Elementary Sampling theory

← Back to Chapters

3.26, 3.27 The most probable value of r

The sequence $h(r|N, M, n)$ for fixed N, M, n is unimodal, meaning it first increases, then decreases. To see this we argue as follows.

We want to see wether $h(r+1|N, M, n)$ is bigger than $h(r|N, M, n)$, so we need to compare their fraction to 1. We compute using 3.22

$$h(r+1|N, M, n)/h(r|N, M, n) =$$

$$\frac{(M-r)/(r+1)}{(N-M-n+r+1)/(n-r)} =$$

$$\frac{(r-M)(r-n)}{(r+1)(r+N-M-n+1)} \overset{?}{\gtrless} 1$$

$$(r-M)(r-n) \overset{?}{\gtrless} (r+1)(r+N-M-n+1)$$

$$Mn-(M+n)r \overset{?}{\gtrless} (N-M-n+1)+r(N-M-n+2)$$

$$Mn-(N-M-n+1) \overset{?}{\gtrless} r(N+2)$$

$$\frac{Mn-(N-M-n+1)}{N+2} \overset{?}{\gtrless} r$$

$$\frac{Mn+M+n+1}{N+2} \overset{?}{\gtrless} r+1$$

$$\frac{(M+1)(n+1)}{N+2} \overset{?}{\gtrless} r+1$$

The sequence $h(r)$ increases while the left hand side is bigger.

Thus denoting by r' the number $\frac{(M+1)(n+1)}{N+2}$ we see that if r' is an integer, then $h(r)$ increase until $r = r' - 1$, then $h(r' - 1) = h(r')$, then the $h(r)$ decrease. If r' is not an integer, then $h(r)$ increase until $h(INT(r'))$, then decrease.

Remak 1: Note that the expected number of red balls is just the "naive" $n\frac{M}{N}$ (this is not hard to show using linearity of expectation, see Example 4.2.3 in Blitzstein-Hwang "Introduction to Probability").

Remark 2: The above result can be restated in the following way: pretend to add one red and one white ball to the urn (for a total of $N + 2$) and draw $n + 1$ balls from the resulting urn. Compute the "naive" most likely fraction of red balls $\frac{M+1}{N+2}$ and the "naive" most likely number of red balls $\frac{(n+1)(M+1)}{N+2}$. Now subtract 1. This is (up to rounding) the most likely number of red balls drawn in the original procedure. This seems somewhat reminiscent of the correction that putting a beta prior on Bernoulli makes to the posterior expectation, but I have no idea if there is more to this connection than that.

3.29 Symmetry of $h(r|N, M, n)$

Combinatorial proof that

$$h(r|N, M, n) = h(r|N, n, M).$$

Remark: This is Theorem 3.4.5 in Blitzstein - Hwang "Introduction to Probability". See also Theorem 3.9.2.

By definition, $h(r|N, M, n)$ is computed as follows. Lay down N balls, labelled $1, ..., N$. Pick the subset $R_0 = \{1, ..., M\}$ of them and paint it red. Then pick a subset D of size n of all the ball, and compute $r = |D \cap R_0|$. The fraction of Ds that give specific answer r is by definition $h(r|N, M, n)$.

Now suppose instead we pick a different subset R_1 of size M to be red, and repeat the procedure above: pick D, and compute $r = |D \cap R_1|$. We claim that the fraction of Ds that give specific answer r is still $h(r|N, M, n)$. In deed, there exists a permutation of $\{1, ..., N\}$ taking R_1 to R_0 ("sort the reds to be first"); the same permutation takes Ds that give $D \cap R_1 = r$ to those that give $D \cap R_0 = r$. Hence there are the same number of Ds in both circumstances.

The above argument means that $h(r|N, M, n)$ can be also computed as follows. Lay down N balls, labeled $1, ..., N$. Pick **any** subset R of them of size M and paint it red. Then pick a subset D of size n of all the ball, and compute $r = |D \cap R|$. The fraction of **Rs and Ds** that give specific answer r is then $h(r|N, M, n)$.

