Mass function of the number of coin flips to get m consecutive heads

Jan Aalmoes
jan.aalmoes@protonmail.com

May 20, 2024

Abstract

In this paper, we compute the exact probability mass function of the number of coin flips to get m consecutive heads. Meaning that we take D, a random variable modeling the number of coin flips until the coin gives m heads in a row; then we explicit a function f such that for all n, $f(n) = P(D = n)$. $P(D = n)$ is the probability that the m last flips among n total flips are heads while the $n - m$ first flips doesn't contain a string of m uninterrupted heads. Using f we compute the cumulative distribution function (CDF) of D as well as $E(D)$: the expected number of flips. In addition, we provide a python program to compute f, the CDF of D and the average of D at the following repository: git.jaalmoes.com/coin_flip.

This work is intended to help practitioners and domain experts build reliable systems independently of the filed of application: from redundant parts in hardware to device identification in the internet of things.

1 Introduction

Repeating events and their occurrences is of interest in many fields: sports, physics, networks, etc. In baseball, successfully repeating an exceptional action such as hitting a home run can be decisive for a player's carrier [1]. In reliability engineering, redundancy is used to reduce the probability of system failure [5]. In networking, the wildly used transmission control protocol (TCP) uses re transmission to manage packet loss. Consecutive packet loss perturbs the re transmission scheme [9]. Regarding the internet of things, the LoraWAN protocol coupled with a sequential private identifier scheme is highly sensitive to packet loss [10]. In this case, consecutive packet loss results in device desynchronization: a costly process when power is sparse.

Our theoretical study aims at giving quantitative answers to questions such as :

- What are the odds of consecutive failure of redundant part ?
- What are the odds of consecutive packet loss after some time ?

- What are the odds of streaks ?

Answering those questions is key to building reliable systems in both software and hardware. Hence we developed a comprehensive python package which, based on our theoretical work, allows practitioners and systems experts to assess risks of failure and make rational choices of parameters.

There are many theoretical problems considering streaks for instance the number of streaks among n tries, or the waiting time in-between streaks. In this paper we choose to study the number of tries before a streak of size m. In reliability engineering it corresponds to the number of time a system can be used before every redundant part has failed. This problem is usually studied either through simulation or by using the inclusion exclusion principle and the Bonferroni inequality to find an approximation [5]. The issue with numerical simulations is computing time and eventual cost of the simulation if real-world experiments have to be made [4].

In this work, we compute an exact formula of the probability law of the number of tries before a streak of size m. Contrary to previous work, this formula provides a way to assess the risk of having a streak of size m in reasonable time. The formula's computation relies on two key algorithms: root finding [8] and matrix inversion. The degree of the polynomial and the rank of the matrix is m hence complexity of the numerical method depends on m.

2 Problem definition

We consider a coin with a probability $p \in [0,1]$ to land heads up. We set $q = 1-p$ for readability. Let $X_0, \cdots, X_{n-1}$ be n random variables independent and identically distributed that follow a Bernoulli law of probability $p \in [0,1]$. Each X_i models one coin flip and takes the value 1 for head and 0 for tail. Let $m \in \{1, \cdots, n\}$. We are interested in computing the probability of m consecutive heads. We model those events using the following vector of random variables :

$$\forall i \in \{0, \cdots, n-m+1\} \quad Y_i = \prod_{j=i}^{i+m-1} X_i$$

With this model, the ith collection of m consecutive coin flips result in only heads if and only if $Y_i = 1$. On the contrary, if $Y_i = 0$ at least one the m coin flips ends up on tail. This setup is represented in figure 1.

We remark that even if (X_i) is mutually independent, (Y_i) is not. It means that computing the probability of events such as $\{Y_0 = 0\} \cap \cdots \cap \{Y_n = 0\}$ is not immediate (e.g. a product of $P(Y_i = 0)$). As we see in further developments, this kind of event is key for the problems at hand.

