Simulating and Understanding the Birthday Problem in MATLAB

Brandon Y. Tai

June 5, 2018

0.1 Introduction

Consider a set of *n* people and a year with *d* days. (Realistically, d = 365 for non leap years and d = 366 leap years.) **We want to find the probability that, from *n* people, at least two of them share a birthday.** We will denote the event that ”at least two people share the same birthday” as **X**, and its probability of occurring as P(X).

In order to have P(X) = 1, by the pigeonhole principle, we would need to have n ¿ d, i.e. the number of people exceed the number of days in the year. In real life, this would mean that $n \geq$ 366 for non leap years and $n \geq$ 367. However, we shall demonstrate that it is possible to have P(X) can be quite close to 1 — with less than 100 people required.

We will presume that, for the purposes of this exercise, it is not a leap year, so d = 365.

0.2 Mathematically Computing The Probability

Here, we will again use the definition of X defined previously. Instead of computing P(X) directly, however, we will instead calculate the complement of X — P(X’) , the probability that **nobody in the room shares a birthday with another** — and use that to find P(X) = 1 - P(X’).

Suppose we have *n* people, and that the year has 365 days. Since we don’t want anybody to share a birthday, we choose *n* days from 365 days, and then arrange them n! ways. (It is important to note here that a combination of *n* from 365 birthdays is insufficient, as the birthdays must be assigned to each of *n*). This result is equivalent to C(365,n) * n! = P(365,n). It follows, then, that P(X) = 1 - P(X’) = 1 - P(365,n).

The *birthdaymath.m* function will calculate, using the mathematical definition derived here, the probability of at least two people sharing the same birthday given *n* people and a year of length 365 days. The *birthdaymath-script.m* generates a vector, named **prob**, whose *n*th element can be called to determine the probability of at least two people sharing a birthday given *n* people. (Due to Matlab storage limitations and the way Matlab handles large numbers, however, you will find that the given Matlab script will become inaccurate after n = 70. This doesn’t stop you from calculating the result by hand, although the result will tend really close to 1 for the range where Matlab fails to calculate it properly.)

0.3 Simulating the Birthday Problem

While knowing how to determine mathematically the probability for this problem, there may be times where finding a nice mathematical definition for it may be infeasible. In such cases, it may be easier to simulate the problem using a program. The *birthday.m* function returns a boolean value — 0 if false, 1 if true — after generating n discrete random numbers from 1 to 365 inclusive and checking if at least two generated numbers are equal. The *birthdaysim.m* script uses the *birthday.m* script which logs the number of times the function returns 1 for an assigned number of simulations and then determines the experimental probability. This process is repeated for each number of people $n = people$ as desccribed in the for loop. The *birthdayscript.m* generates a vector, named **simulated_prob**, whose nth element can be called to determine the probability of at least two people sharing a birthday given n people. **Note that the result returned by this script will be more accurate if you increase the number of simulations to run this for, at the cost of time required to run the program. For most realistic applications, an adequate tradeoff between accuracy and computation time must be determined.**

0.4 Closing

- In both cases, you will find that it takes only 23 people to have a 50% chance, and only 40% to have at leasst a 90% chance.
- If we want to find the probability of event X P(X), it may be easier to find P(X') and use 1 - P(X') = P(X)
- The approach to simulating a problem does not necessarily have to follow the mathematically defined way. Here, we found the mathematical definition to find the birthday probability, but also simulated the birthday problem by simulating and generating random groups of "people."

0.5 Feedback

If you have any questions, comments, or concerns, such as if you feel some things were omitted or could have been explained better, feel free to contact me at *byt1998+writeups@gmail.com*.