An asymptotically optimal Bernoulli factory for certain functions that can be expressed as power series

Luis Mendo

Information Processing and Telecommunications Center, Universidad Politécnica de Madrid.
Avenida Complutense, 30. 28040 Madrid, Spain.
E-mail: `luis.mendo@upm.es`

Abstract

Given a sequence of independent Bernoulli variables with unknown parameter p, and a function f expressed as a power series with non-negative coefficients that sum to at most 1, an algorithm is presented that produces a Bernoulli variable with parameter $f(p)$. In particular, the algorithm can simulate $f(p) = p^a$, $a \in (0,1)$. For functions with a derivative growing at least as $f(p)/p$ for $p \to 0$, the average number of inputs required by the algorithm is asymptotically optimal among all simulations that are fast in the sense of Nacu and Peres. A non-randomized version of the algorithm is also given. Some extensions are discussed.

Keywords: Bernoulli factory, Simulation, Power series
2000 MSC: 65C50

1. Introduction

Let $\mathsf{X} = (X_i)_{i \in \mathbb{N}}$ denote a sequence of independent, identically distributed (i.i.d.) Bernoulli random variables X_i with unknown parameter p. A *non-randomized stopping rule* on X is a sequence of *stopping functions* $\tau_1(x_1)$, $\tau_2(x_1, x_2)$, ..., $\tau_i(x_1, \ldots, x_i)$, ... with $\tau_i : \{0,1\}^i \to \{0,1\}$. Given a realization $x_1, x_2, \ldots$ of X, the *stopping time N* is the smallest integer i for which $\tau_i(x_1, \ldots, x_i)$ equals 1, or infinity if this does not occur. N is assumed to be finite almost surely; that is, for almost all realizations of X at least one of the stopping functions takes the value 1.

A *randomized stopping rule* uses, in addition to X, a *randomizing sequence* $\mathsf{U} = (U_i)_{i \in \mathbb{N}}$ of independent random variables with uniform distribution on $(0,1)$. The stopping function τ_i depends on $x_1, u_1; x_2, u_2; \ldots; x_i, u_i$, where each u_j represents the value taken by the random variable U_j; and is assumed to be measurable. The stopping time N is defined as before, and is again assumed to be finite almost surely.

Another possible definition of a randomizing stopping rule would be to specify that at each i, given $X_1 = x_1, \ldots, X_i = x_i$, there is a certain probability of stopping that depends on $x_1, \ldots, x_i$. This corresponds to the above definition if the output of τ_i is obtained from comparing U_i with a threshold that depends on $x_1, \ldots, x_i$. Thus the definition based on the auxiliary sequence U captures all the randomness that can be

effected by a randomized stopping rule (and is more convenient for the purposes of this paper).

Let $f : S \to (0,1)$ denote a function defined on a set $S \subseteq (0,1)$, and let X be a sequence of independent Bernoulli variables with parameter $p \in S$. A *non-randomized Bernoulli factory* of f based on X is an algorithm that, using values from the sequence X as inputs, generates a Bernoulli variable Y with parameter $f(p)$. Specifically, the number N of required inputs is a stopping time on X dictated by a non-randomized stopping rule; and the output value Y depends on $X_1, \ldots, X_N$.

A *randomized Bernoulli factory* uses, in addition to the sequence X, an auxiliary sequence U of independent random variables uniformly distributed on $(0,1)$. Specifically, N is given by a randomized stopping rule on X with U as randomizing sequence. The output Y is also possibly randomized, that is, it depends on $X_1, \ldots, X_N$ and $U_1, \ldots, U_N$. Specifically, there exists a sequence of functions $\gamma_1, \gamma_2, \ldots$ such that for $N = n$ and for $X_1 = x_1$, $U_1 = u_1$; $\ldots$; $X_n = x_n$, $U_n = u_n$ the output is given as $\gamma_n(x_1, u_1; \ldots; x_n, u_n)$. These functions are assumed to be measurable.

The use of the term "randomized" in the above definitions of randomized stopping rules or Bernoulli factories refers to the fact that the stopping time N and the factory output Y are random even if conditioned on the input sequence X, due to the additional source of randomness represented by U. In the following, a Bernoulli factory will also be referred to as a *simulation*.

One of the earliest references of a Bernoulli factory, for the specific case $f(p) = 1/2$, appears in a work by von Neumann [1]. For general f, Keane and O'Brien [2] give a necessary and sufficient condition for a simulation of f to be possible, namely that the function either is constant, or is continuous and satisfies a certain polynomial bound. Nacu and Peres [3] define a non-randomized simulation to be *fast* if the distribution of N has an exponential tail, that is, for any $p \in S$ there exist values $A > 0$, $\beta < 1$ (which may depend on p) such that

$$\Pr[N > n] \leq A\beta^n. \tag{1}$$

The authors prove that a fast simulation exists for any f real analytic on any closed interval contained in $(0,1)$. In this paper, the definition will be extended to randomized simulations, which will be considered fast if they satisfy (1).

Several works on Bernoulli factories present simulation algorithms for specific functions. Considerable attention has been given to $f(p) = \min\{2p, 1 - 2\varepsilon\}$, $\varepsilon > 0$, which is an important building block for simulating other functions [3, 4]; and to linear functions $f(p) = cp$, $c > 1$ defined on a suitable set $S \subset (0, 1/c)$; see the work by Huber [5, 6].

A crucial parameter of a Bernoulli factory is $\mathrm{E}[N]$, that is, how many inputs X_i are required on average to generate a sample of Y. In applications, observing the variables X_i is usually costly, and thus $\mathrm{E}[N]$ should be as small as possible. For randomized Bernoulli factories, the auxiliary random variables U_i are assumed to be cost-free.

This paper deals mainly with functions of the form $f(p) = 1 - \sum_{k=1}^{\infty} c_k (1-p)^k$ where the coefficients c_k are non-negative and sum to 1. A randomized algorithm to simulate any such function is presented in Section 2. The algorithm is shown to be fast in the sense of (1), and its average number of inputs, $\mathrm{E}[N]$, is computed. The algorithm can be particularized to functions $f(p) = p^a$, $a \in (0,1)$. For $a = 1/2$ the algorithm is

similar to that given by Wästlund [7]; and the presented results affirmatively answer question 1 from [3], i.e. establish that $\sqrt{p}$ can be simulated with finite $E[N]$.

An interesting subclass of functions is formed by those that, in addition to having a power series expression as above, satisfy the following two conditions: $f(p)/p \to \infty$ as $p \to 0$, and the derivative $f'(p)$ asymptotically grows at least as fast as $f(p)/p$. In particular, this includes all functions that behave asymptotically like bp^a, $a \in (0,1)$, $b \in (0,\infty)$. For these functions, it will be seen that any fast simulation algorithm has an average number of inputs $E[N]$ that grows without bound as $p \to 0$. Therefore it is important to analyse the asymptotic rate of growth of $E[N]$. This analysis is presented in Section 3. The results show that the proposed algorithm is asymptotically optimal for the mentioned subclass of functions, in the sense that for any other fast algorithm $E[N]$ grows at least as fast with p.

