Pólya Urn Models

Matthew Rathkey, Roy Wiggins, and Chelsea Yost

May 27, 2013

1 Urn Models

In terms of concept, an urn model is simply a model for simulating chance occurrences and thereby representing many problems in probability. In terms of general design, an urn model consists of one or more urns filled with variously colored balls that are added to or removed from the urn over time according to some schema. Urn models are usually studied through attempts to characterize the state of the balls in the urn(s) as the system evolves over time (either by discrete steps or by continuous time).

Urn models are quite powerful in terms of range of utility, as both they and the computational tools associated with them can be applied to many concepts within mathematics and other fields. For instance, urn models can help simulate genetic mutations within evolution or the growth of binary search trees in computer science (a topic that will be further discussed shortly). To help showcase the flexible applications of urn models within the realm of pure probability, however, let us briefly introduce a few examples of how urn models can be used to simulate some common probability distributions.

1.1 Discrete Uniform Distribution

Let's say we have a single urn filled with n balls labelled $a_1, \ldots, a_n$. We reach into the urn and pull out one ball "at random", which here means that all choices of a single ball are equally likely. If we let U denote the ball chosen,

then U is a discrete uniform random variable over the set $\{a_1, \ldots, a_n\}$. In other words, $U \sim \text{Uniform}(a_1, \ldots, a_n)$.

1.2 Binomial Distribution

This time let's say we have a single urn filled with w white balls and b blue balls (and no other balls). We perform n draws from the urn, sampling with replacement. "Sampling with replacement" means that once we draw a ball from the urn we replace it back within the urn so that each draw has equal chances of picking either a white or blue ball. If we let Y denote the number of white balls drawn from the urn after n draws, then Y is a binomial random variable with n trials (draws from the urn) and probability of success $\frac{w}{w+b}$ (the chance of drawing a white ball). This can be written via notation as $Y \sim \text{Binomial}(n, \frac{w}{w+b})$.

1.3 Geometric Distribution

As a final example, let's again say that we have a single urn filled with w white balls and b blue balls. We sample balls from the urn one at a time and with replacement as before, but in this scenario we perform an unspecified number of draws until we first obtain a white ball. If we let Z denote the draw on which a white ball is first selected, then Z is a geometric random variable with probability of success $\frac{w}{w+b}$ in picking a white ball. In terms of notation once again, $Z \sim \text{Geometric}(\frac{w}{w+b})$.

Let us now hone in our focus to Pólya urn models more specifically.

2 Pólya Urn Models

A Pólya urn model consists of a single urn containing balls of up to k different colors (Mahmoud, p. 45). The urn evolves in discrete time steps. At each step we reach into to urn and sample a ball uniformly at random. We note the color of the ball, then return it to the urn. If the ball drawn is of color i, where $i = 1, \ldots, k$, we add $a_{i,j}$ balls of color j to the urn. Urn replacement schemes are typically represented as square matrices or schemas:

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,k} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k,1} & a_{k,2} & \cdots & a_{k,k} \end{pmatrix}$$

In general the entries in the matrix can be deterministic or random, positive or negative. For the purposes of this paper we will be dealing entirely with deterministic urn schemas. We will also primarily show results for a two-color urn scheme, however most of mathematics generalizes easily to a n-color urn. Throughout this paper we will use the schema:

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

to represent an unspecified two-color urn, and let color one be white and color two be blue.

2.1 Goals

Now that we have defined Pólya models we begin to analyze their behavior. Ideally we would like to find results about the distribution of the number of balls of each color in the urn after n draws which we write as $R_n = \binom{W_n}{B_n}$, where W_n is the number of white balls in the urn after n draws, and B_n is the number of blue balls in the urn after n draws. It may or may not be possible to find an exact distribution for any given urn scheme. Thus we may alternatively work towards a limit result as $n \to \infty$ or simply attempt to find the expectation of R_n.

