An Introduction to Bernoulli Factories

Renato Paes Leme
Google Research

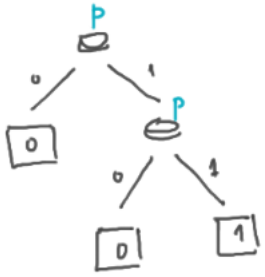
February 2022
Status: work in progress

Bernoulli factories are a tool from applied probability that is both entertaining as a puzzle and has applications in various fields like statistical simulation, mechanism design and quantum physics. Bernoulli refers to a Bernoulli random variable, of a p-coin:

$$X = \begin{cases} 1, & \text{with probability } p \\ 0, & \text{with probability } 1-p \end{cases}$$

We will assume we have access to this coin and can flip it as many times as we want, obtaining iid samples from the distribution, but the bias p is *unknown*. And factory refers to a procedure (or an algorithm) to transform a p-coin into an $f(p)$-coin. Before we give any formal definition, it is instructive to consider some examples:

Example 0.1 (Squaring)**.** To sample from a $f(p)$-coin with $f(p) = p^2$, we can simply sample the coin twice and output 1 is both coin tosses come up 1. Otherwise we output 0. We can represent with the following decision tree:



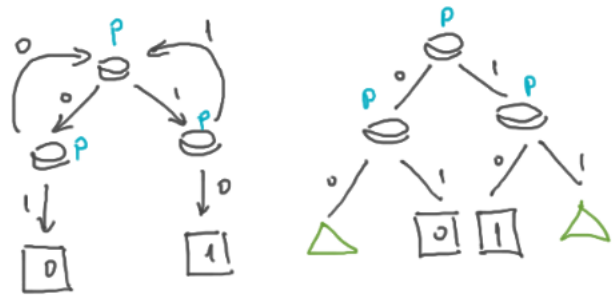
Similarly, we can also sample from $f(p) = p(1-p)$, $f(p) = (1-p)^2p^3$ and so on... This form an important class of functions we will refer as *Bernstein monomials*. A Bernstein monomial is a function of the type:

$$f(p) = p^a(1-p)^b$$

for non-negative integers a and b. To sample from it, we toss $a+b$ coins and output 1 if the first a coins come up 1 and the last b coins come up 0.

Example 0.2 (Von Neumann's Problem: unbiased coin from a biased one)**.** Von Neumann asked how to sample from an unbiased random variable having only access to a biased coin of unknown bias. In other words, how to construct a factory for the constant function $f(p) = 1/2$. The solution is as follows: flip the p-coin twice. If the coins come up 01 output 0, if they come up 10 output 1. Otherwise, retry.

We can represent this procedure either as a finite automaton (a finite state machines whose transitions are given by the outputs of the p-coin) or as an infinite tree, depicted in the right, where each triangle has a copy of the tree.



Example 0.3 (A coin of any bias)**.** Now consider the problem where we ask to produce a coin of known bias $c \in (0, 1)$ where c is a constant like $3/4$, $1/3$ or $1/\pi$.

A coin of bias $3/4$ is easy to sample. We already know how to sample a $1/2$-coin. So we can toss this coin twice and if it comes up 11 we output 0 otherwise we output 1. Using the same idea we can sample any rational number with denominator 2^k for some integer k.

Taking this idea a bit further, we don't need to stop at a finite k. Think of constant c as:

$$c = \sum_{k=1}^{\infty} \frac{b_k}{2^k} \quad \text{with} \quad b_k \in \{0, 1\}$$

Now consider the algorithm that starts at $k = 1$ and samples the $1/2$-coin. Let $X_k \in \{0, 1\}$ be the k-th draw of the coin. If $X_k < b_k$ we output 1. If $X_k > b_k$ we output 0. If $X_k = b_k$ we increase k and sample the $1/2$-coin again.

We note that all this algorithm is doing is sampling an uniform random variable u from $[0, 1]$ by sampling its binary digits one by one: $u = \sum_{k=1}^{\infty} X_k/2^k$. Then we output 1 if $u < c$ and output 0 if $u > c$. If $u = c$ the procedure never terminates, but since u is a continuous random variable, this happens with probability zero.

One thing to note is that this procedure requires in expectation only 2 samples from the $1/2$-coin (even though arbitrarily long sequences can happen with very small probability). Even though our thought process is that we are sampling an uniform variable, we only need to sample enough of its digits to decide whether $u < c$ or $u > c$.

Example 0.4 (Bernstein polynomials)**.** From the last two examples, we know that if we have a p-coin for any $p \in (0, 1)$ we can simulate any coin of known bias. From now on, we will assume that besides having access to a p-coin of *unknown* p, we can also flip any c-coin for a *known* $c \in (0, 1)$. We will call those *helper coins*.

Using helper coins, we can now sample any Bernstein polynomial with normalized non-negative coefficients, i.e.:

$$f(p) = \sum_{i=1}^{n} c_i p^{a_i} (1-p)^{b_i}$$

for non-negative integers $a_i, b_i \in \mathbb{Z}_+$ and non-negative reals $c_i \in \mathbb{R}_+$ such that $\sum_i c_i \leq 1$. For convenience define $c_0 = 1 - \sum_{i=1}^{n} c_i$. Now consider the following procedure.