A non-randomized version of the proposed algorithm is given in Section 4, and is also shown to be fast. Section 5 discusses some extensions of the algorithms to cover a broader range of functions. Section 6 presents conclusions and open problems. Section 7 contains proofs to all results.

The following notation is used throughout the paper. $x^{(m)}$ represents the rising factorial $x(x+1)\cdots(x+m-1)$ for $m \in \mathbb{N}$, and $x^{(0)} = 0$. For $m \in \mathbb{N}$, $m!!$ denotes the double factorial, that is, $m(m-2)(m-4)\cdots 2$ if m is even, or $m(m-2)(m-4)\cdots 1$ if m is odd. Given two positive functions f_1 and f_2, $f_1(x)$ is said to be $\Omega(f_2(x))$ for $x \to x_0$ if there are constants $C, \delta > 0$ such that $f_1(x) \geq Cf_2(x)$ for all x such that $|x - x_0| < \delta$.

2. Simulation algorithm. Average number of inputs

Consider an i.i.d. sequence X of random variables X_i that take the value 1 with probability p and 0 with probability $1 - p$. Let $f : (0,1) \to (0,1)$ be a function that can be expressed as a power series

$$f(p) = 1 - \sum_{k=1}^{\infty} c_k(1-p)^k \tag{2}$$

with

$$c_k \geq 0, \tag{3}$$

$$\sum_{k=1}^{\infty} c_k = 1. \tag{4}$$

Note that this implies that f is differentiable, and $\lim_{p\to 0} f(p) = 0$, $\lim_{p\to 1} f(p) = 1$.

The randomized algorithm to be presented yields a Bernoulli random variable Y with parameter $f(p)$. It takes as inputs a number of random variables from the sequence X, as well as from an auxiliary sequence U of i.i.d. random variables U_i uniformly distributed on $(0,1)$ and independent from the X_i variables. The algorithm makes use of coefficients d_k computed from c_k as follows:

$$d_k = \frac{c_k}{1 - \sum_{j=1}^{k-1} c_j}. \tag{5}$$

3

From (3) and (4) it stems that $0 \leq d_k \leq 1$. If the number of non-zero coefficients c_k is finite, i.e. if there exists K such that $c_K > 0$ and $c_k = 0$ for $k > K$, (5) gives $d_K = 1$; and for $k > K$ the coefficient d_k is not defined (and is not necessary, as will be seen).

Algorithm 1. *Let f be given by (2)–(4), and let d_k be defined by (5). The input of the algorithm is a sequence X of i.i.d. Bernoulli random variables. The output is a Bernoulli random variable Y.*

1. *Set $i = 1$.*
2. *Take one input X_i.*
3. *Produce U_i uniform on $(0,1)$. Let $V_i = 1$ if $U_i < d_i$ or $V_i = 0$ otherwise.*
4. *If V_i or X_i are 1, output $Y = X_i$ and finish. Else increase i and go back to step 2.*

The idea of this algorithm is similar to that presented by Wästlund [7] to simulate $f(p) = \sqrt{p}$, namely, decompose the event $Y = 0$ as an infinite sum of mutually exclusive events, each with probability $c_k(1-p)^k$. However, there are two differences here. First, the referenced paper treats the c_k and $(1-p)^k$ parts separately. Namely, an auxiliary random variable L is first generated with $\Pr[L = k] = c_k$, $k \in \mathbb{N}$. This variable represents the amount of inputs X_i that need to be taken. Then, $(1-p)^L$ is simulated using the product $\prod_{i=1}^{L}(1 - X_i)$. On the other hand, Algorithm 1 reduces the number of required inputs X_i by stopping as soon as one of the X_i variables is 1. This can be done because in that case the above product is 0 irrespective of the values of the remaining X_i variables.

The second difference is that Algorithm 1 uses auxiliary Bernoulli variables V_i with respective parameters d_i, instead of a random variable L with the distribution given by coefficients c_k. The latter approach, used in [7], was probably motivated by the fact that for $f(p) = \sqrt{p}$ the coefficients c_k in (2) are

$$c_k = \frac{1}{2^{2k-1}k}\binom{2k-2}{k-1} \tag{6}$$

and thus simulating L is particularly easy, as it lends itself to a simple probabilistic interpretation. Specifically, if a fair coin is flipped until the total number of tails exceeds the total number of heads, the probability that this happens after $2k-1$ steps is precisely (6). For other functions it is still possible to simulate L from fair coin flips, or from a uniform random variable, as long as conditions (3) and (4) are satisfied; but a simple probabilistic experiment analogous to that for $f(p) = \sqrt{p}$ may not exist. The Bernoulli random variables V_i provide an alternative, which can also be used for any coefficients c_k that satisfy the indicated conditions. Effectively, each d_k represents the conditional probability that $L = k$ given that $L \geq k$. The following proposition clarifies this.

Proposition 1. *The coefficients d_k defined by (5) satisfy*

$$c_k = d_k \prod_{j=1}^{k-1}(1 - d_j). \tag{7}$$

Additionally, this interpretation of d_k as a conditional probability makes it clear that if $d_K = 1$ for some K (which occurs if the series in (2) is finite with $c_K > 0$, $c_k = 0$ for $k > K$) it is unnecessary to define coefficients d_k for $k > K$.

The algorithm can also be phrased as a particular case of the reverse-time martingale approach of Łatuszyński, Kosmidis, Papaspiliopoulos and Roberts with random bounds [4, algorithm 3]. Specifically, from (2) it is possible to obtain monotone sequences of random upper bounds and lower bounds that depend on the observed inputs, such that the sufficient conditions for the referenced algorithm are satisfied. This approach has the additional advantage that condition (4) is not required. Note, however, that this restriction of Algorithm 1 is immaterial, because the coefficients can always be scaled to sum 1 and then the output Y can be multiplied by an independent Bernoulli variable with parameter equal to the desired sum (see Section 5).

Theorem 1. *Given a sequence* X *of i.i.d. Bernoulli random variables with parameter* $p \in (0, 1)$, *a function* f *of the form* (2)–(4), *and coefficients* d_k *computed from* (5), *the Bernoulli random variable* Y *produced by Algorithm 1 has* $\Pr[Y = 1] = f(p)$.

Algorithm 1 takes a new input for each iteration i. Thus the number of used inputs, N, coincides with the value of i when the algorithm finishes.

Theorem 2. *For* f *given by* (2)–(4) *and* $p \in (0, 1)$, *the average number of inputs required by Algorithm 1 is*

$$\mathrm{E}[N] = \frac{f(p)}{p}. \tag{8}$$

In addition, the algorithm is fast in the sense of (1).

It is interesting to consider the case $f(p) = p^a$, $a \in (0, 1)$. This can be expressed in the form (2) with

$$c_k = \frac{(1-a)^{(k-1)}a}{k!}, \tag{9}$$

from which $d_k = a/k$. Algorithm 1 can be applied, and $\mathrm{E}[N] = p^{a-1}$. In particular, $f(p) = \sqrt{p}$ can be simulated with $\mathrm{E}[N] = 1/\sqrt{p}$. This solves [3, question 1], which asks if there is an algorithm that simulates $\sqrt{p}$ on $(0, 1)$ for which the number of required inputs has finite expectation for all p.

