CS6140 Data Mining: A1 Hash Functions and PAC Algorithms

Vai Suliafu, u0742607

January 15, 2020

1 Birthday Paradox

1.1 A

Consider a domain size $n = 5000$. Then, let k represent the number of trials necessary such that we experience our first *collision*. After simulating the Birthday Paradox problem over the domain of n, we observe that $K = 169$ trials.

1.2 B

We can then repeat the simulation m times, recording values of k for each repeated simulation. The cumulative distribution for values of k over m simulations can then be plotted. Figure 1 at the bottom of the section shows the cumulative density of k when $n = 5{,}000$ and $m = 300$.

1.3 C

We can also empirically estimate the expected number of k random trials in order to have a collision by adding up all values of k and dividing by m. For the simulation data where $n = 5{,}000$ and $m = 300$, we observed that $\mathrm{E}[k] =$ 87.97 trials.

1.4 D

Now suppose we were interested in the time it took to run the Birthday Paradox simulation from earlier, where $n = 5{,}000$ and $m = 300$. For this particular simulation, our program reported a time of approximately 1.47 ms. However, to get a better sense of the algorithm's computational time behavior, more simulations are required. Thus, the following steps were followed:

1. Wrote a function, F1, to randomly generate numbers in the domain [1, n] until a duplicate draw - then return the number of draws necessary for the duplication.

2. Wrote a second function, F2, to repeatedly call F1 m times, storing the result k for each repeated simulation.

3. Wrote a third function, F3, which stepped through every combination of $n \in [50000,\ 1000000]$ using steps of 50,000 and $m \in [1000,\ 10000]$ using steps of 1,000, storing the results in a Pandas DataFrame.

The time required for each Birthday Paradox simulation given different combinations of n and m is visualized in Figure 2 at the end of the section.