2.2 Tenability

As we work towards asymptotic results we introduce the condition of tenability to ensure our urn can withstand the test of time. A tenable urn is one from which we can continue drawing and replacing balls indefinitely on every possible stochastic path without ever getting "stuck" in a state where

we cannot follow the replacement rules. (Mahmoud, p. 46) For example, if all entries in an urn schema are positive, such as in the following urn:

$$\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$

then the urn is tenable under any nonempty initial state. While a schema such as:

$$A = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$$

is never tenable, as after each drawing a ball is removed from the urn. Thus, if we begin with n white balls and m blue balls, the urn is depleted after $n + m$ drawings.

In general, tenability is depends on both the urn schema and the initial conditions. It is possible to classify all two-color schemas by the number and arrangement of their negative entries, and to determine the necessary and sufficient conditions for tenability in each case.

2.3 Pólya-Eggenberger

Consider the Pólya-Eggenberger urn with the schema $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

(Mahmoud, Theorem 3.1, p. 51) Let $\widetilde{W}_n$ be the number of white balls drawn after n draws and $\tau_0 = W_0 + B_0$. Then

$$\mathbf{P}\left(\widetilde{W}_n = k\right) = \frac{W_0(W_0+1)\ldots(W_0+(k-1)))B_0(B_0+1)\ldots(B_0+(n-k-1)))}{\tau_0(\tau_0+1)\ldots(\tau_0+(n-1))}\binom{n}{k}$$

$$= \frac{\langle W_0\rangle_k \langle B_0\rangle_{n-k}}{\langle \tau_0\rangle_n}\binom{n}{k}$$

Where $\langle a\rangle_k = a(a+1)(a+2)\cdots(a+k-1)$ is the kth rising factorial of a.

Proof (based roughly on Mahmoud):

We consider one arrangement of drawing k white balls in n draws: drawing k white balls followed by $n-k$ blue balls.

The probability of getting a white on the first draw is $\frac{W_0}{\tau_0}$. Given the previous event, the probability of then getting a white on the second is $\frac{W_0+1}{\tau_0+1}$, since we have added one more white to the urn. So the probability of getting k whites in k draws is $\frac{W_0}{\tau_0}\frac{W_0+1}{\tau_0+1}\ldots\frac{W_0+(k-1)}{\tau_0+(k-1)}$. By the same logic, the chance of then getting $n-k$ blues in the remaining $n-k$ draws is $\frac{B_0}{\tau_0+k}\frac{B_0+1}{\tau_0+k+1}\ldots\frac{B_0+(n-1)}{\tau_0+(n-1)}$.

We can show that the probability of each other arrangement with k whites will have this probability. At each draw $j=1,\ldots n$, we are picking from τ_0+j balls, so this draw's contribution to the entire quantity will look like $\frac{\alpha}{\tau_0+j}$. There are n draws so the total denominator will be $\langle\tau_0\rangle_n$.

Each α will look like W_0+i or B_0+i, where i is the number of white or blue balls previously drawn, since that is how many white or blue balls there are in the urn to be picked from. It will not matter when, for example, the first blue ball is drawn: since it is the first blue ball, there are B_0 blue balls to choose from. Each of W_0+i, for $i\in 0\ldots k$ will appear in the numerator, and similarly for blues, so we arrive at the same probability for each ordering. There are $\binom{n}{k}$ such orderings. $\square$

2.4 Pólya-Eggenberger (Asymptotic)

In the same Pólya-Eggenberger scheme as before, the following limiting distribution holds (Mahmoud, Theorem 3.2, p. 53):

$$\frac{\widetilde{W}_n}{n} \xrightarrow{\mathcal{P}} \text{Beta}\left(W_0, B_0\right)$$

$$\text{So } W_n \text{ grows like } n$$

Proof:

