Lecture 15. Sequential Monte Carlo and Self Avoiding Walk (SAW)

Note Title 4/27/2006

The previous lectures have discussed sampling from distributions of form $\pi(\underline{x}) = \pi(x_1)\pi(x_2|x_1)\pi(x_3|x_2)\ldots\pi(x_d|x_{d-1})$

$x_1 - x_2 - \ldots - x_{d-1} - x_d$

This corresponds to graph structure

→ no closed loops, causal/Markov structure.

Now suppose we have a general distribution $\pi(\underline{x})$ then we can always express it in sequential form:

$\pi(\underline{x}) = \pi(x_1)\pi(x_2|x_1)\pi(x_3|x_1,x_2)\ldots\pi(x_d|x_1\ldots x_{d-1})$

Note: typically $\pi(\underline{x})$ will be specified by a Gibbs distribution $\pi(\underline{x}) = \frac{1}{Z}e^{-E(\underline{x})}$ and so it may be computationally very expensive to translate it in sequential form. Recall that we need to use dynamic programming to translate models like $\pi(\underline{x}) = \frac{1}{Z}e^{-\sum_i \phi_i(x_i,x_{i+1})}$ into sequential form.

(Note: that our earlier models are special cases where $\pi(x_3|x_1,x_2) = \pi(x_3|x_2)$, $\ldots \pi(x_d|x_1,\ldots,x_{d-1}) = \pi(x_d|x_{d-1})$)

If we have $\pi(\underline{x})$ in sequential form then we can sample from it by $x_1^{(1)}$ sampled from $\pi(x_1)$

$x_2^{(1)}$ from $\pi(x_2|x_1^{(1)})$, $x_3^{(1)}$ from $\pi(x_3|x_1^{(1)},x_2^{(1)})$

$\ldots$ $x_d^{(1)}$ from $\pi(x_d|x_1^{(1)},x_2^{(1)},\ldots x_{d-1}^{(1)})$

to give sample $x^{(1)} = (x_1^{(1)}, x_2^{(1)}, \ldots, x_d^{(1)})$.

But often it is hard to sample from $\pi(x_t|x_{t-1},\ldots x_1)$.

Alternatively, we can pick a trial distribution $g(\underline{x})$ which is of form:

$$g(x_1)\,g(x_2|x_1)\,\ldots\,g(x_d|x_1,\ldots,x_{d-1})$$

so that we can sample from each $g(x_t|x_1,\ldots x_{t-1})$

We get a sample $x^{(1)} = (x_1^{(1)},\ldots x_d^{(1)})$ by sampling $x_1^{(1)}$ from $g(x_1)$, $x_2^{(1)}$ from $g(x_2|x_1^{(1)})$, $x_t^{(1)}$ from $g(x_t|x_1^{(1)},\ldots x_{t-1}^{(1)})$, $\ldots x_d^{(1)}$ from $g(x_d|x_1^{(1)}\ldots x_{d-1}^{(1)})$

We assign this sample a weight

$$w(\underline{x}) = \frac{\pi(\underline{x})}{g(\underline{x})}$$

— i.e. importance sampling.

This weight is of form $w(\underline{x}) = \dfrac{\pi(x_1)\,\pi(x_2|x_1)\ldots\pi(x_d|x_1\ldots x_{d-1})}{g_1(x_1)\,g_2(x_2|x_1)\ldots g_d(x_d|x_1,\ldots x_{d-1})}$

and be computed recursively:

$$w_1(x_1) = \frac{\pi(x_1)}{g(x_1)}$$

$$w_t(\underline{x}_t) = w_{t-1}(\underline{x}_{t-1})\,\frac{\pi(x_t|\underline{x}_{t-1})}{g_t(x_t|\underline{x}_{t-1})}$$

where $\underline{x}_t = (x_1,\ldots x_t)$, $\underline{x}_{t-1} = (x_1,\ldots,x_{t-1})$

This method is good if: (a) we know the sequential form of $\pi(\underline{x})$ and (b) we can evaluate terms like $\pi(x_t|\underline{x}_{t-1})$

But often we don't know the sequential form and can't compute it: i.e. we don't know $\pi(x_t|\underline{x}_{t-1})$

Sequential Monte Carlo is a method to obtain weighted samples from $\pi(\underline{x})$ without knowing the sequential form of $\pi(\underline{x})$ (i.e. don't know $\pi(x_t|x_1,..x_{t-1})$.

Instead we need:

(i) a trial distribution

$$g(\underline{x}) = g(x_1)\,g(x_2|x_1)\,\ldots\,g(x_d|x_1,..x_{d-1})$$

(ii) a set of distributions

$\pi_1(\underline{x}_1), \pi_2(\underline{x}_2), \ldots \pi_d(\underline{x}_d)$ (where $\underline{x}_t = (x_1,..x_t)$)

which are approximations to the marginals of $\pi(x)$

i.e. $\pi_t(\underline{x}_t) \sim \pi(\underline{x}_t) = \sum_{x_{t+1},..x_d} \pi(x_1,..x_t x_{t+1},..x_d)$

such that $\pi_d(\underline{x}_d) = \pi(\underline{x}_d)$. //

SIS Algorithm:

(A) Draw sample $x_t^{(i)}$ from $g(x_t|\underline{x}_{t-1}^{(i)})$

and set $\underline{x}_t^{(i)} = (\underline{x}_{t-1}^{(i)}, x_t^{(i)})$

(B) Compute $u_t^{(i)} = \dfrac{\pi_t(\underline{x}_t^{(i)})}{\pi_{t-1}(\underline{x}_{t-1}^{(i)})\, g(x_t^{(i)}|\underline{x}_{t-1}^{(i)})}$

and set $w_t^{(i)} = w_{t-1}^{(i)}\, u_t^{(i)}$ ← distribution you want to sample from

Observe that, because $\pi_d(\underline{x}_d) = \pi(\underline{x}_d)$, that

$w_d^{(i)} = \dfrac{\pi(\underline{x}_d)}{g(\underline{x}_d)}$ (i.e. the correct weight for importance sampling.

