Contents

1 Introduction

Random Walk is a mathematical object used to describe how a random set of steps behaves on a mathematical space. Many historians trace this study of stochastic processes back to Soviet mathematician Andrey Kolmogorov. Who published Foundations of the Theory of Probability in 1933, laying the modern axioms for studying the uncertain. Before Kolmogorov, probability had a bad name in Math and was seen as the "Theory of misfortune" because it lacked a solid foundation. Today, because of the work of Kolmogorov, Blasé Pascal, Pierre Laplace, and many others, the study of probability is used in a very wide range of disciplines, from Physics to Economics. In this paper, I will discuss the various applications of a Random Walk after giving it some background; moving from general Markov Processes to a specific Random Walk case.

2 Markov Process

A Markov chain is a stochastic model describing a sequence of possible events in which the probability of each event depends only on the state attained in

the previous event. Markov processes are the stochastic analogs to a the most useful deterministic processes- differential equations and recurrence relations. To define the language I will use to discuss Markov Processes, consider the series X.

$$X = (X_0, X_1, X_2, ..., X_n, X_{n+1}, ...) \tag{1}$$

This series is comprised of states X and times n. If we take the general sequence of discrete states $n \in N$, then the only thing that that state X_{n+1} depends on is state X_n. This discrete case is known as a Markov Chain, and it is relatively simple, but still proves useful and has many interesting implications. We can also discuss a Markov process that takes place in a continuous space $n \in [0, \infty)$. We can formally define a Markov Process if it fulfills the following relation

$$P(X_{t+1} \in A \mid X_t) = P(X_{s+t} \in A \mid X_0, ..., X_t) \tag{2}$$

Meaning the probability of a being in a state given the previous state is the same as the probability of being in that state given all information regarding all previous states. A random walk is a simple example of a Markov Chain.

3 Random Walk

Now that we have an idea of what a Markov process is, we can look deeper into the idea of random walk. The 1D random walk is probably the simplest example of a random walk and helps us understand some of the important principles of studying randomness. A game of blackjack is a great example of a real life 1D random walk. You have a probability of winning

$$P(win) = a, a \in [0, 1] \tag{3}$$

And a probability of losing

$$P(lose) = b = 1 - a, a, b \in [0, 1] \tag{4}$$

If you play N times, then your chance of winning m of them is given by a binomial distribution

$$B_N(m) = a^m b^{N-m} \binom{N}{m} \tag{5}$$

Now that we have created a distribution for this random process, we can start to extract information about the distribution. For example, we can study what your winnings will be after N games. This is mainly influenced by the number of games you win, or the expected value of m.

$$\langle m \rangle = \sum_{m=0}^{N} m a^m b^{N-m} \binom{N}{m} \tag{6}$$

Using the fact that a probability distribution always integrates to 1, we can differentiate with respect to a, then multiply by a to find a closed form solution to the expected value of m. This is easier if we temporarily forget about the relation between a and b.

$$\sum_{m=0}^{N} ma^m b^{N-m} \binom{N}{m} = a\partial_a [\sum_{m=0}^{N} a^m b^{N-m} \binom{N}{m}] = a\partial_a (a+b)^N = N(a+b)^{N-1} \tag{7}$$

Now this sum is trivial for any a and b, however we only care about when $b = 1 - a$. This yields an expected value of

$$\langle m \rangle = Na \tag{8}$$

Similarly we can calculate standard deviation through a similar expected value calculation

$$\sigma = \sqrt{\langle m^2 \rangle - \langle m \rangle^2} = \sqrt{Nab} \tag{9}$$

When $a = b$, we call this situation an unbiased 1D random walk. After extracting useful information like mean and standard deviation, we note that the binomial distribution becomes a Gaussian as N approaches infinity

$$\lim_{N \to \infty} B_N(m) \to \frac{1}{2\pi Nab} exp[-\frac{(m - Na)^2}{2Nab}] \tag{10}$$

This is a result of the central limit theorem, which states when any probability distribution is sampled N times, the average of the samples approaches a Gaussian as N approaches infinity, with the width scaling like $\sigma \sim \frac{1}{\sqrt{N}}$. Thus, because we know mean and standard deviation we can describe the normal distribution very precisely. The large N approximation becomes very helpful for studying random walks in Physics because we typically deal with number of molecules on the scale of 10^{24}.

4 Physics

The principles of random walk are very useful in studying diffusion in Physics. Diffusion refers to the net spreading of the distribution of molecules due to random molecular motion. Consider a hydrogen molecule being ejected from a gun. The molecule moves until it strikes another molecule, then bounces in a random direction until it hits another molecule. It is useful to compute the probability distribution $P_t(x)$ of the location of a hydrogen molecule after time t. It can be useful to define some physical quantities before going into the mathematics behind diffusion.

