Consecutive Heads or Tails

A Run of Heads(or Tails) in a Number of Coin Tosses

Fuyu Tang
University of California, Berkeley
April 10, 2019

Abstract

In this report, we try to calculate the probability that a run of at least M consecutive heads(or tails) in a series of N coin tosses. And the extended problem is the probability that a run of at least either M consecutive heads or M consecutive tails in a series of N coin tosses. Aimed to the first problem, we try to solve it using both mathematical methods and computer simulation. But for the extended problem, it is hard to calculate mathematically, so we show the result by simulation.

Problem formulation

If we toss a fair coin N times, we can get the following sequence:

$$\underbrace{HTTHHTHH...HHT}_{N},$$

where H denote the HEAD of the coin, T denote the TAIL of the coin. In this note, we will calculate(or estimate) the probability of finding runs of length M in sequences of $N(N \geq M)$ coin tosses for different values of M and N.

$$\underbrace{HTTTHHTHTH}_{10}$$

In this example, the longest run of tails has length 3 and the longest run of head s has length 2.

Now, there are two problems we are interested in:

1. The probability of getting a run of (at least) M heads in N tosses.
2. The probability of getting a run of length M, where the run can be either heads or tails.

In this note, we will use **P1** to denote Problem 1 and **P2** to denote Problem2.

Method

We solve the problems from two different views. On one hand, we use mathematical calculation, on the other hand, we simulate in the computer.

Mathematical Calculation

Aimed to P1, we denote the probability of getting a run of M heads in N coin tosses by $p_{N,M}$.

Recalling the law of total probability, if $\{B_n : n = 0, 1, 2, 3, ...\}$ is a finite partition of a sample space, which means $\{B_n\}$ are mutually disjoint and exhausts all probability, then for event A of the same probability space, we have the following formula:

$$Pr(A) = \sum_n Pr(A \cap B_n) = \sum_n Pr(A|B_n)Pr(B_n)$$

In this case, we make partitions of the sample space by considering the last M trials. We construct B_i as follows:

$$B_0 = \{thesequenceends\ with\ \underbrace{HHH...H}_{M}\}$$

$$B_1 = \{thesequenceends\ with\ T\}$$

$$B_2 = \{thesequenceends\ with\ TH\}$$

$$B_3 = \{thesequenceends\ with\ THH\}$$

$$B_4 = \{thesequenceends\ with\ THHH\}$$

$$...$$

$$B_M = \{thesequenceends\ with\ T\underbrace{HHH...H}_{M-1}\}$$

It can be proved that

$$B_i \cap B_j = \emptyset \quad \forall i \neq j$$

$$B_0 \cup B_1 \cup B_2 ... \cup B_M = \Omega$$

And we can discuss each $Pr(A|B_i)$.
For B_0, we already have a run of M heads in the last M trials.
For B_1, in order to achieve A, we must have a run of M heads in the first $N-1$ trials.
For B_2, in order to achieve A, we must have a run of M heads in the first $N-2$ trials.
...
For B_M, in order to achieve A, we must have a run of M heads in the first $N-M$ trials.
So we have:

$$Pr(A|B_0) = 1 \quad Pr(B_0) = (\frac{1}{2})^M$$

$$Pr(A|B_1) = p_{N-1,M} \quad Pr(B_1) = \left(\frac{1}{2}\right)$$

$$Pr(A|B_2) = p_{N-2,M} \quad Pr(B_2) = (\frac{1}{2})^2$$

$$...$$

$$Pr(A|B_M) = p_{N-M,M} \quad Pr(B_M) = (\frac{1}{2})^M$$

Then according to law of total probability, we can get a recursive formula to calculate $p_{N,M}$.

$$p_{N,M} = (\frac{1}{2})^M + \sum_{i=1}^{\min\{M,N-M\}} (\frac{1}{2})^i p_{N-i,M} \quad M \neq 0 \& N > M$$

$$p_{N,0} = 0$$

$$p_{N,N} = (\frac{1}{2})^N$$

Based on the recursive formula, we can calculate $p_{N,M}$ for each (N, M) easily using Python. The code is as below:

```python
def f1(n, m, p=1/2, saved=None):
    if saved == None:
        saved = {}
    ID = (n,m,p)
    if ID in saved:
        return saved[ID]
    else:
        result = 0
        if(n < m or n <= 0):
            return result
        else:
            result = pow(p, m)
            for i in range(1, m + 1):
                if(n - i < m):
                    break;
                factor = pow(p, i)
                result += factor * f1(n - i, m, p)
            saved[ID] = result
        return result
```

It is difficult to solve P2 through mathematical calculation. So we will solve it using simulation method.