We rewrite $\mathbf{P}\left(\widetilde{W}_n = k\right)$ as a ratio of gammas:

$$\mathbf{P}\left(\widetilde{W}_n = k\right) = \frac{\frac{\Gamma(W_0+k)}{\Gamma(W_0)}\frac{\Gamma(B_0+n-k)}{\Gamma(B_0)}}{\frac{\Gamma(\tau_0+n)}{\Gamma(\tau_0)}}\binom{n}{k}$$

$$\mathbf{P}\left(\widetilde{W}_n = k\right) = \frac{\frac{\Gamma(W_0+k)}{\Gamma(W_0)}\frac{\Gamma(B_0+n-k)}{\Gamma(B_0)}}{\frac{\Gamma(\tau_0+n)}{\Gamma(\tau_0)}} \frac{\Gamma(n+1)}{\Gamma(k+1)\Gamma(n-k+1)}$$

$$\mathbf{P}\left(\widetilde{W}_n = k\right) = \frac{\Gamma(k+W_0)}{\Gamma(k+1)} \frac{\Gamma(n-k+B_0)}{\Gamma(n-k+1)} \frac{\Gamma(n+1)}{\Gamma(n+\tau_0)} \frac{\Gamma(\tau_0)}{\Gamma(W_0)\Gamma(B_0)}$$

This can be found using Stirling's approximation to the ratio of gamma functions:

$$\frac{\Gamma(x+r)}{\Gamma(x+s)} = x^{r-s} + O\left(x^{r-s-1}\right) \text{ as x}\to\infty$$

and taking the limit as $n \to \infty$. Varying the initial conditions produces the probability densities in Figures 1 - 3.

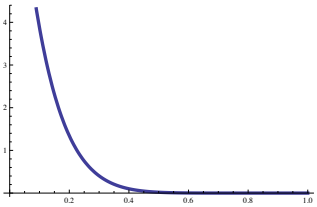

Figure 1:
$W_0 = 10, B_0 = 1$

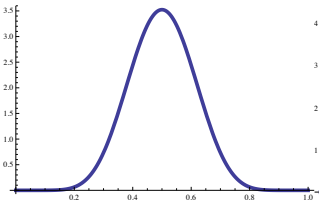

Figure 2:
$W_0 = 10, B_0 = 10$

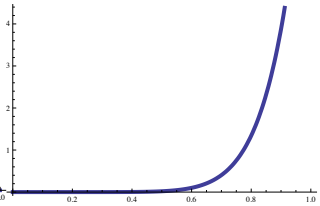

Figure 3:
$W_0 = 1, B_0 = 10$

Notice that even though the color with more balls at the beginning tends to win out in the limit, there is still a decent chance of a non-negligible number of balls of the other color being drawn.

3 Binary Search Trees

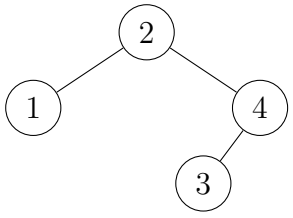

Figure 4: A small BST

To motivate the next sections, we are going to examine a property of binary search trees (BSTs) which can be investigated using a Pólya urn scheme, under the right conditions (Mahmoud, 8.1.1, p. 136). BSTs are a type of data structure considered in computer science. The idea behind a BST is that we want to store data in nodes; each node can have some child nodes; and we want some easy way to be able to search for a piece of data in the tree, as long as we have a total ordering over the data.

Figure 4 shows a small BST containing the values 1, 2, 3, 4. The node containing 2 is the root; it has two children. In a binary tree, each node has no more than two children, denoted "left" and "right" for convenience. We want to look at how many nodes are going to be leaves, that is, with no children, for large, "random" trees.

To talk about random trees, we are going to think of a tree as "growing" from an empty tree, according to an insertion rule:

- If there is no root, create one and store the value there.
- If there is a root, examine it.
  - If the value to insert is less, insert it as the left child of the root. If there is already a node there, examine it as we did the root, and recurse until we do find an empty slot.
  - If the value to insert is greater, insert it as the right child if it does not already exist, and recurse if it does.

To generate a tree of size n, we will take a permutation of the first n integers and insert them into the tree in that order. This is different from simply taking each tree of size n as equally likely: a given tree might be generated by more than one distinct permutation. But this notion of a random tree is amenable to representation by a Pólya urn.