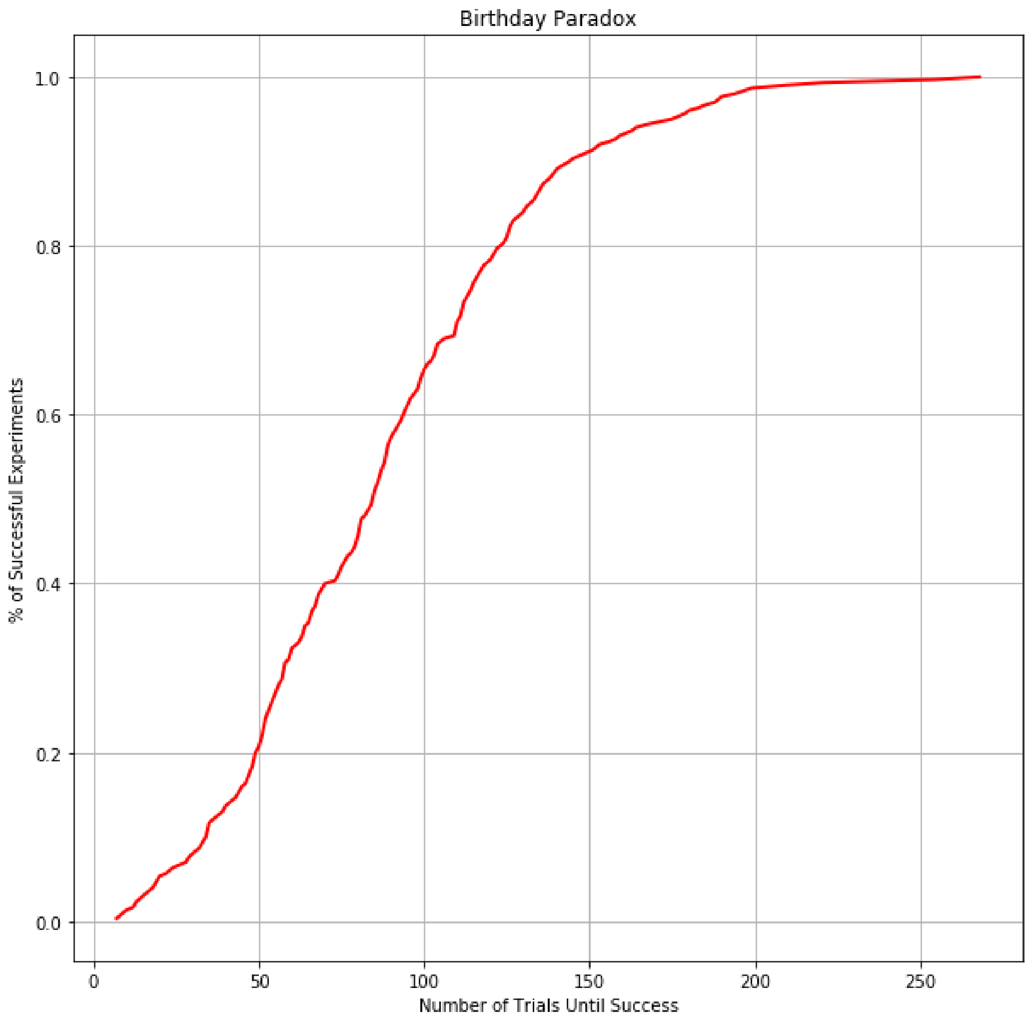

Figure 1: Birthday Paradox

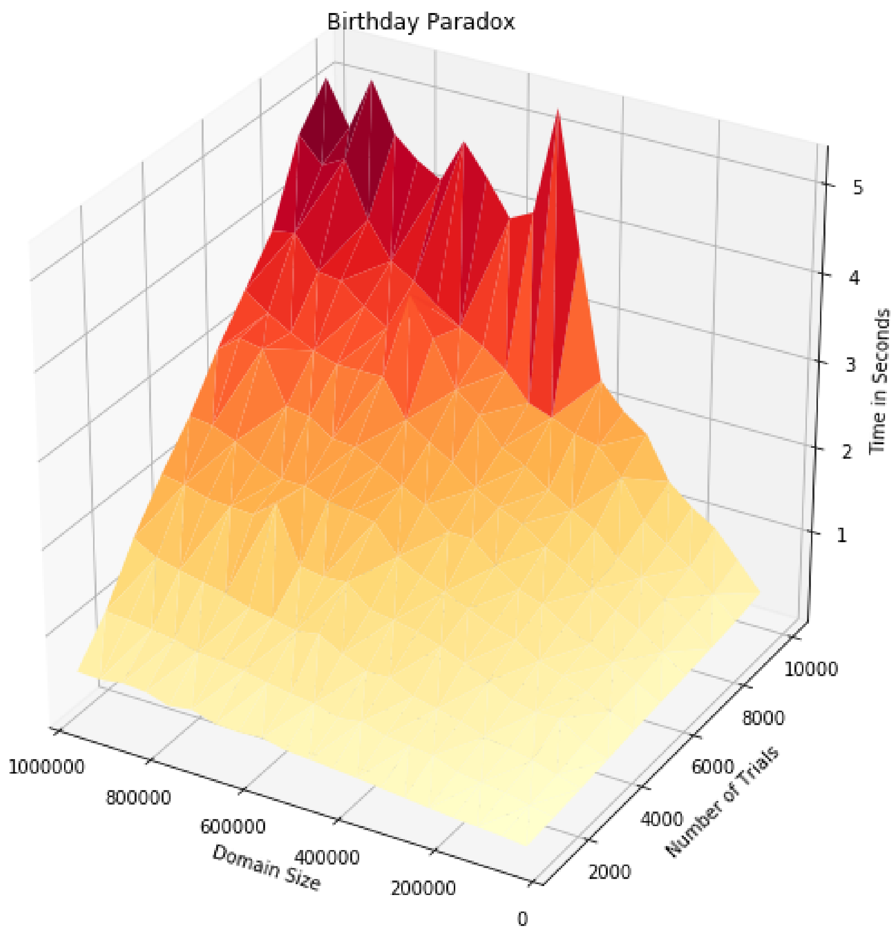

Figure 2: Birthday Paradox Simulation Times

2 Coupon Collectors

2.1 A

Consider a domain size $n = 300$. Then, let k represent the number of trials necessary such that every integer $x \in [1,\ n]$ has been drawn at least once. After simulating the Coupon Colletors problem over the domain of n, we observe that $K = 1707$ trials.

2.2 B

We can then repeat the simulation m times, recording values of k for each repeated simulation. The cumulative distribution for values of k over m simulations can then be plotted. Figure 3 at the bottom of the section shows the cumulative density of k when $n = 300$ and $m = 400$.

2.3 C

We can also empirically estimate the expected number of k random trials in order to have a collision by adding up all values of k and dividing by m. For the simulation data where $n = 300$ and $m = 400$, we observed that $\mathrm{E}[k] = 1896.2375$ trials.

