National Technical University of Athens
School of Electrical and Computer Engineering
Inter-Faculty Postgraduate Studies Programme
Data Science and Machine Learning

# One-Dimensional Random Walks

The first assignment written in partial fulfillment of the requirements for the completion of the Data Driven Models in Engineering elective course of the Data Science & Machine Learning post-graduate studies programme.

*Instructor*
Prof. V. Papadopoulos

*Written by*
[redacted]

March 13, 2022

# 1 Theoretical Approach

Before presenting the simulation's results, we provide a basic theoretical approach to the problem of the one-dimensional uniform random walk. A one-dimensional random walk is a stochastic process where the position $X(t_n)$ of a body at time $t_n$ (i.e. after making $n$ steps) is given by

$$X(t_n) = X(t_0) + \sum_{i=1}^{n} Z_i, \tag{1.1}$$

where $Z = \pm 1$ are i.i.d. random variables with $p(Z = 1) = p$ and $p(Z = -1) = 1 - p$. In the case of the uniform random walk, $p = 0.5$ holds, therefore the body has an equal chance of moving towards either direction. Supposing that the body starts at $X(t_0) = 0$, the expectation value of its position after $n$ steps is

$$\mathbb{E}[X(t_n)] = X(t_0) + \sum_{i=1}^{n} \mathbb{E}[Z_i] = 0 + n[0.5 \cdot 1 + 0.5 \cdot (-1)] = 0 \tag{1.2}$$

This is an expected result in the case of the uniform random walk, as the body is expected to have an overall zero displacement from its initial position. Of course, the result is different for $p \neq 0.5$. As far as the variance is concerned:

$$\begin{aligned} \mathrm{Var}[X(t_n)] &= \mathbb{E}\left[X^2(t_n)\right] - \{\mathbb{E}[X(t_n)]\}^2 = \mathbb{E}\left[\sum_{i=1}^{n} Z_i \sum_{j=1}^{n} Z_j\right] - 0 = \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbb{E}\left[Z_i Z_j\right] \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbb{E}\left[{Z_i}^2\right]\delta_{ij} = \sum_{i=1}^{n} \mathbb{E}\left[{Z_i}^2\right], \end{aligned} \tag{1.3}$$

where we used the fact that $\mathbb{E}\left[Z_i Z_j\right] = \mathbb{E}[Z_i]\,\mathbb{E}\left[Z_j\right] = 0$, for $i \neq j$, since $Z$ are i.i.d.
Since ${Z_i}^2 = (\pm 1)^2 = 1$, Eq. (1.3) becomes

$$\mathrm{Var}[X(t_n)] = \sum_{i=1}^{n} 1 = n. \tag{1.4}$$

This result indicates that the one-dimensional uniform random walk is not an ergodic process (in the mean): while $\mathbb{E}[X(t_n)]$ is constant (specifically, equal to zero), the variance $\mathrm{Var}[X(t_n)]$ increases linearly with the number of time steps. As a result, the time average of $X$ is a random variable with divergent variance, therefore $X(t_n)$ and $X(t_{n+m})$ cannot become independent as $m \to \infty$.

# 2 Simulation & Results

Moving on to the simulation (written in Python), a relatively large number of random walks ($2 \cdot 10^5$) is generated in order to ensure statistical significance of the final results. Each walk begins at $X(t_0) = 0$ and consists of a total of 1000 steps. At the start of each step, the position increases/decreases by 1, depending on the result of a coin flip, i.e. whether a randomly generated number, uniformly distributed in the range $[0, 1]$, is greater than 0.5 or not. Figure 1 shows five examples of the generated random walks.


Figure 1: Five random examples of one-dimensional uniform random walks with 1000 time steps.

Using the ensemble of the $2 \cdot 10^5$ random walks, the mean value $\mathbb{E}[X(t_n)]$ as well as the variance $\mathrm{Var}[X(t_n)]$ are calculated for all values of $n = 1, \ldots, 1000$. The corresponding plots can be seen in Figure 2.

