Exercises NMSM 2024-2025

Name Surname
matr. 0123456

November 9, 2024

Lecture 1: Crude Monte Carlo and inversion sampling

Sampling random points within D-dimensional domains by hit and miss

Sampling random numbers from a given distribution: inversion method We start from the following PDF's:

$$\rho_1(x) = cxe^{-x^2} \,, x \in \mathbb{R}^+ \qquad \rho_2(x) = bx^4 \,, x \in [0,3]$$

We first proceed by doing the normalization to find the coefficients c and b:

$$\int_{\mathbb{R}^+} \rho_1(x)dx = 1 \iff c = 2 \qquad \int_0^3 \rho_2(x)dx = 1 \iff b = \frac{5}{243}$$

Then one can evaluate the CDF's by definition:

$$F_1(t) = \int_0^t \rho_1(x)dx = 1 - e^{-t^2}, \ t \in \mathbb{R}^+ \qquad F_2(t) = \int_0^t \rho_2(x)dx = \frac{t^5}{243}, \ t \in [0,3]$$

Finally by inversion:

$$F_1^{-1}(y) = \sqrt{\ln\left(\frac{1}{1-y}\right)} \qquad F_2^{-1}(y) = \sqrt[5]{243y}$$

For both $y \in (0,1]$.
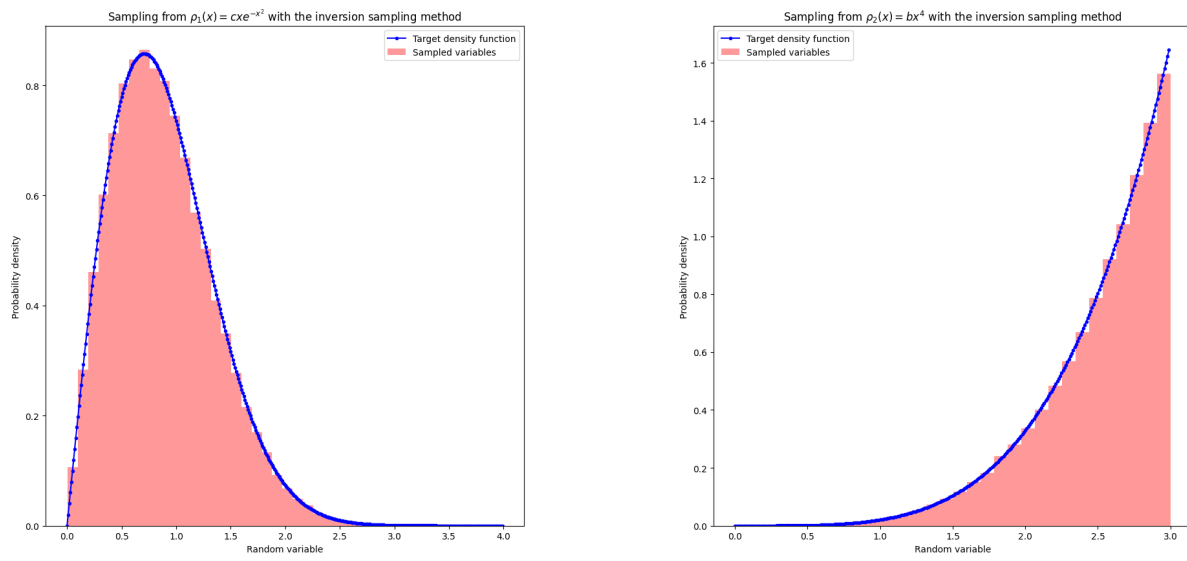We report in the following the obtained graphs:

Figure 1.1: Sampling from distribution ρ_1 and ρ_2 using the inversion method.

