Sampling

For large, multiply connected Bayes nets, it's intractable to calculate the exact inference probability. The calculation has exponential runtime. In these cases, the approximate inference is calculated instead using sampling from the distribution from the relevant nodes in the network. The approximate distribution approaches the true distribution as more samples are generated.

Sampling is faster than computing the exact answer and can also be used for learning a distribution if the distribution is unknown.

How sampling works:

Step 1: Generate a sample *u* from a uniform distribution over [0, 1). For example, use the *random()* functionality in python to get a number between [0, 1).

Step 2: Convert the sample *u* into an outcome for the given distribution by associating the sample with a sub-interval size equal to the probability of the outcome.

Example: Assume you have 100 socks, where 60 socks are red, 10 are green, and 30 are blue. The distribution of sock colors is:

Color	P(color)
red	0.60
green	0.10
blue	0.30

To sample from that distribution, generate a random number [0, 1).

0 <= u < 0.60 -> Color = red
0.60 <= u < 0.70 -> Color = green
0.70 <= u < 1 -> Color = red

Ex: if random returns 0.83, then our sample color = red.
Ex: Assume we repeat the sampling 8 times, generating 8 random numbers:

.34, .45, .57, .12, .25, .67, .73, .85

that map to 5 red, 2 blue, and 1 green. Then P(blue) = 2/8. (Note: the samples are grouped into red, green, blue, but that's for readability in the notes. The order doesn't matter in the probability calculation.)

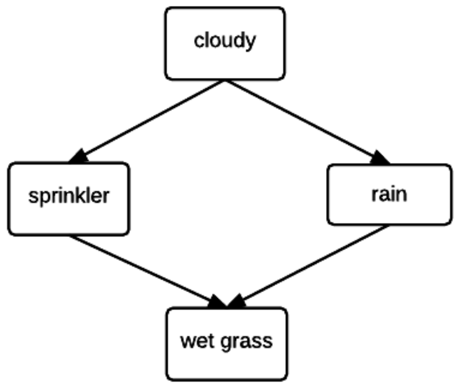
Prior sampling

Given a Bayes net, we want to generate samples to estimate the joint and conditional distributions with calculating them.

Assume we have the following Bayes net with four nodes. Cloudy is the parent of sprinkler and rain, which are both the parent of the wet grass. The prior probability for cloudy P(c) = 0.5: it's cloudy half the time. Given that it is cloudy, it's raining 80% of the time, and 20% of the time when it's not cloudy, it's still raining. The rain and the sprinkler both affect whether the grass is wet. If it's raining and the sprinkler is on, the grass is wet 99% of the time.

P(C) = 0.5

C	P(S)
t	0.10
f	0.50



C	P(R)
t	0.80
f	0.20

S	R	P(W)
t	t	0.99
t	f	0.90
f	t	0.90
f	f	0.00

Ex: Calculate P(w) using sampling.
Start with cloudy and generate number on [0, 1). Assume:

0 <= u < 0.5, cloudy = true
0.5 <= u < 1, cloudy = false

Assume sample u = 0.45, meaning c = true.

Choose S, or R to sample next. We're in the state of +c, which determines our S distribution.

0 <= u < 0.1, sprinkler = true
0.1 <= u < 1, sprinkler = false

Sample S, and assume u = .56, meaning s = false.

Choose R to sample next. We're in the state of +c, -s.

The R distribution for c = true:

0 <= u < 0.8, rain = true
0.8 <= u < 1, rain = false

Sample R, and assume u = .75, meaning r = true.

Finally, sample W next. We're in the state of +c, -s, +r.

The W distribution for s = false, r = true:

0 <= u < 0.9, wet grass = true
0.9 <= u < 1, wet grass = false

Sample W, and assume u = .56, meaning w = true.

The full sample is +c, -s, +r, +w.

Generate a bunch of samples using this process.

Ex: Assume the next sample generated with the numbers:

.43, .09, .67, .89

The sample is +c, +s, +r, +w.