3. Asymptotic optimality

A natural question is whether the average number of inputs required by Algorithm 1 can be improved by some other algorithm. The following proposition and theorem are useful steps towards the answer.

Proposition 2. *Given an open set* $S \subseteq (0, 1)$, *consider a function* $f : S \to (0, 1)$ *and a (possibly randomized) Bernoulli factory for* f *that is fast, as defined by* (1). *Then* $\mathrm{E}[N]$ *is finite and is a continuous function of* $p \in S$.

This proposition, which will be used to prove Theorem 3, can be given an intuitive interpretation as follows. Since the stopping functions only depend on the values of the sequences X and U and the distribution of those values varies smoothly (or is constant) with p, it seems reasonable to expect $\mathrm{E}[N]$ to be a continuous function of p. As established by the proposition, fastness of the Bernoulli factory is indeed sufficient to ensure this.

Theorem 3. *Consider an open set $S \subseteq (0,1)$, a differentiable function $f : S \to (0,1)$ and a (possibly randomized) Bernoulli factory for f that is fast in the sense of* (1). *Then*

$$E[N] \geq (f'(p))^2 \frac{p(1-p)}{f(p)(1-f(p))}. \tag{10}$$

The main idea in the proof of this theorem is as follows. Given an arbitrary, possibly randomized factory for f that uses the input sequence X with parameter p, the output Y can be seen as a sequential estimator of $f(p)$. Wolfowitz's extension of the Cramér-Rao bound to sequential estimators [8] can be applied to Y. This sets a lower bound on $\mathrm{Var}[Y]$ that depends on $E[N]$. Comparing with the actual variance gives the claimed lower bound on $E[N]$.

The proof technique has some similarities with those used by Huber [5, 6]. Specifically, the method employed in [5] to establish a lower bound on $E[N]$ for linear functions $f(p) = cp$, $c > 1$, $p \in (0, (1-\varepsilon)/c)$ is also based on considering the Bernoulli factory as an estimator of $f(p)$; but instead of the Cramér-Rao inequality, a different bound is used which relates $E[N]$ to the confidence level for a given interval. In [6] the Cramér-Rao inequality is used, albeit informally, to provide evidence for a lower bound on $E[N]$ for linear functions.

In order to compare Algorithm 1 with others in terms of $E[N]$, the most interesting case is that of functions f for which this algorithm gives $E[N] \to \infty$ as $p \to 0$; that is, functions of the form (2)–(4) with

$$\lim_{p \to 0} \frac{f(p)}{p} = \infty. \tag{11}$$

In this case, since the average number of inputs used by Algorithm 1 grows without bound, it is important to know if the growth rate could be reduced by using some other algorithm. As will be established by Theorem 4, for a certain subclass of these functions the average number of inputs required by any fast algorithm increases, as $p \to 0$, at least as fast as it does with Algorithm 1, which is thus asymptotically optimal.

Consider the class of functions $f : (0,1) \to (0,1)$ given by (2)–(4) that satisfy (11) and

$$f'(p) = \Omega(f(p)/p) \quad \text{for } p \to 0. \tag{12}$$

Conditions (11) and (12) mean that, asymptotically, $f(p)$ increases faster than p and $f'(p)$ increases at least as fast as $f(p)/p$. In particular, they are fulfilled by any differentiable function f such that

$$\lim_{p \to 0} \frac{f(p)}{p^a} = b \quad \text{for some } a \in (0,1), \ b \in (0,\infty). \tag{13}$$

Indeed, it is clear that (11) holds if (13) does. As for (12), observe that (13) implies $\lim_{p \to 0} f(p) = 0$, and thus by L'Hôpital's rule

$$\lim_{p \to 0} f'(p) \, p^{1-a} = ab. \tag{14}$$

Dividing (14) by (13) it is seen that $\lim_{p \to 0} f'(p)p/f(p) = a$, which implies (12).

Some examples of functions of the form (2)–(4) for which (11) and (12) hold are given by the next proposition.

Proposition 3. *The following functions can be expressed as in* (2)–(4) *and satisfy* (11) *and* (12)*:*

$$f(p) = p^a \quad for\ a \in (0,1) \tag{15}$$

$$f(p) = \frac{2\sqrt{p}}{1 + \sqrt{p}} \tag{16}$$

$$f(p) = \log_2(1 + \sqrt{p}) \tag{17}$$

$$f(p) = \frac{1 - e^{-\sqrt{p}}}{1 - e^{-1}} \tag{18}$$

$$f(p) = p(1 - \log p). \tag{19}$$

Note that functions (15)–(18) satisfy the more specialized condition (13), whereas (19) does not.

The following theorem and its corollary establish that, for the class of functions defined above, Algorithm 1 is asymptotically optimum among all Bernoulli factories that are fast in the sense of Nacu and Peres.

Theorem 4. *Let $S \subseteq (0,1)$ be an open set that has 0 as a limit point. Consider a differentiable function $f : S \to (0,1)$ for which* (12) *holds. Any (possibly randomized) Bernoulli factory for f that is fast in the sense of* (1) *satisfies*

$$\mathrm{E}[N] = \Omega(f(p)/p) \quad for\ p \to 0. \tag{20}$$

Note that this theorem does not require (11). However, for functions that do not satisfy this condition the result (20) is less interesting, and indeed a stronger bound on E[N] can be found. Namely, for a non-constant function any algorithm needs to take at least one input from X, and thus $\mathrm{E}[N] \geq 1$, which is $\Omega(f(p)/p)$ if (11) does not hold. On the other hand, a constant function only satisfies the hypotheses of the theorem if it is the null function; and in any case, constant functions can be simulated by a randomized algorithm without observing X (only U is needed).

Corollary 1. *For any function f that can be expressed as* (2)–(4) *and satisfies conditions* (11) *and* (12)*, Algorithm 1 is asymptotically optimal as $p \to 0$ among all fast algorithms; that is, for any algorithm that satisfies condition* (1)*, there exist $C, \delta > 0$ such that $\mathrm{E}[N] \geq Cf(p)/p$ for all $p < \delta$.*

The results presented in this section are somewhat related to other results that have appeared in previous works. Elias [9] considers a non-randomized Bernoulli factory for the function $f(p) = 1/2$, obtains an entropy-based bound on the average number of outputs per input, and gives an algorithm that approaches that bound. Peres [10] shows that iterating von Neumann's procedure achieves the same efficiency. Stout and Warren [11] carry out a similar analysis for the average number of inputs per output required for simulating $f(p) = 1/2$, and also propose several algorithms. Huber [5, 6], as previously mentioned, considers linear functions $f(p) = cp$, $c > 1$, $p \in (0, (1 - \varepsilon)/c)$, and gives upper and lower bounds on the average number of inputs per output.

For constant functions Theorem 3 reduces to the trivial $\mathrm{E}[N] \geq 0$. On the other hand, for $f(p) = cp$ it yields $c(1 - p)/(1 - cp)$ as a lower bound for E[N]. This is the

bound that was conjectured in [6, section 4] based on an informal argument, which has thus been formalized (and generalized) by Theorem 3. As for Theorem 4, it cannot be directly compared with the bounds in the above referenced works because, as previously mentioned, the theorem does not apply to the constant function $f(p) = 1/2$, and for linear functions it reduces to the trivial $\liminf_{p \to 0} E[N] > 0$.