Lecture 2: Rejection method

The rejection sampling method is useful when our pdf is not easily invertible, and so the inversion method cannot be applied.
Suppose to have as pdf:

$$\rho(x) = A_f \exp\left(-8\left(\frac{x^2}{2} + \frac{x^4}{4}\right)\right)$$

Imposing the normalization condition we obtain:

$$\int_{\mathbb{R}} \rho(x)dx = 1 \qquad \Longleftrightarrow \qquad \frac{eA_f}{\sqrt{2}}K_{1/4}(1) = 1$$

Where we denote with $K_{1/4}(x)$ the Bessel function of the second kind. We find that $A_f \simeq 1.21$. Now we want to find an invertible pdf $g(x)$ (that we'll call candidate density) with the property:

$$\rho(x) \leq cg(x) \qquad \forall x \in S$$

Where S is the sample space of $\rho(x)$. The closer the function $cg(x)$ is to $\rho(x)$ the more efficient the sampling is.

There can be different ways to choose the candidate density. We explore different possibilities in 1D and higher dimensions.
First we must start with a guess for the candidate density and the c parameter. Then we see if this guess respects the proposed inequality: if it does we are done, if not we do another guess and try again.
In 1D the easiest way to to test our guess is the graphical one: we plot $\rho(x)$ and $cg(x)$ in the same graph and evaluate if the inequality holds. Another way can be to evaluate point-wise the inequality. In dimensions where no graphical rapresentation is available, the second method can be

Figure 1.2: Graphical method for evaluating the inequality $\rho(x) < cg(x)$ in the rejection method

easily extended and in order to reduce the computational complexity instead of starting with a proper point-wise evaluation, we can roughly grid the hyper-domain to check if our guess of the candidate density is sensible. If it is, we refine the grid and iterate the process until we are satisfied.
For our exercise, we can choose as candidate density $\mathcal{N}(0, \frac{1}{2})$ with c $= 1.6$, as shown in the picture 1.2

Then the sampling is done with algorithm 1. One can show graphically that the sample is good by plotted the properly binned random variables obtained by the process together with the target density, as showed in figure 1.3
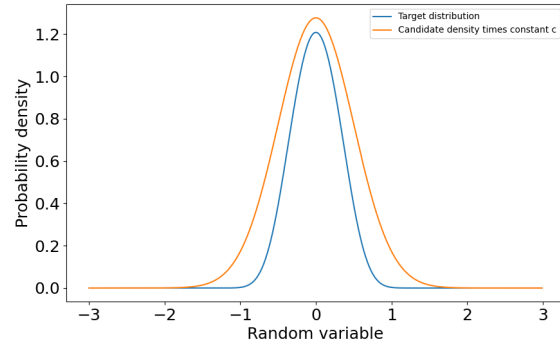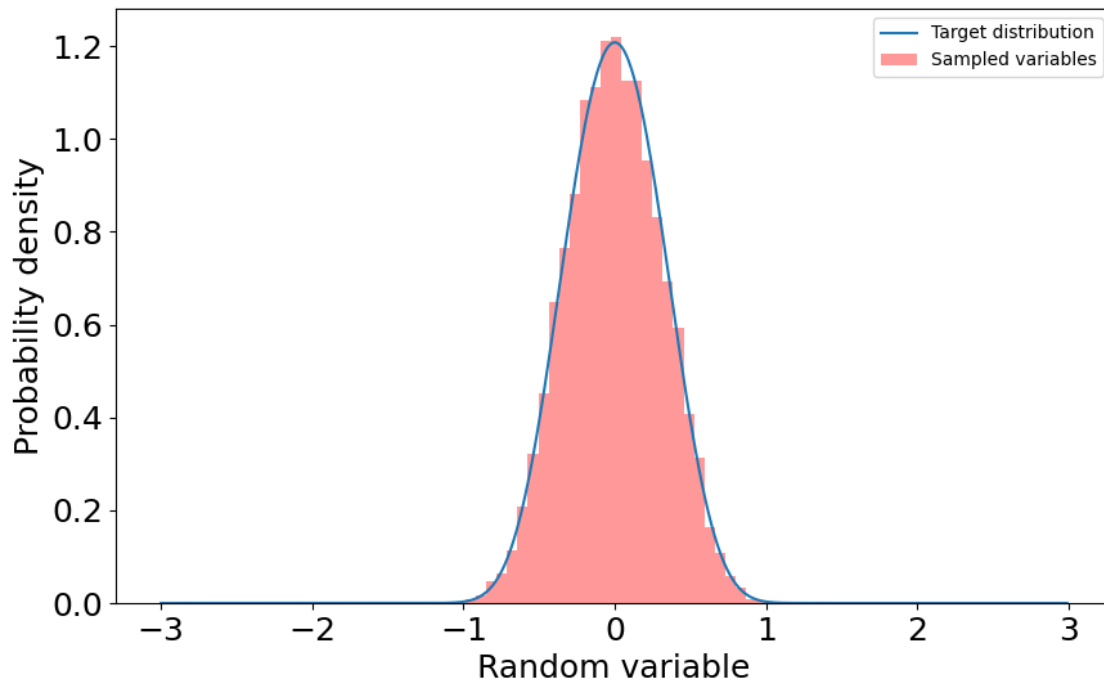
Figure 1.3: Sampled variables with target density