2.4 D

Now suppose we were interested in the time it took to run the Coupon Colllectors simulation from earlier, where $n = 300$ and $m = 400$. For this particular simulation, our program reported a time of approximately 20.5 ms. However, to get a better sense of the algorithm's computational time behavior, more simulations are required. Thus, the following steps were followed:

1. Wrote a function, F1, to randomly generate numbers in the domain $[1,\ n]$ until all integers $x \in [1,\ n]$ had been drawn at least once - then return the number of draws required to achieve this.
2. Wrote a second function, F2, to repeatedly call F1 m times, storing the result k for each repeated simulation.

3. Wrote a third function, F3, which stepped through every combination of $n \in [1000,\ 20000]$ using steps of 1,000 and $m \in [500,\ 5000]$ using steps of 500, storing the results in a Pandas DataFrame.

The time required for each Coupon Collectors simulation given different combinations of n and m is visualized in Figure 4 at the end of the section.

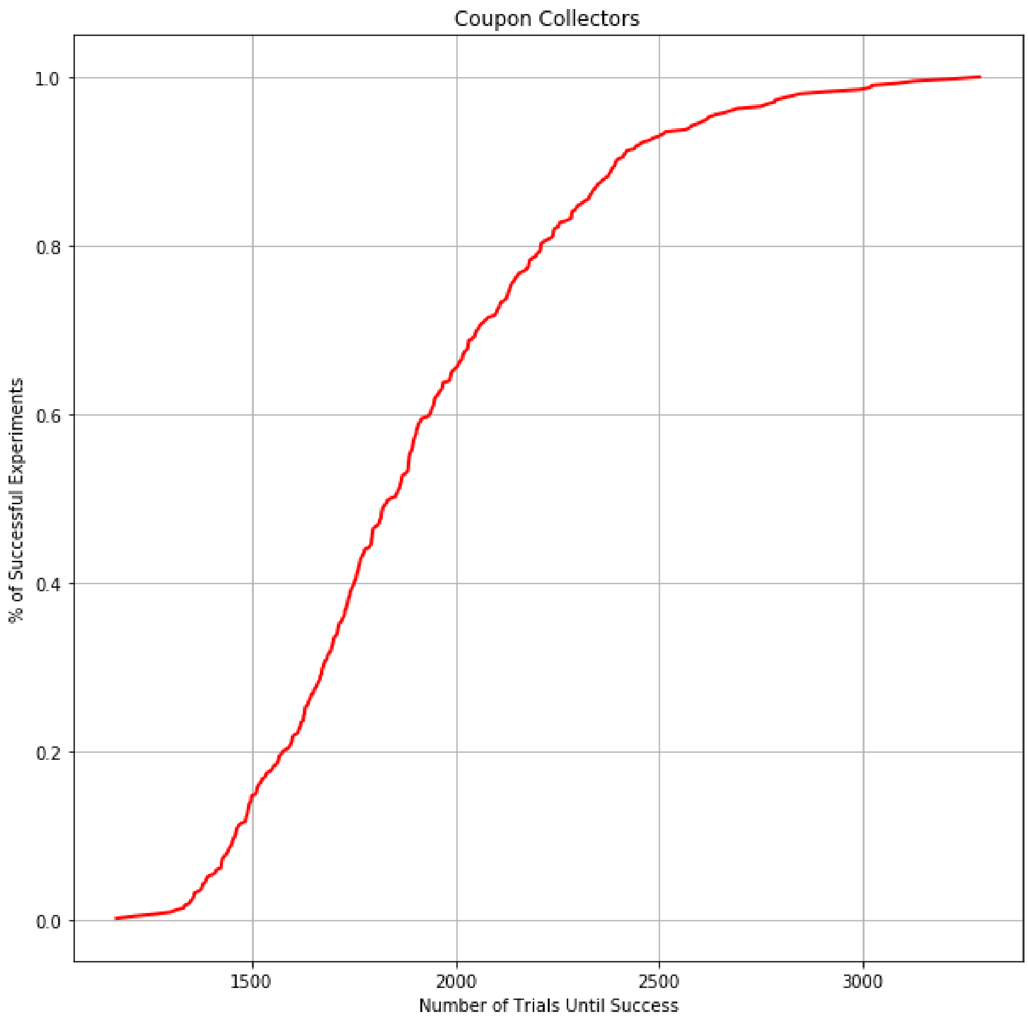

Figure 3: Coupon Collectors

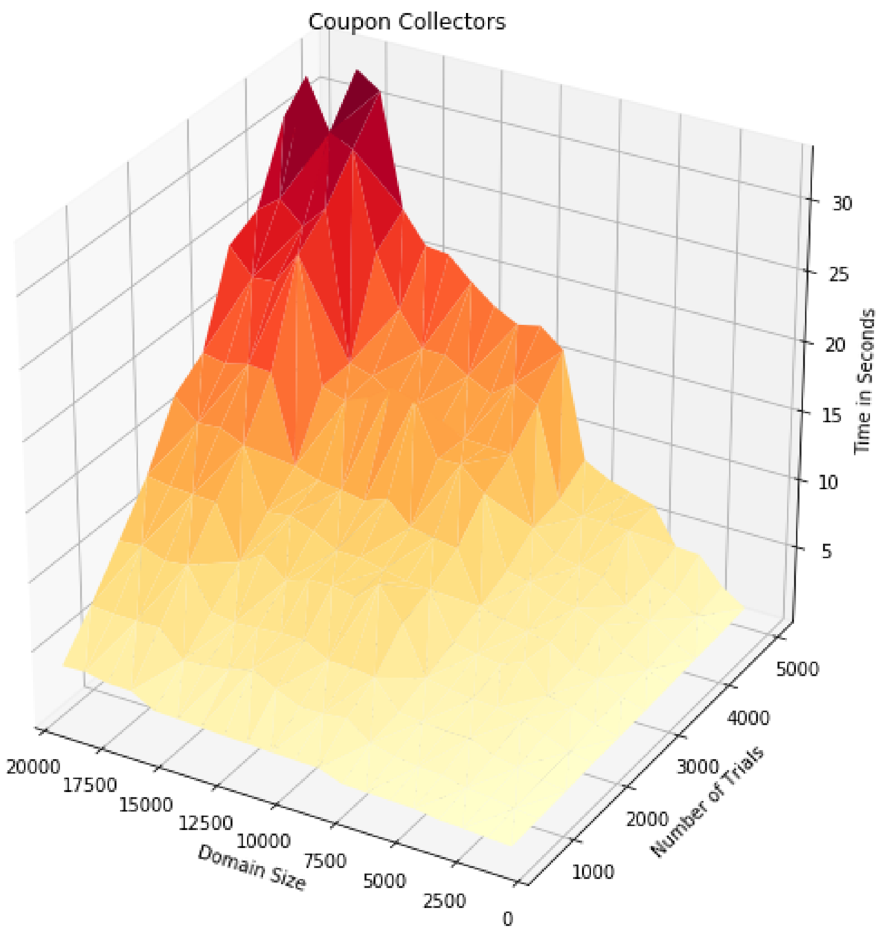

Figure 4: Coupon Collectors Simulation Times

3 Comparing Experiments to Analysis

3.1 A