Simulation

Both questions can be solved by finding an approximate value by using simulation.

Assuming we want to calculate the probability of getting either a run of M heads or a run of M tails in N coin tosses, we can simulate Q sequences, with each sequence consisting of heads and tails and having length of N. Then we

can check each sequence to figure out if it contains a run of length M, where the run can be either heads or tails.
Because the frequency approximates probability when the number of Q is large. So we have:

$$p_{N,M} \approx \frac{K}{Q}$$

, where K is the number of sequences which contains either a run of M heads or a run of M tails.

Result and Comparison

We solve Problem 1 by using both mathematical calculation and simulation. In the case of simulation, we let Q to be 100000. The result can be shown through picture:

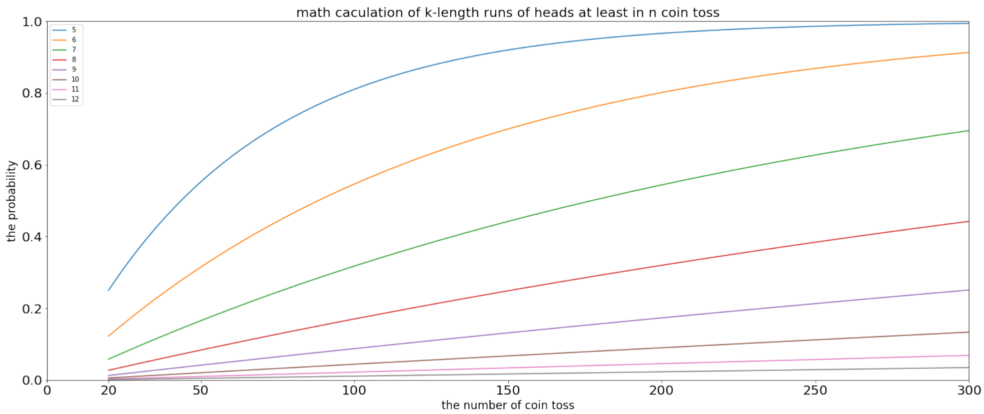

Figure 1: Problem 1 by mathematical calculation

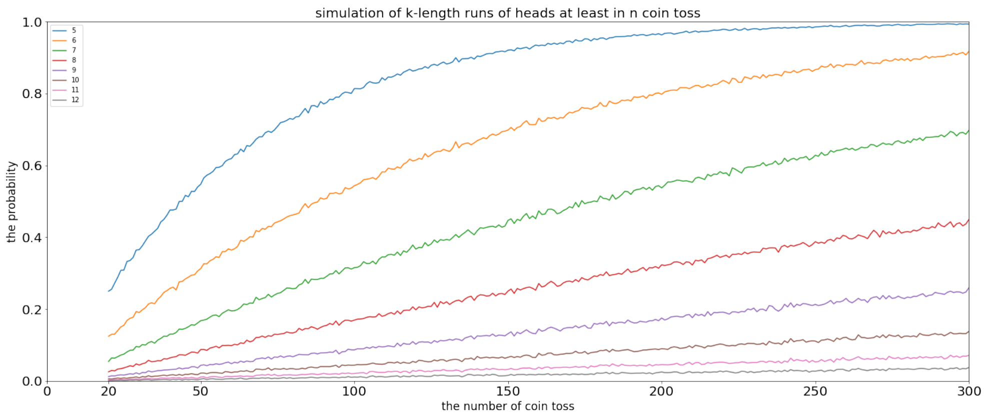

Figure 2: Problem 1 by simulation

It can be figured out that the curve in two different pictures is similar. But in Figure2, the curve fluctuates because the simulation is not as accurate as math calculation. But the whole trend in Figure 2 is similar to Figure1, which proves our method is correct.