Lecture 3: Importance sampling

Part 1   We want to estimate the following integral:

$$\int_0^{\pi/2} \sin(x)dx$$

First we observe that the integral is easily evaluable analitically and is equal to 1.

We then proceed in doing a crude Monte Carlo estimate, obtaining the result in figure 1.4, leading to the evaluation:

$$\langle f \rangle_{\text{Crude MC}} \simeq 1.0010 \pm 0.0153$$

In order to reduce the error on the Monte Carlo evaluation one common approach is to use the importance sampling method.

The main algorithm is described in 2.

We propose as the new sampling function the family of functions defined such that:

Figure 1.4: Crude Monte Carlo distribution (1000 iterations per point)

$$g_{a,b}(x) = a + bx^2 \qquad a, b \in \mathbb{R}$$

We first make some consideration on the family of functions $g_{a,b}(x)$. First we impose, as they must be PDF's, the normalization condition on $[0, \pi/2]$, so that we obtain the constraint:

$$b = \frac{24}{\pi^3} - \frac{12a}{\pi^2}$$

In this way $g_{a,b}(x)$ collapses to a one parameter family:

$$g_a(x) = a + \left(\frac{24}{\pi^3} - \frac{12a}{\pi^2} \right) x^2$$

Moreover if we impose the condition $g_a(x) > 0$ we obtain for a the restriction $0 < a < 1$

There are now many recipes for choosing the function $g_a(x)$. We prescribe to the one that choose $g(x)$ if satisfies the condition $\rho(x) < g(x)$ when the product $f^2(x)\rho(x)$ is "large" and $\rho(x) > g(x)$ when the product $f^2(x)\rho(x)$ is "small" ,
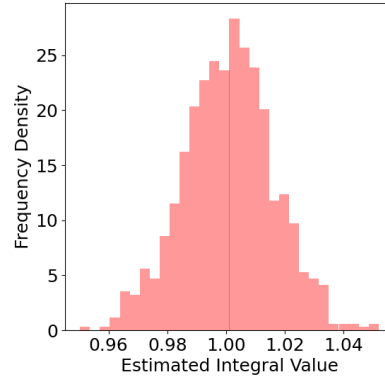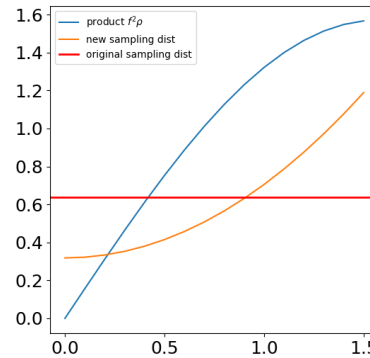