We begin by extending the tree as in Figure 5, that is, we fill out each original node so it has two children. We color each extended node white if its sibling is also an extended node, or blue if its sibling is an original node. Note that insertions will each place the new data in one of the extended nodes, and generate new extended nodes, and the number of white and blue nodes will change (see Figures 6 and 7).

Specifically, if it lands on a white node, its sibling (formerly white) turns

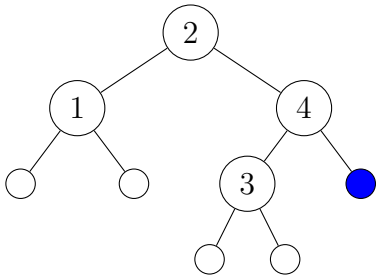

Figure 5: Our extended BST

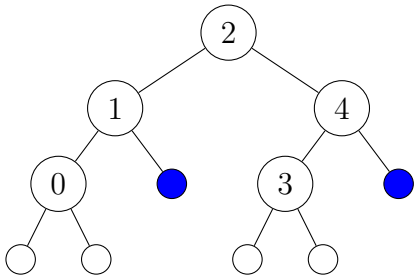

Figure 6: After inserting 0.

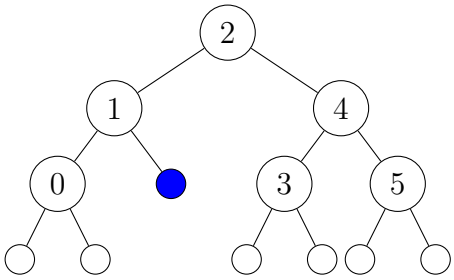

Figure 7: After inserting 5.

blue, and it gains two white extended children, leaving the number of whites the same and increasing blues. If it lands on a blue, the number of blues goes down by one and the number of whites goes up by two. We can represent this using a Pólya urn schema:

$$A = \begin{matrix} \text{W} & \text{B} \\ \begin{pmatrix} 0 & 1 \\ 2 & -1 \end{pmatrix} & \begin{matrix} \text{to insert on a } \textbf{white} \text{ draw} \\ \text{to insert on a } \textbf{blue} \text{ draw} \end{matrix} \end{matrix}$$

Note that this assumes that the growth of a tree behaves as if each node is equally likely to be picked, since each ball in the urn is picked with a uniform distribution. This schema does not much resemble the Pólya-Eggenberger scheme analyzed previously. A limit result for a set of cases that includes this one does exist, but we'd prefer a general method for solving a larger number of Pólya urns.

4 Urns in a Continuous Setting (Poissonization)

To expand towards such a general method, we will introduce the idea of embedding our discrete urn setting in continuous time (Mahmoud, ch. 4). This will ultimately allow us to characterize the expected state of any two-color Pólya urn scheme after any period of time.

To introduce the idea of an urn scheme evolving in continuous time, let us first consider a very trivial case. Let us say we have a single urn with a single white ball inside, and each time a draw is performed we simply remove the lone white ball from the urn and then immediately replace it back within the urn. To make things at least somewhat interesting, however, the time at which we make those draws is randomly determined according to an exponential random variable with parameter $\lambda = 1$. In other words, the amount of time (in units) that passes between draws is randomly determined according to this Exponential(1) random variable. Thus if we begin at time $t = 0$ time units, our first draw could conceivably occur after a randomly determined 1.023 time units, the next draw another 2.889 time units after that, the next draw after 0.337 time units, then 1.214 time units, and so on. The fact that the parameter $\lambda = 1$ simply means that on average draws will be 1 time unit apart, and this value is chosen for convenience.

To expand this idea to an urn with multiple balls, let us first imagine that the randomly determined times between draws in the trivial case were determined by an exponential "stopwatch" or "clock" that stops or "goes off" after the amount of time (in units) determined by the exponential random variable. In the general case with multiple balls in an urn, then, we can assign each ball its own independent exponential "stopwatch". The ball with the corresponding stopwatch that stops with the earliest time is to be drawn from the urn at that time, and then the various necessary replacements of colored balls in the urn occur instantaneously while all the stopwatches are reset. Furthermore, the stopwatches begin anew with no memory of what has occurred previously. This idea is appropriately termed "memorylessness" and is unique to the exponential distribution within the realm of continuous distributions. The fact that the exponential distribution is memoryless is why it has been selected to denote the time passing between draws in our continuous urn setting, since we want to approximate the discrete setting as

much as possible and the discrete setting is memoryless as well.