Assume the set of 5 samples includes:
+c, -s, +r, +w
+c, +s, +r, +w
-c, +s, +r, -w
+c, -s, +r, +w
-c, -s, -r, +w

To determine P(W), count the samples with +w and -w and divide by total number of samples.

<+w:4, -w:1>, P(W) = <4/5, 1/5> = <.80, .20>

With these five samples, P(W = true) = 0.80. If we calculate P(W) exactly, we use the formula:

$$P(w) = \sum_i \sum_j P(w \mid s_i, r_j) P(s_i) P(r_j)$$

Example: P(C | +w). Find the samples with +w, and from those, find the samples with +/-c. +c = 3/4 and -c = 1/4.
P(+c | +w) = .75
P(-c | +w) = .25

Rejection sampling
With prior sampling, we don't know what the question is before we start sampling. We just generate a bunch of samples and ask questions later. Imagine the scenario where you want to answer P(+c | +s, +r). We only care about samples that have +s, but with prior sampling, we would continue generating samples even if we get -s. We would also sample all variables in the net, even if we didn't need them.

However, any sample with -s can be rejected and we don't need to sample R. If there were other variables, those wouldn't need to be sampled either.

Ex: We want P(C). We sample from the top down from C, which means that S, R, W are irrelevant. We can sample C only and ignore the other variables. Each number generated samples from the C distribution.

Ex: Calculate P(+c | +s)
Generate samples for C and S. If -s, then reject the sample. All samples we keep will have +s, and from those, we can find the ones with +c to calculate P(+c | +s). With 4 variables, it's not a huge time savings. But, with 1 million variables, you ignore the last 999,998 variables.

Problems with rejection sampling
For unlikely evidence, reject lots of samples. Samples for unlikely evidence deep in the Bayes net generates lots of wasted computation for all of the samples needed to get to the unlikely evidence sample.

Likelihood weighting
Likelihood weighting is another sampling technique, where, like rejection sampling, the query is known ahead of time. However, with likelihood weighting, we fix some values to be known, and sample the remaining variables.

Weight the sample using probabilities of known variables, where known variables are the evidence.

Setup:
Variable W = 1.
Variable X = { }.

Evidence nodes:
W = W * P(node | parents), from the table for the node.

For non-evidence nodes:

Sample to get the state. Doesn't contribute to W.

Example:
Evidence = Cloudy = true
W = W * P(c = true) = 1.0 * 0.5 = 0.5
x = {-c}

Sample the remaining variables using c = true
Assume r = false, s = true, w = true
x = {-c, -r, +s, +w}

Sample 1:
x = {-c, -r, +s, +w}, weight = 0.5

Sample 2:
Evidence = Wet grass = true, rain = true
W = 1.0
x = { }

Sample cloudy, assume true
x = {+c}

Sample sprinkler, assume false
x = {+c, -s}

Rain is evidence
W = W * P(+r | +c). We use +c because we've already sampled it.
W = 1.0 * .80 = .80

Wet grass is evidence
W = W * P(+w | +r, -s)
W = 0.8 * 0.9 = 0.72

Sample 2:
x = {+c, -s, +r, +w}, weight = 0.72

Assume the following five samples are generated with the following weights.
x = {-c, -r, +s, +w}, weight = 0.5
x = {+c, -s, +r, +w}, weight = 0.72
x = {-c, +r, -s, +w}, weight = 0.1
x = {-c, -r, +s, +w}, weight = 0.5
x = {+c, -r, +s, -w}, weight = 0.1

To generate probabilities, sum the weights associated with each sample and divide by total weights.
P(+c) = (.72 + .1) / (.5 + .72 + .1 + .5 + .1) = .82 / 1.92 = .43

Problems with likelihood weighting
Sampling can be inconsistent with the evidence.

Ex: P(r | s), given that s is fixed. R is conditioned on C, not s. So, sampling r and fixing s doesn't really reflect the Bayes net structure.