
  0.5 & \text{if } s, \neg c \\
  0.5 & \text{if } \neg s, \neg c 
\end{cases}
\]

\[
P(R \mid C) = \begin{cases} 
  0.8 & \text{if } r, c \\
  0.2 & \text{if } \neg r, c \\
  0.2 & \text{if } r, \neg c \\
  0.8 & \text{if } \neg r, \neg c 
\end{cases}
\]

\[
P(W \mid S,R) = \begin{cases} 
  0.99 & \text{if } w, s, r \\
  0.01 & \text{if } \neg w, s, r \\
  0.90 & \text{if } w, \neg s, r \\
  0.10 & \text{if } \neg w, \neg s, r \\
  0.90 & \text{if } w, s, \neg r \\
  0.10 & \text{if } \neg w, s, \neg r \\
  0.01 & \text{if } w, \neg s, \neg r \\
  0.99 & \text{if } \neg w, \neg s, \neg r 
\end{cases}
\]

Samples: \( c, s, r, w \) \( w = 1.0 \times 0.1 \times 0.99 \)
Likelihood Weighting

- Input: evidence $e_1,..,e_k$
- $w = 1.0$
- for $i=1, 2, ..., n$
  - if $X_i$ is an evidence variable
    - $x_i =$ observed value for $X_i$
    - Set $w = w * P(x_i | parents(X_i))$
  - else
    - Sample $x_i$ from $P(X_i | parents(X_i))$
- return $(x_1, x_2, ..., x_n), w$
**Likelihood Weighting**

- Sampling distribution if $\mathbf{z}$ sampled and $\mathbf{e}$ fixed evidence
  \[ S_{WS}(\mathbf{z}, \mathbf{e}) = \prod_j P(z_j \mid \text{parents}(Z_j)) \]

- Now, samples have weights
  \[ w(\mathbf{z}, \mathbf{e}) = \prod_k P(e_k \mid \text{parents}(E_k)) \]

- Together, weighted sampling distribution is consistent
  \[ S_{WS}(\mathbf{z}, \mathbf{e}) \cdot w(\mathbf{z}, \mathbf{e}) = \prod_j P(z_j \mid \text{parents}(Z_j)) \prod_k P(e_k \mid \text{parents}(E_k)) = P(\mathbf{z}, \mathbf{e}) \]

- Likelihood weighting is an example of *importance sampling*
  - Would like to estimate some quantity based on samples from $P$
  - $P$ is hard to sample from, so use $Q$ instead
  - Weight each sample $x$ by $P(x)/Q(x)$
Car Insurance: \( P(PropertyCost \mid e) \)
Likelihood Weighting

- Likelihood weighting is good
  - All samples are used
  - The values of downstream variables are influenced by upstream evidence

- Likelihood weighting still has weaknesses
  - The values of upstream variables are unaffected by downstream evidence
    - E.g., suppose evidence is a video of a traffic accident
  - With evidence in $k$ leaf nodes, weights will be $O(2^{-k})$
  - With high probability, one lucky sample will have much larger weight than the others, dominating the result

- We would like each variable to “see” all the evidence!
MCMC (Markov chain Monte Carlo) is a family of randomized algorithms for approximating some quantity of interest over a very large state space. A Markov chain is a sequence of randomly chosen states (“random walk”), where each state is chosen conditioned on the previous state. Monte Carlo is either a very expensive city in Monaco with a famous casino, or an algorithm (usually based on sampling) that has some probability of producing an incorrect answer. MCMC is defined as ‘wander around for a bit, average what you see’. That’s it. No more, no less. Some of it will be of use to you. Some will not be of use.
Gibbs sampling

- A particular kind of MCMC
  - States are complete assignments to all variables
    - (Cf local search: closely related to simulated annealing!)
  - Evidence variables remain fixed, other variables change
  - To generate the next state, pick a variable and sample a value for it conditioned on all the other variables: \( X_i' \sim P(X_i \mid x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) \)
    - Will tend to move towards states of higher probability, but can go down too
  - In a Bayes net, \( P(X_i \mid x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) = P(X_i \mid \text{markov\_blanket}(X_i)) \)

- Theorem: Gibbs sampling is consistent*
  - Provided all Gibbs distributions are bounded away from 0 and 1 and variable selection is fair
Why would anyone do this?

Samples soon begin to reflect all the evidence in the network.

Eventually they are being drawn from the true posterior!
How would anyone do this?

- Repeat many times
  - Sample a non-evidence variable $X_i$ from
    \[
P(X_i \mid x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) = P(X_i \mid \text{markov\_blanket}(X_i))
    = \alpha \ P(X_i \mid \text{parents}(X_i)) \prod_j P(y_j \mid \text{parents}(Y_j))
    \]
Gibbs Sampling Example: \( P(S \mid r) \)

- Step 1: Fix evidence
  - \( R = \text{true} \)

- Step 2: Initialize other variables
  - Randomly

- Step 3: Repeat
  - Choose a non-evidence variable \( X \)
  - Resample \( X \) from \( P(X \mid \text{markov_blanket}(X)) \)

Sample \( S \sim P(S \mid c, r, \neg w) \)
Sample \( C \sim P(C \mid s, r) \)
Sample \( W \sim P(W \mid s, r) \)
Markov chain given $s, w$
Car Insurance: $P(Age \mid mc, lc, pc)$
Gibbs sampling and MCMC in practice

- The most commonly used method for large Bayes nets
  - See, e.g., BUGS, JAGS, STAN, infer.net, BLOG, etc.

- Can be compiled to run very fast
  - Eliminate all data structure references, just multiply and sample
  - ~100 million samples per second on a laptop

- Can run asynchronously in parallel (one processor per variable)
- Many cognitive scientists suggest the brain runs on MCMC
Suppose I perform a random walk on a graph, following the arcs out of a node uniformly at random. In the infinite limit, what fraction of time do I spend at each node?

Consider these two examples:
Why does it work? (see AIMA 13.4.2 for details)

- Suppose we run it for a long time and predict the probability of reaching any given state at time $t$: $\pi_t(x_1',...,x_n)$ or $\pi_t(x)$
- Each Gibbs sampling step (pick a variable, resample its value) applied to a state $x$ has a probability $k(x' \mid x)$ of reaching a next state $x'$
- So $\pi_{t+1}(x') = \sum_x k(x' \mid x) \pi_t(x)$ or, in matrix/vector form $\pi_{t+1} = K\pi_t$
- When the process is in equilibrium $\pi_{t+1} = \pi_t = \pi$ so $K\pi = \pi$
- This has a unique* solution $\pi = P(x_1',...,x_n \mid e_1',...,e_k)$
- So for large enough $t$ the next sample will be drawn from the true posterior
  - “Large enough” depends on CPTs in the Bayes net; takes longer if nearly deterministic
Bayes Net Sampling Summary

▪ Prior Sampling \( P \)

▪ Likelihood Weighting \( P( Q \mid e) \)

▪ Rejection Sampling \( P( Q \mid e) \)

▪ Gibbs Sampling \( P( Q \mid e) \)