Now, let's try to calculate analytically the number of random trials needed so that there is a collision with probabiliy of at least 0.5 when the domain size $n = 5000$. For any two people, the probability that they have the same birthday can be represented by:

$$P[x_1 = x_2] = 1 - P[x_1 \neq x_2] = [1 - \frac{365 \times 364}{365 \times 365}] = [1 - \frac{364}{365}]$$

Moreover, given a sample size n, the number of possible pairs is represented by:

$$\#pairs = \binom{n}{2}$$

Thus, the probability that amongst n people, at least two people have the same birthday is represented by:

$$P[x_1 = x_2] = 1 - [1 - \frac{364}{365}]^{\binom{n}{2}}$$

More generally, we can say that for any domain size s, the probability of two repeated observations after n observations is represented by:

$$P[x_1 = x_2] = 1 - [1 - \frac{s-1}{s}]^{\binom{n}{2}}$$

If we let $s = 5{,}000$ and set equation equal to the desired probability of 0.5, then solving for n gives us: $n = 84$, which is about in line with the simulated Birthday Paradox results from earlier.

3.2 B

Now, let's try to calculate analytically the expected number of random trials before all elements are witnessed in a domain of size $n = 300$. To do so, we will use the formula:

$$E(T) = n \times H_n$$

where H_n is the n-th harmonic number. Doing so gives us:

$$E(300) = 300(1 + 1/2 + 1/3 + ... + 1/300) = 1883.7991640898508$$

, which is about in line with the simulated Coupon Collectors results form earlier.