- Collision time, τ - this is the average time a molecule undergoes before colliding with another molecule

- Mean free path, l - average distance a molecule covers between collisions
- Average molecular velocity, $\bar{v}$
- Number density, n - number of molecules per unit volume
- Mass density, ρ - average mass of molecule times the number density

Now let us consider the probability distribution of a hydrogen molecule after time t. We can start by treating this as an unbiased 1D random walk. Since this process is unbiased, the expected value does not tell us much, so we will examine the root mean squared displacement. The RMS displacement is given by

$$x_{rms} = \sqrt{N}l = \sqrt{l\bar{v}t} \tag{11}$$

Knowing the mean is zero and $x_{rms} = \sigma = \sqrt{l\bar{v}t}$, we can immediately write down the probability distribution as a Gaussian ($t << \tau$)

$$P_t(x) = \sqrt{1/(2\pi l\bar{v}t)}e^{-x^2/(2tl\bar{v})} \tag{12}$$

Note that this distribution satisfies the following differential equation

$$\frac{\partial P_t(x)}{\partial t} = D\frac{\partial^2 P_t(x)}{\partial x^2} \tag{13}$$

This is known as the 1D Diffusion Equation, with $D = \frac{1}{2}l\bar{v}$. We also notice that, because there are so many particles (about 10^{24}) we can think of the number density as the time average of the probability distribution, $n(x,t) = NP_t(x)$. This leads us to the following differential equation, if we have rotational symmetry in 3D

$$\frac{\partial n(\vec{x},t)}{\partial t} = D\vec{\nabla}^2 n(\vec{x},t) \tag{14}$$

D is referred to as the diffusion constant and given by the Einstein-Smoluchowski equation $D = \frac{1}{2}\frac{l^2}{\tau}$. Although the diffusion equation looks simple, it has very many important implications and is actually mathematically identical to the Schrödinger Equation. Furthermore, if we remove the spatial dependence from D, then we get a more specific version of the Diffusion Equation, called the heat equation. The diffusion equation is linear so if $n_1(\vec{x},t)$ and $n_2(\vec{x},t)$ are solutions then $n_1(\vec{x},t)+n_2(\vec{x},t)$ will also be a solution. This makes the diffusion equation much easier to solve. We can start with a single point which is described by $P_t(x)$. Then we can observe that as $t \to 0$, $P_t(x)$ approaches a delta function $\delta^3(\vec{x})$, by definition. Then since any function can be described as a set of points, we can use Green's function method to come up with a general solution at all time if we know the number density at time $t = 0$, $n_0(\vec{x},0)$, by simply summing over all starting points

$$n(\vec{x},t) = \int d^3y \sqrt{\frac{1}{2\pi l\bar{v}t}} exp[\frac{-x^2}{2tl\bar{v}}] n_0(\vec{x},0) \tag{15}$$

The physical interpretation of this solution is that the number of molecules at point $\vec{x}$ are the molecules that walked from $\vec{y}$ after time t. This ability to approximate microscopic randomness allows us to see patterns in macroscopic phenomena, like Brownian Motion. Which refers to random walk of large objects due to microscopic phenomena. Robert Brown first put forth the idea of Brownian Motion by observing random motion of pollen in a glass of water due to random movement of the water molecules. Brownian motion has many interesting applications in physics and Albert Einstein used it, combined with the diffusion equation, to measure Avogadro's number in 1905.

Now that we have a good physical understanding of how particles diffuse we can take a more mathematical approach the analysis. The heat equation is a nice differential equation to use because we can use principles from Math 110, specifically Fourier analysis, and random walk theory to come up with a solution that is extremely applicable from a physical standpoint. We will solve the heat equation in a discrete setting.

Let A be a finite subset of the integers Z with boundary ∂A. The temperature at time n at location $x \in A$ is given by $p_n(x)$. We can think of this temperature as being determined by the density of "heat particles" from the $2d$ neighbors of x. These particles perform random walks on A until they leave A, and are no longer considered. Thus, we can give the temperature at x by summing over the heat flow from the neighbors of x:

$$p_{n+1}(x) = \frac{1}{2d} \sum_{|y-x|=1} p_n(y) \tag{16}$$

Then we can define $\partial_n p_n(x) = p_{n+1}(x) = p_n(x)$ yielding our heat equation:

$$\partial_n p_n(x) = L p_n(x) \tag{17}$$

Where L is defined to be the discrete Laplacian.

