Stochastic Simulation

- Idea: probabilities $\leftrightarrow$ samples
- Get probabilities from samples:

X	$count$
x_1	n_1
$\vdots$	$\vdots$
x_k	n_k
$total$	m

$\leftrightarrow$

X	$probability$
x_1	n_1/m
$\vdots$	$\vdots$
x_k	n_k/m

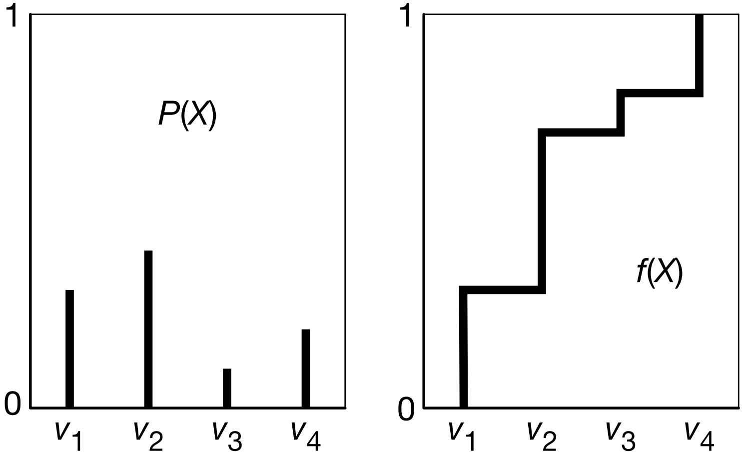
- If we could sample from a variable's (posterior) probability, we could estimate its (posterior) probability.

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



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 ϵ in less than δ of the cases, solve $2e^{-2n\epsilon^2} < \delta$ for n, which gives

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

ϵ	δ	n
0.1	0.05	185
0.01	0.05	18,445
0.1	0.01	265

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 $Y_1 = v_1 \wedge \ldots \wedge Y_j = v_j$:
- 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:

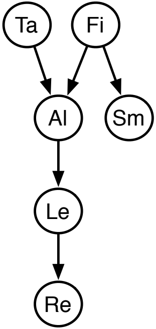
$$P(\alpha \mid evidence) \approx \frac{\sum_{\alpha \text{ is true in sample}} 1}{\sum_{sample} 1}$$

where we consider only samples consistent with evidence.

Rejection Sampling Example: $P(ta \mid sm, re)$

Observe $Sm = true$, $Re = true$

	Ta	Fi	Al	Sm	Le	Re	
s_1	false	true	false	true	false	false	✘
s_2	false	true	true	true	true	true	✔
s_3	true	false	true	false	—	—	✘
s_4	true	true	true	true	true	true	✔
...							
s_{1000}	false	false	false	false	—	—	✘



$P(sm) = 0.02$
$P(re \mid sm) = 0.32$
There are 1000 samples.

How many are rejected?
How many are used?

Doesn't work well when evidence is unlikely.

Importance Sampling

- 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:

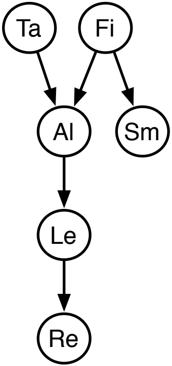
$$P(\alpha \mid evidence) \approx \frac{\displaystyle\sum_{sample:\alpha \text{ is true in sample}} weight(sample)}{\displaystyle\sum_{sample} weight(sample)}$$

- Mix exact inference with sampling: don't sample all of the variables, but weight each sample according to $P(evidence \mid sample)$.