Using the helper coins, we sample an index $i \in \{0, 1, \ldots, n\}$ with probability c_i. To do that we sample a c_0-coin. If it comes up 1 we choose index $i = 0$. If not, we sample a $c_1/(1-c_0)$-coin. If comes up 1, we choose $i = 1$. If not, we sample a $c_2/(1-c_0-c_1)$-coin and so on. Notice that if we reach the last index without choosing anyone, we will sample the last coin with probability $c_n/(1-c_0-\ldots-c_{n-1}) = c_n/c_n = 1$. So we will always choose an index this way. It is easy to see that each index is chosen with the correct probability.

After we sample an index i, we choose the output according to the Bernstein monomial $p^{a_i}(1-p)^{b_i}$ if $i > 0$. If $i = 0$, we simply output 0. With that, the probability of sampling 1 is exactly $f(p)$.

Example 0.5 (Series and Moment Generating Functions)**.** An interesting puzzle is how to sample from $f(p) = p/(2-p)$. The answer becomes immediate once we write it as a Taylor series around zero:

$$f(p) = \sum_{i=1}^{\infty} \frac{1}{2^i} p^i$$

The idea is exactly the same as in the previous example except that n is not infinity. We sample an index $i \in \{1, 2, \ldots\}$ with probability $1/2^i$ and the flip the p-coin i times. If all come up 1 we output 1. Otherwise we output zero.

This is a special case of a moment generating function, i.e, a function that can be written as: $f(p) = \mathbb{E}[p^X]$ for some random variable X taking values in $\mathbb{Z}_+$. For example, $f(p) = p/(2-p)$ is the moment generating function of a geometric random variable with parameter $1/2$.

Another case of interest is $f(p) = \exp(p-1)$ which is the moment generating function of a Poisson random variable with parameter 1. Any function of that type admits a factory by sampling X using the helper coins and then flipping the p-coin X times.

Why exact sampling? An important aspect of Bernoulli factories is that it asks for *exact* sampling. The original motivation is to do exact simulation of stochastic processes. In those, small sampling errors quickly compound, sometimes exponentially – hence the need for exact simulation. The same situation happens in Bayesian inference, where sampling is a sub-routine in an iterative procedure. Finally, in Mechanism Design the fact that sampling is exact allows us to design black-box-reductions that are Bayesian-incentive compatible. Before the introduction of this machinery, the known reduction in the general case was ϵ-Bayesian-incentive-compatible, i.e. agents had still a small incentive to deviate from truth-telling.

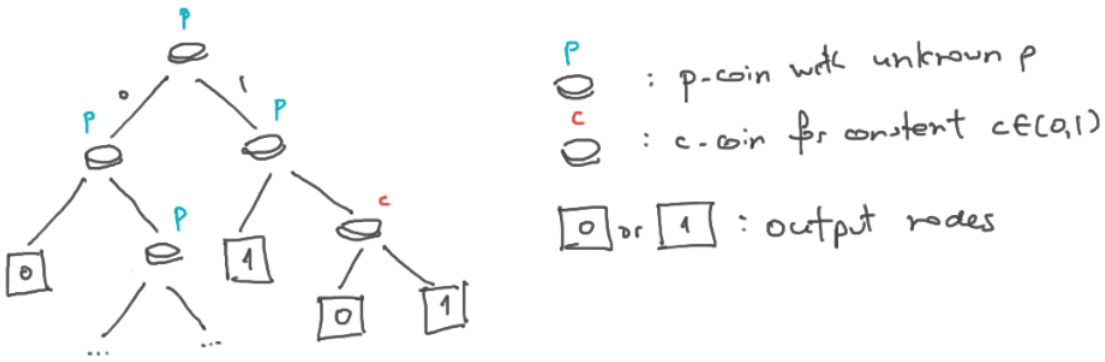
This discussion is to motivate why in certain situations *approximately* simulating an $f(p)$-coin is not enough. Approximately sampling can be easily done by the following method: let $X_1, \ldots, X_n$ be n draws from the p-coin and define its empirical average as:

$$\hat{p}_n = \frac{X_1 + \ldots + X_n}{n}$$

We know by the Chernoff bound hat for $n = O(\epsilon^{-2}\log(1/\delta))$ we have $\mathbb{P}[|p - \hat{p}_n| > \epsilon] < \delta$. Hence if f is continuous, estimating p by $\hat{p}_n$ and using the helper coins to sample from a $f(\hat{p}_n)$-coin produces a reasonable approximation of the $f(p)$-coin.

1 Necessary and sufficient conditions (Keane and O'Brien 1994)

We are now ready to give a formal definition of a single-parameter Bernoulli factory. Single-parameter refers to the fact that we have only one coin with unknown bias. We will define a Bernoulli factory as a (potentially infinite) decision tree. Each internal node corresponds to either flipping the p-coin of unknown bias or flipping a helper c-coin of known bias. Those have two outgoing edges labelled 0 and 1. The leaves of the trees correspond to output nodes labelled with 0 and 1.



