Idea: probabilities $\leftrightarrow$ samples

Get probabilities from samples:

<table>
<thead>
<tr>
<th>$X$</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>$n_1$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$x_k$</td>
<td>$n_k$</td>
</tr>
<tr>
<td>total</td>
<td>$m$</td>
</tr>
</tbody>
</table>

$\leftrightarrow$

<table>
<thead>
<tr>
<th>$X$</th>
<th>probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>$n_1/m$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$x_k$</td>
<td>$n_k/m$</td>
</tr>
</tbody>
</table>

If we could sample from a variable’s (posterior) probability, we could estimate its (posterior) probability.
Generating samples from a distribution

For a variable $X$ with a discrete domain or a (one-dimensional) real domain:

- Totally order the values of the domain of $X$.
- Generate the cumulative probability distribution: $f(x) = P(X \leq x)$.
- Select a value $y$ uniformly in the range $[0, 1]$.
- Select the $x$ such that $f(x) = y$. 
Cumulative Distribution

\[ P(X) \]

\[ f(X) \]
Hoeffding’s inequality

Theorem (Hoeffding): Suppose $p$ is the true probability, and $s$ is the sample average from $n$ independent samples; then

$$P(|s - p| > \epsilon) \leq 2e^{-2n\epsilon^2}.$$ 

Guarantees a probably approximately correct estimate of probability.

If you are willing to have an error greater than $\epsilon$ in less than $\delta$ of the cases, solve $2e^{-2n\epsilon^2} < \delta$ for $n$, which gives

$$n > \frac{-\ln \frac{\delta}{2}}{2\epsilon^2}.$$ 

<table>
<thead>
<tr>
<th>$\epsilon$</th>
<th>$\delta$</th>
<th>$n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.05</td>
<td>185</td>
</tr>
<tr>
<td>0.01</td>
<td>0.05</td>
<td>18,445</td>
</tr>
<tr>
<td>0.1</td>
<td>0.01</td>
<td>265</td>
</tr>
</tbody>
</table>
Forward sampling in a belief network

- Sample the variables one at a time; sample parents of $X$ before sampling $X$.
- Given values for the parents of $X$, sample from the probability of $X$ given its parents.
Rejection Sampling

- To estimate a posterior probability given evidence $Y_1 = v_1 \land \ldots \land Y_j = v_j$:
- Reject any sample that assigns $Y_i$ to a value other than $v_i$.
- The non-rejected samples are distributed according to the posterior probability:

$$P(\alpha \mid \text{evidence}) \approx \frac{\sum_{\alpha \text{ is true in sample}} 1}{\sum_{\text{sample}} 1}$$

where we consider only samples consistent with evidence.
Rejection Sampling Example: \( P(\text{ta} \mid \text{sm}, \text{re}) \)

Observe \( Sm = true, Re = true \)

<table>
<thead>
<tr>
<th></th>
<th>Ta</th>
<th>Fi</th>
<th>Al</th>
<th>Sm</th>
<th>Le</th>
<th>Re</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s_1 )</td>
<td>false</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>( s_2 )</td>
<td>false</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>( s_3 )</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>false</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>( s_4 )</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>( s_{1000} )</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>—</td>
</tr>
</tbody>
</table>

\[ P(sm) = 0.02 \]
\[ P(re \mid sm) = 0.32 \]

There are 1000 samples.

How many are rejected?

How many are used?

Doesn’t work well when evidence is unlikely.
Samples have weights: a real number associated with each sample that takes the evidence into account.

Probability of a proposition is weighted average of samples:

\[
P(\alpha \mid \text{evidence}) \approx \frac{\sum_{\text{sample:} \alpha \text{ is true in sample}} \text{weight(sample)}}{\sum_{\text{sample}} \text{weight(sample)}}
\]

Mix exact inference with sampling: don’t sample all of the variables, but weight each sample according to \(P(\text{evidence} \mid \text{sample})\).
Importance Sampling (Likelihood Weighting)

procedure $likelihood\_weighting(Bn, e, H, n)$:

# Approximate $P(H \mid e)$ in belief network $Bn$ using $n$ samples.
# $H$ has some real domain (e.g., $\{0, 1\}$)

mass := 0  # mass of all samples
hmass := 0  # weighted sum of value of $H$

repeat $n$ times:

weight := 1  # weight of current sample

for each variable $X_i$ in order:

if $X_i = o_i$ is observed

weight := weight $\times$ $P(X_i = o_i \mid \text{parents}(X_i))$

else assign $X_i$ a random sample of $P(X_i \mid \text{parents}(X_i))$

mass := mass + weight

hmass := hmass + weight $\times$ (value of $H$ in current assignment)

return $hmass / \text{mass}$
Importance Sampling Example: $P(ta \mid sm, re)$

<table>
<thead>
<tr>
<th></th>
<th>Ta</th>
<th>Fi</th>
<th>Al</th>
<th>Le</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_1$</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>false</td>
<td>$0.01 \times 0.01$</td>
</tr>
<tr>
<td>$s_2$</td>
<td>false</td>
<td>true</td>
<td>false</td>
<td>false</td>
<td>$0.9 \times 0.01$</td>
</tr>
<tr>
<td>$s_3$</td>
<td>false</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>$0.9 \times 0.75$</td>
</tr>
<tr>
<td>$s_4$</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>$0.9 \times 0.75$</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$s_{1000}$</td>
<td>false</td>
<td>false</td>
<td>true</td>
<td>true</td>
<td>$0.01 \times 0.75$</td>
</tr>
</tbody>
</table>