But the above procedure remains the same if we exchange M and n and rename "paint red" into "pick" and "pick" into "paint red". Then it computes $h(r|N, n, M)$. So the two numbers are equal.

Remark: This view of hypergeometric distribution as giving probabilities of overlap of two subsets ("the red" and "the picked") removes all time dependence and, in my opinion, sheds a lot of light on the discussion at the end of Section 3.2.

Exercise 3.2

Modified from stackexchange.

Let A_i be the statement "color i was not drawn". Then

$$P(A_i) = \frac{\binom{N-N_i}{m}}{\binom{N}{m}}$$

This is the number of ways of drawing m balls from $N - N_i$ non-i colored balls, divided by the number of ways of drawing m balls from all N colored balls. Similarly,

$$P(A_iA_j) = \frac{\binom{N-N_i-Nj}{m}}{\binom{N}{m}}, P(A_iA_jA_k) = \frac{\binom{N-N_i-Nj-N_k}{m}}{\binom{N}{m}}...$$

The probability we want is $1 - P(A_1 + A_2 + A_3 + A_4 + A_5)$, the probability that no colors will be missing from our draw.

By the sum rule this can be calculated as

$$1 - \sum_i P(A_i) + \sum_{i<j} P(A_iA_j) - \sum_{i<j<k} P(A_iA_jA_k)...$$

where the first sum term is over all subsets of 1 color, the second is over all subsets of 2 colors, etc.

This calculation can be done in python like so:

```python
from itertools import combinations
from scipy.special import comb, perm

def prob_all_colors_drawn(m, N):
    '''
    m is number of balls drawn
    N is a list containing how many of each color is in the urn
    '''
    k = len(N) # number of colors

    total = 1.0 # start with 1
    for i in range(1,k+1):

        # calculate each sum term
        conjunction_prob = 0
        for Ns in combinations(N, i): # for all combinations of i colors
            conjunction_prob += comb(sum(N) - sum(Ns), m,True)/comb(sum(N),m,True)

        # alternately add or subtract the sum term
        total += ((-1)**i)*conjunction_prob
    return total
```

You can modify and run this code on Google Colab, and see a monte carlo approximation of the same problem.

The code shows that to be 90% confident of getting all 5 colors we need 15 draws.

Exercise 3.3

First we assume a uniform prior over k: $p(k) = \frac{1}{50}$. We can obtain an upper bound on $p(k|colors = 3)$ like so:

$$\begin{aligned}
p(k|colors = 3) &= \frac{\sum_{allN_1...N_k} p(colors = 3|k, N_1, N_2, ...N_k)p(N_1, N_2, ...N_k|k)p(k)}{\sum_k p(colors = 3|k)p(k)} \\
&< \sum_{allN_1...N_k} p(colors = 3|k, N_1, N_2, ...N_k)p(N_1, N_2, ...N_k|k)\frac{1}{50} \times 334 \\
&< \max_{N_1,...,N_k} [p(colors = 3|k, N_1, N_2, ...N_k)] \times 6.66 \\
&< \binom{k}{3} \max_{N_1,...,N_k} [p(\overline{A_1A_2A_3}A_4...A_k|k, N_1, N_2, ...N_k)] \times 6.66
\end{aligned}$$

The first step uses the following lower bound on the denominator.

$$\begin{aligned}
p(colors = 3|k = 3) &= \sum_{allN_1...N_k} p(colors = 3|k = 3, N_1, N_2, N_3)p(N_1, N_2, ...N_k|k) \\
&> \min_{allN_1...N_k} p(colors = 3|k = 3, N_1, N_2, N_3) \\
&= p(colors = 3|k = 3, N_1 = 48, N_2 = 1, N_3 = 1) \\
&> p(\overline{A_1A_2A_3}|k = 3, N_1 = 48, N_2 = 1, N_3 = 1) \\
&= 0.15
\end{aligned}$$

so the denominator must contain a term greater than $0.15/50 = 1/334$, hence the sum is also greater than that value. The value 0.15 is calculated in the Colab code.

