Numerical Simulation and Theory of 1D Random Walk

Central Limit Theorem and Gaussian Distribution

Prepared by: Ali Bijari

Introduction

Random walks are fundamental stochastic processes used to model a variety of physical, chemical, and biological systems, from Brownian motion to diffusion in solids. In one dimension, a random walk consists of a sequence of steps, each of which is taken to the right or left with equal probability.

This report presents both the mathematical theory and the numerical simulation of the 1D random walk, culminating in an explicit demonstration of the Central Limit Theorem (CLT) via Python code.

1 Physical Theory of the 1D Random Walk

Consider a particle starting at the origin ($x = 0$). At each discrete time step, it moves either $+\Delta$ or $-\Delta$ with equal probability ($p = q = 0.5$). After N steps, the position is:

$$x_N = \sum_{i=1}^{N} s_i, \qquad s_i = \pm\Delta$$

The probability of being at position x after N steps is given by the binomial distribution:

$$P(x, N) = \binom{N}{n} p^n q^{N-n}$$

where n is the number of steps to the right, $m = N - n$ to the left, and $x = (n - m)\Delta$.

For large N, the binomial distribution can be approximated by a Gaussian (Normal) distribution via the Central Limit Theorem:

$$P(x, N) \simeq \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

with

$$\mu = \langle x_N \rangle = N(p - q)\Delta = 0, \qquad \sigma^2 = N\Delta^2$$

2 Central Limit Theorem (CLT)

The CLT states that the sum of a large number of independent, identically distributed (i.i.d.) random variables (with finite variance) tends toward a normal distribution, regardless of the original distribution. In our case, as N increases, the distribution of x_N approaches:

$$P(x, N) = \frac{1}{\sqrt{2\pi N}\Delta} \exp\left(-\frac{x^2}{2N\Delta^2}\right)$$

3 Algorithm and Numerical Simulation

Simulation Steps

1. **Single Walk:** Simulate one random walk of N steps, each step $+1$ or -1, and record the position at each step.
2. **Local Time Histogram:** Count how many times the walker visits each position in that path (not necessarily Gaussian).
3. **Multiple Walks:** Repeat the experiment M times. Each time, record only the *final* position after N steps.
4. **Empirical Distribution:** Make a histogram of the final positions. Normalize the histogram to get a probability distribution.
5. **Gaussian Comparison:** Calculate the theoretical Gaussian PDF and compare it to the empirical distribution.

4 Numerical Simulation: Modern Visualization of 1D Random Walk

In this section, we present a Python simulation of the 1D symmetric random walk, using `numpy` and `matplotlib` with a modern visualization style. The code produces three visual outputs: a single trajectory, a histogram of local visit counts, and the distribution of final positions compared to the Gaussian prediction of the Central Limit Theorem (CLT).

4.1 Annotated Python Code

```
        import numpy as np
        import matplotlib.pyplot as plt
        from collections import Counter
        import math

```

```python
    # PARAMETERS
    N = 1000           # Number of steps in each random walk
    num_trials = 5000  # Number of independent random walks

    # SINGLE RANDOM WALK - TRAJECTORY AND LOCAL TIME HISTOGRAM
    steps = np.random.choice([-1, 1], size=N)
    x_positions = np.cumsum(np.insert(steps, 0, 0))

    visit_hist = Counter(x_positions)
    unique_positions = sorted(visit_hist.keys())
    frequencies = [visit_hist[pos] for pos in unique_positions]

    # MANY RANDOM WALKS - FINAL POSITION DISTRIBUTION
    final_positions = np.array([
    np.sum(np.random.choice([-1, 1], size=N))
    for _ in range(num_trials)
    ])

    final_hist = Counter(final_positions)
    final_unique_positions = np.array(sorted(final_hist.keys()))
    final_freqs = np.array([final_hist[pos] for pos in final_unique_positions])
    final_freqs = final_freqs / final_freqs.sum()

    # THEORETICAL GAUSSIAN (CLT PREDICTION)
    mu = 0
    sigma = math.sqrt(N)
    x_gauss = np.linspace(final_unique_positions.min(), final_unique_positions.max(), 300)
    gauss_pdf = (1/(sigma * np.sqrt(2 * np.pi))) * np.exp(-((x_gauss - mu) ** 2) / (2 * sigma ** 2))

    # VISUALIZATION (MODERN & PROFESSIONAL)
    plt.style.use('seaborn-v0_8-darkgrid')
    fig, axs = plt.subplots(1, 3, figsize=(18, 5))

    # (A) Trajectory of a single random walk
    axs[0].plot(np.arange(N+1), x_positions, lw=2, color='navy')
    axs[0].set_xlabel('Step', fontsize=12)
    axs[0].set_ylabel('Position', fontsize=12)
    axs[0].set_title('1D Random Walk: Single Trajectory', fontsize=14)
    axs[0].tick_params(axis='both', labelsize=10)

    # (B) Local time histogram for a single walk
    axs[1].bar(unique_positions, frequencies, color='orange', width=1)
    axs[1].set_xlabel('Position', fontsize=12)
    axs[1].set_ylabel('Visit Count', fontsize=12)
    axs[1].set_title('Visit Histogram (Single Walk)', fontsize=14)
    axs[1].tick_params(axis='both', labelsize=10)

    # (C) Final position distribution over many walks vs. Gaussian
    axs[2].bar(final_unique_positions, final_freqs, width=1, color='#0081a7', alpha=0.8, label='Empirical')
    axs[2].plot(x_gauss, gauss_pdf, 'r-', lw=2, label='Gaussian (CLT)')
    axs[2].set_xlabel('Final Position', fontsize=12)
    axs[2].set_ylabel('Probability', fontsize=12)
    axs[2].set_title(f'Final Position Distribution ($N={N}$)\nCentral Limit Theorem', fontsize=14)
```

```
        axs[2].legend(fontsize=11)
        axs[2].tick_params(axis='both', labelsize=10)

        plt.tight_layout()
        plt.suptitle('Random Walk: Trajectory, Local Time, and Central Limit Theorem', fontsize=16, y=1.04)
        plt.show()
```

Key Points

- The `steps` array contains $+1$ or -1 for each random walk.
- For a single walk, the `x_positions` records the walker's position at each step.
- The `final_positions` list collects the endpoints from many walks, used to build an empirical histogram.
- The code normalizes the histogram to compare with the probability density function of the normal distribution.
- The width (standard deviation) of the Gaussian grows as $\sqrt{N}$.

5 Results and Discussion

- The histogram of the final positions over many independent walks forms a bell-shaped curve, well-approximated by the normal distribution predicted by the CLT.
- For a single walk, the local time histogram does **not** follow a Gaussian.
- This simulation demonstrates how random microscopic events give rise to macroscopic deterministic laws (diffusion, Brownian motion).

6 References

- D. S. Lemons, *A Student's Guide to Entropy*, Cambridge University Press.
- Wikipedia: Random walk
- Wikipedia: Central Limit Theorem