Notation: $\underline{X}_t = (X_1, \ldots, X_t)$

Note Title 4/27/2006

# Sequential Monte Carlo.

The Book says Kalman / Particle Filtering methods are special cases of what Liu calls Sequential Monte Carlo.

$$\underline{X} = (X_1, X_2 \ldots, X_d)$$

We can always express the full distribution as $\pi(\underline{X}) = \pi(X_1)\pi(X_2|X_1)\pi(X_3|X_1,X_2) \ldots \pi(X_d|X_1 \ldots X_{d-1})$.

(Previous lectures treated the special Markov case where $\pi(X_3|X_1,X_2) = \pi(X_3|X_2)$, $\pi(X_4|X_1,X_2,X_3) = \pi(X_4|X_3)$)

We can choose a trial (importance) distribution:

$$g(\underline{X}) = g_1(X_1)\, g_2(X_2|X_1)\, g_3(X_3|X_1,X_2) \cdots g_d(X_d|X_1 \ldots X_{d-1})$$

The importance weight $w(\underline{X}) = \dfrac{\pi(X_1)\pi(X_2|X_1)\ldots\pi(X_d|X_1\ldots X_{d-1})}{g_1(X_1)g_2(X_2|X_1)\ldots g_d(X_d|X_1,\ldots X_{d-1})}$

can be computed recursively. $w_t(\underline{X}_t) = w_{t-1}(\underline{X}_{t-1}) \dfrac{\pi(X_t|\underline{X}_{t-1})}{g_t(X_t|\underline{X}_{t-1})}$

(2)

(Each $x_i$ can take $k$ values)

Strategy: sample $x_1$ from $g_1(x_1)$, $x_2$ from $g_2(x_2|x_1)$, $x_3$ from $g_3(x_3|x_1,x_2)$, etc. to get a sample $x_1,\dots,x_d$ with weight $w_d(\underline{x}_d)$

But this is usually impractical, because it may be computationally intractible (e.g. $O(k^d)$ operations) to determine the $\pi(x_t|\underline{x}_{t-1})$ (even if $\pi(\underline{x})$ is known)

Instead, find good approximations $\pi_1(x_1), \pi_2(x_2) \dots \pi_d(x_d)$

so that each $\pi_t(\underline{x}_t)$ is a good approximation to the marginal $\pi(\underline{x}_t)$.

and $\pi_d(\underline{x}_d) = \pi(\underline{x}_d)$

(Only need to know $\pi_t(\underline{x}_t)$ up to a normalization constant)

SIS Step:

(A) Draw $X_t = x_t$ from $g(x_t|\underline{x}_{t-1})$ and let $\underline{x}_t = (\underline{x}_{t-1}, x_t)$.

(B) Compute $$u_t = \frac{\pi_t(\underline{x}_t)}{\pi_{t-1}(\underline{x}_{t-1})\, g_t(x_t|\underline{x}_{t-1})}$$

and let $w_t = w_{t-1} u_t$

But how to find good $\pi_t(\underline{x}_t)$? This is problem specific.

(3)

## Example: Self-Avoiding Walk (SAW)

Biopolymer - polyester
- polyethylene

2-D lattice model.

An "idealized" polymer of length $N$

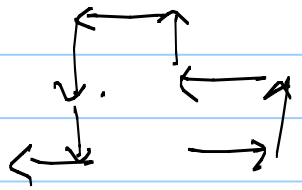
$\underline{x} = (x_1, \ldots, x_N)$ $\quad x_i = (a, b)$ $a, b$ integers.

Distance between $x_i$ & $x_{i+1}$ must be 1.

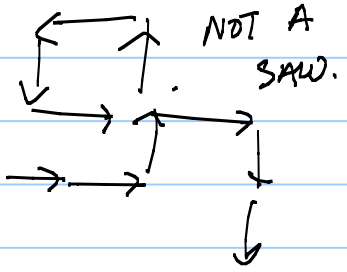
$x_{i+1} \neq x_k$, $\forall\ k < i$

Line connecting $x_i$ and $x_{i+1}$ is (covalent) bond.

SAW.



NOT A SAW.



The target distribution is the uniform distribution $\pi(\underline{x}) = 1/Z_N$ ← constant.

### Simplest Design:

Start at (0,0)

At each step, choose one of three neighbors with equal probability.

If that neighbor has been visited, then abort - go back to (0,0) and start again.

Inefficient - many walks are aborted.

(4)

"Rosenbluth method" one-step-look-ahead.

Let $X_1 = (0,0)$ & $X_2 = (1,0)$

At time $t$, examine all neighbors of $X_t = (i,j)$

(i.e. $(i \pm 1, j)$ & $(i, j \pm 1)$)

If all neighbors have been occupied — abort

Otherwise, select one of the available (i.e. unoccupied) neighbors. with equal probability

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

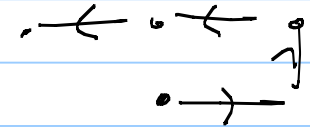
$(i',j')$ is unoccupied neighbor of $X_t$

$n_t$ — no. of unoccupied neighbors.

These walks are rarely aborted — BUT THIS METHOD DOES NOT GENERATE UNIFORMLY DISTRIBUTED SAW'S.

Probability of generating

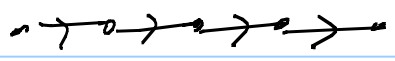
(A)



$\frac{1}{4} \times \frac{1}{3} \times \frac{2}{3} \times \frac{1}{2}$

$= \frac{1}{72}$

(B)



$\frac{1}{4} \times \frac{1}{3} \times \frac{1}{3} \times \frac{1}{3}$

$= \frac{1}{108}$

Must correct for this bias.

(5)

Note: Some errors in lecture. Corrected in these notes.

Intuitively,

To correct assign a weight:

$$w(\underline{x}) = n_1 \times n_2 \times \ldots \times n_{d-1}$$

The probability of generating the sample is

$$g_1(x_1) g_2(x_2|x_1) g(x_3|x_1,x_2) \ldots = \frac{1}{n_1 n_2 \ldots n_{d-1}}$$

So the weights balance the probabilities.

Alternatively, - in terms of sequential Monte Carlo.

select $\Pi_t(\underline{x}_t)$ to be the uniform distribution on paths of length $t$, $\Pi_t(\underline{x}_t) = 1/Z_t \leftarrow$ constant

Then $g_t(x_{t+1}|\underline{x}_t) = 1/n_t$

weight update. $w_t = w_{t-1} \frac{(1/Z_t)}{(1/Z_{t-1})(1/n_t)} = w_{t-1} n_t \frac{Z_{t-1}}{Z_t}$.

Hence $w(\underline{x}) \propto n_1 \times n_2 \times \ldots \times n_{d-1}$ (normalization does not matter for weights - because we can normalize by $\sum_i w(\underline{x}^{(i)})$, as for importance sampling)

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 $g_t(x_{t+1}|\underline{x}_t) = \frac{\Pi_{t+1}(\underline{x}_{t+1})}{\Pi_{t+1}(\underline{x}_t)} = \Pi_{t+1}(x_{t+1}|\underline{x}_t)$

Called 1-step lookahead. //