$$Lf(x) = \frac{1}{2d} \sum_{y \in Z, |y-x|=1} [f(y) - f(x)] \tag{18}$$

Also, helpful to define Q

$$Qf(x) = \frac{1}{2d} \sum_{y \in Z, |y-x|=1} f(y) \tag{19}$$

We now want to come up with some boundary conditions and initial conditions such as

Initial:

$$p_0(x) = f(x), x \in A \tag{20}$$

Boundary:

$$p_n(x) = 0, x \in \partial A \tag{21}$$

With these initial conditions we can imagine a unique solution that satisfies the initial conditions and simply follows the differential equation. We can set

$p_n(x) = 0$ if $x \in \partial A$, $p_0(x) = f(x)$ if $x \in A$, and find $p_n(x)$ if $x \in A$ recursively. Now we simply must find the function! Let's start by finding p_n for $A = (1, ..., N-1)$. This becomes a linear algebra problem and we must, essentially, diagonalize the matrix Q. Let's start by finding functions that satisfy

$$p_n(x) = \lambda\phi(x) \tag{22}$$

Use Equation (9) to see that if $p_n(x)$ is of this form then

$$\partial_n p_n(x) = \lambda^{n+1}\phi(x) - \lambda^n\phi(x) = (\lambda - 1)\lambda^n\phi(x) \tag{23}$$

We can then try to find the eigenfunctions and eigenvalues of Q to find λ and $\phi(x)$ such that

$$Q\phi(x) = \lambda\phi(x) \tag{24}$$

This is where Fourier analysis comes in. We can make this a lot easier for ourselves if we guess good eigenfunctions of Q.

Noticing that $sin((x \pm 1)\theta) = sin(\theta x)cos(\theta) \pm cos(\theta x)sin(\theta)$, we can find an eigenfunction of Q

$$Qsin(\theta x) = \lambda_\theta sin(\theta x), \lambda_\theta = cos(\theta) \tag{25}$$

Now we simply choose a θ_j that satisfies the boundary conditions. If $\theta_j = \pi j/N$, then $\phi_j(x) = sin(\pi jx/N)$. Which satisfies $\phi_j(0) = \phi_j(N) = 0$. Note that these are eigenfunctions with different eigenvalues for the symmetric matrix Q, so they must be orthogonal and linearly independent. Now, every function $f(x)$ on A can be written uniquely as the sum of these eigenfunctions as follows

$$f(x) = \sum_{j=1}^{N-1} c_j sin(\frac{\pi jx}{N}) \tag{26}$$

This sum is called a finite Fourier series and allows us to find the solution to the heat equation with $f(x)$ as the initial condition. If we sum over both sides of Equation (15) and plug in our values of λ, θ, and $\phi(x)$, then we get a general solution to the heat equation

$$p_n(y) = \sum_{j=1}^{N-1} c_j[cos(\frac{\pi j}{N})]^n\phi_j(y) \tag{27}$$

In summary we can say that the solution to the heat equation

$$\partial_n p_n(x) = \partial_x^2 p_n(x) \tag{28}$$

Is given by

$$p_n(y) = \sum_{j=1}^{N-1} c_j[cos(\frac{\pi j}{N})]^n\phi_j(y) \tag{29}$$

With c_j that satisfy

$$f(x) = \sum_{j=1}^{N-1} c_j \phi_j(x) \tag{30}$$

5 Data Science

Another application of Random Walk and Markov processes is in machine learning. The specific example that I'd like to dive into is Reinforcement Learning which leverages the Markov Decision Process. At a high level, the process of reinforcement learning consists of an agent observing its environment, which consists of surrounding states and rewards. This process is known as Reinforcement learning (RL), which is an area of machine learning concerned with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward. Reinforcement learning is considered as one of three machine learning paradigms, alongside supervised learning and unsupervised learning. We can formally define an RL problem as a Markov Decision Process. We have already defined the Markov property, but what is a Markov Decision Process? First, in RL we like to define a probability that the agent moves on to the next state

$$P_{ss'} = P[S_{t+1} = s' | S_t = s] \tag{31}$$

This would be the probability that the agent moves from state s to s'. Given this probability we can say that the Markov process is a tuple, containing state and probability (S, P). However, we are concerned with a Markov Reward Process (MRP), which contains a value judgment about how much reward a given path yields. This MRP is also a tuple, but includes a few more criteria (S, P, R, λ). We can define these criteria as follows.