The decision tree induces a random variable $\mathcal{F}$ corresponding to the output we obtain from the decision tree. The variable $\mathcal{F}$ is implicitly parametrized by $p \in (0,1)$ since the output distribution will depend on which p-coin is used. For all random variables related to the factory $\mathcal{F}$ we will use $\mathbb{P}_p[\cdot]$ and $\mathbb{E}_p[\cdot]$ to denote the probability and expectation when a p-coin is used to execute the tree.

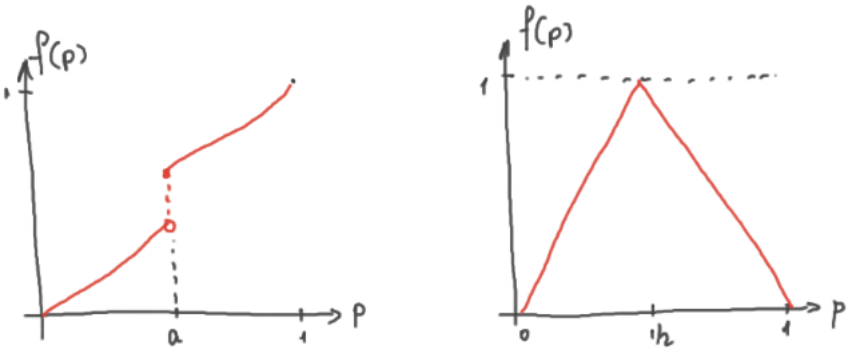
The variable $\mathcal{F}$ can take values 0, 1 or $\emptyset$. The first two correspond to reaching an output node of that value and we use $\emptyset$ to denote the event that it never reaches an output node.

A decision tree is valid if it terminates almost surely, i.e. $\mathbb{P}_p[\mathcal{F} = \emptyset] = 0$. Given a function $f : S \subseteq (0,1) \to (0,1)$ we say that the decision tree is a Bernoulli factory for f whenever

$$\mathbb{P}_p[\mathcal{F} = 1] = f(p), \forall p \in S$$

1.1 Necessary Conditions

The question we will try to answer next is what are the functions f that admit a Bernoulli factory. We saw already many examples for which it is possible to construct a factory. It is also instructive to consider two examples for which this is impossible:



Example 1.1 (Discontinuous function)**.** In the first example we have a discontinuous function f. The intuitive reason why we can't implement f is that the factory has only access to samples $X_1, X_2, \ldots$ from the p-coin. For sufficiently close points $a - \delta$ and a the distribution of the samples is practically indistinguishable if we only look at finitely many samples. Hence it is impossible to output with different probabilities at those point as $\delta \to 0$. The next lemma formalizes that intuition.

Lemma 1.2. *If $f : S \subseteq (0,1) \to [0,1]$ admits a Bernoulli factory, then f is continuous in S.*

Proof. Fix a point $a \in S$. We want to show that for every $\epsilon > 0$, there is δ such that if $|p - a| < \delta$ then $|f(p) - f(a)| < \epsilon$.

To show that, let N be a random variable showing the number of coins flipped before the output if the decision tree is executed using an a-coin (this is equal to the depth of the output node reached in the tree). Now, fix n such that $\mathbb{P}_a[N > n] < \epsilon/4$. Now, for each tuple $x = (x_1, \ldots, x_n) \in \{0,1\}^n$ we define function $F(x) \in \{0, 1, \emptyset\}$ indicating whether the decision tree outputs 0, 1 or doesn't yet terminate after seeing inputs $x_1, \ldots, x_n$. Also, let $X = (X_1, \ldots, X_n)$ be the random output of the coins. With that, we can re-write $\mathbb{P}_a[N > n] < \epsilon/4$ as:

$$\sum_{x \in \{0,1\}^n; F(x) = \emptyset} \mathbb{P}_a[X = x] \leq \frac{\epsilon}{4} \tag{1}$$

Now, choose δ small enough such that the total variation distance between the sequences $X = (X_1, \ldots, X_n)$ generated under p and a is at most $\epsilon/3$ for any $|p - a| < \delta$. More formally:

$$\sum_{x \in \{0,1\}^n} |\mathbb{P}_a[X = x] - \mathbb{P}_p[X = x]| < \frac{\epsilon}{4}, \forall p \in (a - \delta, a + \delta) \tag{2}$$

Now, we can bound $f(a)$ and $f(p)$ for $|p - a| < \delta$ as follows:

$$\left| f(a) - \sum_{x \in \{0,1\}^n; F(x) \in \{0,1\}} F(x) \mathbb{P}_a[X = x] \right| \leq \sum_{x \in \{0,1\}^n; F(x) = \emptyset} \mathbb{P}_a[X = x] < \frac{\epsilon}{4}$$

and similarly:

$$\left| f(p) - \sum_{x \in \{0,1\}^n; F(x) \in \{0,1\}} F(x) \mathbb{P}_p[X = x] \right| \leq \sum_{x \in \{0,1\}^n; F(x) = \emptyset} \mathbb{P}_p[X = x] < \frac{\epsilon}{2}$$

where the last bound follows from combining equations (1) and (2). Now, taking it all together, we have:

$$|f(a) - f(p)| \leq \left| \sum_{x \in \{0,1\}^n; F(x) \in \{0,1\}} F(x)(\mathbb{P}_a[X = x] - \mathbb{P}_p[X = x]) \right| + \frac{3\epsilon}{4} < \frac{\epsilon}{4} + \frac{3\epsilon}{4} = \epsilon$$