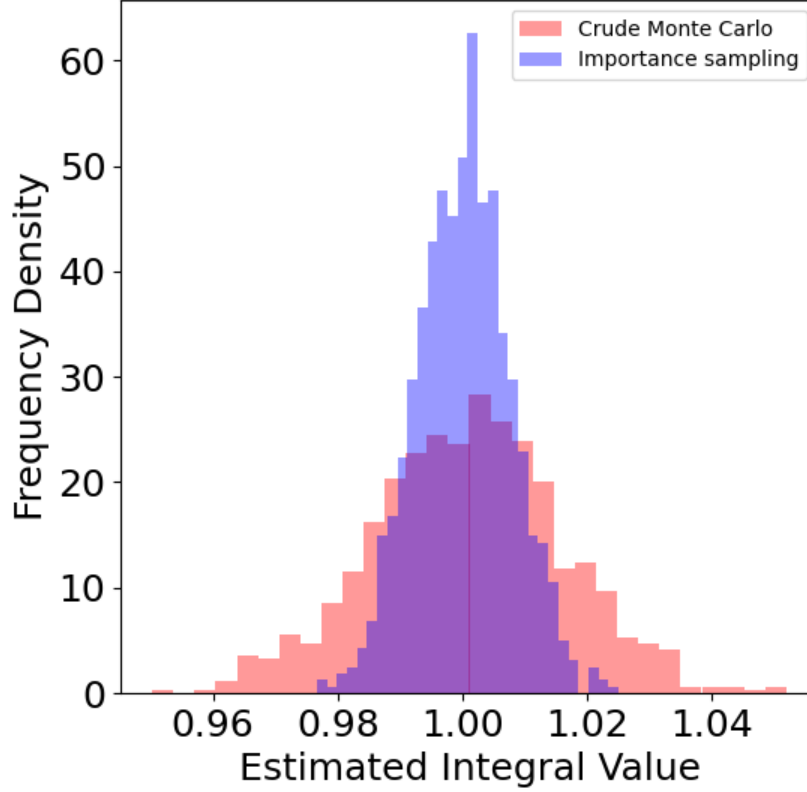Figure 1.5: Graphical criterion for choosing a in the $g_a(x)$ family

Figure 1.6: Distribution comparison between crude Monte Carlo and importance sampling with the same number of iterations

where $\rho(x)$ in our case is the original sampling distribution, which is $U(0, \frac{\pi}{2})$.
We show in fig 1.5 that for $a = \frac{1}{\pi}$ these conditions are satisfied.

To sample from $g(x)$ we will use a rejection method algorithm using as a candidate distribution the truncated normal $\mathcal{N}(\frac{\pi}{2}, 1)_{|[0, \frac{\pi}{2}]}$. Note that this specific sampling presents some interesting technical details in Python discussed in the relative algorithm section
Now we perform the importance sampling for the same number of iterations as the crude Monte Carlo, and compare the results, seen in figure 1.6.
As expected the variance is reduced. Specifically we have for this Monte Carlo evaluation:

$$\langle f \rangle_{\text{Importance sampling}} \simeq 1.0001 \pm 0.0078$$

Which is an error reduction of $\sim 51\%$.

Part 2   We want now to evaluate the integral

$$\int_{\mathbb{R}} \left[e^{-(x-3)^2/2} + e^{-(x-6)^2/2} \right] \mathcal{N}(0,1) dx = \int_{\mathbb{R}} f(x) \mathcal{N}(0,1)$$

Where $\mathcal{N}(0,1)$ is the normal distribution with mean 0 and variance 1. One can prove (to prove) that the integral has the analytical value:

$$\int_{\mathbb{R}} \left[e^{-(x-3)^2/2} + e^{-(x-6)^2/2} \right] \mathcal{N}(0,1) dx = \frac{1 + e^{27/4}}{\sqrt{2}e^9} \simeq 0.075$$

Instead of going through the analytical route, one can think to evaluate it by using a crude Monte Carlo method, by sampling respect the normal distribution, i.e. by evaluating $\langle f(x) \rangle_{\mathcal{N}(0,1)}$. We obtain:

$$\langle f(x) \rangle_{\mathcal{N}(0,1)} = 0.07431809274645616 \pm 0.004796287626550513$$

As also shown in the picture 1.7.