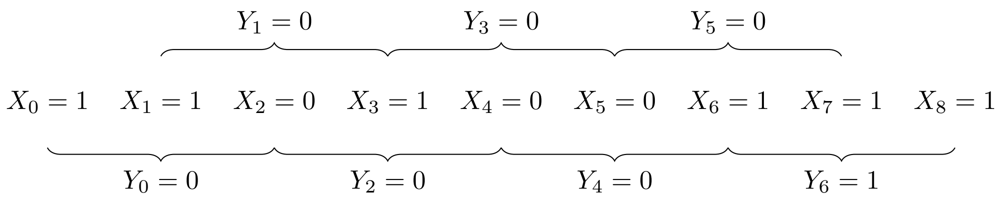

Figure 1: Repeating coin flip until $m = 3$ heads on a row

3 Mass function

The probability to obtain m heads in a row after flipping the coin n times, and not before, is the probability of the following event: "The first collection of m flips to give only heads is the $(n-m+1)$th". Let $f(n)$ be the probability of this event. We start by computing $f(n)$ for $n \leq 2m$.

$$f(n) = \begin{cases} 0 & \text{if} \quad n < m \\ p^m & \text{if} \quad n = m \\ qp^m & \text{if} \quad m < n \leq 2m \end{cases}$$

We now study the case where $n > 2m$.

$$\begin{aligned}
&P\left(Y_0 = 0 \cap \cdots \cap Y_{n-m-1} = 0 \cap Y_{n-m} = 1\right) && (1)\\
=&P\left(\bigcap_{i=0}^{n-m-1} Y_i = 0 \cap \bigcap_{i=0}^{m-1} X_{n-m+i} = 1\right) && (2)\\
=&P\left(\bigcap_{i=0}^{n-2m-1} Y_i = 0 \cap X_{n-m-1} = 0 \cap \bigcap_{i=0}^{m-1} X_{n-m+i} = 1\right) && (3)\\
=&P\left(\bigcap_{i=0}^{n-2m-1} Y_i = 0\right) P\left(X_{n-m-1} = 0 \cap \bigcap_{i=0}^{m-1} X_{n-m+i} = 1\right) && (4)\\
=&P\left(\bigcap_{i=0}^{n-2m-1} Y_i = 0\right) (1-p)p^m && (5)
\end{aligned}$$

Let's now study the sequence $u_n = P\left(\bigcap_{i=0}^{n} Y_i = 0\right)$.

Case 1 $n = 0$ $\quad u_0 = P(Y_0 = 0) = 1 - P(X_0 = 1 \cap \cdots \cap X_{n+m-1} = 1)$. Since $(X_n)_{n\in\mathbb{N}}$ is mutually independent, $u_0 = 1 - p^m$.

Case 2 $n \leq m$

$$u_n = \left(1 - P\left(Y_n = 1 \quad \middle| \quad \bigcap_{i=0}^{n-1} Y_i = 0\right)\right) u_{n-1} = u_{n-1} - (1-p)p^m$$

In this case we can solve the sequence :

$$u_n = 1 - p^m - n(1-p)p^m$$

Case 3 $n > m$ According it Bayes' formula

$$u_n = P\left(\bigcap_{i=0}^{n} Y_i = 0\right) = P\left(Y_n = 0 \quad | \quad \bigcap_{i=0}^{n-1} Y_i = 0\right) u_{n-1}$$

Since Y_n takes values in $\{0, 1\}$,

$$u_n = \left(1 - P\left(Y_n = 1 \quad | \quad \bigcap_{i=0}^{n-1} Y_i = 0\right)\right) u_{n-1} \tag{6}$$

To compute $P\left(Y_n = 1 \quad | \quad \bigcap_{i=0}^{n-1} Y_i = 0\right)$ we use Bayes' formula once more and obtain:

$$\frac{P\left(Y_n = 1 \cap \bigcap_{i=0}^{n-1} Y_i = 0\right)}{u_{n-1}}$$

By using the same method as in equations 1 to 5 and by substitution in equation 6

$$u_n = u_{n-1} - (1-p)p^m u_{n-m-1}$$

By using the initial conditions described in cases 1 and 2, we can solve the sequence in case 3 by first finding the roots of the characteristic polynomial $Q(\lambda) = \lambda^{m+1} - \lambda^m + (1-p)p^m$ for $\lambda \in \mathbb{C}$. Let those roots be $\lambda_0, \cdots, \lambda_m$. Then $u_n = \sum_{i=0}^{m} c_i \lambda_i^n$ where $(c_i)_{i \in \{0, \cdots, m\}} \in \mathbb{C}^{m+1}$ are found by using the initial conditions of cases 1 and 2. Namely :

$$\begin{pmatrix} c_0 \\ \vdots \\ c_m \end{pmatrix} = \Lambda^{-1} \begin{pmatrix} u_0 \\ \vdots \\ u_m \end{pmatrix} \quad \text{with } \Lambda = \begin{pmatrix} \lambda_0^0 & \cdots & \lambda_m^0 \\ \vdots & \vdots & \vdots \\ \lambda_0^m & \cdots & \lambda_m^m \end{pmatrix}$$