□

Example 1.3. The second example in the figure of a function for which we can't construct a factory is the function $f(p) = 1 - |1 - 2p|$. The important fact is that $f(\frac{1}{2}) = 1$ but $f(p') < 1$ for some other p' (e.g. $p' = \frac{1}{3}$). Since the factory can output zero when executed with a (1/3)-coin, there must be a node outputting zero in the tree that is reached given a finite sequence of coin flips: $(x_1, \ldots, x_n) \in \{0,1\}^n$. But the same sequence happens with positive probability when we execute the factory with a (1/2)-coin. Hence when $p = 1/2$ the factory must output 0 with positive probability, contradicting the fact that $f(\frac{1}{2}) = 1$.

In particular this means that if f is implementable by a factory and $f(p) \in \{0,1\}$ for some $p \in (0,1)$, then f must be the constant function. The next lemma formalized this discussion.

Lemma 1.4. *If $f : S \subseteq (0,1) \to [0,1]$ is implementable by a Bernoulli factory and is not constant, then it must be* polynomially bounded*, i.e., exists $n \in \mathbb{Z}_+$ such that:*

$$\min(f(p), 1 - f(p)) \geq \min(p, 1 - p)^n$$

Proof. If f is not constant, then a factory for f needs to contain at least one node outputting 0 and one node outputting 1. Let n_1 be the depth of some node outputting 1. In the path to that node, we take a_1 1-edges and b_1 0-edges with $a_1 + b_1 = n_1$. So:

$$f(p) \geq p^{a_1}(1 - p)^{b_1} \geq \min(p, 1 - p)^{n_1}$$

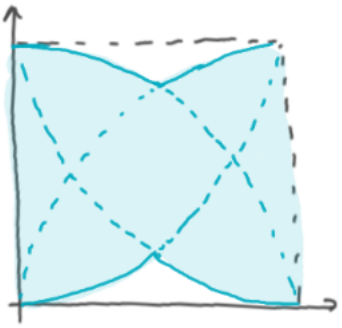
If we define n_0, a_0 and b_0 similarly for some node outputting 0 we have:

$$1 - f(p) \geq p^{a_0}(1 - p)^{b_0} \geq \min(p, 1 - p)^{n_0}$$

Hence the statement holds for $n = \max(n_0, n_1)$. □

1.2 Sufficient Conditions

Those two conditions (continuity and polynomially-boundedness) turn out to be (essentially) sufficient as well. In other words, any continuous function we can draw that is between $\min(p, 1 - p)^n$ and $1 - \min(p, 1 - p)^n$ for some n (shaded region below) can be implemented.



We will add the following technical definition: consider a continuous function $f : S \subseteq (0,1) \to [0,1]$ and let $\bar{S}$ be the (topological) closure of S. We say that f is an *extensible continuous* function if there is a function $\tilde{f} : \bar{S} \to [0,1]$ such that $f(x) = \tilde{f}(x)$ for all $x \in S$. Note that the function in the previous open problem in continuous on $(0,1)$ but not extensible continuous. Another example is a function $f : (0,1/2) \cup (1/2,1) \to [0,1]$ that is piecewise constant in $(0,1/2)$ and $(1/2,1)$. This function is again continuous in its domain, but can't be extended continuously to the closure $[0,1]$.

An important property for us that is that extensible continuous function have the *no-zigzag* property: given any $a < b$ and any polynomial $g(x)$ there are at most finitely pairs (x_i, y_i) with $x_1 < y_1 < x_2 < y_2 < x_3 < y_3 < \dots$ such that $h(x_i) = a$ and $h(y_i) = b$ for function $h(x) = f(x) + g(x)$.

Theorem 1.5. Let $f : S \subseteq (0,1) \to (0,1)$ that is an extensible continuous and polynomially bounded function. Then there exists a Bernoulli factory implementing f.

A complete proof will be added in the next version. The following example is a nice application of the previous theorem:

Example 1.6 (Sum of two coins)**.** Given two coins with biases p and q where p and q are unknown but we are promised that $p + q < 1 - \epsilon$ for some small but positive constant ϵ (e.g. $\epsilon = 0.0001$), design a factory for a coin of bias $p + q$.

The solution is as follows: first we design a factory that samples from a coin of bias $\frac{1}{2}(p+q)$. This is easy to do: with probability half, sample from the p-coin and output its result. With half probability, do the same for the q-coin.

Now, consider a factory for the function $f(x) = \min(2x, 1-\epsilon)$. Notice that f is continuous, Lipschitz (and hence no-zigzag) and polynomially-bounded, so a factory exists. Using this factory on the $\frac{1}{2}(p+q)$-coin, we obtain a sample from the $(p+q)$-coin.