If, as proposed by the exercise, we now try to evaluate the same integral by the importance sampling method using as sampling function the uniform distribution $U(-8, -1)$ for 1000 iterations, we will see nothing.

It's easy to see way if we show the involved functions together in a graph 1.8

The uniform distribution samples in that interval values for which the integrand function assumes values between 10^{-41} to 10^{-5}
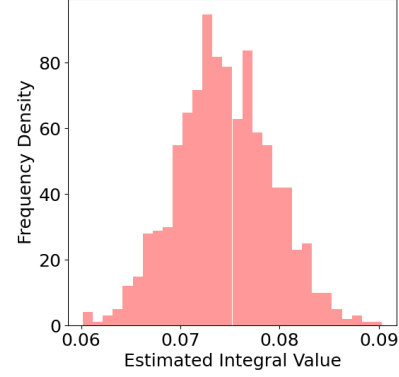

Figure 1.7: Monte Carlo distribution for the evaluation of $\langle f(x) \rangle_{\mathcal{N}(0,1)}$
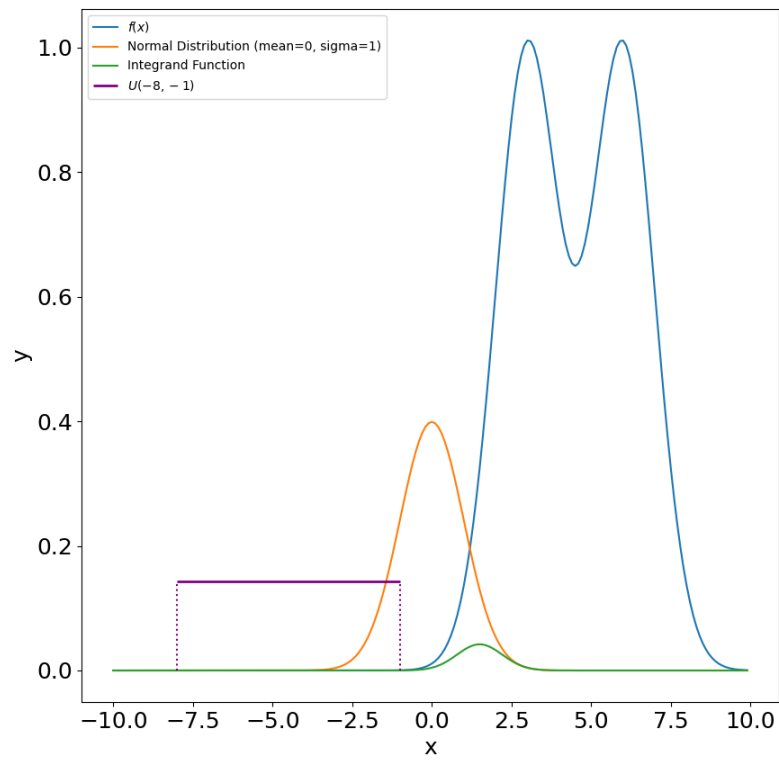
Figure 1.8: Graphical analysis for the bad choice of $U(-8, -1)$ as a sampling function for the integral $\langle f(x) \rangle_{\mathcal{N}(0,1)}$

Lecture 4: Markov chains

Part 1 We report the stochastic matrixes with the relative digraphs in figure 1.9.
One can prove from the digraph representation that chain A is irreducible, while chain B is not (for example in chain B from state 2 is not possible to reach any other state).

