# Sequential Importance Sampling

STAT 525

9/18/18

## Review: Importance Sampling

Procedure: Write $\mu = \int h(\mathbf{x})\pi(\mathbf{x})d\mathbf{x} = \int \left[h(\mathbf{x})\frac{\pi(\mathbf{x})}{g(\mathbf{x})}\right] g(\mathbf{x})d\mathbf{x}$.

- Draw $\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(m)}$ i.i.d. from a *proposal distribution* $g(\mathbf{x})$;
- Calculate the *importance weights*

$$w^{(i)} = \frac{\pi(\mathbf{x}^{(i)})}{g(\mathbf{x}^{(i)})}, \text{ for } i = 1, \cdots, m.$$

- Estimate $\mu$ by

$$\hat{\mu} = \frac{w^{(1)}h(\mathbf{x}^{(1)}) + \cdots + w^{(m)}h(\mathbf{x}^{(m)})}{m}$$

$$\text{or} \quad \tilde{\mu} = \frac{w^{(1)}h(\mathbf{x}^{(1)}) + \cdots + w^{(m)}h(\mathbf{x}^{(m)})}{w^{(1)} + \cdots + w^{(m)}}.$$

- Importance sampling is very useful for Monte Carlo computaiton with high-dimensional models, such as those in statistical physics, molecular simulation, and Bayesian statistics.
- However it’s hard to find a good proposal distribution in high dimensional problems.
- In high dimensional models, the “important” region is often very small.

## Sequential Importance Sampling (SIS)

- Build up proposal $g(\mathbf{x})$ sequentially. “Divide and Conquer”.
- Basic framework: Write $\mathbf{x} = (x_1, \cdots, x_d)$,

$$\boxed{\begin{aligned} g(\mathbf{x}) &= g_1(x_1)g_2(x_2|x_1)\cdots g_d(x_d|x_1,\cdots,x_{d-1}) \\ \pi(\mathbf{x}) &= \pi(x_1)\pi(x_2|x_1)\cdots\pi(x_d|x_1,\cdots,x_{d-1}) \end{aligned}}$$

$$w(\mathbf{x}) = \frac{\pi(x_1)\pi(x_2|x_1)\cdots\pi(x_d|x_1,\cdots,x_{d-1})}{g_1(x_1)g_2(x_2|x_1)\cdots g_d(x_d|x_1,\cdots,x_{d-1})}$$

- Define *current weight* as

$$w_t(\mathbf{x}) = \frac{\pi(x_1)\pi(x_2|x_1)\cdots\pi(x_t|x_1,\cdots,x_{t-1})}{g_1(x_1)g_2(x_2|x_1)\cdots g_t(x_t|x_1,\cdots,x_{t-1})}$$

then

$$w_t(\mathbf{x}) = w_{t-1}(\mathbf{x})\frac{\pi(x_t|x_1,\cdots,x_{t-1})}{g_t(x_t|x_1,\cdots,x_{t-1})}.$$

- In order to get $\pi(x_1), \pi(x_2|x_1), \cdots, \pi(x_t|x_1, \cdots, x_{t-1})$ , one needs to have the marginal distribution

$$\pi(x_1, \ldots, x_t) = \int \pi(x_1, \ldots, x_d) dx_{t+1} \cdots dx_d,$$

  whose computation involves integrating out components $x_{t+1}, \ldots, x_d$ in $\pi(\mathbf{x})$ and is often as difficult as the orginal problem.

- Suppose we can find a sequence of “auxiliary distributions,” $\pi_1(x_1), \pi_2(x_1, x_2), \ldots, \pi_d(x_1, \ldots, x_d)$, so that

$$\pi_t(x_1, \ldots, x_t) \approx \pi(x_1, \ldots, x_t) \ \text{ for } t = 1, \ldots, d-1,$$

and $\pi_d(x_1, \ldots, x_d) = \pi(x_1, \ldots, x_d)$. Then we can write

$$w_t(\mathbf{x}) = w_{t-1}(\mathbf{x}) \frac{\pi_t(x_1, \ldots, x_t)}{\pi_{t-1}(x_1, \ldots, x_{t-1}) g_t(x_t | x_1, \ldots, x_{t-1})}.$$

- The final weight is still correct

$$\begin{aligned} w_d(\mathbf{x}) &= \frac{\pi_d(x_1, \ldots, x_d)}{g_1(x_1) g_2(x_2|x_1) \cdots g_d(x_d|x_1, \cdots, x_{d-1})} \\ &= \frac{\pi(x_1, \cdots, x_d)}{g_1(x_1) g_2(x_2|x_1) \cdots g_d(x_d|x_1, \cdots, x_{d-1})}. \end{aligned}$$

- The “auxiliary distributions” $\pi_1(x_1), \pi_2(x_1, x_2), \ldots$ are only required to be known up to a normalizing constant, and they *only* serve as “guides” to our construction of the whole sample $\mathbf{x} = (x_1, \ldots, x_d)$.