1.3 Fast Simulation

Theorem 1.5 guarantees that essentially any function we can hope to implement is implementable. However, it says nothing about how efficiently this can be done. By efficiency we mean: how many times do we need to flip the p-coin until we can decide the output of the $f(p)$-coin.

To make this more precise, fix a Bernoulli factory and define N to be the random variable denoting how many coins are flipped until we output. In other words, if we execute the factory, what is the depth of the output node we reach.

Example 1.7. To give one example, consider the factory for $f(p) = 1/2$ in Example 0.2. With probability $2p(1-p)$ we output after the first 2 coin flips. With remaining probability $(p^2 + (1-p^2))$ we retry. This means that:

$$\mathbb{P}_p[N = 2k] = [p^2 + (1-p^2)]^{k-1} 2p(1-p) \text{ for } k = 1, 2, ...$$

In particular, N has finite expectation for each $p \in (0,1)$:

$$\mathbb{E}_p[N] = \frac{p^2 + (1-p^2)}{p(1-p)} < \infty \text{ for } p \in (0,1)$$

The only guarantee given by the construction in Theorem 1.5 is that $N < \infty$ almost surely. We don't even know if $\mathbb{E}_p[N] < \infty$ for example. In the next lemma (due to Nacu and Peres) we will

show that the expected number of coin flips is $\mathbb{E}_p[N]$ is tied to the derivatives of f. In particular, a function that is continuous but not Lipschitz continuoius must be implementable with $\mathbb{E}_p[N] = \infty$ at certain points.

Lemma 1.8. *If a function f admits a Bernoulli factory on a closed interval $I \subseteq (\epsilon, 1-\epsilon)$ and $\sup_{p\in I} \mathbb{E}_p[N] = C < \infty$ then function f must be (C/ϵ)-Lipschitz.*

Proof. Define $f_n(p) = \mathbb{P}_p(\mathcal{F} = 1 \text{ and } N = n)$, i.e., the probability that the factory outputs 1 after exactly n coin flips of a p-coin. Observe that $f(p) = \sum_{n=0}^{\infty} f_n(p)$ and that $f_n(p)$ can be obtained by summing over all nodes outputting 1 at level n of the decision tree the probability of reaching that node, which is $p^k(1-p)^{n-k}$ for some k. Hence, $f_n(p)$ is a Bernstein polynomial of the type:

$$f_n(p) = \sum_{k=0}^{n} a_{k,n} p^k (1-p)^{n-k}$$

Derivating it, we obtain:

$$f_n'(p) = \sum_{k=0}^{n} a_{k,n}[kp^{k-1}(1-p)^{n-k} - (n-k)p^k(1-p)^{n-k-1}]$$

Hence:

$$\begin{aligned}
|f_n'(p)| &\le \sum_{k=0}^{n} a_{k,n}[kp^{k-1}(1-p)^{n-k} + (n-k)p^k(1-p)^{n-k-1}] \\
&\le \sum_{k=0}^{n} a_{k,n}[kp^{k-1}(1-p)^{n-k}(p/\epsilon) + (n-k)p^k(1-p)^{n-k-1}((1-p)/\epsilon)] \\
&= n f_n(p)/\epsilon
\end{aligned}$$

Now, we are ready to bound the Lipschitz constant. Given two points $p, q \in I$ we have that:

$$|f(q) - f(p)| \le \sum_{n=0}^{\infty} |f_n(p) - f_n(q)| \le \sum_{n=0}^{\infty} \int_p^q |f_n'(x)| dx \le \sum_{n=0}^{\infty} n \int_p^q \frac{f_n(x)}{\epsilon} dx$$

In the last expression we have the term $\sum_{n=0}^{\infty} n f_n(p)$ which we can relate to N as follows:

$$\sum_{n=0}^{\infty} n f_n(p) = \sum_{n=0}^{\infty} n \mathbb{P}_p(\mathcal{F} = 1, N = n) \le \sum_{n=0}^{\infty} n \mathbb{P}_p(N = n) = \mathbb{E}_p[N] \le C$$

The two last expressions together imply that $|f(q) - f(p)| \le |p - q| \cdot C/\epsilon$ as desired. □

The lemma now allows us to exhibit a function that admits a Bernoulli factory but must necessarily have unbounded $\mathbb{E}_p[N]$. For example, consider:

$$f(p) = \begin{cases} \frac{1}{4}, & p \le \frac{1}{4} \\ \frac{1}{4} + \frac{1}{2}\sqrt{2x - \frac{1}{2}}, & \frac{1}{4} \le p \le \frac{3}{4} \\ \frac{3}{4}, & p \ge \frac{3}{4} \end{cases}$$

The function has essentially a copy of $\sqrt{x}$ in the interval $[1/4, 3/4]$ and hence it is not Lipschitz since its derivative blows up at $p = 1/4$ From the previous theorem, we can only guarantee that $\sup_{p\in[1/4-\epsilon,1/4+\epsilon]} \mathbb{E}_p[N] = \infty$ for any $\epsilon > 0$. Can we argue that the expectation is unbounded exactly at $p = 1/4$?