Finding Λ and inverting it can be done on a case by case basis numerically. We use Numpy's implementations for numerical matrix inversion and root finding [7]. Hence, for all $n \in \mathbb{N}^*$, we have

$$f(n) = \begin{cases} 0 & \text{if} \quad n < m \\ p^m & \text{if} \quad n = m \\ (1-p)p^m & \text{if} \quad m < n \leq 2m \\ u_{n-2m-1}(1-p)p^m & \text{if} \quad n \geq 2m+1 \end{cases}$$

We display f for $m = 5$, $p = 0.9$ and $m = 20$, $p = 0.2$ on figure 2. In sequential private identifiers for LoRaWAN, those values means that each device has a list of 5 and 20 addresses with a packet loss rate of 0.9 and 0.2. It represents two

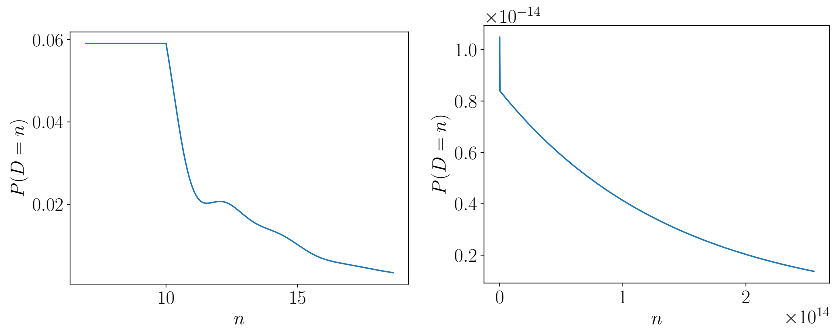

Figure 2: Mass function for $p = 0.9$, $m = 5$ on the left and $p = 0.2$, $m = 20$ on the right.

extreme case: in the first one we expect frequent desynchronization contrary to the second case. Comparing the two graphs, we indeed observe that the first instance has a high chance to desynchronize around 10 packets sent.

While the mass function gives the exact probability of the streak occurring exactly after n tries, the cumulative distribution function gives a more visual interpretation. For instance we find a range such that the streak occurs in this range with a probability of 99%.

4 Cumulative distribution function

We build the random variable D such that $\forall n \in \mathbb{N}$ $P(D_n) = f(n)$. Let F be the CDF of D: for all $N \in \mathbb{N}^*$, $F(N) = P(D \leq N)$. Hence

$$F(N) = \sum_{n=1}^{N} f(n) = \begin{cases} 0 & \text{if} \quad N < m \\ p^m & \text{if} \quad N = m \\ p^m + (N-m)(1-p)p^m & \text{if} \quad m < N \leq 2m \\ p^m + m(1-p)p^m + (1-p)p^m \sum_{n=1}^{N-2m-1} u_n & \text{if} \quad N < m \end{cases} \tag{7}$$

We compute the sum of u_n. Let $n \in \mathbb{N}$

$$\sum_{i=0}^{n} u_i = \sum_{i=0}^{n} \sum_{j=0}^{m} c_j \lambda_j^i = \sum_{j=0}^{m} c_j \sum_{i=0}^{n} \lambda_j^i = \sum_{j=0}^{m} c_j \frac{1-\lambda_j^{n+1}}{1-\lambda_j} \tag{8}$$

By substitution of the sum in the last line of equation 7 we obtain:

$$F(N) = \sum_{n=1}^{N} f(n) = \begin{cases} 0 & \text{if} \quad N < m \\ p^m & \text{if} \quad N = m \\ p^m + (n-m)(1-p)p^m & \text{if} \quad m < N \leq 2m \\ p^m + m(1-p)p^m + (1-p)p^m \sum_{j=0}^{m} c_j \frac{1-\lambda_j^{N-2m}}{1-\lambda_j} & \text{if} \quad N < m \end{cases}$$