The second step is taken because it contains a weighted average, same as above. We can find an upper bound over the weighted average by finding the $N_1, N_2, ...N_k$ that maximises $p(colors = 3|k, N_1, N_2, ...N_k)$. Note that this makes the proof entirely independent of the prior $p(N_1, N_2, ...N_k|k)$.

The third step is found because the statement $colors = 3$ is the logical sum of $\binom{k}{3}$ conjunctions of the form $\overline{A_1A_2A_3}A_4...A_k$, each of which has 3 A's negated. This sum is bounded by $\binom{k}{3} \max\limits_{N_1,...,N_k} [p(\overline{A_1A_2A_3}A_4...A_k|...)]$.

$p(\overline{A_1A_2A_3}A_4...A_k|k, N_1, N_2, ...N_k)$ can be calculated with:

$$p(\overline{A_1A_2A_3}|A_4...A_k...)p(A_4...A_k|...) = [1-p(A_1+A_2+A_3|A_4...A_k...)]p(A_4...A_k|...)$$

An equivalent but more efficient formula for this likelihood can be found in the alternative approach below. From either formula it's similar to the calculations from Exercise 3.2, and straightforward to implement in python.

We run the calculations using python in the same Colab as above, and show that we can be *at least* 99% confident that $3 \le k \le 20$. Changing the prior $p(k)$ to one that favors lower k will tighten the bound.

Alternative approach: Data likelihood estimates, no prior. Suppose the color counts in the bin are given by the tuple $\vec{N} = (N_1, N_2, \dots, N_k)$ with $N_i \ge 1$, $\sum N_i = 50$ and $k \ge 1$. Of course under this assumption the number of colors in the bin is just k.

If we had a prior $p(\vec{N})$ for various $\vec{N}$ tuples, we would compute posterior over same tuples by multiplying the $p(\vec{N})$ by likelihood of getting 3 colors from a sample of 20 balls $L(\vec{N})$ (which is fully determined by $\vec{N}$, see below), and renormalizing.

Let's try to see what the data likelihood would be $L(\vec{N})$ for different N_i tuples (to see how much probability of each N_i tuple is suppressed/boosted by the data).

So, again, we suppose the numbers of balls of different colors in the urn are $N_1, N_2, \dots, N_k$. What is the probability of event "a sample of 20 contains balls of exactly 3 colors"? First we choose the 3 colors, i_1, i_2 and i_3 and then apply Exercise 3.2 to the triple $N_{i_1}, N_{i_2}, N_{i_3}$ to get

$\binom{N_{i_1}+N_{i_2}+N_{i_3}}{20} - \binom{N_{i_1}+N_{i_2}}{20} - \binom{N_{i_2}+N_{i_3}}{20} - \binom{N_{i_1}+N_{i_3}}{20} + \binom{N_{i_1}}{20} + \binom{N_{i_2}}{20} + \binom{N_{i_3}}{20}$

possible draws that satisfy this, summing over all the selections of i_1, i_2, i_3 to get the total number of draws with 3 colors. Each N_i will be chosen in $\binom{k-1}{2}$ triples and each pair N_i, N_j in $k-2$ triples. So the sum is

$\sum_{triples} \binom{N_{i_1}+N_{i_2}+N_{i_3}}{20} - (k-2)\sum_{pairs} \binom{N_{j_1}+N_{j_2}}{20} + \binom{k-1}{2} \sum_l \binom{N_l}{20}$

(The total number of draws is always the same $\binom{50}{20}$.)

