Monte Carlo Methods Homework 2

Bernice Feng (bf1318@nyu.edu)

Exercise 28. *Write a routine to use $\mathcal{N}(0,1)$ random variables to generate approximate $\mathcal{N}(0,\sigma^2)$ random variables via an application of each of the three resampling methods (multinomial, Bernoulli, and systematic) discussed in this section. Numerically estimate the variance of the $N^{(k)}$ from each method. What do you observe? Are your observations robust to changing σ^2? Note that this test corresponds to a single resampling step: first sample from $\mathcal{N}(0,1)$, then weight the samples by the appropriate normalized importance weights, then resample.*

The pdf for $N(0,1)$ distribution is

$$\pi'(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

The pdf for $N(0,\sigma^2)$ distribution is

$$\pi(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x}{\sigma})^2}$$

Thus, the weight for the resampling process is

$$w(x) = \frac{\pi}{\pi'} = \frac{1}{\sigma} e^{\frac{x^2}{2} - \frac{x^2}{2\sigma^2}}$$

We calculate the variance of important sampling estimator N^k, where $k = 1,2,3,4,5$ and $\sigma = 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75$. The resampling method used in figure [1], figure [2], and figure [3], are multinomial resampling, Bernoulli resampling, and systematic resampling, respectively.

k\sigma	0.25	0.5	0.75	1.0	1.25	1.5	1.75
1	1.578e-05	4.405e-05	9.943e-05	0.00020141	0.0006259	0.00684042	0.0506068
2	1.71e-06	2.047e-05	9.513e-05	0.00040941	0.00381775	0.1149309	0.91334511
3	5.4e-07	2.791e-05	0.00039016	0.00297821	0.07536336	3.47854	30.37326789
4	2.1e-07	4.495e-05	0.00118487	0.01960656	1.23672998	80.45757538	706.15395439
5	1.1e-07	9.69e-05	0.00659407	0.18592521	28.90503551	2239.57898931	20346.80934978

Figur 1: Variances of important sampling estimator under various k and σ using multinomial resampling

k\sigma	0.25	0.5	0.75	1.0	1.25	1.5	1.75
1	1e-05	2.796e-05	5.757e-05	9.792e-05	0.0004235	0.00423679	0.02778263
2	1.22e-06	1.424e-05	6.677e-05	0.00019955	0.00287333	0.05820151	0.42056995
3	4e-07	2.381e-05	0.00024894	0.00152244	0.05306099	1.56891056	12.81982162
4	1.6e-07	3.699e-05	0.00092988	0.00983916	0.82315142	30.50959228	253.99962609
5	9e-08	9.608e-05	0.00503946	0.09498541	16.77857166	751.74729533	6647.56356316

Figur 2: Variances of important sampling estimator under various k and σ using Bernoulli resampling

k\sigma	0.25	0.5	0.75	1.0	1.25	1.5	1.75
1	1.046e-05	3.038e-05	5.863e-05	9.092e-05	0.0004051	0.00432545	0.02931664
2	1.04e-06	1.377e-05	7.007e-05	0.00021015	0.00291815	0.0616427	0.45996504
3	3.8e-07	2.333e-05	0.00027876	0.00143784	0.05346047	1.68881474	14.16525748
4	1.4e-07	4.073e-05	0.00097547	0.0099545	0.83979794	33.97220787	292.7664714
5	9e-08	9.142e-05	0.00537666	0.09356557	17.2272148	853.65246107	7778.54917287