(cancellation $\dfrac{\pi_t(x_t^{(i)})}{\pi_{t-1}(x_{t-1}^{(i)})} \cdot \dfrac{\pi_{t+1}(x_t^{(i)})}{\pi_t(x_t^{(i)})}$)

But this is only efficient if the $\pi_t(\underline{x}_t)$ are good approximations to $\pi(\underline{x}_t)$ and the trial dist. $g(x_t|\underline{x}_{t-1})$ is good. (problem specific)

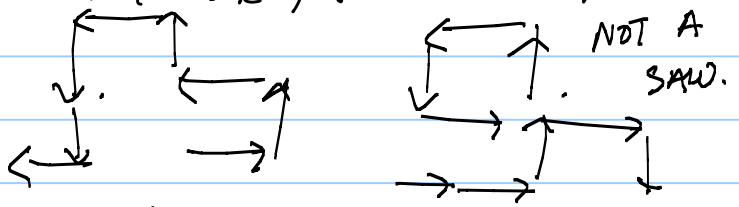
Example: Self-Avoiding Walk (SAW)

This is an idealized model of a polymer/protein. It is a random walk in a two-dimensional lattice which must avoid any lattice site that has already been visited.

$\underline{X} = (X_1, \ldots, X_N)$ $\quad X_i = (a, b)$ $\quad a$ & b are integers

The distance between X_i and X_{i+1} must be 1. So X_{i+1} must be a neighbouring point on the lattice: $(a \pm 1, b)$, $(a, b \pm 1)$.

- Also $X_{i+1} \neq X_k$, for $k = 1 \ldots, i$ (self-avoiding)



The target distribution $\pi(\underline{x})$ is uniform over all self-avoiding paths of length N. So $\pi(\underline{x}) = 1/Z_N$ ← constant.

How to sample a SAW?

Simplest Idea: start at $(0,0)$

At each time step, choose a neighbor with equal probability if the chosen neighbor has been occupied, then abort the entire walk. Return to $(0,0)$ and start again.

This is very inefficient because many partial walks have to be aborted.

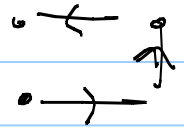
Next Try: Let $X_1 = (0,0)$.

- At time t, examine all neighbors of $X_t = (i,j)$ (ie. $(i\pm1, j)$ & $(i, j\pm1)$).
- If all neighbors have been occupied, then abort.
- Otherwise, select one of the unoccupied neighbors with equal probability.

This is better because fewer walks are aborted, BUT IT DOESN'T GENERATE UNIFORMLY DISTRIBUTED SAWS — and so does not generate samples from $\pi(X)$.

To see this observe that

Prob of generating walk (A) $= \frac{1}{4} \times \frac{1}{3} \times \frac{1}{3} \times \frac{1}{2} = 1/72$



Prob of generating walk (B) $= \frac{1}{4} \times \frac{1}{3} \times \frac{1}{3} \times \frac{1}{3} = 1/108$

So the process above is biased.

But we can use it as a trial distribution.

$g(X_{t+1} = (i',j') \mid X_1 \ldots X_t) = 1/n_t$

where (i',j') is any occupied neighbor of $X_t = (i,j)$ and n_t is the number of un-occupied neighbors.

Intuitively, should assign a weight $w(X_d) = n_1 n_2 \ldots n_{N-1}$ to each sample X_d to remove the bias — ie. importance sampling.

Claim this intuition is simply SIS with trial dist. $g(\cdot|\cdot)$ and approximates distribution $\pi_t(X_t) = 1/Z_t$ (ie. SAW of length t).

page 6.

Justify this claim.

$g(X_t | X_1 \ldots X_{t-1}) = 1/n_t$ (see page 5 half way down)

$\Pi_t(\underline{X}_t) = \frac{1}{Z_t}$ is distribution of SAW of length t

Clearly $\Pi_N(\underline{X}_N) = \frac{1}{Z_N} = \Pi(\underline{X})$ as required for the approximate distribution

SIS updates the weights by:

$$w_t = \frac{w_{t-1}\,\Pi_t(\underline{X}_t)}{\Pi_{t-1}(\underline{X}_{t-1})\, g(X_t | X_1 \ldots X_{t-1})} = w_{t-1}\, n_t \frac{Z_{t-1}}{Z_t}$$

Hence $w(\underline{X}) = \frac{n_1 \cdots n_{N-1}}{Z_N} \propto n_1 \cdots n_{N-1}$ (proportional to)

This justifies our claim that the "intuitive algorithm" on page 5 is an SIS algorithm and hence performs importance sampling for SAW's.

This is called 1-step lookahead, or Rosenbleth's algorithm.

Note: $g_t(X_{t+1} | \underline{X}_t)$ is related to $\Pi_t(\underline{X}_t)$ because

$$\Pi_{t+1}(\underline{X}_t) = \sum_{X_{t+1}} \Pi_{t+1}(\underline{X}_{t+1}) = n_t / Z_{t+1}$$

so $$g_t(X_{t+1} | \underline{X}_t) = \frac{\Pi_{t+1}(\underline{X}_{t+1})}{\Pi_{t+1}(\underline{X}_t)} = \Pi_{t+1}(X_{t+1} | \underline{X}_t)$$