Birthday Paradox

Khudayar Farmanli
Politecnico Di Torino
Data Science and Engineering
Computer Aided Simulations and Performance Evaluation
Torino, Italy
s276310@studenti.polito.it

***Abstract*—In this homework we simulate Birthday Paradox problem where we try to find probability of collision which means at least 2 of m people have the same birthday within provided n number of days and compare simulation results with theoritical ones.**

I. Assumptions

- There are n number of days and m number of people where m is smaller than n.
- At least two of them should have same birthday for collision.
- r number of runs should be repeated for each e number of experiment for calculating collision probability for each experiment based on runs and finding mean and standard deviation (std) of experiments based on each m group of people.

II. Input Parameters

- m - number of people groups which is numpy array
- n - number of days and defined 365, number of days in a year
- r - number of runs which is defined 100
- e - number of experiments which is defined 5
- q value - upper limit of Confidence Interval (CI) which is defined 0.975 for CI of 0.95
- Seed - for initializing the random number generator

III. Output parameters

- Mean of each m group of people's experiments' collision probabilities which is calculated based on r number of runs:

$$Probability = \frac{\#RunsWithCollision}{\#Runs}$$

- Confidence Interval of 0.95 based on obtained means and standard deviations
- Theoritical results of array of m number of people groups for n days to compare with simulation

$$p(n) \approx 1 - e^{-\frac{m^2.}{2n}}.$$

IV. Main data structures and algorithms

For developing this simulation I have mainly used lists and numpy arrays. In the beginning, I have assigned all input parameters with particular numbers and generated 8 different group of numbers with the help of np.arange() function provided by numpy library. Then I have created empty lists and filled them with randomly generated values uniformly, after doing all the operations and getting related values, again I convert lists to numpy arrays for easily getting proper statistical features. At the end, with the help of matplotlib library I visualized the results obtained.

V. Development of Simulation

- Assignment of the values to particular input parameters.
- Generation of random dates in a uniform way for particular group of m people for 100 runs for 5 experiments.
- Calculation of repetition of numbers in each list of particular group of m people for each 100 runs and each 5 experiments.
- After getting list of all those numbers I divide them to particular groups for calculation needed probability.
- Calculation of collision probability and storing them.
- Converting particular results to numpy arrays and calculating mean and std and storing them in particular lists.
- Calculation of CI based on these statistical features.
- Calculation of theoritical results based on m number of people groups for n days.
- Visualization of obtained simulation results and theoritical results to see how much they differ.

VI. Results

The results based on simulation did not surprise us, since the visualization of them, with respect to theoritical results are almost the same. From below figure we can easily observe this.

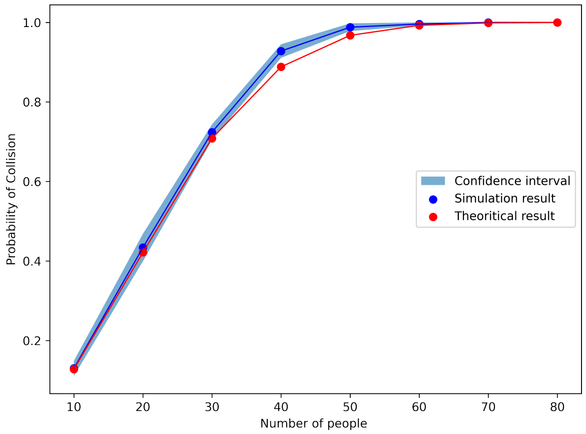

Fig. 1: The change in the probability of collision based on group of people attended in simulation and the comparison of simulation results with theorithical ones.

VII. Conclusion

From the results we see that the simulation results almost totally overlaps with theoritical ones and after around 60 people in the same group it can be observed that almost at every single run we guaratee to get birthday collision of at least two people.