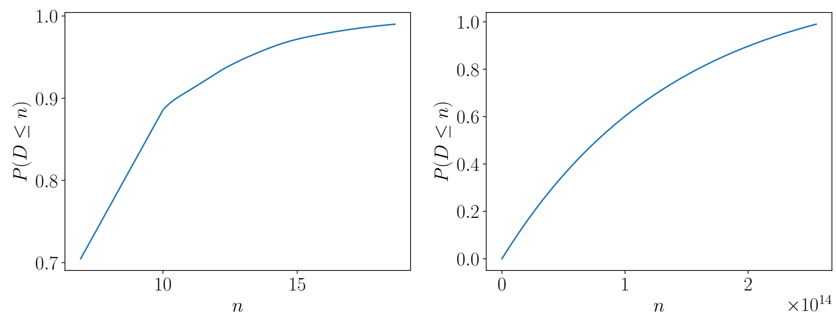

Figure 3: CDF for $p = 0.9$, $m = 5$ on the left and $p = 0.2$, $m = 20$ on the right.

We display the CDF of D, F in figure 3. To analyse those CDF, we keep the example of sequential private identifiers. With the CDF formula, we can state that in the first instance desynchronization happens before 19 packet sent with a probability of 99%. In the second instance where desynchronization is less likely, it happens after 10^{12} packet sent with a probability of 99%.

Finally, we use this development to compute the expectancy to allow us to answer question such as: "On average, a streak of size m happens after how many tries ?"

5 Average number of coin flip to get m heads on a row

We compute the expected value of D, noted $E(D)$.

$$E(D) = \sum_{i=m}^{+\infty} iP(D = i) = mp^m + (1-p)p^m \sum_{i=m+1}^{2m} i + (1-p)p^m \sum_{i=2m+1}^{+\infty} iu_{i-2m-1} \tag{9}$$

We study the partial sums of the last equation. Let $A \in \mathbb{N}$ such that $A > 2m+1$.

$$\sum_{i=2m+1}^{A} iu_{i-2m-1} = \sum_{i=0}^{A-2m-1} (i + 2m + 1)u_i = \sum_{i=0}^{A-2m-1} iu_i + (2m+1) \sum_{i=0}^{A-2m-1} u_i$$

We study separately the two previous sums.
On one hand: According to equation 8:

$$(2m+1) \sum_{i=0}^{A-2m-1} u_i = (2m+1) \sum_{j=0}^{m} c_j \frac{1 - \lambda_j^{A-2m}}{1 - \lambda_j}$$

Hence:

$$\lim_{A \to +\infty} (2m+1) \sum_{i=0}^{A-2m-1} u_i = (2m+1) \sum_{j=0}^{m} \frac{c_j}{1 - \lambda_j} \tag{10}$$

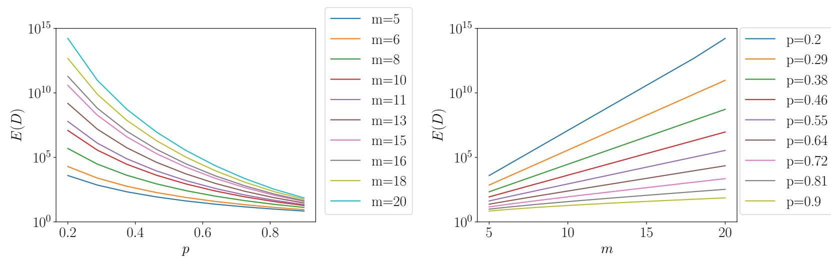

Figure 4: Average number of coin flips to get m head on a row where the probability to get head is p. On the left, average in function of p, on the right, average in function of m. Both figure are in log-scale for the y-axis.

On the other hand:

$$\begin{aligned}
&\sum_{i=0}^{A-2m-1} i u_i \\
&= \sum_{i=0}^{A-2m-1} i \sum_{j=0}^{m} c_j \lambda_j^i \\
&= \sum_{j=0}^{m} c_j \sum_{i=0}^{A-2m-1} i \lambda_j^i \\
&= \sum_{j=0}^{m} c_j \frac{1}{1-\lambda_j} \left(\frac{\lambda_j - \lambda_j^{A-2m}}{1-\lambda_j} - (A-2m-1)\lambda_j^{A-2m} \right)
\end{aligned}$$

Hence:

$$\lim_{A \to +\infty} \sum_{i=0}^{A-2m-1} i u_i = \sum_{j=0}^{m} \frac{c_j \lambda_j}{(1-\lambda_j)^2} \tag{11}$$

