PHSX815: Computational Methods in Physical Sciences
Project 3
Project Paper
Title: Parameter Estimation from Experimental Data: Random Walk

Vishakha Lingadahally

May 16, 2021

Introduction

In Science, we are often interested in a set of outcomes rather than a single event. Stochastic simulations allow us to generate and examine a series of simulations of a system in which the steps are governed by random choice. A simple algorithm for flipping of coin can be a useful tool to 'choose' if a step in a phenomenon will occur or not. Consider a drunkard trying to walk along the pavement in the middle of the night. In their drunk state of mind, they can only move sideways-either to the left or the right from their current position. Whether they take a step to the left or the right can be dictated by flipping a coin and checking if the outcome is 'Heads' or 'Tails'. The philosophy behind the seemingly simple thought experiment has a wide range of application in physics, chemistry as well as biology; to describe the diffusion of gas molecules, simulate bacterial motion and learn the pattern of genetic drift.

In Project 1, we explored 1D random walk with the hypothesis of equal probabilities of stepping on either side, i.e., the coin had equal probabilities of showing either heads or tails. In Project 2, we explored two different hypotheses or variations of the experiment. In Project 3, we estimate the probability of the drunkard either to the left or to the right, using data from experiment, by maximizing the likelihood of the probability distribution describing the event. The likelihood for a binomial distribution is given by the following formula:

$$L(p|n) = \frac{n!}{k!(n-k)!} p^{(k)} q^{(n-k)}$$

In order to find p, we maximize the above expression by taking its log and and equating its first derivative to 0. Which would give us the following expression to simplify:

$$k(\frac{1}{p}) - (n-k)(\frac{1}{1-p}) = 0$$

For this, python's SymPy library is utilized for solving the equation.

Understanding the code

Flipping the coin

The first section of the code consists of importing the necessary modules for computation and plotting. We then begin by flipping the coins and understanding their output. To see why this is relevant in random walk, think of a gas molecule that has equal probabilities of moving either to the left of the right. To dictate whether the molecule moves to the left or the right by flipping a coin. This will obviously change every time that we run the code cell. To convert this to 'heads' and 'tails' readout, we can assume that this is a totally fair coin. This means that the probability of getting 'heads' to get P(H) is the same as flipping a 'tails' P(T) such that P(H)+P(T)=1 . This means that for a fair coin, P(H)=P(T)=0.5 . To convert our coin flips to 'heads' or 'tails', we simply have to test if the flip is above or below 0.5 . If it is below 0.5, we say that the coin was flipped 'heads', otherwise, it is 'tails'.

Testing if the coin-toss is fair

Now, in order to check if our coin-toss is fair, we toss it multiple times, say 1000 times. Let's flip a coin one thousand times and compute the probability of getting 'heads'. The simulated probability is very close to our expected P(H)=0.5, but not exactly. This is the nature of stochastic simulations-it's based on repetitions. If we were to continue to flip a coin more number of times, our simulated P(H) would get closer and closer to 0.5 . This is why doing many repetitions of stochastic simulations is necessary to generate reliable statistics.

The random walk and parameter estimation

We start at position zero and flip a coin at each time step. If it is less than 0.5, we take a step left. Otherwise, we take a step to the right. At each time point, we keep track of our position and then plot our trajectory.
First, we perform an experiment of the random walk, similar to project 1 and 2, with the hypothesis that p=q=0.5. Of course, one can change the probabilities of stepping to the right or left and explore alternate hypotheses. As our steps are based on the generation of random numbers, this trajectory will change every time the code is run. The power of stochastic simulation comes from doing them many times over. The next step would be to run the steps, say, one thousand times and plot all of the traces. We plot the mean at each time point as a blue line.

Understanding the output

The user sees the following results as the output of the code:

- The results of the of three consecutive coin flips that our code simulates in the beginning is displayed. Then, these probability outputs are converted to 'heads' and 'tails'.
- Then, the predicted and simulated probabilities are then displayed. The predicted probability, naturally, is 0.5 and the simulated probability is the ratio of the number of

heads to that of tails.