4. Non-randomized algorithm

A non-randomized version of Algorithm 1 will be given next. Instead of using an auxiliary variable U_i to produce a Bernoulli variable V_i with parameter d_i in step 3, the required V_i is obtained from additional input samples, using a well known procedure based on the binary expansion of d_i [3, proof of proposition 13]. Note also that a variation of [4, algorithm 3] could be used for the same purpose, with progressively finer truncations of the binary representation of d_i providing the lower and upper bounds required therein. Consider the binary expansion of $d_i \in [0, 1]$ assuming zero as integer part and an infinite amount of digits in the fractional part. Thus the decimal values 0, 0.75 and 1 are respectively expressed in binary as $0.000\cdots$, $0.11000\cdots$ (or equivalently $0.10111\cdots$) and $0.11111\cdots$.

Algorithm 2. *The algorithm uses the same steps 1–4 from Algorithm 1 except that step 3 is replaced by the following:*

3.1. *Set $j = 1$.*
3.2. *Keep taking pairs of values from the sequence X until the two values of a pair are different. Let T be the first value of that pair.*
3.3. *If $T = 0$ increase j and go back to step 3.2. Else set V_i equal to the j-th digit in the fractional part of the binary expansion of d_i.*

The total number of inputs taken from X is obviously greater than with Algorithm 1. However, the final value of i in Algorithm 1 has an exponential tail, which can be used for establishing that Algorithm 2 is fast as defined by Nacu and Peres.

Theorem 5. *Algorithm 2 requires an average total number of inputs*

$$E[N] = \frac{f(p)}{p} \left(1 + \frac{2}{p(1-p)} \right). \tag{21}$$

In addition, the algorithm satisfies (1).

The value of $E[N]$ attained by Algorithm 2 could be improved in several ways. Firstly, if $d_i \in [0, 1]$ happens to be a dyadic number, i.e. its binary expansion has an infinite trail of zeros or ones starting at the m-th digit, step 3.2 is unnecessary for $j \geq m$ (once j reaches m, the output V_i is known to be the repeating digit). This can lead to a lower $E[N]$ for certain functions. Secondly, step 3.2, which is von Neumann's procedure for obtaining a Bernoulli variable with parameter $1/2$, can be replaced by more efficient approaches; see for example [9] and [10]. Lastly, instead of Algorithm 2 a modification of [4, algorithm 3] could be used, replacing the auxiliary uniform random variable required therein by a procedure similar to step 3 of Algorithm 2.

8

5. Extensions of the algorithms

The presented algorithms can be modified in several ways to extend the range of functions that can be simulated. Algorithm 1 will be considered in the following, but the discussion also applies to its non-randomized version given by Algorithm 2.

An obvious modification is to change step 4 of the algorithm so that instead of $Y = X_i$ it outputs the complementary variable $Y = 1 - X_i$. This simulates the function $g(p) = 1 - f(p)$. The average number of inputs and asymptotic optimality of the modified algorithm are unaffected. Equation (2) and conditions (11) and (12) are modified replacing $f(p)$ by $1 - g(p)$, whereas (3) and (4) are maintained.

The same operation can be applied to the input variables in step 2. This allows simulation of functions $g(p) = f(1 - p)$, where f satisfies the conditions of the original algorithm; and the simulation is asymptotically optimal for $p \to 1$.

It is also possible to simulate a function obtained from applying certain operations to two constituent functions f_1 and f_2. Define the functions f (composition of f_1, f_2), g (product with complement) and h (convex combination) as follows:

$$f(p) = f_2(f_1(p)) \tag{22}$$
$$g(p) = 1 - (1 - f_1(p))(1 - f_2(p)) \tag{23}$$
$$h(p) = \alpha f_1(p) + (1 - \alpha)f_2(p) \quad \text{for } \alpha \in (0, 1). \tag{24}$$

Proposition 4. *Consider f_1, f_2 that can be expressed as in (2)–(4).*

- *f, g and h can also be expressed as in (2)–(4).*

- *f, g and h satisfy (11) if f_1 and f_2 do.*

- *f, g and h satisfy (12) if f_1 and f_2 do.*

By Proposition 4, if f_1 and f_2 satisfy (2)–(4) Algorithm 1 can be used to directly simulate f, g or h. Alternatively, it is possible to simulate f_1 and f_2 separately and then combine the results to obtain the desired function. In the three cases this alternative approach is easily seen to require the same average number of inputs as applying Algorithm 1 to f, g or h. Consider for example the case of function f. In the alternative approach, Algorithm 1 is first applied to simulate f_1 with input sequence X. This produces a sequence of independent Bernoulli variables with parameter $f_1(p)$. Then the algorithm is applied again to simulate f_2 on this sequence. The first stage requires $f_1(p)/p$ inputs on average to produce each output. The second uses on average $f_2(f_1(p))/f_1(p)$ outputs of the first stage as inputs. The average number of original inputs is the product of those two numbers, which equals $f(p)/p$.

Of course, other combinations of functions may be realizable, even if the resulting function cannot be simulated directly by a single application of the algorithm. For example, if f_1, f_2 satisfy the conditions for Algorithm 1, it is immediate to simulate $f(p) = f_1(p)f_2(p)$ by multiplying the outputs for f_1 and f_2 (the average number of required inputs can be reduced by not computing the second output if the first is 0). However, it may not be possible to simulate f directly because its coefficients c_k are not necessarily non-negative. Similarly, given $\alpha \in (0, 1)$, multiplying the output for f_1 by a Bernoulli variable with parameter α simulates αf_1. This allows relaxing the restriction (4) to $\sum_{k=1}^{\infty} c_k \le 1$.

6. Conclusions and future work

An algorithm has been presented that can simulate certain functions f using an average number of inputs that, within the class of fast algorithms (in the sense of [3]), is asymptotically optimal for p vanishingly small. This algorithm generalizes that given in [7] for $f(p) = \sqrt{p}$, uses fewer inputs, and admits a non-randomized version.

In future research, it would be interesting to relax the sufficient condition (12) for asymptotic optimality (Theorem 4), perhaps using a different proof technique. The class of Bernoulli factories to which Theorems 3 and 4 apply (namely, fast factories) could be extended if the continuity of $E[N]$ as a function of p (Proposition 2) could be proved under more general conditions. It would also be useful to extend the algorithm to a more general class of functions, especially in relation to condition (3). In this regard, [4, section 3.1] gives a method to simulate functions with alternating series expansions.

