An Interesting Solution to the Decaying Number Game Using a Continuous Markov Chain

Sena Ghobadi

December 13, 2023

Scenario

Consider the uniform probability distribution from [0,1]. Let us imagine a game somewhat analogous to blackjack where we draw numbers from this distribution in series. We keep drawing numbers as long as they are smaller than the previous draw. The game is finished when a number greater than the previous number is drawn. We may then ask the following question.

Problem

What is the expected number of draws that you make before "busting"? We define "busting" as drawing a number smaller than the previous one drawn.

Approach

Intuition suggests that you shouldn't be able to draw more than a couple numbers without busting. We expect to draw a value of 0.5 first on average which already cuts the space of possibilities in half. One may then naively suggest that the answer may be the sum of $\frac{1}{2^n}$ from 1 to ∞. However, this solution simply ignores the space of all possibilities, and one may easily disprove this by considering what the expected second draw would be. This may be found by manually integrating over the space of possibilities after two draws $0 < y < x < 1$ which encompasses the volume below

$$\int_0^1 \int_0^{1-x} \int_0^{1-x-y} dz\, dy\, dx = \frac{1}{6}$$

Therefore, a more sophisticated solution is necessary. The method that I initially solved this problem with (and the one that seems to be the most common) is based on an analysis similar to what we described above. However, I will take a more unique approach using the machinery of Markov processes that we discussed in the latter half of the semester. In particular, we will derive a unique solution that utilizes an infinite number of Markov states.

The best way to solve a difficult problem is to ignore it and solve an easier problem. Let us consider a discrete uniform distribution instead with only 3 possible values to be drawn (one may imagine a small deck of cards): 1, 2, and 3. We will play the same game as described above with us drawing numbers from this deck (with replacement) until we obtain a number greater than or equal to the current one (we set the drawing of 1 as a bust condition too). However, even this involves an annoying amount of counting, so we'll simplify it even further by considering the case of only 1 value and building up to 3 where we will hopefully have obtained some information that allows us to solve the original problem. For the case of a single card, the answer is trivial. We only have 1 value to draw and after that there are none left, so the expected number of draws must be 1. We will call this m_1 and, in general, refer to the expected number of draws at any value i as m_i. For the case of 2 cards, we still get 1 draw for free, but if we draw the smaller one, we are done. There is a $\frac{1}{2}$ probability of drawing the larger one, after which we either bust by picking 2 again, or bust by exhausting the deck and drawing 1. Therefore, $m_2 = \frac{3}{2}$. However, I will elect to represent this as

$$m_2 = 1 + \frac{1}{2}m_1$$

. For the case of m_3, we again note that we get 1 draw for free. There is a $\frac{1}{3}$ chance of it being any of the 3 possible cards. In the case of drawing 1, we are finished. In the case of drawing 2, we have a $\frac{1}{3}$ chance of drawing 1 next and a $\frac{2}{3}$ chance of busting. We will represent this as a possible state transition from m_3 to m_1. Likewise, in the case of drawing 3, we have a $\frac{2}{3}$ probability of not busting by drawing 3 again, which would transition us to state m_2. Of course, there is a $\frac{1}{3}$ probability of busting by drawing 3. We can then represent all these possibilities in the following state transition equation for m_3.

$$m_3 = 1 + \frac{1}{3}((\frac{1}{3}m_1 + \frac{2}{3}) + (\frac{2}{3}m_2 + \frac{1}{3}))$$

$$m_3 = 1 + \frac{1}{3}(\frac{1}{3}(m_1 + 2m_2) + \frac{1}{3}(1 + 2))$$

$$m_3 = 1 + \frac{1}{9}((m_1 + 2m_2) + (1 + 2))$$

This would mean that m_3 evaluates to $\frac{16}{9}$ which is about 1.78. The Monte Carlo simulation of this game in the appendix of this paper validates these results as well as other computations we make later on. This same reasoning can be used for m_4, m_5, and in general m_i. Therefore, we may inductively represent the expected value of draws (proving this is an exercise at the end) before busting from a deck of n cards as

