Approximate Inference 1

Forward Sampling

- This section on approximate inference relies on samples / particles
- Full particles: complete assignment to all network variables eg. \(X_1 = x_1, X_2 = x_2, \ldots, X_N = x_N\)

Forward Sampling

- Topological sort or order: An ordering of the nodes in the DAG where X comes before Y in the ordering if there is a directed path from X to Y in the graph.
- A topological order is equivalent to a partial order on the nodes of the graph
- There may be several topological orderings

Examples of Topological orders:
- A,B,C,D
- B,A,C,D

Student Example
Forward Sampling

Topological ordering: D, I, G, S, L
1. Sample D from P(D) (Say you get D=high)
2. Sample I from P(I) (Say you get I=low)
3. Sample G from P(G|I=low,D=high) (Say you get G=C)
4. Sample S from P(S|I=low) (Say you get S=low)
5. Sample L from P(L|G=C) (Say you get L=weak)

You now have a sample (D=high, I=low, G=C, S=low, L=weak)

Forward Sampling

Suppose you want to calculate $P(X_1=x_1, X_2=x_2, \ldots, X_n=x_n)$ using forward sampling on a Bayesian network. The algorithm:
1. Do a topological sort of the nodes in the Bayesian network.
2. For $j = 1$ to NUM_SAMPLES
   - For each node $i$ in the ordering (starting from the top of the Bayesian network down)
     - Sample the value $\tilde{x}_i$ from the distribution $P(X_i | \text{Parents}(X_i))$
     - Add $\{\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_i\}$ to your collection of samples
3. Let $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n$ be the samples with $X_1=x_1, X_2=x_2, \ldots, X_n=x_n$
   - $\hat{x}_i$ is the $i$th sample.

Forward Sampling

- How do you sample from $P(X_i | \text{Parents}(X_i))$?
- Note: $P(X_i | \text{Parents}(X_i))$ is a multinomial distribution $P(x_i^1, \ldots, x_i^K | \theta_1, \ldots, \theta_K)$?

- Generate a sample $s$ uniformly from $[0,1]$
- Partition interval into $k$ subintervals: $[0, \theta_1), [\theta_1, \theta_1+\theta_2), \ldots$
- More generally, the $i$th interval is $[\sum_{j=1}^{i-1} \theta_j, \sum_{j=1}^{i} \theta_j)$
- If $s$ is in the $i$th interval, the sample value is $x_i$
- Use binary search to find the interval for $s$ in time $O(\log k)$
Forward Sampling

Suppose your list of samples looks like the following table:

<table>
<thead>
<tr>
<th>D</th>
<th>I</th>
<th>G</th>
<th>S</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>low</td>
<td>low</td>
<td>B</td>
<td>low</td>
<td>weak</td>
</tr>
<tr>
<td>low</td>
<td>high</td>
<td>A</td>
<td>high</td>
<td>strong</td>
</tr>
<tr>
<td>low</td>
<td>high</td>
<td>A</td>
<td>high</td>
<td>weak</td>
</tr>
<tr>
<td>high</td>
<td>high</td>
<td>A</td>
<td>high</td>
<td>strong</td>
</tr>
<tr>
<td>high</td>
<td>low</td>
<td>C</td>
<td>low</td>
<td>weak</td>
</tr>
</tbody>
</table>

$P(\text{I=high}) = 3/5 = 0.6$

Note that this value becomes a lot more accurate as the number of samples heads to infinity.

Forward Sampling

• From a set of particles $D = \{\xi[1], ..., \xi[M]\}$, we can estimate the expectation of any function $f$ as:

$$\hat{E}_D(f) = \frac{1}{M} \sum_{m=1}^{M} f(\xi[m])$$

• To estimate $P(y)$

$$\hat{P}_D(y) = \frac{1}{M} \sum_{m=1}^{M} I\{y[m] = y\}$$

This is the values of the variables in $Y$ in the particle $\xi[m]$

Forward Sampling

How accurate is this estimate? Using the Hoeffding bound:

$$P_D(\hat{P}_D(y) \not\in [P(y) - \varepsilon, P(y) + \varepsilon]) \leq 2e^{-2M\varepsilon^2}$$