This continuous urn setting that we have described is called a Pólya Process and is denoted by very similar notation to what we have already encountered. Let the state of the urn at time t be represented by $R(t) := \binom{W(t)}{B(t)}$, where $W(t)$ denotes the number of white balls in the urn at time t and $B(t)$ denotes the number of blue balls in the urn at time t. As a further piece of terminology, this whole idea of embedding a discrete system into a continuous time setting with events separated in time by an exponential random variable is called "poissonization". The reason for this name is that although the amount of time that passes between draws is determined by an exponential random variable, the amount of draws that have occurred up to a certain point in time is determined by a Poisson random variable.

4.1 Characterizing Urns in a Continuous Setting

Now that we have described the Pólya Process, let us explain why it is beneficial in characterizing urn schemes. The merits of the continuous Pólya Process lie in moment generating functions. Thus, we shall take a brief detour to explain what moment generating functions are and why they are useful.

4.1.1 Moment Generating Functions

In terms of concept, a moment generating function is simply an alternative way to define a probability distribution as opposed to the more familiar ways of probability density functions and cumulative distribution functions. In terms of mathematical specifity, a moment generating function for a random variable X is

$$\mathbf{E}[e^{uX}] = 1 + u\mathbf{E}[X] + \frac{u^2\mathbf{E}[X^2]}{2!} + \frac{u^3\mathbf{E}[X^3]}{3!} + \cdots .$$

Moment generating functions are so named because they generate the moments of a random variable (the k^{th} moment of a random variable X is defined as $\mathbf{E}[X^k]$). The j^{th} moment of X can be obtained from the moment generating function by taking the j^{th} derivate with respect to u (a dummy variable) and then setting $u = 0$.

Moment generating functions can often be represented in a nice direct, finite form, but many probability distributions only have indirect representations of moment generating functions and some distributions have no way to characterize their moment generating functions at all. As an example of a simple and clean moment generating function, for a binomial random variable X we have $\mathbf{E}[e^{uX}] = (1 - p + pe^u)^n$. To obtain the first moment we take the first derivative with respect to u and then set $u = 0$, as follows:

$$\begin{aligned}\mathbf{E}[X] &= \frac{d}{du}(1 - p + pe^u)^n|^{u=0} \\ &= npe^u(1 - p + pe^u)^{n-1}|^{u=0} \\ &= np.\end{aligned}$$

4.1.2 Obtaining Moments of the Pólya Process

By definition, the moment generating function of the Pólya Process is defined by $\mathbf{E}[e^{uW(t)+vB(t)}]$, and for convenience we can represent this function by $\phi(t, u, v)$, where t denotes time and u, v are the respective dummy variables for $W(t), B(t)$. Notice that we have expanded the definition of a moment generating function to allot for more than one random variable, but this generalizaton is perfectly sound.

Unfortunately, we cannot obtain a simple finite representation of ϕ (as we could for the binomial distribution). However, we can derive an equation containing ϕ that still allows us to obtain the moments of the Pólya Process by deriving the whole equation with respect to u and v and then setting $u, v = 0$. The equation is as follows:

$$\frac{\partial\phi}{\partial t} + (1 - e^{au+bv})\frac{\partial\phi}{\partial u} + (1 - e^{cu+dv})\frac{\partial\phi}{\partial v} = 0.$$

The proof for deriving this equation is involved, but let it suffice to say that we condition $\phi(t + \Delta t, u, v)$ over all possible urn states $R(t)$ and then take the limit as $\Delta t \to 0$. Furthermore, the fact that we are able to derive this equation at all is due to specific characteristics of the Poisson distribution, which is precisely why we have undertaken poissonization in the first place (for additional details, see Mahmoud, pp. 72-73). Recall that $a, b, c,$ and d are the entries in the matrix A for a general two-color replacement scheme.

