Asymptotic Behavior of Candy Collectors

Iman Hosseini

December 2020

1 The Problem

Jane Street puts pseudo-monthly puzzles on their website, October 2020 they put up a puzzle titled 'Candy Collectors'[1] I read the puzzle and tried to solve it only with pen and paper. As is the case with their puzzles, there is no pen and paper solution, and it is intended that you use computers to solve the puzzle, I relented, and solved the puzzle with the aid of a python script. The original problem statement:

> *Five children went trick-or-treating together and decided to randomly split their candy haul at the end of the night. As it turned out, they got a total of 25 pieces of candy, 5 copies each of 5 different types (they live in a small town). They distribute the candies by choosing an ordering of the 25 uniformly at random from all shufflings, and then giving the first 5 to the first child, the second 5 to the second, and so on.*
> *What is the probability that each child has one type of candy that they have strictly more of than every other trick-or-treater? Give your (exact!) answer in a lowest terms fraction.*

My python and ocaml scripts are accessible here[2]. My solution was automating the process by which I solved the simpler, pen-paper tractable case of 3-kids 3-candy type by hand. I got my name on the solvers board of the challenge and lived happily ever after right? No! I was now wondering what about the case that we have N-kids N-candy, and N is very large? This sort of problem reminds me of the thermodynamics and statistical physics course I took years ago, so many stirling approximations, hand-waving expressions around, you know good 'ol days :)

Also in that same repository my experiment code is included, which gives an estimate of the probability by simulating candy distributions and counting how many times the condition is satisfied. The probability for the case of large N, would be very very tiny: as you increase N, very soon, you can run the experiment for hours and yet not even once, would the condition be satisfied

[1] https://www.janestreet.com/puzzles/candy-collectors/

[2] https://github.com/ImanHosseini/Puzzles/tree/main/JaneStreet/Oct_2020

and this is interesting, put N=1000 and very soon that shiny computer of yours cannot help you even for an estimate of the probability, maybe pen and paper, our old friend can help with this?

2 The Solution

Let us denote $H(n, a)$ the number of ways we can distribute an candy, which come in a types and we have n of each, to n kids. Now observe that without loss of generality we can assume we want the ith kid to get at least the fraction $f = 1/2$ of the ith type of candy. What ever the result, due to symmetry we multiply it by $n!$ to get take into account for different permutations. Now after each kid gets their majority of their designated type, we are left with $(1 - f)n$ of each type and now we have to distribute these candies to the n kids, which is $H(n, (1 - f)n)$ so the probability we are after, in terms of f (later we can plug in $f = 1/2$) would be:

$$p(N, f) = n!\frac{H(n,(1-f)n)}{H(n,n)} \tag{1}$$

So we want to know how this function acts for large N. If you see my post on github, for $N = 5$ we managed to calculate this fraction, that same code explodes for large N. The idea to calculate $H(n, a)$ is that we can represent a configuration of candies-to-kids via $n \times n$ matrix where m_{ij} denotes how many candies of type j, the ith kid got, and $H(n, a)$ is the number of matrices such that the following holds:

$$\begin{aligned}\sum_j m_{ij} &= a\\ \sum_i m_{ij} &= a\end{aligned} \tag{2}$$

It is difficult to count these matrices, and yet it is not hard to find a generating function for it:

$$G(x_1, .., x_n; y_1, .., y_n) = \sum_m x_1^{\sum_j m_{1j}} \dots x_n^{\sum_j m_{nj}} y_1^{\sum_j m_{i1}} \dots y_n^{\sum_j m_{in}} \tag{3}$$

Where m is all $n \times n$ matrices with non-negative elements, it is easy to see that the coefficient of $(x_1^a x_n^a y_1^a y_n^a)$ would be the result of our desired sum: all those powers i.e. $\sum_j m_{ij}$ become a. Now we rewrite equation (3) as:

$$G = \prod_{ij} \sum_m (x_i y_j)^{m_{ij}} \tag{4}$$

Possible m_{ij} are $0, 1, ..$ which means we get the series $1 + (x_i y_j) + (x_i y_j)^2 + ...$ which to a my eyes, having spent couple of years, gruesomely applying taylor expansions and perturbation methods to anything that moves is nothing but $\frac{1}{1-x_i y_j}$ so:

$$G = \prod_{ij} \frac{1}{1 - x_i y_j} \tag{5}$$

Now this is the fun part, now we diverged from that other way of seeing this problem, this new generating function view, despite being actually unsuitable to solve the case for $n = 5$ and get our names on Jane Street website, is suitable for asymptotic analysis for large n, to find the desired coefficient in G we can write it as contour integrals which go round origin and once counterclockwise (interior of unit disk):

$$H(n,a) = \prod_j \oint \frac{dx_j}{2\pi i x_j^{a+1}} \tag{6}$$

And this should remind you of the infamous steepest descent method[3] the integrand is e^A where:

$$A = -(a+1)\sum_i \ln x_i - (a+1)\sum_j \ln(y_j) - \sum_{ij} \ln\left(1 - x_i y_j\right) \tag{7}$$

Steepest descent basically says, in this integral only part which matters is the part where A is extremum. So extremizing A:

$$\begin{aligned} \frac{\partial A}{\partial x_i} &= 0 = -\frac{a+1}{x_i} + \sum_j \frac{y_j}{1 - x_i y_j} \\ \frac{\partial A}{\partial y_j} &= 0 = -\frac{a+1}{y_j} + \sum_j \frac{x_i}{1 - x_i y_j} \end{aligned} \tag{8}$$

This suggests that x_i and y_j are independent of the actual indices (as symmetry suggests) and so:

$$\frac{xy}{1 - xy} = \frac{a+1}{n} \tag{9}$$

Solving for xy we get:

$$xy = \frac{a+1}{n+a+1} \tag{10}$$

And those $2n$ integrals squash to 1 integral where the integrand is:

$$B(n,a) \approx (xy)^{-n(a+1)}(1 - xy)^{-n^2} \tag{11}$$

Plugging in the value of xy and disregarding the 1 as both n and a are assumed to be large we get:

$$B(n,a) \approx (\frac{(n+a)^{n+a}}{a^a n^n})^n \tag{12}$$

[3] https://en.wikipedia.org/wiki/Method_of_steepest_descent

And ultimately remember that $n!$ we discussed first? That will also not contribute to the asymptotic behavior (that behaves like $n^n e^{-n}$) and so lo and behold:

$$p(N, f) \approx (4^{-n} \frac{(2-f)^{n(2-f)}}{(1-f)^{(1-f)n}})^n \tag{13}$$

for $f = 1/2$ we get:

$$p(N) \approx (\frac{3\sqrt{3}}{8})^{n^2} \approx 0.649^{n^2} \tag{14}$$

As it can be seen, this value decreases very fast, even for a value as small as $n = 10$ we get $p \approx 10^{-19}$ and it would take so many experiment runs until you can expect to see a random configuration satisfy the condition. Also don't forget that this whole expression was derived assuming large N, so of course don't expect this to work for small N, the $N = 10$ example was to demonstrate how fast this expression gets tiny.