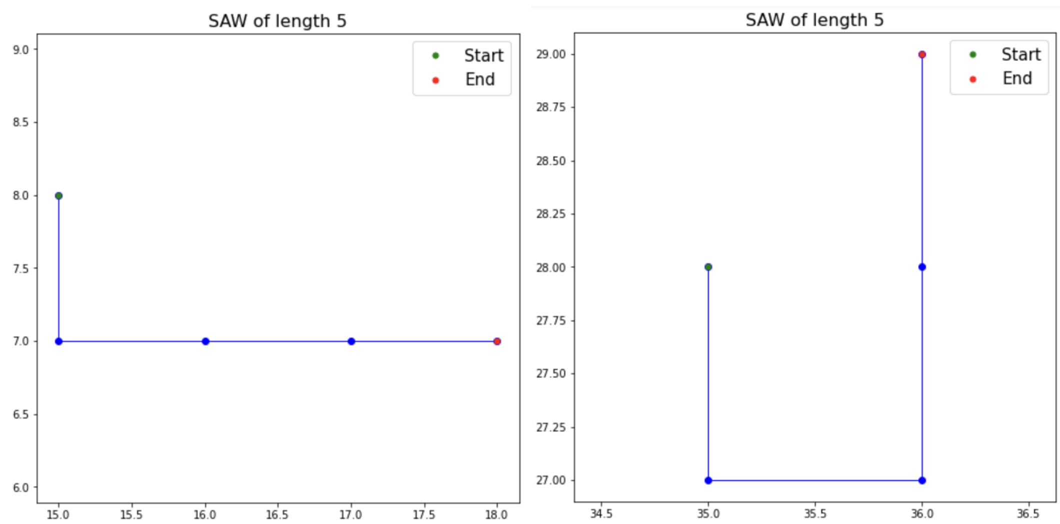
Figur 3: Variances of important sampling estimator under various k and σ using systematic resampling

From all the three tables above, I observe that variances of N^k for $k = 1,2,3,4,5$ all increase as σ increases. This observation is robust to changing σ^2, because it holds true for all the σs tried. However, for fixed *sigma*, the variances increase as k increases only when $\sigma \geq 0.75$; they actually decrease when $\sigma = 0.25$, and increase after decrease when $\sigma = 0.5$.

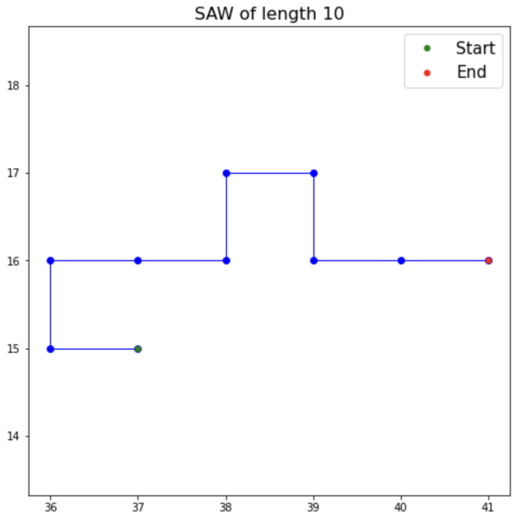
Comparing the three resampling methods, the difference in variance is not very significant when σ and k are small, but becomes more and more significant as σ and k grow larger. The formula for multinomial resmapling is $Var(N^k) = NW_n^k(1-W_n^k)$ of $\mathscr{O}(N)$, and formula for Bernoulli resmapling is $Var(N^k) = (\lceil NW_n^k \rceil - NW_n^k)(NW_n^k - \lfloor NW_n^k \rfloor)$ of $\mathscr{O}(1)$, agree with the numerical results.

Exercise 29. *Write a routine that uses sequential importance sampling with and without resampling to compute averages with respect to the uniform measure on SAW(d) for a sequence of increasing d. Use the same reference density described in Example 11 in both schemes. Produce plots of a single sample path for each value of d. In validating your algorithms you may find it useful to check, e.g. the expected number of times a lattice site is visited (this should be the same for every site). Can you think of other statistics that might help you validate/debug your codes? How can you compare the two methods? Can you think of a way to use these simulations to estimate the normalization constants* $\mathcal{Z}_d$*? Estimate how quickly* $\mathcal{Z}_d$ *grows with d.*

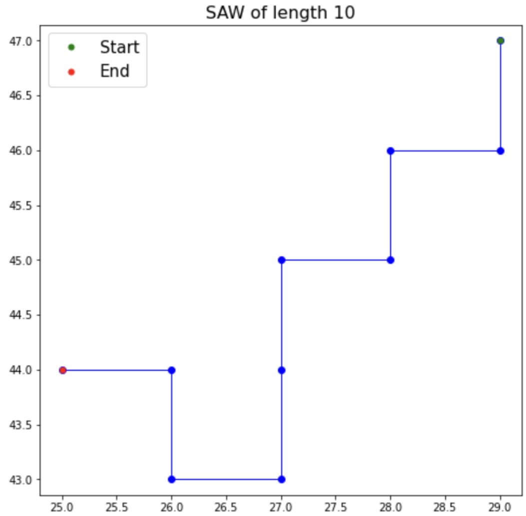
Here are plots of a sample path for given d:



