MCMC methods

A simple technique is to introduce a random walk, so

$$\mathbf{w}_{i+1} = \mathbf{w}_i + \epsilon$$

where ϵ is zero mean spherical Gaussian and has small variance. Obviously the sequence $\mathbf{w}_i$ does not have the required distribution. However, we can use the *Metropolis algorithm*, which does *not* accept all the steps in the random walk:

1. If $p(\mathbf{w}_{i+1}|\mathbf{y}) > p(\mathbf{w}_i|\mathbf{y})$ then accept the step.
2. Else accept the step with probability $\frac{p(\mathbf{w}_{i+1}|\mathbf{y})}{p(\mathbf{w}_i|\mathbf{y})}$.

In practice, the Metropolis algorithm has several shortcomings, and a great deal of research exists on improved methods, see:

R. Neal, "Probabilistic inference using Markov chain Monte Carlo methods," University of Toronto, Department of Computer Science Technical Report CRG-TR-93-1, 1993.

Approximate inference for Bayesian networks

MCMC methods also provide a method for performing *approximate inference* in *Bayesian networks*.

Say a system can be in a state s and moves from state to state in discrete time steps according to a probabilistic transition

$$\Pr(\mathrm{s} \rightarrow \mathrm{s}')$$

Let $\pi_t(\mathrm{s})$ be the probability distribution for the state after t steps, so

$$\pi_{t+1}(\mathrm{s}') = \sum_{\mathrm{s}} \Pr(\mathrm{s} \rightarrow \mathrm{s}')\pi_t(\mathrm{s})$$

If at some point we obtain $\pi_{t+1}(\mathrm{s}) = \pi_t(\mathrm{s})$ for all s then we have reached a *stationary distribution* π. In this case

$$\forall \mathrm{s}' \pi(\mathrm{s}') = \sum_{\mathrm{s}} \Pr(\mathrm{s} \rightarrow \mathrm{s}')\pi(\mathrm{s})$$

There is exactly one stationary distribution for a given $\Pr(\mathrm{s} \rightarrow \mathrm{s}')$ provided the latter obeys some simple conditions.

The condition of *detailed balance*

$$\forall s, s' \pi(s)\Pr(s \rightarrow s') = \pi(s')\Pr(s' \rightarrow s)$$

is sufficient to provide a π that is a stationary distribution. To see this simply sum:

$$\begin{aligned}\sum_{s} \pi(s)\Pr(s \rightarrow s') &= \sum_{s} \pi(s')\Pr(s' \rightarrow s) \\ &= \pi(s')\underbrace{\sum_{s} \Pr(s' \rightarrow s)}_{=1} \\ &= \pi(s')\end{aligned}$$

If all this is looking a little familiar, it's because we now have an excellent application for the material in *Mathematical Methods for Computer Science*. That course used the alternative term *local balance*.

Approximate inference for Bayesian networks

Recalling once again the basic equation for performing probabilistic inference

$$\Pr(Q|e) = \frac{1}{Z}\Pr(Q \wedge e) = \frac{1}{Z}\sum_{u}\Pr(Q, u, e)$$

where

- Q is the query variable.
- e is the evidence.
- u are the unobserved variables.
- $1/Z$ normalises the distribution.

We are going to consider obtaining samples from the distribution $\Pr(Q, U|e)$.

Approximate inference for Bayesian networks

The evidence is fixed. Let the *state* of our system be a specific set of values for the *query variable and the unobserved variables*

$$\mathrm{s} = (q, u_1, u_2, \ldots, u_n) = (s_1, s_2, \ldots, s_{n+1})$$

and define $\overline{\mathrm{s}}_i$ to be the state vector *with* s_i *removed*

$$\overline{\mathrm{s}}_i = (s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_{n+1})$$

To move from s to s' we replace one of its elements, say s_i, with a new value s_i' sampled according to

$$s_i' \sim \Pr(S_i | \overline{\mathrm{s}}_i, e)$$

This has detailed balance, and has $\Pr(Q, U | e)$ as its stationary distribution.

Approximate inference for Bayesian networks

To see that $\Pr(\mathrm{Q}, \mathrm{U}|e)$ is the stationary distribution

$$\begin{aligned}
\pi(\mathrm{s})\Pr(\mathrm{s} \to \mathrm{s}') &= \Pr(\mathrm{s}|e)\Pr(s_i'|\bar{\mathrm{s}}_i, e) \\
&= \Pr(s_i, \bar{\mathrm{s}}_i|e)\Pr(s_i'|\bar{\mathrm{s}}_i, e) \\
&= \Pr(s_i|\bar{\mathrm{s}}_i, e)\Pr(\bar{\mathrm{s}}_i|e)\Pr(s_i'|\bar{\mathrm{s}}_i, e) \\
&= \Pr(s_i|\bar{\mathrm{s}}_i, e)\Pr(s_i', \bar{\mathrm{s}}_i|e) \\
&= \Pr(\mathrm{s}' \to \mathrm{s})\pi(\mathrm{s}')
\end{aligned}$$

As a further simplification, sampling from $\Pr(S_i|\bar{\mathrm{s}}_i, e)$ is equivalent to sampling S_i conditional on its parents, children and children's parents.

So:

- We successively sample the query variable and the unobserved variables, conditional on their parents, children and children's parents.
- This gives us a sequence $s_1, s_2, \ldots$ which has been sampled according to $\Pr(Q, U|e)$.

Finally, note that as

$$\Pr(Q|e) = \sum_{u} \Pr(Q, u|e)$$

we can just ignore the values obtained for the unobserved variables. This gives us $q_1, q_2, \ldots$ with

$$q_i \sim \Pr(Q|e)$$

Approximate inference for Bayesian networks

To see that the final step works, consider what happens when we estimate the expected value of some function of Q.

$$\begin{aligned}\mathbb{E}[f(Q)] &= \sum_q f(q)\Pr(q|e)\\ &= \sum_q f(q)\sum_u \Pr(q,u|e)\\ &= \sum_q \sum_u f(q)\Pr(q,u|e)\end{aligned}$$

so sampling using $\Pr(q,u|e)$ and ignoring the values for u obtained works exactly as required.