Open Problem 1.1. *In the previous example, is $\mathbb{E}_p[N] = \infty$ for $p = 1/4$?*

The function $f(x) = \sqrt{x}$ is very interesting. We know that we can construct a factory by Theorem 1.5. In fact, Mossel and Peres construct an explicit factory for it, but the expectation is not bounded. Nacu and Peres ask the following question:

Open Problem 1.2. *Is there a Bernoulli factory for $f(p) = \sqrt{p}$ defined for $p \in (0, 1)$ such that $\mathbb{E}_p[N] < \infty$ for all $p \in (0, 1)$?*

Above, we saw how to related the Lipschitzness of the f to the expected number of coin flips. Similarly, we can realated the analyticity of f to the properties of the tail of N. Nacu and Peres say that a Bernoulli factory is fast if N has an exponential tail, i.e., if for every p there is a constant $r(p) < 1$ such that:

$$\mathbb{P}_p(N \leq n) \leq O(r(p)^n)$$

They prove that a function $f(p)$ admits a fast Bernoulli factory iff it is real-analytic, i.e., if for every point p in the domain of f the Taylor series around p converges on a neighborhood of p.

A special case of real-analytic functions are rational functions, i.e. function of the type $f(p) = a(p)/b(p)$ for polynomials $a(p)$ and $b(p)$. Mossel and Peres show that those functions admit fast factories with a particularly nice structure, which we will discuss in the next section.

2 Multiple Coins and Dice

So far we considered with a single p-coin. We will now consider we have access to many p-coins or to a dice. A dice is a random variable X with values in $[n] := \{1, 2, \ldots, n\}$. It is represented by a vector p in the simplex $\Delta_n := \{p \in [0, 1]^n; \sum_{i=1}^n p_i = 1\}$ such that $\mathbb{P}[X = i] = p$.

2.1 Bernoulli Race

Our first observation is that having a p-dice is equivalent to having n coins with biases $p_1, \ldots, p_n$. One direction is obvious: if we have a dice we can simulate a p_i-coin by tossing the dice and outputting 1 if $X = i$ and outputting zero otherwise.

For the other direction, we will use a procedure known as the *Bernoulli Race*. Given coins with biases $p_1, \ldots, p_n$ we want to sample from a dice such that $\mathbb{P}(X = i) = p_i/(p_1 + \ldots + p_n)$. The procedure is as follows: choose a $i \in [n]$ uniformly at random and flip the p_i-coin. If it comes up 1, then output $X = i$. Otherwise, retry.

To see why $\mathbb{P}(X = i)$ is proportional to p_i, first compute the probability of retry. We have that $\mathbb{P}(\text{retry}) = 1 - \sum_{j=1}^n \frac{p_j}{n}$. Now, the probability that we output i is the probability we output it right away after retrying k times for $k = 0, 1, \ldots$:

$$\mathbb{P}(X = i) = \frac{p_i}{n} \sum_{k=0}^{\infty} \left(1 - \sum_{j=1}^n \frac{p_j}{n}\right)^k = \frac{p_i}{\sum_{j=1}^n p_j}$$

2.2 Pólya's Theorem

Our next goal is to design a factory for rational functions. The main ingredient is to understand how polynomials that are positive on the $(0,1)$ interval (and more generally on the opens simplex in higher dimensions) looks like. This is given by the following theorem by Pólya which is a type of *Positivstellensatz*.

Theorem 2.1 (Pólya)**.** Let $f(x_1,\ldots,x_n)$ be an homogeneous polynomial with real coefficients such that $f(x) > 0$ for all points in $\{x \in \mathbb{R}^n; x_i \geq 0, \forall i \text{ and } \sum_i x_i > 0\}$. Then there is an integer $m \geq 0$ such that all the coefficients of the polynomial $(\sum_i x_i)^m f(x)$ are non-negative.

Proof. It will be useful to introduce some notation. Let $\alpha = (a_1,\ldots,a_n)$ be a multi-index and $x^\alpha = x_1^{a_1}x_2^{a_2}\ldots x_n^{a_n}$. Also define $|\alpha| = a_1 + \ldots + a_n$ and $\alpha! = a_1!a_2!\ldots a_n!$. Given two multi-indices α and β we say $\alpha \geq \beta$ if the inequality holds componentwise. Otherwise we write $\alpha \ngeq \beta$. With this notation, we can write f of degree d as:

$$f(x) = \sum_{|\alpha|=d} f_\alpha \frac{x^\alpha}{\alpha!}$$

for some coefficients f_α. Now, define the following for a variable x_i and an index a_i:

$$P(x_i, a_i, t) = \frac{x_i(x_i - t)(x_i - 2t)\ldots(x_i - (a_i - 1)t)}{a_i!}$$

And with some abuse of notation, for the variable vector x and a multi-index α define:

$$P(x, \alpha, t) = \prod_{i=1}^{n} P(x_i, a_i, t)$$

And finally extend f as follows:

$$F(x,t) = \sum_{|\alpha|=d} f_\alpha P(x, \alpha, t)$$

such that $F(x,0) = f(x)$.