$$m_n = 1 + \frac{1}{n^2}\sum_{i}^{n-1}(m_i + 1)i$$

We now have an expression that will compute the expected number of draws for a deck with finite elements. To extrapolate this analysis to our original problem, we will take the limit as the number of elements approaches infinity. However, our expression is unfortunately a recurrence relation that relates every single previous state, leaving us with no closed form for m_n. There is a creative workaround to this issue. Though we may not be able to solve this incredibly complicated difference equation, in the limit of $n \to \infty$, we may be able to express this (Riemann) sum as an integral and derive a solvable differential equation. It is important that we consider the most general case of a draw where we might not be bounded between [0,1], but rather between 0 and some intermediate pick x. Therefore, we consider

$$m_x = 1 - x + \frac{1}{n^2}\sum_{i}^{x-1}(m_i + 1)i$$

The $(1 - x)$ term represents the probability of bust (with 1 being the resulting expected number of draws) which is something we have to consider in the general case where we've already used our "free" draw. The only real work left is to transform the sum into an integral. Though the machinery of real analysis is likely necessary to perform this rigorously, there are a few heuristics from calculus that we may use. We note that in the limit of n approaching infinity, $\frac{1}{n}$ represents a very small change in summing window if we divide our integration interval into n pieces which we can intuit as dx'. We evaluate this sum over bounds determined by the previous draw x, which corresponds to us integrating the value of $m + 1$ over the non-busting range [0,x]. Thus, we have the integral equation

$$m(x) = 1 - x + \int_0^x (m(x') + 1)dx'$$

This satisfyingly corresponds to the same idea we explored in the discrete case. The expected number of draws from a particular state $m(x)$ is the probabilistic weighted sum of the states $m(< x)$. The $1 - x$ term corresponds to the idea that there is a $1 - x$ probability of busting and leaving with only one draw. We observe the advantages of this approach by differentiating this expression to get a very simple differential equation,

$$\frac{dm}{dx} = m(x)$$

which has the well-known solution

$$m(x) = Ce^x$$

You may have guessed that e was involved in this problem from the progression of expected draws (i.e. 1/2, 1/6, etc) and indeed, this is another way of solving the problem. But to finally obtain an answer we need to define our integration constant using boundary conditions. As described in our setup, we required a bust once the smallest value was reached. However, we always get 1 draw for free, which specifies that $m(0) = 1$, implying $C = 1$. The value encompassing the entire interval is 1, so our answer is simply

$$\boxed{m(1) = e}$$

The fact that e shows up seemingly out of nowhere is quite interesting, and I encourage you to solve this problem with a more traditional probability approach, inductively evaluating the probability that the draws $x_1 > x_2 > x_3 \ldots$ using counting arguments rather than the Markovian approach above.

Related Problems

1.

Prove the general solution for the expected number of discrete draws that we defined above using induction.

2.

Solve the decaying number problem by considering the exact probability of busting at each given step and taking a sum of the results (this will be easy to evaluate unlike the one we derived). For example, consider the probability of getting exactly 4 picks as the probability of $x_4 > x_3$ and $x_3 < x_2 < x_1$ where x_n is the nth draw.

3.

What is the lowest value you expect to draw from this game?

4.

Now consider a ”different” problem where you are still drawing values from the uniform distribution, but you keep drawing until the sum of your values exceeds 1. What is the expected number of draws before busting?

5.

Given the scenario in problem 4, what is the expected sum of the values you draw? Hint: This should be really easy! However, it's worth thinking about why such a naive approach works.

Appendix: Monte Carlo Code

```
import random
import numpy as np
#rset = [1, 2, 3, 4] #for discrete case
rset = np.linspace(0,1,10000) #for continuous case
N = 100000 #number of MC iterations
pcount = 0 #record draws for each trial


random.seed(1)
for i in range(N):
  prev = 1
  n = 0
  while True:
      draw = random.choice(rset)
      n += 1
      if draw >= prev:
          break
      else:
          prev = draw
  pcount += n

print(pcount/N)

#evaluates to 2.718 ~ e
```