7. Proofs

7.1. Proof of Proposition 1

From (4) and (5) it stems that

$$d_k = \frac{c_k}{\sum_{j=k}^{\infty} c_j}, \tag{25}$$

$$1 - d_k = \frac{\sum_{j=k+1}^{\infty} c_j}{\sum_{j=k}^{\infty} c_j}. \tag{26}$$

Combining (25) and (26),

$$d_k \prod_{j=1}^{k-1} (1 - d_j) = \frac{c_k}{\sum_{j=k}^{\infty} c_j} \frac{\sum_{j=2}^{\infty} c_j}{\sum_{j=1}^{\infty} c_j} \frac{\sum_{j=3}^{\infty} c_j}{\sum_{j=2}^{\infty} c_j} \cdots \frac{\sum_{j=k}^{\infty} c_j}{\sum_{j=k-1}^{\infty} c_j} = \frac{c_k}{\sum_{j=1}^{\infty} c_j} = c_k. \tag{27}$$

$\square$

7.2. Proof of Theorem 1

The algorithm ends after taking n inputs producing output $Y = 0$ if and only if $X_i = 0$, $V_i = 0$ for $i \leq n - 1$; $V_n = 1$; and $X_n = 0$. Since all these variables are independent,

$$\Pr[N = n, Y = 0] = (1 - d_1)(1 - d_2) \cdots (1 - d_{n-1}) d_n (1 - p)^n, \tag{28}$$

which by Proposition 1 equals $c_n (1 - p)^n$. Therefore

$$\Pr[Y = 1] = 1 - \sum_{n=1}^{\infty} \Pr[N = n, Y = 0] = 1 - \sum_{n=1}^{\infty} c_n (1 - p)^n = f(p). \tag{29}$$

$\square$

7.3. Proof of Theorem 2

The algorithm requires at least n inputs if and only if $X_i = 0$, $V_i = 0$ for $i \leq n-1$; that is,

$$\Pr[N \geq n] = (1-d_1)(1-d_2)\cdots(1-d_{n-1})(1-p)^{n-1}. \tag{30}$$

Thus

$$\mathrm{E}[N] = \sum_{n=1}^{\infty} \Pr[N \geq n] = \sum_{n=1}^{\infty} (1-d_1)(1-d_2)\cdots(1-d_{n-1})(1-p)^{n-1}. \tag{31}$$

Making use of Proposition 1 and equation (5),

$$\mathrm{E}[N] = \sum_{n=1}^{\infty} \frac{c_n}{d_n}(1-p)^{n-1} = \sum_{n=1}^{\infty} (1-p)^{n-1}\left(1 - \sum_{j=1}^{n-1} c_j\right) = \frac{1}{p} - \sum_{n=1}^{\infty} \sum_{j=1}^{n-1} c_j(1-p)^{n-1}. \tag{32}$$

Since all the terms in the double series are non-negative, the order of summation can be changed. This gives, taking into account (2),

$$\mathrm{E}[N] = \frac{1}{p} - \sum_{j=1}^{\infty} \sum_{n=j+1}^{\infty} c_j(1-p)^{n-1} = \frac{1}{p} - \sum_{j=1}^{\infty} \frac{c_j(1-p)^j}{p} = \frac{f(p)}{p}. \tag{33}$$

Using the fact that all coefficients d_i are upper-bounded by 1, (30) yields

$$\Pr[N > n] = (1-d_1)(1-d_2)\cdots(1-d_n)(1-p)^n \leq (1-p)^n, \tag{34}$$

and thus (1) holds with $A = 1$, $\beta = 1 - p < 1$. $\qquad\square$

7.4. Proof of Proposition 2

Let S be an open subset of $(0,1)$, and let $f : S \to (0,1)$ be a function. Consider an arbitrary, randomized Bernoulli factory $\mathscr{B}$ for f based on the sequence X with parameter p, randomizing sequence U, stopping functions $\tau_i(x_1, u_1; \ldots; x_i, u_i)$, and output functions $\gamma_i(x_1, u_1; \ldots; x_i, u_i)$, $i \in \mathbb{N}$, all assumed to be measurable.

For clarity, the following definitions will be used, which explicitly show the dependence of certain probabilities on p: $\Phi_n(p) = \Pr[N \geq n]$; $\varphi_n(p) = \Pr[N = n]$; and $\varphi_{n,y}(p) = \Pr[N = n, Y = y]$ for $y \in \{0,1\}$.

The randomized factory $\mathscr{B}$ can be replaced by an equivalent, non-randomized sequential procedure $\mathscr{B}'$ that produces the same output using an input sequence Z defined by $Z_i = X_i + U_i$. The equivalence is clear from the fact that X_i and U_i can be retrieved from Z_i as $X_i = \lfloor Z_i \rfloor$, $U_i = Z_i - \lfloor Z_i \rfloor$. The stopping functions of $\mathscr{B}'$, denoted by τ_i', are related to those of $\mathscr{B}$ as $\tau_i'(z_1, \ldots, z_i) = \tau_i(\lfloor z_1 \rfloor, z_1 - \lfloor z_1 \rfloor; \ldots; \lfloor z_i \rfloor, z_i - \lfloor z_i \rfloor)$. Each τ_i' is measurable because τ_i is. Similarly, $\gamma_i'(z_1, \ldots, z_i)$ is defined as $\gamma_i(z_1 - \lfloor z_1 \rfloor, \ldots, z_i - \lfloor z_i \rfloor)$. Let the random variable N represent the number of Z_i inputs used by $\mathscr{B}'$ (or of X_i inputs used by $\mathscr{B}$). The sequence Z will be said to have parameter p if the underlying X sequence has parameter p. Note that the stopping time N is randomized from the point of view of X, but is non-randomized with respect to Z.

11

For a given n, the random variables $Z_1, \ldots, Z_n$ are independent; and their joint probability density, with respect to Lebesgue measure, is $\lambda_n(z_1, \ldots, z_n; p) = \prod_{i=1}^{n} \lambda(z_i; p)$ with

$$\lambda(z; p) = \begin{cases} 1 - p & \text{if } z \in (0, 1) \\ p & \text{if } z \in (1, 2). \end{cases} \tag{35}$$

Defining $r = \lfloor z_1 \rfloor + \cdots + \lfloor z_n \rfloor$, $\lambda_n(z_1, \ldots, z_n; p)$ can be expressed as $p^r(1-p)^{n-r}$, and

$$\frac{\partial \lambda_n(z_1, \ldots, z_n; p)}{\partial p} = (r - np)p^{r-1}(1-p)^{n-r-1} = \frac{r - np}{p(1-p)}\lambda_n(z_1, \ldots, z_n; p). \tag{36}$$

Let $R_n \subseteq (0, 2)^n$ be the set of all n-tuples $(z_1, \ldots, z_n)$ such that $\mathscr{B}'$ with inputs $Z_1 = z_1, \ldots, Z_n = z_n$ stops at $N = n$. This set is defined by the conditions $\tau_1'(z_1) = 0$, $\tau_2'(z_1, z_2) = 0$, $\ldots$, $\tau_n'(z_1, \ldots, z_n) = 1$. Since the stopping functions τ_i' are measurable, the region R_n is measurable too, and

$$\varphi_n(p) = \int_{R_n} \lambda_n(z_1, \ldots, z_n; p) \, dz_1 \cdots dz_n. \tag{37}$$