Compactness argument: Let $\Delta = \{x \in \mathbb{R}^n_{\geq 0}; \sum_i x_i = 1\}$ be the simplex and $\mu = \min_{x\in\Delta} f(x) > 0$. By the fact that f is strictly positive on the orthant and Δ is compact, its minimum must also be strictly positive. Since $F(x,t)$ is continuous, there is some ϵ for which:

$$\min_{x\in\Delta} F(x,t) \geq \frac{\mu}{2}, \quad \forall t \in [0,\epsilon]$$

Newton's binomial formula: Now, we can use Newton's binomial formula to write:

$$\left(\sum_i x_i\right)^m = m! \cdot \sum_{|\beta|=m} \frac{x^\beta}{\beta!}$$

Combining with f we have:

$$\left(\sum_i x_i\right)^m f(x) = m! \cdot \sum_{|\alpha|=d} \sum_{|\beta|=m} \frac{x^{\alpha+\beta}}{\alpha!\beta!} = m! \cdot \sum_{|\gamma|=m+d} \frac{x^\gamma}{\gamma!} \sum_{\alpha,\beta;\alpha+\beta=\gamma} f_\alpha \frac{\gamma!}{\alpha!\beta!}$$

Let's focus on the last term. For each fixed γ observe that:

$$\sum_{\alpha,\beta;\alpha+\beta=\gamma} f_\alpha \frac{\gamma!}{\alpha!\beta!} = \sum_{\alpha\leq\gamma} f_\alpha \frac{\gamma!}{\alpha!(\gamma-\alpha)!} = \sum_{\alpha\leq\gamma} f_\alpha \cdot P(\gamma,\alpha,1) = \sum_{\alpha} f_\alpha \cdot P(\gamma,\alpha,1) = F(\gamma,1)$$

where the second inequality follows from the fact that $P(x,\alpha,t)$ becomes a product of binomial coefficients when x is an integer and $t=1$. The third inequality follows from that fact that $P(\gamma,\alpha,1)=0$ whenever $\alpha \not\leq \gamma$ since it has at least one index for which $\alpha_i \geq \gamma_i + 1$ and hence one of the terms in the numerator of $P(\gamma_i,\alpha_i,1)$ will be zero.

If we put the last two display equations together and note that $F(x,t)$ is also d-homogeneous, we have:

$$\left(\sum_i x_i\right)^m f(x) = m! \cdot \sum_{|\gamma|=m+d} F(\gamma,1)\frac{x^\gamma}{\gamma!} = m!(m+d)^d \cdot \sum_{|\gamma|=m+d} F\left(\frac{\gamma}{m+d},\frac{1}{m+d}\right)\frac{x^\gamma}{\gamma!}$$

Since $|\gamma| = m+d$ we have that $\gamma/(m+d) \in \Delta$. Now, for large enough m we have that $1/(m+d) < \epsilon$ and so $F\left(\frac{\gamma}{m+d},\frac{1}{m+d}\right) > 0$ for all γ. Therefore, all coefficients in the right hand side are non-negative. □

The previous theorem requires positivity on the *closed* positive orthant. For our purposes, it will be convenient to work on the *open* positive orthant. Note that while the previous theorem has conditions $x_i \geq 0$ the lemma below asks for $x_i > 0$.

Corollary 2.2. *Let $f(x,y)$ be an homogeneous polynomial with real coefficients such that $f(x,y) > 0$ for all points in $\{(x,y) \in \mathbb{R}^2; x > 0, y > 0\}$. Then there is an integer $m \geq 0$ such that all coefficients of $(x+y)^m f(x)$ are non-negative.*

Proof. Define $g(p) = f(p,1-p)$. If $n = \text{degree}(f)$, then:

$$f(x,y) = (x+y)^n \cdot f\left(\frac{x}{x+y},\frac{y}{x+y}\right) = (x+y)^n \cdot g\left(\frac{x}{x+y}\right)$$

If $g(0)=0$ then $g(p)$ must be divisible by p. Similarly if $g(1)=0$ then $g(p)$ must be divisible by $1-p$. So we can re-write:

$$g(p) = p^a(1-p)^b h(p)$$

where for integers $a,b \geq 0$ and a polynomial $h(p)$ that is strictly positive on $[0,1]$. Note that $h(p)$ is strictly positive for $p \in (0,1)$ since $h(p) = g(a)/(p^a(1-p)^b)$. It is strictly positive for $p = 0,1$ since it is non-zero (since we removed p and $1-p$ factors) and it is continuous since it is a polynomial. Hence, we can write:

$$f(x,y) = x^a y^b H(x,y)$$

for some homogeneous polynomial $H(x,y)$ satisfying the conditions of Pólya's Theorem. □

Polya's Theorem (Theorem 2.1) is about homogeneous polynomials that are strictly positive in the closed positive orthant $\mathbb{R}^n_{\geq 0} \setminus 0$. For $n=2$ variables, we showed in Corollary 2.2 that it is enough to ask for it to be strictly positive on the *open* positive orthant $\mathbb{R}^2_{>0}$. For $n > 2$. however, the result breaks if we only ask for positivity in the open positive orthant as shown in the next example (which is based on a suggestion by Jon Schneider).