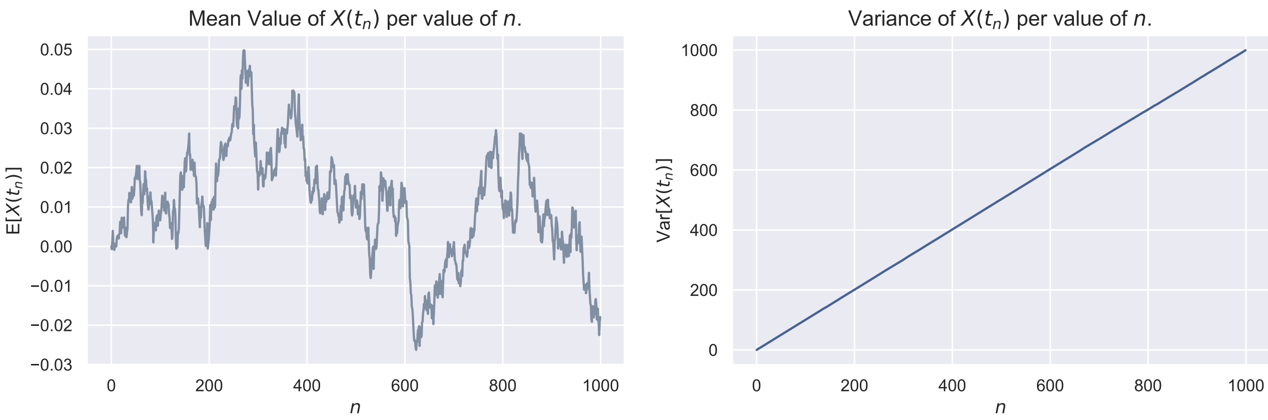


Figure 2: The mean value and the variance of $X(t_n)$ as calculated from the random walks ensemble.

These graphs seem to confirm Eqs. (1.2), (1.4). The graph for the variance is clearly almost identical to the curve $y = x$. As far as the mean-value graph is concerned, it does not appear to be constant and equal to zero, but upon a closer look it becomes evident that this is simply an artifact of the y-axis' scale, since the range of values is $[-0.03, 0.05]$. Depicting both of these curves on the same set of axes (see Figure 3) resolves the scaling problem and illustrates why Eqs. (1.2) and (1.4) hold.

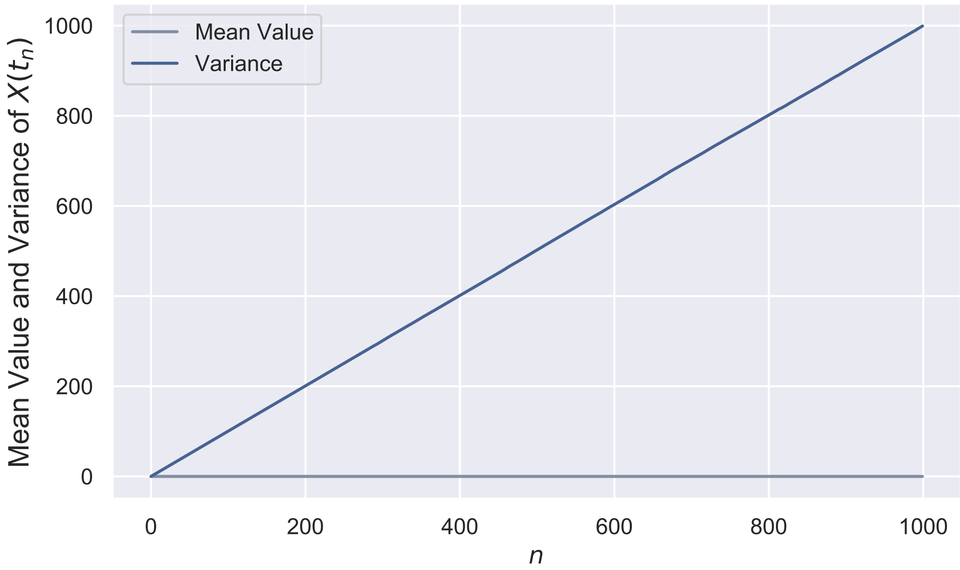


Figure 3: The mean value and the variance of $X(t_n)$ depicted on the same set of axes.

The code used to perform the simulation can be seen below.

```
import numpy as np
import random

num_walks = 200000
prob_up = 0.5
walks = []
for i in range(num_walks):
    t = []
    t.append(0.0)
    for n in range(1,1000):
        coin = random.uniform(0,1)
        t_n = t[-1]
        t.append(t_n+1) if coin >= prob_up else t.append(t_n-1)
    walks.append(np.asarray(t))
walks = np.asarray(walks)

MeanVal = []
Variance = []
for n in range(1000):
    first_moment = walks[:,n].sum()/num_walks
    MeanVal.append(first_moment)
    second_moment = (walks[:,n]**2).sum()/num_walks
    Variance.append(second_moment-first_moment**2)
```

Python Code Snippet 1: 1D Random Walk Simulation.