Homework Form for Data Mining [*]

Alex Clemmer

January 26, 2012

1 Overview

This is a sample latex file to use for completing assignments. This particular file is not required. In fact, there are many cool ways to spruce up this plain look. Feel free to use them.

2 Q1: Birthday "Paradox"

A: For domain $n = 1000$, it took 58 random trials.

B: Please see figure 1.

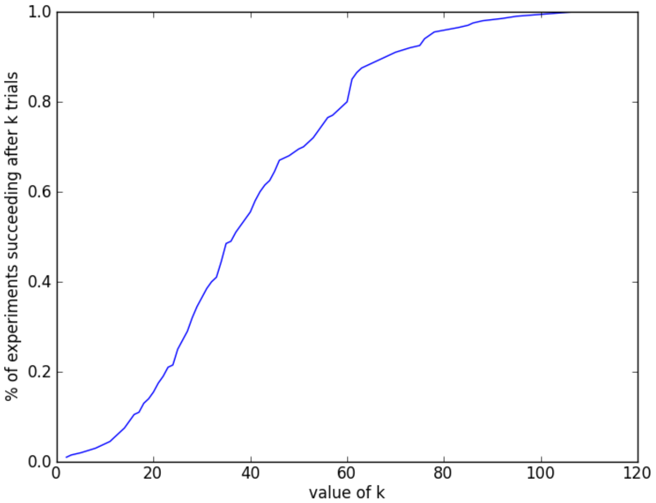


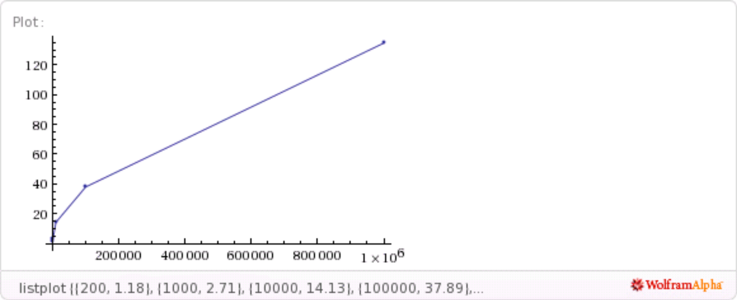
Figure 1: The % of experiments requiring k tests before collision, plotted as a function of k.

C: For some $m = 200$ random variables $X_1 \ldots X_m$ representing the outcomes of m random repetitions of the experiment, the expected value $\mathbf{E}[\vec{X}] = 38.005$

[*] CS 6955 Data Mining; Spring 2012 Instructor: Jeff M. Phillips, University of Utah

D: I check uniqueness using a bit vector of length n, where each place is 0 if we haven't seen the corresponding element before, and is 1 if we have. As long as this bit vector fits in memory, it should scale pretty well. This is probably roughly optimal for exact solutions, though for approximate solutions, you might be able to relax the memory constraint.

The algorithm as coded should run in at most linear time on n, but in practice, it runs sublinearly. In figure 2, we show that even when we exponentially increase m Extending the domain n should not majorly impact run time as long as we assume n is small enough to fit in the traditional constant-time operations. If you're curious, I've plotted the running time as we increase m exponentially, noted by figure 2. It looks to be sub-linear, at least as long as we're assuming constant-time operations.



```
listplot {{200, 1.18}, {1000, 2.71}, {10000, 14.13}, {100000, 37.89},...
```

Figure 2: The running time (x-axis) increases by successive powers of 10; the y-axis denotes the running time of seconds it took to complete.

3 Q2: Coupon Collectors

A: For the domain $n = 60$, the required trials $k = 198$.

B: As we can see in figure 3, the highest bar was pinged 10 times.

C: Please see figure 4.

D: For some $m = 300$ random variables $X_1 \ldots X_m$ representing the outcomes of m random repetitions of the experiment, the expected value $\mathbf{E}[\vec{X}] = 283.61$

E: I implemented this on a streaming basis.

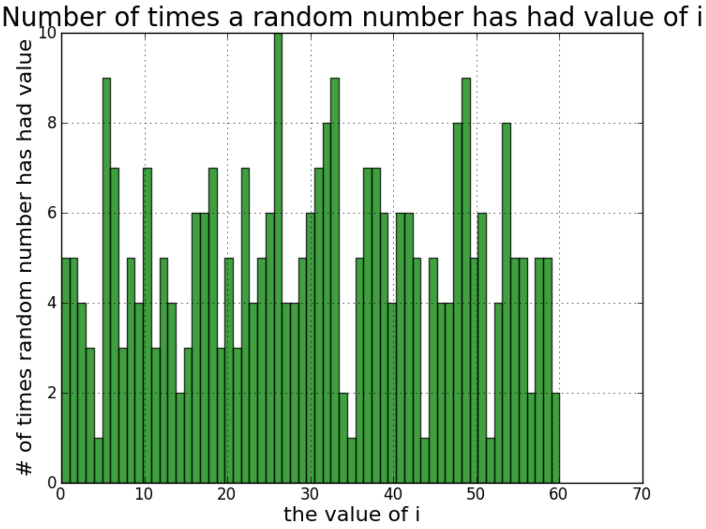


Figure 3: The number of times a random number ends up with value i.

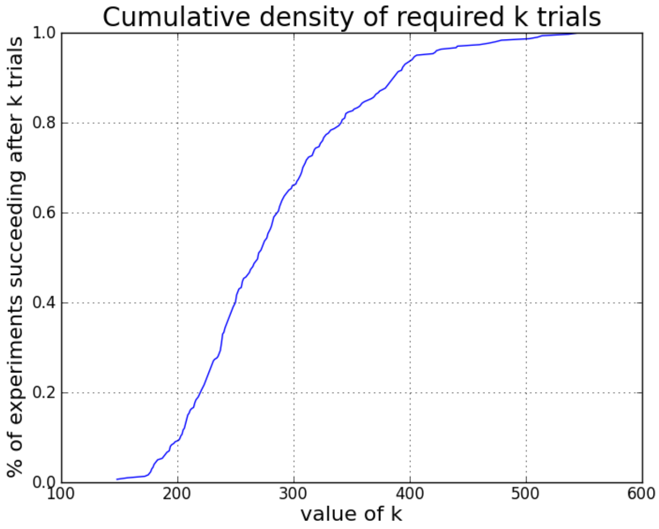


Figure 4: The number of times a random number ends up with value i.