If we take the first derivative of the above equation with respect to u and v and then set $u, v = 0$ we obtain the expectation of the state of the Pólya Process at time t:

Theorem 1. *Let A be the schema of a two-color Pólya Process of white and blue balls. At time t, the average number of white and blue balls in the process is*

$$\begin{pmatrix} \mathbf{E}[W(t)] \\ \mathbf{E}[B(t)] \end{pmatrix} = e^{A^T t} \begin{pmatrix} W(0) \\ B(0) \end{pmatrix}.$$

Notice how the general wording of Theorem 1 means that we are now able to derive the expected state of *any* two-color urn scheme at any point in time.

By way of example, let us calculate the expectated state of a binary search tree at time t. Recall that $A = \begin{pmatrix} 0 & 1 \\ 2 & -1 \end{pmatrix}$, and we have

$$\begin{aligned}
\begin{pmatrix} \mathbf{E}[W(t)] \\ \mathbf{E}[B(t)] \end{pmatrix} &= e^{A^T t} \begin{pmatrix} W(0) \\ B(0) \end{pmatrix} \\
&= \begin{pmatrix} \frac{e^{-2t}}{3} + \frac{2e^t}{3} & -\frac{2}{3}e^{-2t} + \frac{2e^t}{3} \\ -\frac{1}{3}e^{-2t} + \frac{e^t}{3} & \frac{2e^{-2t}}{3} + \frac{e^t}{3} \end{pmatrix} \begin{pmatrix} W(0) \\ B(0) \end{pmatrix} \\
&= \begin{pmatrix} -\frac{2}{3}\mathrm{B}(0)e^{-2t} + \frac{2\mathrm{B}(0)e^t}{3} + \frac{1}{3}e^{-2t}\mathrm{W}(0) + \frac{2e^t\mathrm{W}(0)}{3} \\ \frac{2}{3}\mathrm{B}(0)e^{-2t} + \frac{\mathrm{B}(0)e^t}{3} - \frac{1}{3}e^{-2t}\mathrm{W}(0) + \frac{e^t\mathrm{W}(0)}{3} \end{pmatrix} \\
&\sim \begin{pmatrix} \frac{2}{3} \\ \frac{1}{3} \end{pmatrix} e^t \left(B(0) + W(0)\right)
\end{aligned}$$

as $t \to \infty$. Thus, in the long run we can expect there to be twice as many white balls in our urn as blue balls. Thinking in terms of binary search trees, this means that in the long run we can expect there to be just as many nodes with no children as there are nodes with one child.

5 Depoissonization

Though we have obtained some nice results from poissonization, they are in terms of t, a continuous variable, when often we are interested in processes

that are really evolving in discrete steps. Returning to our BST, we don't in general expect insertions into the tree to follow a Pólya process. We want to take our continuous results and use them to approximate results in our original discrete setting.

To begin this, define the random variable t_n as the time at which the nth draw occurs. If we can work out how t_n behaves, we can characterize $R(t_n)$, the balls in the urn after the nth draw has occured.

This approximation relies on the observation that t_n is sharply concentrated around its mean, that is $\frac{t_n}{\mathbf{E}[t_n]} \xrightarrow{P} 1$. This is not always possible to show exactly, but in some simple cases it can be proven (Mahmoud, example 5.1, p. 89).

We return to the Pólya-Eggenberger process. Recall that our schema $A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. Let $\begin{pmatrix} W(0) \\ B(0) \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$.