$$(A) \quad \boldsymbol{P} = \begin{pmatrix} 0 & 0 & 0.5 & 0.5 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

$$(B) \quad \boldsymbol{P} = \begin{pmatrix} 0.3 & 0.4 & 0 & 0 & 0.3 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.6 & 0.4 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$
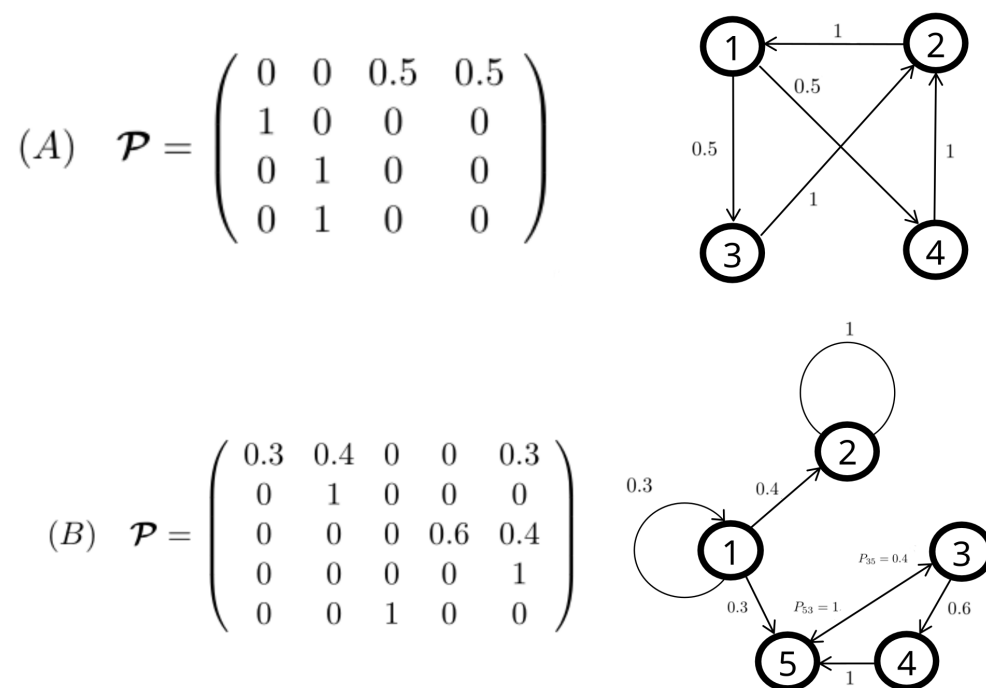

Figure 1.9: Stochastic matrixes with relative digraph representation

Part 2 Suppose to have the Markov chain with the following stochastic matrix:

$$P = \begin{Vmatrix} \frac{1}{2} & \frac{1}{3} & \frac{1}{6} \\ \frac{3}{4} & 0 & \frac{1}{4} \\ 0 & 1 & 0 \end{Vmatrix}$$

One can show with the corresponding digraph that is irreducible and aperiodic (spend some extra word). Now we evaluate the stochastic matrix in the limit $n \to +\infty$ in 2 ways, first by using the analytical approach and the by computation trying to estimate it. First we evaluate the invariant distribution:

$$\bar{\pi} P = \bar{\pi} \iff P^t \bar{\pi}^t = \bar{\pi}^t$$

11

In the end one as to solve this linear system with the normalization constraint $\sum_i \pi_i = 1$. We then obtain the following linear system:

$$\begin{cases} \frac{1}{2}\pi_1 + \frac{3}{4}\pi_2 = \pi_1 \\ \frac{1}{3}\pi_1 + \pi_3 = \pi_2 \\ \frac{1}{6}\pi_1 + \frac{1}{4}\pi_2 = \pi_3 \\ \pi_1 + \pi_2 + \pi_3 = 1 \end{cases} \iff \bar{\pi} = \left(\frac{1}{2}, \frac{1}{3}, \frac{1}{6} \right)$$