Importance Sampling (Likelihood Weighting)

```
procedure likelihood_weighting(Bn, e, H, n):
    # Approximate P(H | 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 × P(X_i = o_i | parents(X_i))
            else assign X_i a random sample of P(X_i | parents(X_i))
        mass := mass + weight
        hmass := hmass + weight ∗ (value of H in current assignment)
    return hmass/mass
```

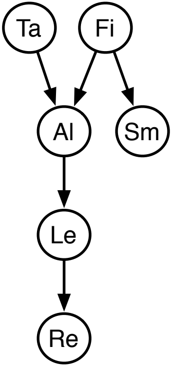
Importance Sampling Example: $P(ta \mid sm, re)$



	Ta	Fi	Al	Le	Weight
s_1	true	false	true	false	0.01×0.01
s_2	false	true	false	false	0.9×0.01
s_3	false	true	true	true	0.9×0.75
s_4	true	true	true	true	0.9×0.75
...					
s_{1000}	false	false	true	true	0.01×0.75

$P(sm \mid fi) = 0.9$
$P(sm \mid \neg fi) = 0.01$
$P(re \mid le) = 0.75$
$P(re \mid \neg le) = 0.01$

Importance Sampling Example: $P(le \mid sm, ta, \neg re)$



$$P(ta) = 0.02$$
$$P(fi) = 0.01$$
$$P(al \mid fi \wedge ta) = 0.5$$
$$P(al \mid fi \wedge \neg ta) = 0.99$$
$$P(al \mid \neg fi \wedge ta) = 0.85$$
$$P(al \mid \neg fi \wedge \neg ta) = 0.0001$$
$$P(sm \mid fi) = 0.9$$
$$P(sm \mid \neg fi) = 0.01$$
$$P(le \mid al) = 0.88$$
$$P(le \mid \neg al) = 0.001$$
$$P(re \mid le) = 0.75$$
$$P(re \mid \neg le) = 0.01$$

Computing Expectations & Proposal Distributions

Expected value of f with respect to distribution P:

$$\mathbb{E}_P(f) = \sum_w f(w) * P(w)$$
$$\approx \frac{1}{n} \sum_s f(s)$$

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

$$\mathbb{E}_P(f) = \sum_w f(w) * P(w)/Q(w) * Q(w)$$
$$\approx \frac{1}{n} \sum_s f(s) * P(s)/Q(s)$$

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:

for each particle:
for each variable:
sample / absorb evidence / update query

where particle is one of the samples.
Instead we could do:

for each variable:
for each particle:
sample / absorb evidence / update query

Why?

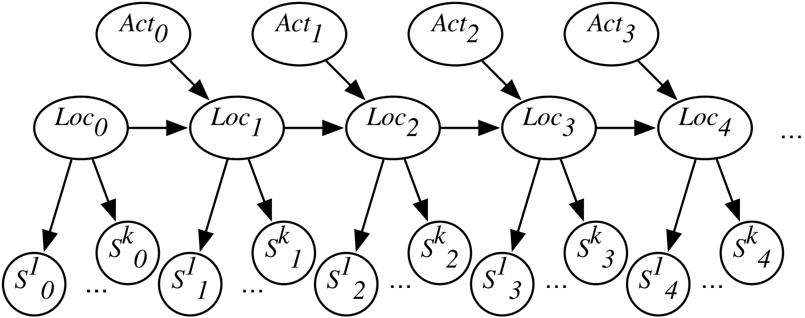
- It works with infinitely many variables (e.g., HMM)
- We can have a new operation of resampling

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, θ) – position and orientation

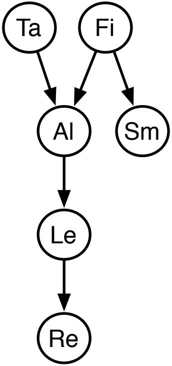
$k = 24$ sonar sensors (all very noisy)

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: $P(ta \mid sm, re)$



	Ta	Fi	Al	Le
s_1	true	false	false	true
Select *Le*. Sample from $P(Le \mid \neg al \wedge re)$				
s_2	true	false	false	false
Select *Fi*. Sample from $P(Fi \mid ta \wedge \neg al \wedge sm)$				
s_3	true	true	false	false
Select *Al*. Sample from $P(Al \mid ta \wedge fi \wedge \neg le)$				
s_4	true	true	false	false
Select *Le*. Sample from $P(Le \mid \neg al \wedge re)$				
s_5	true	true	false	true
Select *Ta*. Sample from $P(Ta \mid \neg al \wedge fi)$				
s_6	true	true	false	true

...