Now, $Q(x) = 20!\binom{x}{20} = x(x-1)\dots(x-19)$ is increasing in $x > 19$, with ratio $Q(x+1)/Q(x) = x/(x-19)$; for x near 50 this is a factor of about 1.7. So, at first glance, those k tuples with largest possible $i_1+i_2+i_3$ will have highest data likelihood. All those with $i_1+i_2+i_3 = 50$ have $k = 3$ of course. The $\vec{N} = [48, 1, 1]$ has $\binom{47}{17}$ sequences, and data likelihood of about 0.06. The $\vec{N} = [17, 17, 16]$ has data likelihood 0.99995.

The $\vec{N} = [51-k, 1, \dots, 1]$ gives data likelihood $\binom{k-1}{2}\binom{50-k}{17}$, which goes

0	0	$5.82 \cdot 10^{-2}$	$1.11 \cdot 10^{-1}$	$1.40 \cdot 10^{-1}$
$1.46 \cdot 10^{-1}$	$1.34 \cdot 10^{-1}$	$1.13 \cdot 10^{-1}$	$9.01 \cdot 10^{-2}$	$6.78 \cdot 10^{-2}$
$5.20 \cdot 10^{-3}$	$2.97 \cdot 10^{-3}$	$1.63 \cdot 10^{-3}$	$8.61 \cdot 10^{-4}$	$4.35 \cdot 10^{-4}$
$2.20 \cdot 10^{-6}$	$6.96 \cdot 10^{-7}$	$1.96 \cdot 10^{-7}$	$4.80 \cdot 10^{-8}$	$9.82 \cdot 10^{-9}$
$1.58 \cdot 10^{-9}$	$1.78 \cdot 10^{-10}$	$1.05 \cdot 10^{-11}$	0	

So when k reaches 16 even the most advantageous color counts $\vec{N}$ are suppressed at least 10^4 times more than the most disadvantageous ones with $k = 3$. So it seems no matter what reasonable prior for the color counts one takes the posterior should be mostly supported on $3 \leq k \leq 16$.

Remark: For exact inference, it is not sufficient to have a prior over k, since same prior over k may correspond to different priors over $\vec{N}$, producing different posteriors (the reason being that data likelihoods are not determined by k). Thus a prior over $\vec{N}$ is needed. Of course, approximate inference a prior over k may be sufficient (as in the first approach above; however it seems likely that a reasonable prior over $\vec{N}$ will not result in a uniform prior over k).

Discussion of priors How to get a prior over $\vec{N}$ is not clear to me. One option is to model the urn being filled by sampling from an (infinite) population. We can use several versions:

Version 1: The population has K colors, with frequencies $p_1, \dots, p_K$. The set of such populations is the union of $K-1$ dimensional simplexes for $K = 1, 2, \dots$.

Version 2: The population has infinitely many colors and probabilities of each color p_i. The set of such populations is the "infinite dimensional simplex" $p_i \geq 0$, $\sum p_i = 1$.

These two versions are actually in a way equivalent, the infinite simplex being union of by finite simplex strata (the difference being that each d-dimensional simplex appears countably many times). Because of this I will only consider Version 1.

For each population, the probability of every $\vec{N}$ is determined. So if we had a prior over the population types it would determine a corresponding prior over fully specified (though maybe still intractable) inference problem.

How to get a prior over population types also seems unclear. One could try to take some maximal entropy priors, or do some further hierarchical modeling, but since I do not plan to actually implement the inference, I will not go into details of this.

Exercise 3.4

Denote by F_i be the event "i is fixed", and, for any $I \subset \{1, ..., n\}$, denote by F_I the event "all i in I are fixed", i.e. $F_I = \prod_{i \in I} F_i$.