$P(sm \mid fi) = 0.9$
$P(sm \mid \neg fi) = 0.01$
$P(re \mid le) = 0.75$
$P(re \mid \neg le) = 0.01$
Importance Sampling Example: $P(\text{le} \mid \text{sm}, \text{ta}, \neg\text{re})$

\[
P(\text{ta}) = 0.02 \\
P(\text{fi}) = 0.01 \\
P(\text{al} \mid \text{fi} \land \text{ta}) = 0.5 \\
P(\text{al} \mid \text{fi} \land \neg\text{ta}) = 0.99 \\
P(\text{al} \mid \neg\text{fi} \land \text{ta}) = 0.85 \\
P(\text{al} \mid \neg\text{fi} \land \neg\text{ta}) = 0.0001 \\
P(\text{sm} \mid \text{fi}) = 0.9 \\
P(\text{sm} \mid \neg\text{fi}) = 0.01 \\
P(\text{le} \mid \text{al}) = 0.88 \\
P(\text{le} \mid \neg\text{al}) = 0.001 \\
P(\text{re} \mid \text{le}) = 0.75 \\
P(\text{re} \mid \neg\text{le}) = 0.01
\]
Expected value of $f$ with respect to distribution $P$:

$$
\mathbb{E}_P(f) = \sum_w f(w) \ast P(w)
$$

$$
\approx \frac{1}{n} \sum_s f(s)
$$

$s$ is sampled with probability $P$. There are $n$ samples. (Expectation of variable with domain $\{0, 1\}$ is its probability.)

$$
\mathbb{E}_P(f) = \sum_w f(w) \ast P(w)/Q(w) \ast Q(w)
$$

$$
\approx \frac{1}{n} \sum_s f(s) \ast P(s)/Q(s)
$$

$s$ is selected according the distribution $Q$. The distribution $Q$ is called a proposal distribution. $P(c) > 0$ then $Q(c) > 0$. Try to make $Q$ so the weights end up far from zero.
Particle Filtering

Importance sampling can be seen as:

\[
\text{for each particle:} \\
\quad \text{for each variable:} \quad \text{sample / absorb evidence / update query}
\]

where particle is one of the samples.

Instead we could do:

\[
\text{for each variable:} \\
\quad \text{for each particle:} \quad \text{sample / absorb evidence / update query}
\]

Why?

- It works with infinitely many variables (e.g., HMM)
- We can have a new operation of resampling
Particle Filtering for HMMs

- Start with random chosen particles (say 1000)
- Sample initial states in proportion to their probability.
- Repeat (as each observation arrives):
  - Absorb evidence: weight each particle by the probability of the evidence observation given the state of the particle.
  - Resample: select each particle at random, in proportion to the weight of the particle. Some particles may be duplicated, some may be removed. All new particles have same weight.
  - Transition: sample the next state for each particle according to the transition probabilities.

To answer a query about the current state, use the set of particles as data.
Example: Localization

Loc consists of \((x, y, \theta)\) – position and orientation

\(k = 24\) sonar sensors (all very noisy)
Markov Chain Monte Carlo

To sample from a distribution $P$:

- Create (ergodic and aperiodic) Markov chain with $P$ as equilibrium distribution.
  Let $T(S_{i+1} \mid S_i)$ be the transition probability.
- Given state $s$, sample state $s'$ from $T(S \mid s)$
- After a while, the states sampled will be distributed according to $P$.
- Ignore the first samples “burn-in”
  — use the remaining samples.
- Samples are not independent of each other “autocorrelation”.
  Sometimes use subset (e.g., 1/1000) of them “thinning”
- **Gibbs sampler**: sample each non-observed variable from the distribution of the variable given the current (or observed) value of the variables in its Markov blanket.
Gibbs Sampling Example: \( P(ta \mid sm, re) \)

<table>
<thead>
<tr>
<th></th>
<th>Ta</th>
<th>Fi</th>
<th>Al</th>
<th>Le</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s_1 )</td>
<td>true</td>
<td>false</td>
<td>false</td>
<td>true</td>
</tr>
<tr>
<td>( s_2 )</td>
<td>true</td>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>( s_3 )</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>( s_4 )</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>( s_5 )</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>( s_6 )</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
</tbody>
</table>

Select \( Le \). Sample from \( P(Le \mid \neg al \land re) \)

Select \( Fi \). Sample from \( P(Fi \mid ta \land \neg al \land sm) \)

Select \( Al \). Sample from \( P(Al \mid ta \land fi \land \neg le) \)

Select \( Le \). Sample from \( P(Le \mid \neg al \land re) \)

Select \( Ta \). Sample from \( P(Ta \mid \neg al \land fi) \)