By substitution of the results of equations 10 and 11 in equation 12, we obtain:

$$E(D) = mp^m + (1-p)p^m \frac{3m^2+m}{2} + (1-p)p^m \left(\sum_{j=0}^{m} \frac{c_j \lambda_j}{(1-\lambda_j)^2} + (2m+1) \sum_{j=0}^{m} \frac{c_j}{1-\lambda_j} \right) \tag{12}$$

Using this equation for the expected value, we get insight on the variations of the average of coin flips needed. We do so using two plots. The first one in function of m: the streak size, the second one in function of p: the probability to get head. We present those two plots in figure 4. We observe that increasing m results in an exponential growth in the average number of coin flips.

6 Related work

6.1 Markov chains

Using Markov chains gives that the expected number of coin flips until m heads on a row is $\frac{p^{-1}-1}{1-p}$ [6]. This formula is easier to work with than the expression we found for $E(D)$ in equation 12; but [6] does not provide an expression for the mass function nor the CDF. Those two functions give more information on the law of D than just the average and can be used to find quantile, moments or even generate sample using inverse transform sampling [3].

6.2 Recency

The phenomenon of decision maker using past streaks to predict the next occurring event is called recency. Positive recency means that after m heads (respectively tails), the decision maker predicts heads (respectively tails). Negative recency is the opposite: an inversion between the streak and the prediction. Studies of recency use sociological experiments to determine how streaks length is in relation to positive and negative recency [2]. To influence the size of streaks, this work leverages p. Hence, they rely on repeating a large number of time the experiment to approach the law of the number of tries needed to obtain a streak. Using our work on the CDF, one could use inverse transform sampling to fine tune this experiment reducing the number of samples while keeping an accurate representation of the streak phenomenon.

7 Conclusion

We have successfully computed an expression of the mass function and the CDF of the number of coin flips required before getting m heads in a row. Those expression rely on root finding and matrix inversion algorithms with a complexity increasing with m. We provide python code at the following repository git.jaalmoes.com/coin_flip that allows to compute both the mass function and the CDF for arbitrary values of p and m by using state of the art base algorithms of the Numpy [7] python package.

Thanks to our theory and with our implementation, practitioners can now compute confidence intervals, statistical testing and random sampling of the number of tries to get m streaks. Domain experts use those informations to select optimal parameters of the system they are designing (e.g. the number of redundant parts, size of identifier list etc).

References

[1] S. Christian Albright. A statistical analysis of hitting streaks in baseball. *Journal of the American Statistical Association*, 88(424):1175–1183, 1993.

[2] Erik M Altmann and Bruce D Burns. Streak biases in decision making: Data and a memory model. *Cognitive Systems Research*, 6(1):5–16, 2005.

[3] Robet Collins. Intro to Sampling Methods. https://www.cse.psu.edu/ rtc12/CSE586/lectures/cse586samplingPreMCMC.pdf, 2005. [Online; accessed 2024-01-23].

[4] Jérôme de Reffye. Numerical simulation of system dynamical reliability. In *2017 2nd International Conference on System Reliability and Safety (ICSRS)*, pages 20–30, 2017.

[5] Margaux Duroeulx. *Évaluation de la fiabilité des systèmes modélisés par arbres de défaillances grâce aux techniques de satisfiabilité*. Theses, Université de Lorraine, March 2020.

[6] Paul Ginsparg. How many coin flips on average does it take to get n consecutive heads? https://www.cs.cornell.edu/ ginsparg/physics/INFO295/mh.pdf, 2005. [Online; accessed 2024-01-17].

[7] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[8] Roger A. Horn and Charles R. Johnson. Matrix analysis, 2nd ed. 2012.

[9] Yoshiaki Nishikawa, Kousuke Nogami, and Eiji Takahashi. A method for reducing tcp latency due to consecutive packet losses in mobile networks. In *2016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pages 1–4, 2016.

[10] Samuel Pélissier, Jan Aalmoes, Abhishek Kumar Mishra, Mathieu Cunche, Vincent Roca, and Didier Donsez. Privacy-preserving pseudonyms for LoRaWAN. In *WiSec 2024 - 17th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pages 1–6, Seoul, South Korea, May 2024. Association for Computer Machinery (ACM), ACM.