The limiting stochastic matrix will have for each row the invariant distribution:

$$P^\infty = \left\| \begin{matrix} \frac{1}{2} & \frac{1}{3} & \frac{1}{6} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{6} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{6} \end{matrix} \right\|$$

Now we should obtain the same result after many steps:

Balls and boxes We can solve this exercise using a Markov chain approach. Called a_i the state in which there are i red balls in box A there can be just 3 possible states. Our chain is then represent by the following digraph:
——— INSERT DIGRAPH ———-
So we need to evaluate the transition probabilities. In general, for a given state i:
- $a_i \to a_{i+1}$ = $\mathbb{P}$(red extraction in B) and $\mathbb{P}$(white extraction in A) - $a_i \to a_{i-1}$ = $\mathbb{P}$(red extraction in A) and $\mathbb{P}$(white extraction in B) - $a_i \to a_i$ = ($\mathbb{P}$(red extraction in B) and $\mathbb{P}$(red extraction in A)) or ($\mathbb{P}$(white extraction in B) and $\mathbb{P}$(white extraction in A))

Now we can proceed to evaluate these probabilities and then evaluate the transition matrix elements. In general one has, for a given state i:

$$\mathbb{P}(\text{red extraction in A at state i}) = \frac{i}{2} \qquad \mathbb{P}(\text{white extraction in A at state i}) = 1 - \frac{i}{2} = \frac{2-i}{2}$$

$$\mathbb{P}(\text{red extraction in B at state i}) = 1 - \frac{i}{3} = \frac{3-i}{3} \qquad \mathbb{P}(\text{white extraction in B at state i}) = \frac{i}{3}$$

We obtain the stochastic matrix

$$P = \left\| \begin{matrix} 0 & 1 & 1 \\ \frac{1}{6} & \frac{1}{2} & \frac{1}{3} \\ 0 & \frac{2}{3} & \frac{1}{3} \end{matrix} \right\|$$

Which is a stochastic matric because all rows sums up to 1.

Lecture 5: Ising 2D

Lecture 6: Polymer MMC

Lecture 7: Brussellator

Appendix : algorithms

Algorithm 1 Rejection Sampling Algorithm

1: $S \leftarrow \emptyset$
2: count $\leftarrow 0$
3: **while** count $<$ N **do**
4:      Sample $X \sim g(x)$ $\triangleright$ Generate sample from candidate distribution
5:      Sample $U \sim \text{Uniform}(0, 1)$
6:      $A \leftarrow \frac{f(X)}{c \cdot g(X)}$ $\triangleright$ Calculate acceptance threshold
7:      **if** $U \leq A$ **then**
8:          Append X to S
9:          count $\leftarrow$ count $+ 1$
10:      **end if**
11: **end while**
12: **Return** S

Implementation detail : Note that to obtain exactly N samples a while loop must be used in the algorithm.

Moreover is interesting to discuss some technical details regarding the implementation of this algorithm in Python when the candidate distribution is a truncated distribution, or in general a distribution that cannot be generated easily but needs the `scipy.stats` package to be generated.

In this case a naive implementation of algorithm 1 can hinder the perfomance greatly, because even if one has in mind to speed up the function with a `numba` decorator, this will not work, due to known incompatibility issues within the 2 cited packages.

The easiest solution to this problem is to vectorize our algorithm using the `numpy` library, overproducing samples and then cutting what is not necessary. This approach has a clear caveat, we risk an `int` overflow, so we must put attention in not choosing an high value of N for this solution. If a big number of samples is required, this approach must be revised.

Algorithm 2 Importance Sampling Algorithm

1: $\mu \leftarrow 0$
2: count $\leftarrow 0$
3: **while** count $<$ N **do**
4:      Sample $X \sim g(x)$ $\triangleright$ Generate sample from proposal distribution $g(x)$
5:      $\mu \leftarrow \mu + \frac{f(X)}{g(X)}\rho(X)$
6:      count $\leftarrow$ count $+ 1$
7: **end while**
8: **Return** μ/N