Similarly, given $y \in \{0, 1\}$, let $R_{n,y}$ denote the region of all n-tuples $(z_1, \ldots, z_n)$ such that $\mathscr{B}'$ with inputs $Z_1 = z_1, \ldots Z_n = z_n$ stops at $N = n$ producing the output $Y = y$. Clearly $R_n = R_{n,0} \cup R_{n,1}$, where $R_{n,0}$ and $R_{n,1}$ are disjoint. The function γ_n' is measurable because the functions γ_i are; and thus $\gamma_n'^{-1}(\{y\})$ is a measurable set for $y \in \{0, 1\}$. The intersection of this set with R_n is precisely the region $R_{n,y}$, which is thus measurable, and

$$\varphi_{n,1}(p) = \int_{R_n} \gamma_n'(z_1, \ldots, z_n) \lambda_n(z_1, \ldots, z_n; p) \, dz_1 \cdots dz_n$$
$$= \int_{R_{n,1}} \lambda_n(z_1, \ldots, z_n; p) \, dz_1 \cdots dz_n \tag{38}$$

The probability $\varphi_n(p) = \Pr[N = n]$ is computed as the sum of 2^n terms $\pi(x_1, \ldots, x_n)$, each associated to an n-tuple $(x_1, \ldots, x_n) \in \{0, 1\}^n$, where $\pi(x_1, \ldots, x_n)$ is the probability that the first n inputs of X are $x_1, \ldots, x_n$ and $\mathscr{B}$ stops at $N = n$. With $r = x_1 + \cdots + x_n$, this can be expressed in terms of the stopping functions τ_i as follows:

$$\pi(x_1, \ldots, x_n) = p^r(1-p)^{n-r} \cdot \Pr[\tau_1(x_1, U_1) = 0]$$
$$\cdot \Pr[\tau_2(x_1, U_1; x_2, U_2) = 0 \mid \tau_1(x_1, U_1) = 0] \cdots$$
$$\cdot \Pr[\tau_{n-1}(x_1, U_1; x_2, U_2; \ldots; x_{n-1}, U_{n-1}) = 0 \mid \tau_{n-2}(x_1, U_1; x_2, U_2; \ldots; x_{n-2}, U_{n-2}) = 0]$$
$$\cdot \Pr[\tau_n(x_1, U_1; x_2, U_2; \ldots; x_n, U_n) = 1 \mid \tau_{n-1}(x_1, U_1; x_2, U_2; \ldots; x_{n-1}, U_{n-1}) = 0]. \tag{39}$$

The factors in (39) involving stopping functions do not depend on p. Thus $\pi(x_1, \ldots, x_n)$ is a polynomial in p, and so is $\varphi_n(p)$. This implies that $\Phi_n(p) = 1 - \sum_{i=1}^{n-1} \varphi_i(p)$ is also a polynomial, and thus a continuous function of p. Therefore, taking into account that

$$\mathrm{E}[N] = \sum_{n=1}^{\infty} \Phi_n(p), \tag{40}$$

12

to establish the finiteness of $E[N]$ and its continuity as a function of $p \in S$ it suffices to show that the above series converges uniformly on any interval $[\zeta, \eta] \subset S$.

Consider $\zeta, \eta > 0$ arbitrary such that $[\zeta, \eta] \subset S$. By assumption $\mathscr{B}$ is fast, that is, $\Phi_{n+1}(p)$ satisfies a bound of the form (1). According to [3, proposition 21], this bound can be made uniform on $[\zeta, \eta]$. Thus, there exist A and β independent of p such that $\Phi_n(p) \leq A\beta^{n-1}$ for all $p \in [\zeta, \eta]$. This implies that, given $t \in \mathbb{N}$,

$$\sum_{n=t}^{\infty} \Phi_n(p) \leq A \sum_{n=t}^{\infty} \beta^{n-1} = \frac{A\beta^{t-1}}{1-\beta} \tag{41}$$

for all $p \in [\zeta, \eta]$. Since $\beta < 1$, the bound (41) can be made as small as desired by choosing t large enough, which shows that the series (40) converges uniformly to $E[N]$ on $[\zeta, \eta]$. This implies [12, theorem 2.11] that $E[N]$ is a continuous function of p on that interval. Since ζ and η are arbitrary, $E[N]$ is continuous on S. $\qquad \square$

7.5. Proof of Theorem 3

Let Z, $\mathscr{B}'$, $\lambda_n(z_1, \ldots, z_n; p)$, $\lambda(z; p)$, R_n, $R_{n,y}$, $\varphi_n(p)$ and $\varphi_{n,y}(p)$ be defined as in the proof of Proposition 2.

Applying the sequential procedure $\mathscr{B}'$ to inputs taken from Z with parameter p produces a Bernoulli variable Y with parameter $f(p)$. The key idea of the proof is to consider Y as an estimator of $f(p)$. Namely, the variance of Y is

$$\mathrm{Var}[Y] = f(p)(1 - f(p)). \tag{42}$$

Substituting this into inequality (47) from Lemma 2, to be proved below, will give (10). This lemma, in turn, uses the result in Lemma 1.

Lemma 1. *Under the hypotheses of Theorem 3, the series $\sum_{n=1}^{\infty} \partial \varphi_{n,1}(p)/\partial p$ converges uniformly on any interval $[\zeta, \eta] \subseteq S$.*

Proof. Consider an arbitrary interval $[\zeta, \eta] \subseteq S$. It will be shown first that differentiation under the integral sign is possible in (38) for $p \in [\zeta, \eta]$. This requires checking certain regularity conditions, so that Leibniz's rule can be applied. To this end, consider an open interval (ζ', η') such that $[\zeta, \eta] \subset (\zeta', \eta') \subset S$, with $\zeta' > 0$, $\eta' < 1$. To ensure the validity of Leibniz's rule it suffices to show that the subintegral function and its derivative are continuous and bounded for $(z_1, \ldots, z_n) \in R_{n,1}$, $p \in (\zeta', \eta')$ [12, page 237].

From (36) it is seen that $\partial \lambda_n(z_1, \ldots, z_n; p)/\partial p$ has a discontinuity when any z_i approaches 1 (which causes a jump in r), but is continuous for $(z_1, \ldots, z_n)$ within each of the 2^n hypercubes $(x_1, x_1 + 1) \times \cdots \times (x_n, x_n + 1)$, $x_i \in \{0, 1\}$, $i = 1, \ldots, n$. These hypercubes will be denoted as $H_{x_1, \ldots, x_n}$. In order to differentiate under the integral sign in (38), the region $R_{n,1}$ needs to be divided into 2^n sets resulting from the intersection of $R_{n,1}$ with one of the hypercubes $H_{x_1, \ldots, x_n}$. The resulting sets are disjoint and measurable with respect to Lebesgue measure. The integral in (38) can thus be expressed as

$$\varphi_{n,1}(p) = \sum_{x_1, \ldots, x_n} \int_{R_{n,1} \cap H_{x_1, \ldots, x_n}} \lambda_n(z_1, \ldots, z_n; p) \, \mathrm{d}z_1 \cdots \mathrm{d}z_n. \tag{43}$$

13

Within each of the 2^n integration regions in (43) and for $p \in (\zeta', \eta')$, the function $\partial \lambda_n(z_1, \ldots, z_n; p)/\partial p$ is bounded and continuous, because the region is contained in a single $H_{x_1, \ldots, x_n}$ and p is in (ζ', η') (thus r does not have any jumps and p is bounded away from 0 or 1). The function $\lambda_n(z_1, \ldots, z_n; p)$ is bounded and continuous too. Therefore Leibniz's rule can be applied to each integral in (43); that is,

$$\frac{\partial \varphi_{n,1}(p)}{\partial p} = \sum_{x_1, \ldots, x_n} \int_{R_{n,1} \cap H_{x_1, \ldots, x_n}} \frac{\partial \lambda_n(z_1, \ldots, z_n; p)}{\partial p} \, \mathrm{d}z_1 \cdots \mathrm{d}z_n \tag{44}$$

and $\partial \varphi_{n,1}(p)/\partial p$ is a continuous function of $p \in [\zeta, \eta]$.