- The number of steps taken to the left and the number of steps taken to the right are computed for our hypothesis.
- The plots of the tracks of the random walk for both the hypotheses are generated, along with the random walk simulated multiple times (n=1000).

Output interpretation

The parameters p and q are estimated using the maximum likelihood method. The obtained parameters are plugged into our formula for likelihood, as well as the step probability in the next section to simulate more experiments. We see how well the true value of the parameter for our hypothesis is approximated with increased number of steps of the experiment. Let us take a look at a few examples. Here's how an isolated track looks when the number of steps is 50:

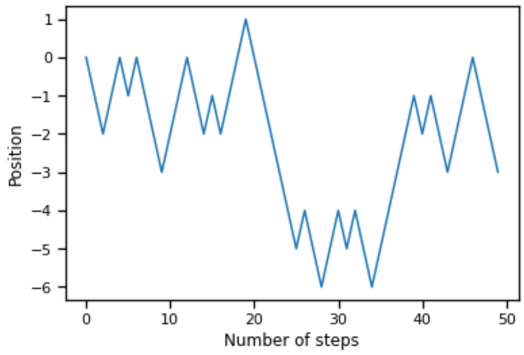

Figure 1: Track for 50 steps

The probability of stepping to the right (p) that was obtained in one of the runs was 0.54. Now for 100 steps:

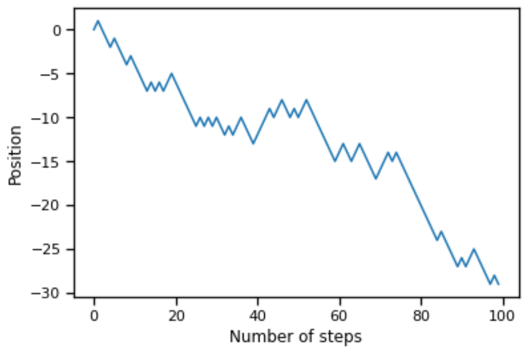

Figure 2: Track for 100 steps

p that was obtained in one of the runs was 0.57.

500 steps:

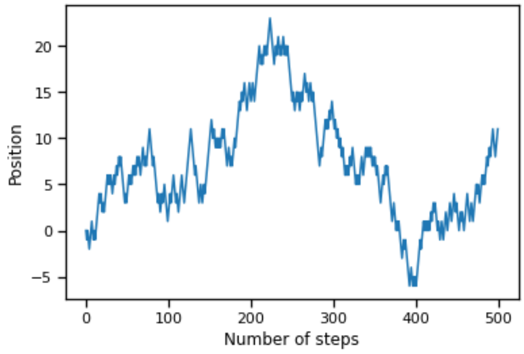

Figure 3: Track for 500 steps

p=0.522

1000 steps:

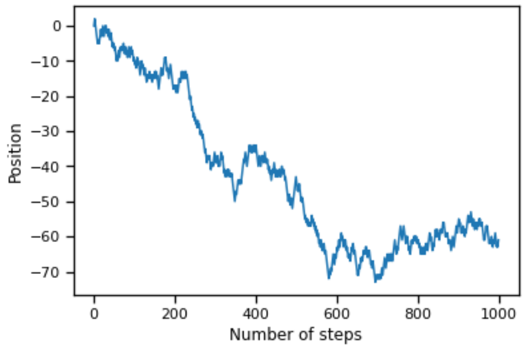

Figure 4: Track for 1000

p=0.497

We see how with increasing number of steps in the experiment, the value of p gets closer and closer to the true value of 0.5.

Summary

- The parameter p, which is the probability of stepping either to the right or to the left has been estimated by maximizing the log likelihood of the probability distribution.
- The parameter gets closer and closer to its true value with increasing number of steps.

References:

- stackoverflow.com
- medium.com

- towardsdatascience.com