Considering the math calculation is faster than the simulation, so in fact we can get a picture where N goes from 20 to 1000.

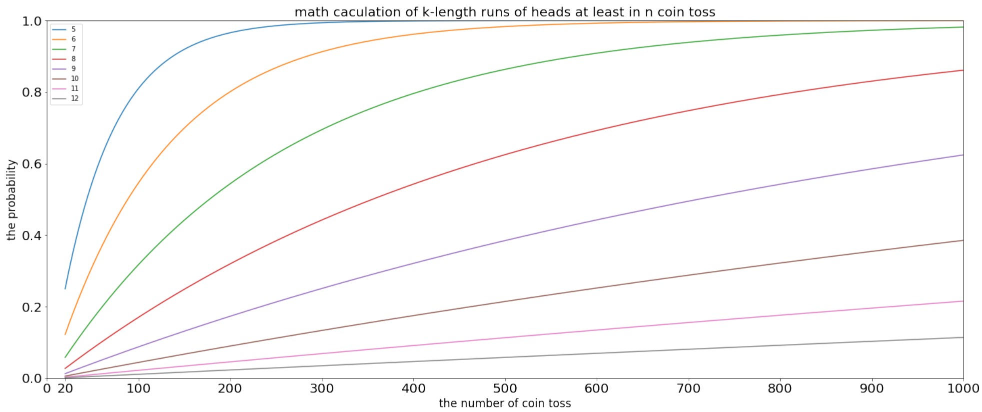

Figure 3: Problem 1 by simulation

We solve Problem 2 by using simulation. The result can be shown through picture:

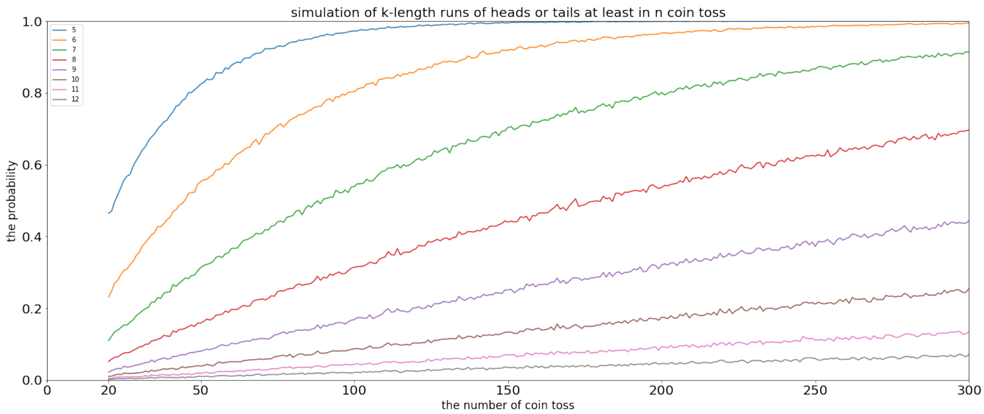

Figure 4: Problem 1 by simulation

Compared to Problem 1, the probability of Problem 2 is higher given fixed M and N, which is what we expected.

Other methods reference

There are also two mathematical we found which can be used to solve Problem 1.

- Greg Egan uses recursion also. But he makes recursion using $A(M, N)$ for the number of sequences of N trials that contain at least a run of length M.
- Feller shows us a a set of numerical constants which describe asymptotic probability by solving:

$$\lim_{n\to\infty} p_{N,M}\alpha_M^{N+1} = \beta_M$$

, where α_M is the smallest positive real root of

$$x^{M+1} = 2^{M+1}(x-1)$$

and

$$\beta_M = \frac{2-\alpha_M}{M+1-M\alpha_M}$$

We can get the approximate formula for $p_{N,M}$.

$$p_{N,M} \approx \frac{F_{n+2}^{(M)}}{2^N}$$

, where F is the generalized Fibonacci sequence.