## SIS Procedure

Let $w_0 = 1$. For $t = 1$ to $d$:

a. Draw $x_t$ from $g_t(\cdot | x_1, \ldots, x_{t-1})$.

b. Update the weight $w_t = w_{t-1} u_t$, where the “incremental weight” $u_t$ is

$$u_t = \frac{\pi_t(x_1, \ldots, x_t)}{\pi_{t-1}(x_1, \ldots, x_{t-1}) g_t(x_t | x_1, \ldots, x_{t-1})},$$

where $\pi_t(x_1, \ldots, x_t)$ is a reasonable approximation to the marginal distribution $\pi(x_1, \ldots, x_t)$, for $t = 1, \ldots, d-1$, and $\pi_d(x_1, \ldots, x_d) = \pi(x_1, \ldots, x_d)$.

## Choice of the Sampling Distribution

- Very important to the efficiency of SIS
- In many problems, a good choice of $g_t$ in light of the sequence of auxiliary distributions $\pi_t$ is

$$g_t(x_t \mid x_1, \dots, x_{t-1}) = \pi_t(x_t \mid x_1, \dots, x_{t-1}),$$

with the incremental weight

$$\begin{aligned} u_t &= \frac{\pi_t(x_1, \dots, x_t)}{\pi_{t-1}(x_1, \dots, x_{t-1})\pi_t(x_t|x_1, \dots, x_{t-1})} \\ &= \frac{\pi_t(x_1, \dots, x_{t-1})}{\pi_{t-1}(x_1, \dots, x_{t-1})}. \end{aligned}$$

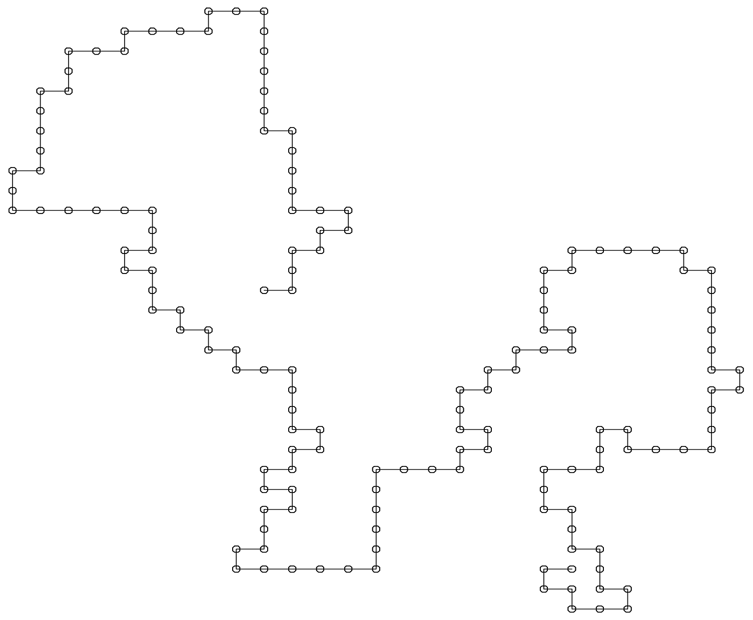
## Application in Molecular Simulation

- Although the history goes back to 1950s, the problem of simulating a long chain of biopolymer still presents a major challenge to the scientific community because of both its difficult nature and its extreme importance in biology and chemistry.

## Self-Avoiding Walk

A simple chain polymer model is the self-avoiding walk (SAW) in a two-dimensional lattice space.

A Self-Avoiding Walk of Length N=150



- The realization of a chain polymer of length $d$ is fully characterized by the positions of all its molecules, $\mathbf{x} = (x_0, x_1, \ldots, x_d)$, where $x_i$ is a point in the 2-D lattice space, i.e. $x_i = (a_i, b_i)$, where $a_i$ and $b_i$ are integers.
- The distance between $x_i$ and $x_{i+1}$ has to be exactly one, and $x_{i+1} \neq x_k$ for all $k < i + 1$.

- A physical model for the simple chain polymer configuration is the *uniform* distribution. i.e., the target distribution of interest is

$$\pi(\mathbf{x}) = \frac{1}{Z_d},$$

  where the space of $\mathbf{x}$ is the set of all SAWs of length $d$ and $Z_d$ is the normalizing constant which equals to the total number of different SAWs of length $d$.