The uniform convergence of $\sum_{n=1}^{\infty} \partial \varphi_{n,1}(p)/\partial p$ on $[\zeta, \eta]$ is easily obtained using (36) and (44) as follows. The term $r - np$ in (36) can be bounded as $|r - np| < n$. In addition, for $p \in [\zeta, \eta]$ the term $p(1-p)$ is lower-bounded by $\zeta(1-\eta)$. Therefore

$$\left| \frac{\partial \lambda_n(z_1, \ldots, z_n; p)}{\partial p} \right| \leq \frac{|r - np|}{p(1-p)} \lambda_n(z_1, \ldots, z_n; p) < \frac{n}{\zeta(1-\eta)} \lambda_n(z_1, \ldots, z_n; p). \tag{45}$$

Using (44) and (45) and taking into account (38),

$$\begin{aligned}
\left| \frac{\partial \varphi_{n,1}(p)}{\partial p} \right| &< \sum_{x_1, \ldots, x_n} \int_{R_{n,1} \cap H_{x_1, \ldots, x_n}} \left| \frac{\partial \lambda_n(z_1, \ldots, z_n; p)}{\partial p} \right| \mathrm{d}z_1 \cdots \mathrm{d}z_n \\
&\leq \frac{n}{\zeta(1-\eta)} \sum_{x_1, \ldots, x_n} \int_{R_{n,1} \cap H_{x_1, \ldots, x_n}} \lambda_n(z_1, \ldots, z_n; p) \, \mathrm{d}z_1 \cdots \mathrm{d}z_n \\
&= \frac{n}{\zeta(1-\eta)} \int_{R_{n,1}} \lambda_n(z_1, \ldots, z_n; p) \, \mathrm{d}z_1 \cdots \mathrm{d}z_n = \frac{n\varphi_{n,1}(p)}{\zeta(1-\eta)} \leq \frac{n\varphi_n(p)}{\zeta(1-\eta)}.
\end{aligned} \tag{46}$$

From Proposition 2, $\mathrm{E}[N] = \sum_{n=1}^{\infty} n\varphi_n(p)$ is a continuous function of p on $[\zeta, \eta]$, and then by Dini's theorem [13, page 29] the series $\sum_{n=1}^{\infty} n\varphi_n(p)$ converges uniformly on that interval. By the dominated uniform convergence theorem [14, theorem 5.9], the bound (46) then implies that $\sum_{n=1}^{\infty} \partial \varphi_{n,1}(p)/\partial p$ converges uniformly on $[\zeta, \eta]$. $\square$

Lemma 2. *Under the hypotheses of Theorem 3, the variable Y produced by the Bernoulli factory satisfies the following (sequential Cramér-Rao) bound for $p \in S$:*

$$\mathrm{Var}[Y] \geq \frac{(f'(p))^2 p(1-p)}{\mathrm{E}[N]}, \tag{47}$$

where $\mathrm{E}[N]$ is the average number of inputs used for producing Y.

Proof. It suffices to prove that, given an arbitrary interval $(\zeta, \eta) \subset S$ with $\zeta > 0$, $\eta < 1$, (47) holds for all $p \in (\zeta, \eta)$. This will be done using Wolfowitz's extension of the Cramér-Rao bound to sequential estimators [8], which particularized to $\mathscr{B}'$ and Y will give the desired result.

Consider $(\zeta, \eta) \subset S$, $\zeta > 0$, $\eta < 1$. The sequential version of the Cramér-Rao bound will hold on this interval if the five regularity conditions enunciated in [8, section 3] are satisfied. The first condition specifies that p must belong to an open interval, which is indeed the case.

The second regularity condition requires that $\partial \lambda(z;p)/\partial p$ exist for all p and almost all z, and that $\mathrm{E}[\partial \log \lambda(Z;p)/\partial p] = 0$ and $\mathrm{E}[(\partial \log \lambda(Z;p)/\partial p)^2] > 0$ for all $p \in (\zeta, \eta)$, where Z is a generic variable from the sequence Z. This easily follows from (35) by computing

$$\frac{\partial \lambda(z;p)}{\partial p} = \begin{cases} -1 & \text{if } z \in (0,1) \\ 1 & \text{if } z \in (1,2). \end{cases} \tag{48}$$

and

$$\frac{\partial \log \lambda(z;p)}{\partial p} = \begin{cases} -1/(1-p) & \text{if } z \in (0,1) \\ 1/p & \text{if } z \in (1,2). \end{cases} \tag{49}$$

Taking into account that $\Pr[Z \in (0,1)] = 1 - p$ and $\Pr[Z \in (1,2)] = p$, from (49) it stems that $\mathrm{E}[\partial \log \lambda(z;p)/\partial p] = 0$ and

$$\mathrm{E}\left[\left(\frac{\partial \log \lambda(z;p)}{\partial p}\right)^2\right] = \frac{1}{1-p} + \frac{1}{p} = \frac{1}{p(1-p)}, \tag{50}$$

which is strictly positive as required.

The third condition requires that, for $n \in \mathbb{N}$ and for variables $Z_1, \ldots, Z_n$ belonging to sequence Z,

$$\mathrm{E}\left[\left(\sum_{j=1}^{n} \left|\frac{\partial \log \lambda(Z_j;p)}{\partial p}\right|\right)^2\right] \tag{51}$$

exists for all $p \in (\eta, \zeta)$. This is satisfied because, according to (49), $|\partial \log \lambda(Z_j;p)/\partial p|$ is upper-bounded by $\max\{1/\zeta, 1/(1-\eta)\}$ for $p \in (\zeta, \eta)$.

The fourth condition states that $|\gamma_n'(z_1, \ldots, z_n) \partial \lambda_n(z_1, \ldots, z_n;p)/\partial p|$ be bounded by a measurable function of $z_1, \ldots, z_n$ with finite integral on R_n. This clearly holds because $|\gamma_n'(z_1 \ldots, z_n)| \leq 1$ and, for $p \in (\zeta, \eta)$, (36) implies

$$\left|\frac{\partial \lambda_n(z_1, \ldots, z_n;p)}{\partial p}\right| < \frac{n}{p(1-p)} < \frac{n}{\zeta(1-\eta)}. \tag{52}$$

Lastly, the fifth regularity condition postulates the uniform convergence of the series $\sum_{n=1}^{\infty} \partial \varphi_{n,1}(p)/\partial p$ on the interval (ζ, η). This is established by Lemma 1.