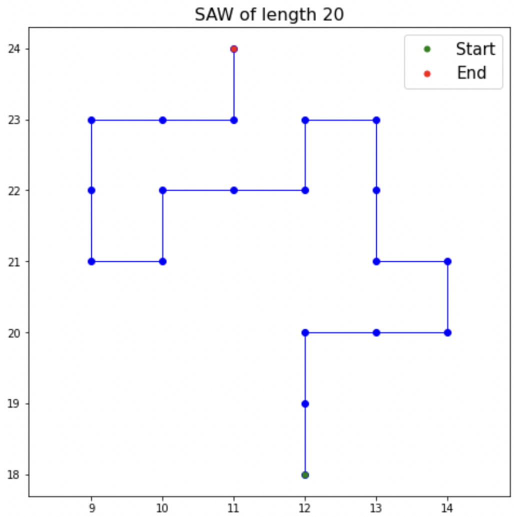
(a) Without resampling (b) With resampling



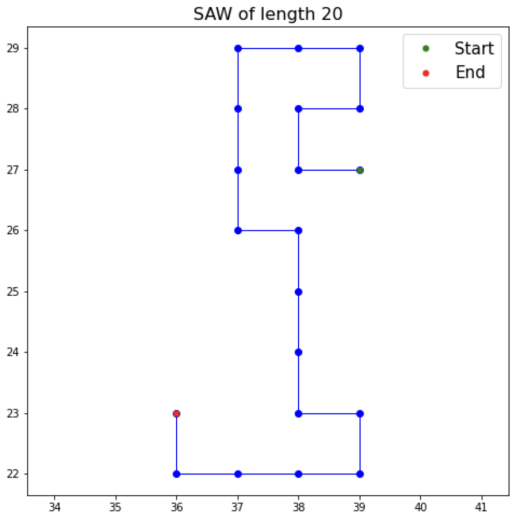
(c) Without resampling



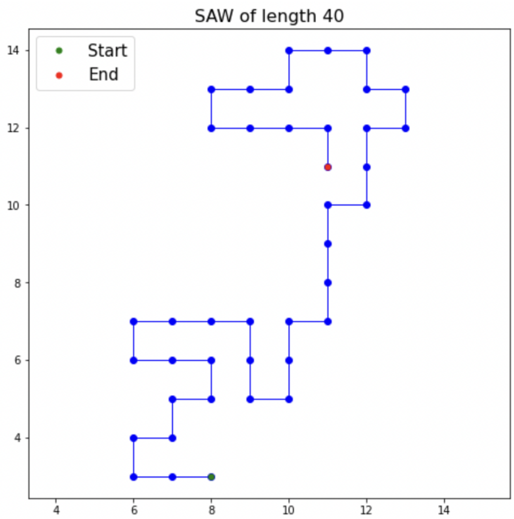
(d) With resampling



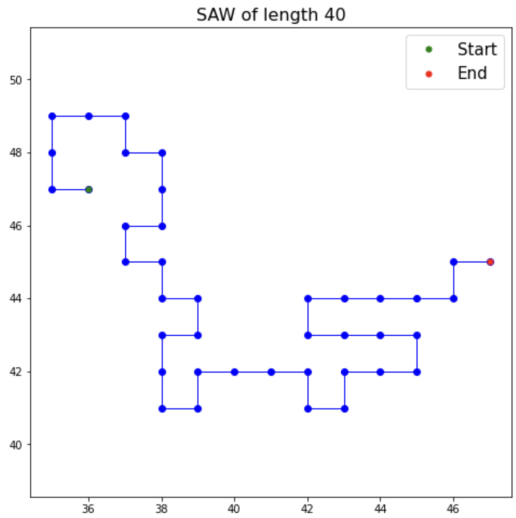
(e) Without resampling



(f) With resampling

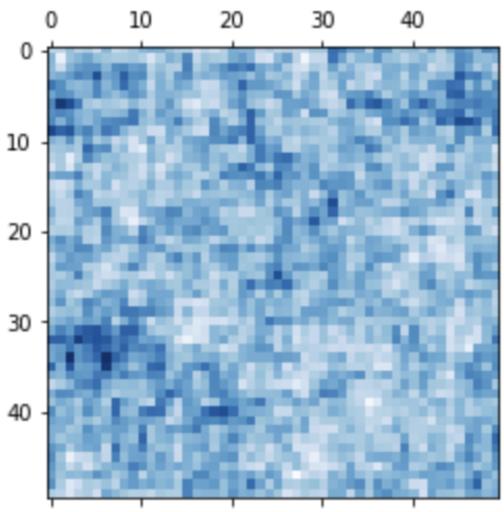


(g) Without resampling

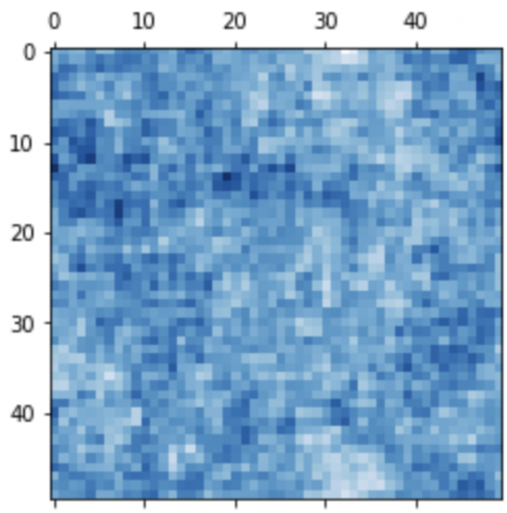


(h) With resampling

Here in figure [4] we checked the expected number of times a lattice site is visited. Darker color indicates more time visited, while light color indicates less time visited. We can see that the sites in the SAW algorithm with resampling are visited more evenly compared to the one without resampling.



(i) Without resampling

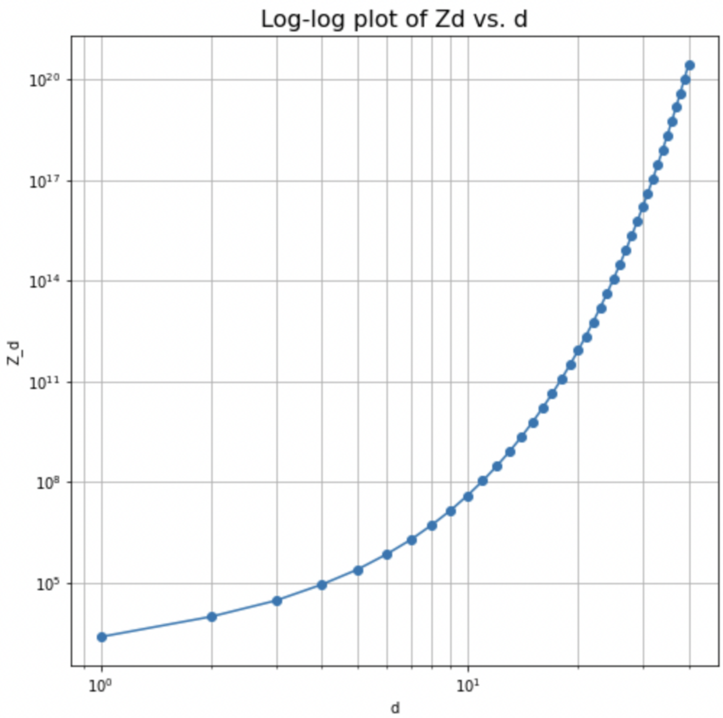


(j) With resampling

Figur 4: Number of time a lattice site is visited

We could also validate the code by checking the distribution of the position of each step, especially the ending position. We could compare the two methods by how fast they are and how accurate they are. Specifically, we could record the time that it takes for running each method, and we could check the variance of the expected number of times all lattice sites are visited.

Here's how Z_d can be calculated: The base case is $d = 1$, where in a $L \times L$ lattice, there are L^2 SAW simple random walks. Assume we have the normalizing constant $\tilde{Z}_n$ for SAW of length n, and we can use the SAW algorithm (with resampling) to generate a collection of $SAW(n)$, where we count the number of possible direction for each sample and calculate their sum M. Thus, estimation for $\tilde{Z}_{n+1}$ is $\frac{1}{N}\tilde{Z}_n M$.



Figur 5: Log-log plot for studying how quickly Z_d grows with d

We can see from figure [5] that Z_d grows with d exponentially, which is especially obvious for small d. For larger d, since there are less choice to go because of self-avoidingness, the proportion of self-avoiding walks will go to 0 quite fast as d goes to ∞.