1. S - state space
2. P - transition probability
3. R - reward function
4. $\lambda \in [0, 1]$ - Discount Factor

Our goal is to maximize the expected reward over the course of our agents journey from state to state. We can quantify this return with the following function.

$$G_t = R_{t+1} + \lambda R_{t+2} + \lambda^2 R_{t+2} + ... = \sum_{k=0}^{\infty} \lambda^k R_{t+k+1} \tag{32}$$

The λ appropriately discounts rewards further away because they should matter less. The higher the value of λ the more far-sighted the agent is (i.e. it cares more about distant rewards). The discount factor is mathematically

helpful as it ensures that our process will converge and we avoid an infinitely looping MDP, which would not find a maximum G. it is also good to discount rewards because of uncertainty regarding the future. Often times, the agent will not be 100 percent certain of its path. Now that we understand the discount factor, we can focus on how an MDP maximizes return. An agent can calculate the value of a state by taking the expected value of its return function G.

$$v(s) = E[G_t|S_t = s] \tag{33}$$

By taking the expected value of the return function G at each state, we can develop a set of values for each state to inform the agent of how to move. But what exactly does this look like? We use the Bellman Equation. Starting from our $v(s)$ function we can expand G_t

$$v(s) = E[R_{t+1} + \lambda(R_{t+1} + \lambda R_{t+2} + ...)|S_t = s] \tag{34}$$

Then we can sub in G_{t+1} and note that expectation value function is linear, and we arrive at the Bellman Equation

$$v(s) = E[R_{t+1} + \lambda v(S_{t+1})|S_t = s] \tag{35}$$

Bellman Equation is linear and can be solved directly, but takes a great deal of computing power, on the order of $O(n^3)$ for n states. Now that we have an idea of how MRP works, we can examine a Markov Decision Process, which can be defined by a tuple containing (S, A, P, R, λ). Where A is the action space that an agent can take. Imagine a mouse in a grid that only goes up, down, left, or right. It's action space is $(up, down, left, right)$. Now we have a state transition probability function of

$$P^a_{ss'} = P[S_{t+1} = s'|S_t = s, A_t = a] \tag{36}$$

Reward function is

$$R^a_s = E[R_{t+1}|S_t = s, A_t = a] \tag{37}$$

The goal of an MDP is to find the policy π that maximizes an agents expected return in a given environment.

$$\pi(a|s) = P[A_t = a|S_t = s] \tag{38}$$

These policies are not dependent on the current state, meaning when an agent arrives at a state, its action is already decided. In order to determine the best policy we need to iteratively figure our which π maximizes our expected return. We can do this by defining the reward function of a given state combined with a policy.

$$R^\pi_s = \sum_{a \in A} \pi(a|s) R^a_s \tag{39}$$

Now that we have developed a method for valuing actions from given states with a given policy, we need to figure out the best policy to tell us how to behave in a Markov Process. We need to first find out how good it is to be in a given state if I'm following a policy π. This is given by

$$v_\pi(s) = E_\pi[\sum_{k=0}^{\infty} \lambda^k R_{t+k+1} | S_t = s], s \in S \tag{40}$$

From this function, we can figure out the best action a to take, given a policy π.

All of this is to show the considerations that a computer will make when navigating a Reinforcement Learning Environment. The computer iteratively operates in a Markov process to maximize expected return for the agent. Next steps would include the process of using gradient descent and other optimization techniques to develop the best policy. An example of this in real life would be a rational mouse (agent) in a maze (environment), operating under a similar technique to find a policy (π) that gets it out of the maze the fastest while collecting the most cheese (rewards).

6 Conclusion

In this paper I defined a Markov Process and showed how Random Walk was a very useful extension of it in Physics. Physical phenomena like diffusion, Brownian motion, and Schrodinger's equation all can follow from Random Walk at a microscopic level. I then took the diffusion constant to not be spatially dependent, yielding the heat equation. I then solved the heat equation using principles of Random Walk and Fourier analysis. Then I moved on to talk about Reinforcement learning and how a Markov Decision Process is built up from a simple Markov Chain. This is one of the 3 most important forms of machine learning and relies heavily on simple principles discussed in this paper. Overall, I found it enjoyable to discuss uncertainty and randomness in physical systems because uncertainty typically has a negative connotation, so its interesting to see how much information we can siphon from such little information. I also enjoyed being able to apply things we learned in the course to real world problems that present themselves in my other classes and research.

7 References

1. http://www.math.uchicago.edu/ lawler/reu.pdf
2. Matthew Schwartz Statistical Mechanics Spring 2019 Lecture 2
3. http://www.randomservices.org/random/markov/Recurrence.html
4. https://www.cs.cmu.edu/ avrim/ML14/lect0326.pdf

5. https://towardsdatascience.com/reinforcement-learning-demystified-markov-decision-processes-part-1-bf00dda41690