Example 2.3. We will exhibit a polynomial $g(x, y, z)$ such that $g(x, y, z) > 0$ whenever $(x, y, z) \in \mathbb{R}^3_{>0}$ but $(x + y + z)^m g(x, y, z)$ has negative coefficients for all non-negative integers m.

We start by defining an auxiliary polynomial:

$$f(x, y, z) = x^3 + y^3 + z^3 - 3xyz$$

By the AM-GM inequality, we have that $f(x, y, z) \geq 0$ for $(x, y, z) \in \mathbb{R}^3_{>0}$ with equality holding only when $x = y = z$. Now, we can defined:

$$g(x, y, z) = f(x, x + y, x + z)$$

Note that for any $(x, y, z) \in \mathbb{R}^3_{>0}$ we have $(x, x+y, x+z) \in \mathbb{R}^3_{>0}$ and we necessarily have $x \neq x+y$. Hence:

$$g(x, y, z) > 0, \forall (x, y, z) \in \mathbb{R}^3_{>0}$$

Now, we will show that for all integers $m \geq 0$ the polynomial $(x + y + z)^m g(x, y, z)$ has at least one negative coefficient. First, let's expand g.

$$g(x, y, z) = 3xy^2 + y^3 - 3xyz + 3xz^2 + z^3$$

The polynomial g is homogeneous of degree 3. Observe that the term $-3xyz$ is the only one where both x and y have degree at most 1. In every other term either y or z appears with a power at least 2. For a given m, the degree of $(x + y + z)^m g(x, y, z)$ is $m + 3$. The coefficient of the monomial $x^{m+1}yz$ in $(x + y + z)^m g(x, y, z)$ must be -3 since it can only be formed by multiplying x^m in the first part with $-3xyz$. Any other monomial in the second part will lead to a term with degree at least two for y or z.

2.3 Factories for Rational Functions

We are now ready to describe a factory for any rational function $f : (0, 1) \to (0, 1)$. A rational function can be written as:

$$f(p) = \frac{a(p)}{b(p)} \quad \text{for} \quad a(p) = \sum_{i=0}^{k} a_i p^i \quad \text{and} \quad b(p) = \sum_{i=0}^{k} b_i p^i$$

First observe that for the function f to be well defined on $(0, 1)$ the function $b(p) \neq 0$ for any $p \in (0, 1)$ and hence it can't change signs. Since we can always replace $a(p)$ and $b(p)$ by $\lambda a(p)$ and $\lambda b(p)$ for any constant $\lambda \neq 0$ we can assume that $b(p) > 0$ and $\sup_{p\in(0,1)} b(p) < 1$. Furthermore, since $0 < f(p) < 1$ we have:

$$0 < a(p) < b(p) < 1 \quad \text{for} \quad p \in (0, 1)$$

Therefore:

$$0 < b(p) - a(p) < 1 \quad \text{for} \quad p \in (0, 1)$$

Now, assume we can build a Bernoulli factory for $a(p)$ and one for $b(p) - a(p)$ we can simply use a Bernoulli Race between the $a(p)$-coin and the $(b(p) - a(p))$-coin and output 1 whenever the $a(p)$ coin is chosen. This will output 1 with probability: $a(p)/(a(p)+(b(p)-a(p))) = a(p)$ as desired.

All we need to do now is to design a factory for $a(p)$ and $b(p)$. This will be particularly nice factory since it will be also finite:

Lemma 2.4. *Let $a(p)$ be a polynomial such that $0 < a(p)$ for $p \in (0,1)$. Then there is a constant $\lambda > 0$ for which $\lambda a(p)$ admits a finite Bernoulli factory.*

Proof. Define: $A(x,y) = \sum_{i=0}^{k} a_i x^i (x+y)^{k-i}$ such that $a(p) = A(p, 1-p)$. Note that for $x, y > 0$ we have:

$$A(x,y) = (x+y)^k \cdot \sum_{i=0}^{k} a_i \left(\frac{x}{x+y}\right)^i = (x+y)^k \cdot a\left(\frac{x}{x+y}\right) > 0$$

So we can apply Corollary 2.2 to write:

$$(x+y)^m \cdot A(x,y) = \sum_i c_i x^{a_i} y^{b_i}$$

with $c_i \geq 0$, $m, a_i, b_i \in \mathbb{Z}_+$. Now, take λ such that $\lambda \sum_i c_i \leq 1$. Hence we can write:

$$\lambda a(p) = \lambda A(p, 1-p) = \sum_i \lambda c_i p^{a_i} (1-p)^{b_i}$$

which is a Bernstein polynomial (Example 0.4) and therefore admits a finite factory. □

Taking all those results together, we obtain that every rational function is implementable via a Bernoulli race between two Bernstein polynomials. In particular, this factory has an exponential tail.