We are looking for $P(\sum F_i)$. By inclusion exclusion this is

$$P(\sum F_i) = \sum_{I \subset \{1,...,n\}, I \neq \emptyset} (-1)^{|I|+1} P(F_I).$$

For a given subset of size k probability that it is fixed is $\frac{(n-k)!}{n!}$, and there are $\binom{n}{k}$ such subsets, so the sum over those I with size k gives $(-1)^{k+1}\frac{1}{k!}$. Plugging this in we obtain

$$h = P(\sum F_i) = \sum_{k=1}^{n} (-1)^{k+1} \frac{1}{k!},$$

as wanted.

Observe that $1 - h$ is the value of k-th order Taylor series for e^x evaluated at $x = -1$, which, as $k \to \infty$, converges to $e^{-1} = 1/e$.

Exercise 3.5

Similarly to 3.4, consider the event E_I=the bins with labels $i \in I$ are left empty; then $P(E_I) = (M - |I|)^N / M^N$ and by inclusion-exclusion $P(\overline{\sum E_i})$ is

$$\frac{1}{M^N} \sum_{k=0}^{M} (-1)^k \binom{M}{k} (M-k)^N.$$

Remark: We are computing probability that a function from a set of size N to a set of size M is onto. There are M^N total functions, and the number of surjective ones is $M!$ times a Stirling number of second kind.

Some of Exercise 3.6

Remark: If the initial distribution (for R_0, i.e. $P(red) = P(R_0) = p$, $P(white) = P(\bar{R}_0) = q$) were the same as the limit distribution π (formula 3.125, $\pi(red) =$

$\lim P(R_k) = \frac{p-\delta}{1-\epsilon-\delta}$, $\pi(white) = \lim P(\bar{R}_k) = \frac{q-\epsilon}{1-\epsilon-\delta}\}$, this would be a steady state Markov chain, whose time-reverse process is also a Markov chain with transition probabilities $M^r_{ij} = \frac{\pi_j}{\pi_i} M_{ji}$ (note that for 2 state chains one always has $M^r_{ij} = M_{ij}$). This is precisely condition 3.131. Under this condition it is easy to compute $P(R_j|R_k)$ with $j < k$, and in the 2-state case that we are considering, they would be the same as $P(R_k|R_j)$ (as in 3.134). However in this exercise the Markov chain starts from the initial distribution that, in general, is not the steady state distribution, so reversing the time produces a process (indexed by negative integers) which is a Markov chain which is not time-homogeneous. Maybe there is still a way to apply general theory of Markov chains to the problem of "backward inference" in this setting; absent that, we proceed by a direct computation (but observe that the reversed process is connected to the limiting behavior of the result, see below).

As usual, all probabilities are conditioned on C. Equation 3.129 is

$$P(R_k|R_j)P(R_j) = P(R_j|R_k)P(R_k)$$

Equation 3.118 is

$$P(R_k) = \frac{(p-\delta) + (\epsilon+\delta)^{k-1}(p\epsilon - q\delta)}{1-\epsilon-\delta}$$

Finally equation 3.128 is

$$P(R_k|R_j) = \frac{(p-\delta) + (\epsilon+\delta)^{k-j}(q-\epsilon)}{1-\epsilon-\delta}$$

Plugging in

$$P(R_j|R_k) = \frac{(p-\delta) + (\epsilon+\delta)^{j-1}(p\epsilon - q\delta)}{(p-\delta) + (\epsilon+\delta)^{k-1}(p\epsilon - q\delta)} \frac{(p-\delta) + (\epsilon+\delta)^{k-j}(q-\epsilon)}{1-\epsilon-\delta}$$

If both j and k go to infinity but $k-j$ is kept constant, this converges to

$$P(R_{\infty+d}|R_\infty) = \frac{(p-\delta) + (\epsilon+\delta)^{d}(q-\epsilon)}{1-\epsilon-\delta},$$

which is precisely the "reversed process" result (for large j and k the influence of the initial distribution not being the stationary one has dissipated; in the 2 state case the reversed process is the same as the forward one, but that's a special feature; when the number of states (i.e. colors) is higher limit behavior of "backward inference" is given by the reversed process).