Consider t_1, the time of the first draw. t_1 will be distributed the minimum of two independent exponentially distributed random variables with parameter 1, because we begin with two clocks. So $t_1 \stackrel{\mathcal{D}}{=} \text{Exp}\left(\frac{1}{2}\right)$. t_2 will be the sum of t_1 and the minimum of the clocks assigned to the three balls now in the urn, so $t_2 \stackrel{\mathcal{D}}{=} t_1 + \text{Exp}\left(\frac{1}{3}\right) = \text{Exp}\left(\frac{1}{2}\right) + \text{Exp}\left(\frac{1}{3}\right)$ and in general,

$$t_n \stackrel{\mathcal{D}}{=} \text{Exp}\left(\frac{1}{2}\right) + \text{Exp}\left(\frac{1}{3}\right) + \cdots + \text{Exp}\left(\frac{1}{n+1}\right)$$

a sum of independent random variables. We take the expectation and variance:

$$\begin{aligned}
\mathbf{E}[t_n] &= \mathbf{E}\left[\mathrm{Exp}\left(\frac{1}{2}\right)\right] + \mathbf{E}\left[\mathrm{Exp}\left(\frac{1}{3}\right)\right] + \cdots + \mathbf{E}\left[\mathrm{Exp}\left(\frac{1}{n+1}\right)\right] \\
&= \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n+1} \\
&\sim \ln n \\
\mathbf{Var}[t_n] &= \mathbf{Var}\left[\mathrm{Exp}\left(\frac{1}{2}\right)\right] + \mathbf{Var}\left[\mathrm{Exp}\left(\frac{1}{3}\right)\right] + \cdots + \mathbf{Var}\left[\mathrm{Exp}\left(\frac{1}{n+1}\right)\right] \\
&= \frac{1}{2^2} + \frac{1}{3^2} + \cdots + \frac{1}{(n+1)^2} \\
&\sim \frac{\pi^2}{6}
\end{aligned}$$

The expectation grows while the variance converges, so t_n is indeed sharply concentrated around its average: the growth in expectation overwhelms the variance, so we expect t_n to behave like its expectation for large n.

5.1 Depoissonization of the Two-Color Pólya Process

We would like to derive a depoissonization result for a general tenable two-color Pólya process (Mahmoud, pp. 91-93). We begin by establishing a connection between the number of drawings of white and blue balls by time t, and the number of balls of each color in the urn after time t. We set $\tilde{W}(t)$ to be the number of white ball drawings by time t , $\tilde{B}(t)$ the number of blue drawings and $\tilde{\mathbf{R}}(t) = \left(\tilde{W}(t), \tilde{B}(t)\right)^T$. Then since each white drawing adds a white balls, while each blue drawing adds c white balls we can write the following relation

$$W(t) = W(0) + a\tilde{W}(t) + c\tilde{B}(t).$$

Similarly for blue,

$$B(t) = B(0) + b\tilde{W}(t) + d\tilde{B}(t).$$

Which gives the matrix equation

$$\mathbf{R}(t) = \mathbf{B}\tilde{\mathbf{R}}(t) + \mathbf{R}(0)$$

where $B = A^T$, and on average

$$\mathbf{E}\left[\mathbf{R}\left(t\right)\right] = \mathbf{B}\mathbf{E}\left[\tilde{\mathbf{R}}\left(t\right)\right] + \mathbf{R}\left(0\right).$$

So at a random time t_n,

$$\mathbf{E}\left[\mathbf{R}\left(t_n\right)|t_n\right] = \mathbf{B}\mathbf{E}\left[\tilde{\mathbf{R}}\left(t_n\right)|t_n\right] + \mathbf{R}\left(0\right),$$

the expectation of which is

$$\mathbf{E}\left[\mathbf{R}\left(t_n\right)\right] = \mathbf{B}\mathbf{E}\left[\tilde{\mathbf{R}}\left(t_n\right)\right] + \mathbf{R}\left(0\right).$$

Finally, if we introduce the assumptions that $\mathbf{B}$ must be invertable, we can rearrange the expression to obtain

$$\mathbf{E}\left[\tilde{\mathbf{R}}\left(t_n\right)\right] = \mathbf{B}^{-1}\left(\mathbf{E}\left[\mathbf{R}\left(t_n\right)\right] - \mathbf{R}\left(0\right)\right). \tag{1}$$

