Theory Club: Fun with Probability

Neil Thistlethwaite

6 August 2021

These notes are provided as-is! They are not intended to be comprehensive, and I haven't checked over them carefully.

How do we think about things with uncertain outcomes? **Random variables**. That is, variables that take on different values according to the outcome of some event. By convention, use capitals for random variables and lowercase for "normal" variables. For example,

$$X \triangleq \text{number on top after rolling a 6-sided die}$$

Then, we need to be able to say what **distribution** X comes from, which we can do in a few ways. Since X is a **discrete** random variable, meaning it only takes on a few different values, we can do this with a **probability mass function**, or pmf. For example,

$$f(x) = \begin{cases} 1/6 & \text{if } x \in \{1, 2, 3, 4, 5, 6\} \\ 0 & \text{otherwise} \end{cases}$$

This is a function that, given the outcome, tells us how likely it is for the random variable to take on that value. For example, $f(3) = 1/6$, meaning there's a 1/6 chance that $X = 3$. Somewhat more formally , $\mathrm{P}(X = x) = f(x)$.

Note that any manipulations we do with X will themselves be random variables. For example, $3 + X$ is also a random variable. $X = 4$ is a special kind of random variable, called an "event", meaning it has a binary outcome (either it happens or it doesn't). The notation $\mathrm{P}(A)$ is usually reserved for events, and means "the probability of event A happening", but sometimes people abuse it to really mean $\mathrm{P}(A = a)$, or "the distribution of A".

The most important concept in probability is that of **expectation**, or **expected value**, which provides us a tool to say something about the "average" value of a random variable. For discrete random variables, it's defined like this:

$$\mathbb{E}\left[X\right] = \sum_x x \cdot \mathrm{P}(X = x)$$

Here's one use of expectation: suppose I say that I'll play a game with you where I'll let you roll a 6-sided die, and I'll pay you in \$ the number that shows up on top. How much would you pay (as an entry fee) to play this game? (Try computing $\mathbb{E}X$). How about a version of the game where I pay you the number that shows up on top *squared*?

Here are some important properties of expectation:

$$\textbf{Linearity of expectation: } \mathbb{E}[X+Y] = \mathbb{E}[X] + \mathbb{E}[Y]$$

$$\textbf{Monotonicity: } \text{If } \mathrm{P}(X \leq Y) = 1, \text{ and } \mathbb{E}[X],\ \mathbb{E}[Y] \text{ exist, then } \mathbb{E}[X] \leq \mathbb{E}[Y]$$

$$\textbf{Non-multiplicativity: } \text{In general, } \mathbb{E}[XY] \neq \mathbb{E}[X]\mathbb{E}[Y]$$

$$\textbf{Non-degeneracy: } \text{If } \mathbb{E}[|X|] = 0, \text{ then } \mathrm{P}(X = 0) = 1$$

Another important concept of probability is **independence**, which I won't go too much into here. Basically, two random variables are independent if and only if $\mathrm{P}(X = x, Y = y) = \mathrm{P}(X = x)\mathrm{P}(Y = y)$. Examples of events that are independent: the results of two separate coin flips or die rolls. Examples of events that are dependent: the first roll of a die and the sum of the first two rolls, whether it rains today and whether it rains tomorrow.

If two events are independent, you can do a lot of things, but one of the nice things about expectation is that you can use **linearity of expectation** on random variables that aren't necessarily independent! Here's some problems that you can solve with expectation:

Problem 1. There are k people in a lift at the ground floor. Each wants to get off at a random floor of one of the n upper floors. What is the expected number of lift stops (i.e., total number of distinct floors chosen by the k people)?

Problem 2. Suppose I want to go on a *random walk*. I start at $x = 0$, and each minute, I randomly move in either the $+1$ direction or the -1 direction. After t minutes, what's the average location I'll be in? What's the average distance I'll be away?

Problem 3. I have a randomized algorithm that succeeds with some probability $p \in (0, 1)$, and distinct runs of the algorithm are independent. Fortunately, I can easily check if the algorithm succeeded. If I can only run it n times, what's the probability it will fail? On average, how many runs will it take me to succeed?