How many samples are required to achieve an estimate whose error is bounded by $\varepsilon$, with probability at least $(1-\delta)$? Setting

$$2e^{-2M\varepsilon^2} \leq \delta$$

we get $M \geq \frac{\ln(2/\delta)}{2\varepsilon^2}$
Forward Sampling

How accurate is this estimate? Using the Chernoff bound:

\[
P_D(\hat{D}(y) \notin P(y)(1 \pm \varepsilon)) \leq 2e^{-\frac{M\varepsilon^2}{2}}
\]

Note: This requires us to know \( P(y) \)

How many samples are required to achieve an estimate whose error is bounded by \( \varepsilon \), with probability at least \( (1-\delta) \)?

\[
M \geq \frac{3}{\varepsilon^2} \frac{\ln(2/\delta)}{P(y)}
\]

Rejection Sampling

What if we want to estimate \( P(y|E=e) \)?

- **Rejection sampling**: do forward sampling but throw out samples where \( E \neq e \)

Example:

<table>
<thead>
<tr>
<th>D</th>
<th>I</th>
<th>G</th>
<th>S</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>low</td>
<td>low</td>
<td>B</td>
<td>low</td>
<td>weak</td>
</tr>
<tr>
<td>low</td>
<td>high</td>
<td>A</td>
<td>high</td>
<td>strong</td>
</tr>
<tr>
<td>low</td>
<td>high</td>
<td>A</td>
<td>high</td>
<td>weak</td>
</tr>
<tr>
<td>high</td>
<td>high</td>
<td>A</td>
<td>high</td>
<td>strong</td>
</tr>
<tr>
<td>high</td>
<td>low</td>
<td>C</td>
<td>low</td>
<td>weak</td>
</tr>
</tbody>
</table>

Rejection Sampling

What if the evidence \( E=e \) is very very rare?

- For example, if \( P(e) = 0.001 \), then for 10,000 samples, we get 10 unrejected samples
- To obtain at least \( M^* \) unrejected samples, we need to generate on average \( M = M^*/P(e) \) samples
- If evidence is rare, we end up generating a lot of samples which wastes time
Rejection Sampling

Bad news:
– Rare evidence is the norm!
– As # of evidence variables \( k = |E| \) grows, the probability of the evidence decreases exponentially with \( k \)

Need something better than rejection sampling!

Likelihood Weighting

Intuition: Weight samples according to probability of the evidence

<table>
<thead>
<tr>
<th>I</th>
<th>S</th>
<th>P(I( \rightarrow )S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>low</td>
<td>low</td>
<td>0.95</td>
</tr>
<tr>
<td>low</td>
<td>high</td>
<td>0.05</td>
</tr>
<tr>
<td>high</td>
<td>low</td>
<td>0.2</td>
</tr>
<tr>
<td>high</td>
<td>high</td>
<td>0.8</td>
</tr>
</tbody>
</table>

Drawing \( I = \text{high} \) and \( S = \text{high} \) should be 80% of a sample

Drawing \( I = \text{low} \) and \( S = \text{high} \) should be 5% of a sample

Likelihood Weighting

Weighted particles:

\[ D = \langle \xi[1], w[1] \rangle, \ldots, \langle \xi[M], w[M] \rangle \]

Estimate:

\[ \hat{P}_D(y | e) = \frac{\sum_{m=1}^{M} w[m] I\{y[m] = y\}}{\sum_{m=1}^{M} w[m]} \]
Likelihood Weighting

Procedure LW-Sample(
    β, // Bayesian network over \( X \)
    Z=z // Event in the network
)
1. Let \( X_1, \ldots, X_n \) be a topological ordering of \( X \)
2. \( w \leftarrow 1 \)
3. for \( i = 1, \ldots, n \)
4. \( u_i \leftarrow x_{Pa_{u_i}} \) // Assignment to \( Pa_{u_i} \) in \( x_1, \ldots, x_{i-1} \)
5. if \( X_i \in Z \) then
6. Sample \( x_i \) from \( P(X_i \mid u_i) \)
7. else
8. \( x_i \leftarrow z_{\setminus X} \) // Assignment to \( X_i \) in \( z \)
9. \( w \leftarrow w \cdot P(x_i \mid u_i) \) // Multiply weight by probability of desired value
10. return \((x_1, \ldots, x_n), w\)