Expected Number of Unique Elements Generated from a Discrete Distribution

Marco A. Wedemeyer - January 2021

Problem Formulation

Assume you have a vector X of unique elements of length n. Paired with this vector is a discrete probability distribution vector P describing the probability of each element being drawn. Although X is a set, P does not need to be.

$$X = [x_1, x_2, x_3, \ \dots\ , x_n]$$
$$P = [p_1, p, p_3, \ \dots\ , p_n]$$

From the vector X, given the distribution P, d elements are drawn *with* replacement into a vector $Y_{n,d}$. An example is shown below.

Example

$$n = 5,\ d = 4,\ P = [0.1, 0.3, 0.2, 0.2, 0.2]$$
$$Y_{5,4} = [\, x_4, x_2, x_3, x_2 \,]$$

Let m be the unique number of elements in Y. What is the expectation of m given n,d and P?

Properties of m

As the number of draws approaches infinity, m becomes equal to the number of elements in X. Similarly, as the number of elements in X goes towards infinity, the probabilities of each element tend towards zero and m equals d.

$$\lim_{d \to \infty} m = n$$
$$\lim_{n \to \infty} m = d$$

These two limits show that m has an upper bound of n and d, depending on which is smaller. Additionally, when X or Y are empty, m is equal to zero.

$$\text{For } 1 \le n < d,\ 1 \le m \le n$$
$$\text{For } 1 \le d < n,\ 1 \le m \le d$$
$$\text{For } n = 0 \text{ or } d = 0,\ m = 0$$

Existing Solutions

According to a post on Math stackexchange[1], when P is a uniform distribution,

$$E(m_{n,d}) = n\frac{n^d - (n-1)^d}{n^d} \tag{1}$$

According to a post on Math stackexchange[2], when P is a uniform distribution and $d = n$,

$$E(m_n) = n - \frac{(n-1)^n}{n^{n-1}} \tag{2}$$

General Approach

When P is any discrete distribution, take the cartesian product of the vector Y, d times. Let y_i be the rows of this matrix converted into sets.

$$y_i = \{Y_i^d\} \tag{3}$$

Let M be the vector of cardinalities (number of unique elements) of the sets. This vector represents the number of unique values found in all of the permutations with replacement that the elements of X can take over d draws.

$$M_i = |y_i| \tag{4}$$

To complement the values found in M, the corresponding probability vector needs to be calculated. Let p be the cartesian product of P, d times. Let R be the row wise product of p. The expectation of m is the dot product of R and M.

$$p = P^d \tag{5}$$

$$R = \prod_{i=1}^{n^d} p_{i,1} \times p_{i,2} \times \ \dots \ \times p_{i,d} \tag{6}$$

$$E(m_{n,d,P}) = R \bullet M \tag{7}$$

Issues with this approach

Exact calculations of the expectation of m become computationally intractable fairly quickly. This is due to the repeated calculations performed on the cartesian products of the vectors Y and P. The shape of these two resulting matrices are (n^d,d), which does not scale well (see time complexity section).

One attempt to remedy this is to use the symmetry of the cartesian product. After the generation of the cartesian products only half of the calculations are needed. Test calculations estimated a reduction of 25% and 33% on the previous method when altering d and n respectively. Although better, it does not address the exponential nature of the problem.

[1] https://math.stackexchange.com/questions/72223/finding-expected-number-of-distinct-values-selected-from-a-set-of-integers

[2] https://math.stackexchange.com/questions/41519/expected-number-of-unique-items-when-drawing-with-replacement?noredirect=1&lq=1

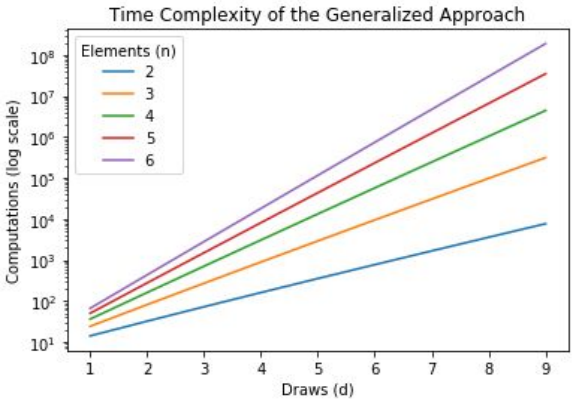
Time Complexity

The time complexity of the total calculation can be found by combining the time complexities of each step. The table below outlines the time complexities per function and the combined total.

Table 1 Time complexities

Function	**Time Complexity**
3	$O(2n^d)$
4	$O(nn^d)$
5	$O(n^d)$
6	$O(dn^d)$
7	$O(n^d)$
Total	$O((4+d+n)n^d)$