- $Z_d$ is large. For example, $Z_{29} \approx 6.3 \times 10^{12}$.

- Of interest to scientists is to understand certain descriptive statistics regarding such SAWs. For example, one may be interested in $E\|x_d - x_0\|^2$, i.e. the mean squared extension of the chain.

- Without loss of generality, we assume that the simulated SAW is always started at position $(0, 0)$. That is, we let $x_0 = (0, 0)$.
- The most naive way of simulating a SAW: Start the walk at $x_0 = (0, 0)$, and at each step $i$, choose with equal probability one of the 3 allowed neighboring positions to move. If that position has already been visited before, the walker has to start with a completely new chain at $(0, 0)$. Otherwise, the walker can keep going until the presumed length $d$ is reached.

- This simulation procedure is inefficient, with the success rate of obtaining a legal SAW decreases exponentially. For the 2-dim lattice, this rate is roughly $\sigma_d = \frac{Z_d}{4\times 3^{d-1}}$.
- For $d = 20$, we estimated that $\sigma_{20} \approx$ 19.3%, and for $d = 48$, this number is as low as 0.79% (Lyklema and Kremer, 1986).

## One-Step-Look-Ahead

- Start with $x_0 = (0, 0)$. Suppose at stage $t$, we are at position $x_t = (i, j)$.
- Then in order to place $x_{t+1}$, the walker first examines all the neighbors of $x_t$, i.e., $(i \pm 1, j)$ and $(i, j \pm 1)$. If all the neighbor has been visited before, the walk is terminated and given a weight $0$, otherwise the walker selects one of the available positions (not visited before) with equal probability and places his $(t + 1)$th step.

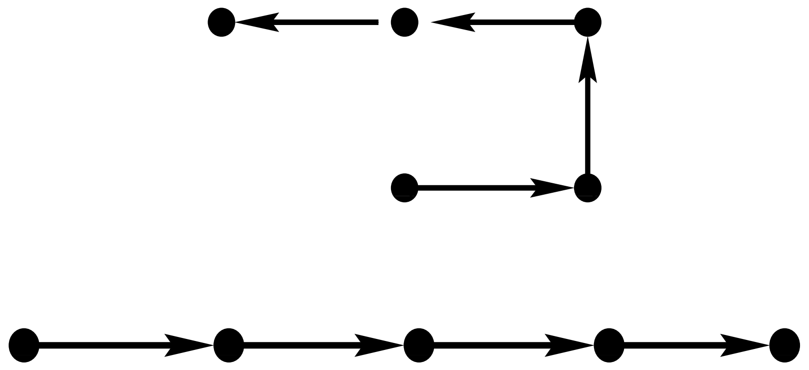
- Mathematically, this sampling method is to draw the position of $x_{t+1}$ conditional on the current configuration of $(x_0, x_1, \ldots, x_t)$, according to the probability distribution

$$P(x_{t+1} = (i', j') \mid x_0, x_1, \ldots, x_t) = \frac{1}{n_t},$$

  where $(i', j')$ is one of the unoccupied neighbors of $x_t$ and $n_t$ is the total number of such unoccupied neighbors.

- This scheme has higher success rate than the simple random walk method.

- However the SAWs produced by this “growth” method is not uniformly distributed.
- For example, the probability of generating the first chain by the growth method is $\frac{1}{4} \times \frac{1}{3} \times \frac{1}{3} \times \frac{1}{2}$, whereas the probability for generating the second chain is $\frac{1}{4} \times \frac{1}{3} \times \frac{1}{3} \times \frac{1}{3}$.



- To correct for the bias, we need to assign each chain a weight computed as

$$w(\mathbf{x}) = \frac{\pi(\mathbf{x})}{g(\mathbf{x})} \propto n_0 \times n_1 \times \cdots \times n_{d-1}.$$

  Because the target distribution is $\pi(\mathbf{x}) \propto 1$ and the sampling distribution of $\mathbf{x}$ is $(n_0 \times n_1 \times \cdots \times n_{d-1})^{-1}$.

- This is a sequential importance sampling procedure.

## Comparison of Two Methods

- Success rate:

| | $d$ | |
|---|---|---|
| | 20 | 50 |
| Naive method | 19.3% | $<$1% |
| One-step-look-ahead | 91% | 58% |

- For $d = 20$, one-step-look-ahead method estimated the mean squared extension to be 71.7 based on 10,000 samples which only took minutes. The naive method took hours.

## References

- Sections 2.6.3 and 3.1 of Jun Liu’s *Monte Carlo Strategies in Scientific Computing*.