Inference by stochastic simulation

Basic idea:
1) Draw $N$ samples from a sampling distribution $S$
2) Compute an approximate posterior probability $P$
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 Prob-Sample(e) returns an event sampled from $P$
inputs: $e$, a belief network specifying joint distribution $P(X_1, \ldots, X_n)$
set $x$ ← an event with $n$ elements
for $i = 1$ to $n$ elements
    $x_i$ ← Prob-Sample($e$)
return $x$

Example

Rejection sampling

$P(X|e)$ estimated from samples agreeing with $e$

function Rejection-Sample($X, e, N, X$) returns an estimate of $P(X|e)$
local variables: $N$, a vector of counts over $X$, initially zero
for $j = 1$ to $N$ do
    $x$ ← Prob-Sample($e$)
    if $x$ is consistent with $e$ then
        $N[j] ← N[j] + 1$ where $j$ is the value of $X$ in $x$
return Normalizer($N$)

E.g., estimate $P(\text{Rain} | \text{Sprinkler} = \text{true})$ using 100 samples:
27 samples have $\text{Sprinkler} = \text{true}$
20 of these, 8 have $\text{Rain} = \text{true}$ and 19 have $\text{Rain} = \false$.

Likelihood weighting

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

function Likelihood-Weighted($X, e, X$) returns an estimate of $P(X|e)$
local variables: $W$, a vector of weighted counts over $X$, initially zero
for $j = 1$ to $N$ do
    $x_i$ ← Weighted-Sample($e$)
    $W[i] ← W[i] + w$ where $w$ is the value of $X$ in $x_i$
return Normalizer($W$)

Likelihood weighting

function Weighted-Sample($X, e$) returns an event and a weight
$x$ ← an event with $n$ elements
for $i = 1$ to $n$ elements
    if $x_i$ has a value $e_i$ then
        $w_i ← w_i \times P(X_i | \text{Parents}(X_i))$
    else $w_i$ ← a random sample from $P(X_i | \text{Parents}(X_i))$
return $x, w$
Likelihood weighting example

\[ w = 1.0 \]

Likelihood weighting example

\[ w = 1.0 \times 0.1 \]

Likelihood weighting example

\[ w = 1.0 \times 0.1 \times 0.1 = 0.009 \]

The Markov chain

With \texttt{Sprinkler} = true, \texttt{WetGrass} = true, there are four states:

Wander about for a while, average what you see
MCMC

- Stochastic process that creates a stationary distribution over the states
- What computation are we performing???
- Metropolis Hastings Algorithm
- Gibbs Sampler
- Issues with MCMC
  - Warm up period
  - Visually verify convergence

![Behavior Over Time](image)

Initial State: TFT

Two Chains that Will Converge but with Differing Efficiencies

A

n(A)=0.5

B

n(B)=0.2

C

n(C)=0.3

Figure 6-7: Simple three state system with transition probabilities.

Let us suppose two Markov chains have been constructed which have converged to the stationary (posterior) distribution. If we were to obtain the sequence of states visited by the chains they could be:


and

A, B, A, B, C, A, C, A, A.

Dynamics of Gibbs and Metropolis Samplers

- Motivating simple two dimensional case.
- Dynamics of movement
- Mixing, poor mixing, Multi-modality, stickiness, sensitivity to initial conditions

![Cause 1 – Posterior Correlation](image)

![Measuring Convergence (1)](image)
Measuring Convergence (3)

Start State

<table>
<thead>
<tr>
<th>Transition</th>
<th>FFF</th>
<th>TTT</th>
<th>TFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>FF</td>
<td>0.451537</td>
<td>0.439024</td>
<td>0.453196</td>
</tr>
<tr>
<td>FT</td>
<td>0.548463</td>
<td>0.560976</td>
<td>0.546804</td>
</tr>
<tr>
<td>TF</td>
<td>0.46875</td>
<td>0.431421</td>
<td>0.476684</td>
</tr>
<tr>
<td>TT</td>
<td>0.53125</td>
<td>0.568579</td>
<td>0.523316</td>
</tr>
</tbody>
</table>

M a r g i n a l P r o b a b i l i t y o f X 2 O v e r V a r i o u s T i m e P e r i o d s

<table>
<thead>
<tr>
<th>Time Period</th>
<th>P(X2=F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[5000, 6000]</td>
<td>0.41</td>
</tr>
<tr>
<td>[6000, 7000]</td>
<td>0.42</td>
</tr>
<tr>
<td>[7000, 8000]</td>
<td>0.43</td>
</tr>
<tr>
<td>[8000, 9000]</td>
<td>0.44</td>
</tr>
<tr>
<td>[9000, 10000]</td>
<td>0.45</td>
</tr>
</tbody>
</table>

Learning Networks

- Four situations
  - Structure known, All variables observed
    - Simple counting exercise!
  - Structure known, some variables unobserved
    - EM
  - Structure unknown, All variables observed
    - Can use BIC
  - Structure unknown, some variables unobserved
    - Structural EM
- Currently focus on finding best model, but will later focus on finding multiple models.
  - How? Why?

Structure Known

- Full Observability
  - Count to work out every conditional probability table stored at a node. Maximum likelihood est.
  - Use Laplace correction to stop zero probabilities
- Partial Observability
  - Postulate a hidden variable
  - E step : calculate expectation of hidden variables
    - How?
  - M step : Maximize likelihood like above.

Structure Unknown

- How complex should the graph be?
- Full Observability
  - How many links to postulate?
  - What graph would be the maximum likelihood?
  - Penalize complex models – Occam’s razor
  - Use BIC/MDL