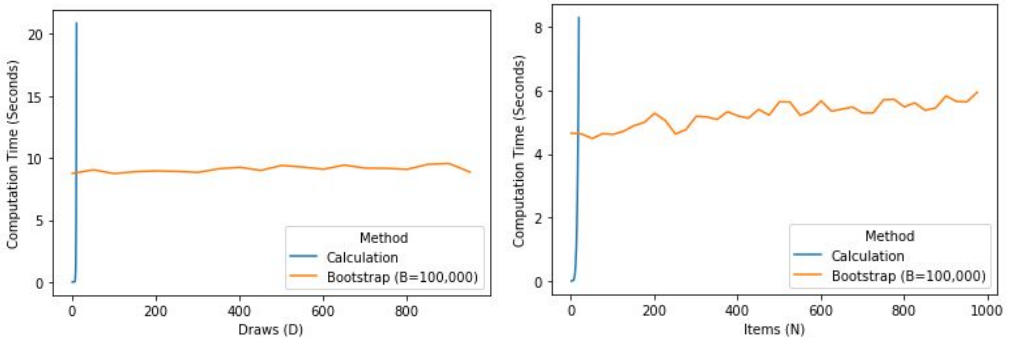
Figure 1 Visualizing Time Complexities



Bootstrapping

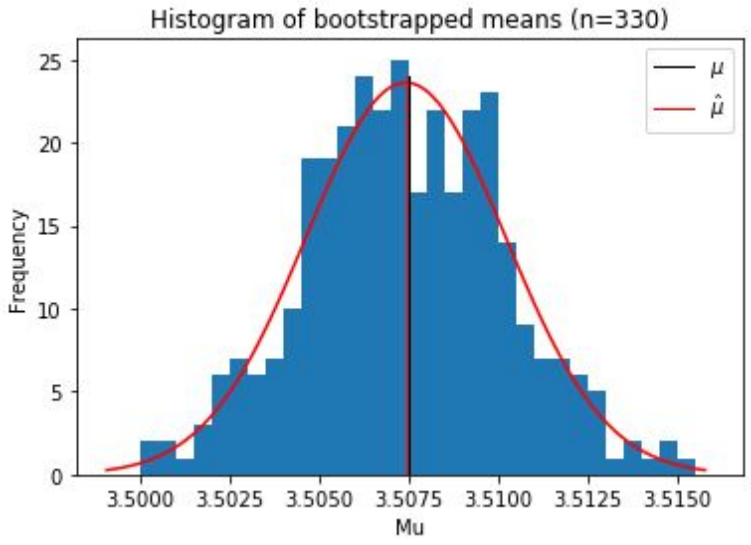
Calculating the value using the existing method is intractable. A simple solution is to bootstrap the value. Using 100,000 bootstraps, the time to calculate m remains tractable. Figure 2 compares the computation times between the two methods for increases in n and d.

Figure 2 Computation times of calculations versus bootstrapping over changing values of n and d



The standard error of the estimator $\widehat{m}$ is 0.00279 based on 330 bootstrapped samples. As Figure 3 below shows, fairly accurate results can be obtained through simulation rather than calculation.

Figure 3 Histogram of bootstrapped sample means



Estimating the Calculation

Even though bootstrapping is a feasible solution its viability is limited with very large numbers for *d* or *n*. Finding a way to approximate the calculated result via a simpler method would be favorable. Inspired by the limits of *m* described at the beginning, a function can be created to approximate *m*. Figure 4 below visualizes the lower limit of 1 and upper limit of *n* as *d* increases for the uniform distribution.

Figure 4 Limits of *m* as *d* increases for different values of *n*

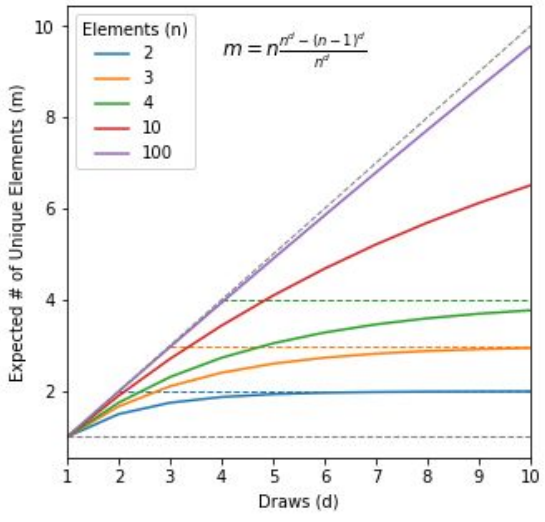
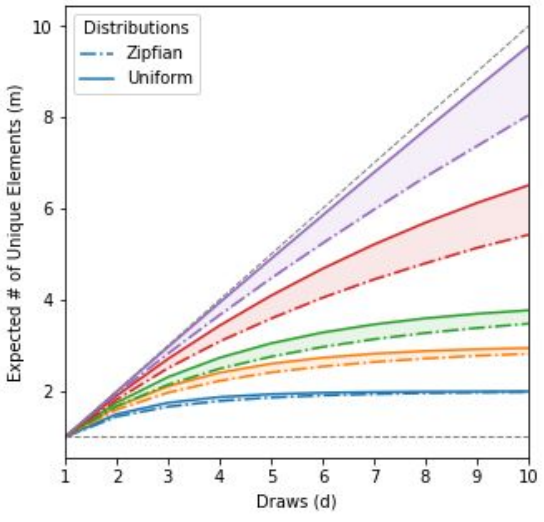


Figure 5 Difference between between Uniform and Zipfian distribution



To create a function that approximates *m* we can start with the function for the uniform distribution that is already known (function 1). As the uniform distribution is the most equal/balanced distribution, any other discrete distribution would result in a lower value of *m* for any pair of *n* and *d*, due to higher chance of duplicate draws. An example of this can be seen in Figure 5, with the Zipfian distribution.

The uniform distribution thus forms an upper bound for m for any pair of n and d for other distributions. As the general bounds described at the beginning still hold, the only difference between distributions is the "speed" of convergence to the limits. With nonparametric distributions, the shape of the function will also be different. Finding unique modifications to the existing function for parametric distributions will probably be easier as their shapes are already defined by certain coefficients.