At this point we look ahead to the goal of our calculations. We want to find an average measure of time $\bar{t}_n$ so that

$$\mathbf{E}\left[\mathbf{R}\left(t_n\right)\right] \approx \mathbf{E}\left[\mathbf{R}\left(\bar{t}_n\right)\right] = e^{\mathbf{A}^T\bar{t}_n}\mathbf{R}\left(0\right) \tag{2}$$

In general for more than one color it is not possible to find a $\bar{t}_n$ that will give an exact depoissonization, because Eq. (1) is a vectorial relation. Thus each component of the vector might need a different average measure. However it can be shown, in a similar way to the Pólya-Eggenberger case, that they are all of the order ln n, which allows us to find the desired approximation.

Continuing with the derivation, notice that on any stochastic path, by time t_n we have had n drawings. We introduce the vector $\mathbf{J} = (1\ 1)$, which when multiplied on the right by a column vector will add up the two components of the vector. Thus,

$$n = \mathbf{E}\left[\mathbf{J}\tilde{\mathbf{R}}\left(t_n\right)\right].$$

By linearity we have,

$$n = \mathbf{J}\mathbf{E}\left[\tilde{\mathbf{R}}\left(t_n\right)\right].$$

Then substituting our result from (1), we have

$$n = \mathbf{J}\mathbf{B}^{-1}\left(\mathbf{E}\left[\mathbf{R}\left(t_n\right)\right] - \mathbf{R}\left(0\right)\right),$$

and replacing $\mathbf{E}[\mathbf{R}(t_n)]$, with the approximation in (2) gives

$$n \approx \mathbf{J}\mathbf{B}^{-1}e^{\mathbf{B}\bar{t}_n}\mathbf{R}(0) + o(n).$$

We can decompose the matrix exponential $e^{\mathbf{B}\bar{t}_n}$ in terms of eigenvalues and idempotent matrices. We consider the case with two real, distinct eigenvalues $\lambda_1 > \lambda_2$. For any real number x

$$e^{\mathbf{B}x} = e^{\lambda_1 x}\varepsilon_1 + e^{\lambda_2 x}\varepsilon_2$$

Then for large n

$$\begin{aligned} n &\approx \mathbf{J}\mathbf{B}^{-1}\left(e^{\lambda_1\bar{t}_n}\varepsilon_1 + e^{\lambda_2\bar{t}_n}\varepsilon_2\right)\mathbf{R}(0) + o(n) \\ &\approx \mathbf{J}\ \mathbf{B}^{-1}\varepsilon_1\mathbf{R}(0)\, e^{\lambda_1\bar{t}_n}. \end{aligned}$$

It follows that

$$e^{\mathbf{B}\bar{t}_n} \approx e^{\lambda_1\bar{t}_n}\varepsilon_1 = \frac{n}{\mathbf{J}\mathbf{B}^{-1}\varepsilon_1\mathbf{R}(0)}\varepsilon_1,$$

giving the approximation,

$$\mathbf{E}[\mathbf{R}_n] \approx e^{\mathbf{B}\bar{t}_n}\mathbf{R}(0) = \frac{\varepsilon_1\mathbf{R}(0)\, n}{\mathbf{J}\ \mathbf{B}^{-1}\varepsilon_1\mathbf{R}(0)}.$$

For the binary search tree example, this formula yields the following approximation, which agrees with our previous results

$$\begin{pmatrix} \mathbf{E}[W_n] \\ \mathbf{E}[B_n] \end{pmatrix} \approx \begin{pmatrix} \frac{2}{3} \\ \frac{1}{3} \end{pmatrix} n.$$

6 Conclusion

Urn models, in particular Pólya urns, are a flexible and powerful method for analyzing a variety of probability problems. We have only presented a brief introduction to some of the results and techniques in this broad field. There are many interesting extensions to the problems we've covered in this paper, such as urn schemes with random entries, models in which multiple balls are drawn at each step, applications to bioscience, and many others.

7 References

Mahmoud, Hosam M. (2009). Pólya Urn Models. Boca Raton, FL: CRC Press.