Since the regularity conditions are satisfied, Wolfowitz's single-parameter inequality [8, equation (4.5)] holds for $p \in (\zeta, \eta)$, namely

$$\mathrm{Var}[Y] \geq \frac{(\mathrm{d}\mathrm{E}[Y]/\mathrm{d}p)^2}{\mathrm{E}[N]\mathrm{E}\left[(\partial \log \lambda(z;p)/\partial p)^2\right]}. \tag{53}$$

The estimator Y is unbiased, that is, $\mathrm{E}[Y] = f(p)$. Thus

$$\frac{\mathrm{d}\mathrm{E}[Y]}{\mathrm{d}p} = f'(p). \tag{54}$$

Substituting (50) and (54) into (53) yields (47). Since the interval $(\zeta, \eta) \subseteq S$ is arbitrary, this holds for any $p \in S$. $\qquad\square$

As indicated, the result in Theorem 3 readily follows from (42) and Lemma 2. $\quad\square$

7.6. Proof of Proposition 3

1. $f(p) = p^a$, $a \in (0,1)$ can be written as in (2) with c_k given by (9). These coefficients clearly satisfy (3), and (4) also holds because $\lim_{p \to 0} f(p) = 0 = 1 - \sum_{k=1}^{\infty} c_k$. Condition (13), which implies (11) and (12), is obviously satisfied with the same a as in the function definition and $b = 1$.

2. For f given by (16),

$$1 - f(1-p) = 1 - \frac{2\sqrt{1-p}}{1+\sqrt{1-p}} = \frac{1-\sqrt{1-p}}{1+\sqrt{1-p}} = 2\frac{1-\sqrt{1-p}}{p} - 1. \qquad (55)$$

Consider the series expansion of $\sqrt{p}$ and let c_k' denote its coefficients: $1 - \sqrt{1-p} = \sum_{k=1}^{\infty} c_k' p^k$, where, according to (6), $c_1' = 1/2$. Substituting into (55),

$$1 - f(1-p) = 2\sum_{k=1}^{\infty} c_k' p^{k-1} - 1 = 2\sum_{k=1}^{\infty} c_{k+1}' p^k. \qquad (56)$$

Thus $f(p)$ can be expressed as in (2) with coefficients $c_k = 2c_{k+1}'$, which are positive and sum to 1. On the other hand, condition (13) is satisfied with $a = 1/2$, $b = 2$.

3. From [15, equation 4.6.32],

$$\begin{aligned}
\operatorname{arcsech} z &= \log\left(\frac{1}{z} + \sqrt{\frac{1}{z^2} - 1}\right) = \log\frac{2}{z} - \sum_{k=1}^{\infty} \frac{(2k-1)!!}{(2k)!!}\frac{z^{2k}}{2k} \\
&= \log\frac{2}{z} - \sum_{k=1}^{\infty} \binom{2k}{k}\frac{z^{2k}}{2^{2k+1}k} \quad \text{for } |z| < 1.
\end{aligned} \qquad (57)$$

Substituting $z^2 = 1 - p$ and rearranging,

$$\log(1+\sqrt{p}) = \log 2 - \sum_{k=1}^{\infty} \binom{2k}{k}\frac{(1-p)^k}{2^{2k+1}k} \quad \text{for } p \in (0,1). \qquad (58)$$

Therefore

$$\log_2(1+\sqrt{p}) = 1 - \sum_{k=1}^{\infty} \binom{2k}{k}\frac{(1-p)^k}{2^{2k+1}k\log 2} \quad \text{for } p \in (0,1). \qquad (59)$$

Comparing with (2), the coefficients c_k for this function are seen to be positive. Since $\lim_{p \to 0} \log_2(1+\sqrt{p}) = 0$, the coefficients sum to 1. Lastly, $\lim_{p \to 0} \log_2(1+\sqrt{p})/\sqrt{p}$ is easily seen to be $\log_2 e$, and thus (13) holds.

4. From [16, pages 41 and 42] (see also [17]),

$$e^{(1-\sqrt{1-2zt})/z} = \sum_{k=0}^{\infty} \frac{y_{k-1}(z)}{k!} t^k \qquad (60)$$

where $y_j(z)$ are the Bessel polynomials, which satisfy [16, pages 18 and 20]

$$y_{-1}(z) = y_0(z) = 1, \qquad y_j(z) = (2j-1)zy_{j-1}(z) + y_{j-2}(z). \tag{61}$$

Setting $z = 1$ and substituting $t = p/2$ in (60), and then using (61) gives

$$e^{1-\sqrt{1-p}} = 1 + \sum_{k=1}^{\infty} \frac{y_{k-1}(1)}{2^k k!} p^k. \tag{62}$$

This implies that

$$f(p) = \frac{1 - e^{-\sqrt{p}}}{1 - e^{-1}} = 1 - \sum_{k=1}^{\infty} \frac{y_{k-1}(1)}{(e-1)2^k k!} (1-p)^k. \tag{63}$$

Comparing with (2) and taking into account (61), the coefficients c_k are seen to be positive. Their sum is 1 because $\lim_{p \to 0} f(p) = 0$; and (13) holds because, as is easily seen, $\lim_{p \to 0} f(p)/\sqrt{p} = e/(e-1)$.

5. Consider the series expansion [15, equation 4.1.26]

$$\log p = -\sum_{k=1}^{\infty} \frac{(1-p)^k}{k} \quad \text{for } p \in (0,1). \tag{64}$$

Multiplying by $1 - p$ in (64), subtracting (64), collecting same-power terms and rearranging gives

$$f(p) = p - p\log p = 1 - \sum_{k=2}^{\infty} \frac{1}{k(k-1)} (1-p)^k \quad \text{for } p \in (0,1). \tag{65}$$

Identifying terms in (2) and (65), it is clear that (3) is satisfied. Since $\lim_{p \to 0} f(p) = 0$, (4) holds as well.

Clearly $\lim_{p \to 0} f(p)/p = \lim_{p \to 0}(1 - \log p) = \infty$, which establishes (11). On the other hand, the derivative of $f(p)$ is $f'(p) = -\log p$. Thus

$$\lim_{p \to 0} \frac{f'(p)p}{f(p)} = \lim_{p \to 0} \frac{\log p}{\log p - 1} = 1, \tag{66}$$

and therefore (12) holds. □

7.7. Proof of Theorem 4

The proof uses a standard argument based on the definition of Ω-type asymptotic growth and on Theorem 3.

By assumption (12), there exist $C, \delta > 0$ such that

$$f'(p) \geq Cf(p)/p \quad \text{for } p \in S \cap (0, \delta). \tag{67}$$

The function f satisfies the hypotheses of Theorem 3. Substituting (67) into (10),

$$\mathrm{E}[N] \geq \frac{C^2 f(p)(1-p)}{p(1-f(p))} \quad \text{for } p \in S \cap (0, \delta). \tag{68}$$

17

The claimed result is equivalent to the statement that there are $C', \delta' > 0$ such that $\mathrm{E}[N] > C'f(p)/p$ for all $p \in S \cap (0, \delta')$. Taking $C' = C^2/2$, $\delta' = \min\{\delta, 1/2\}$, from (68) it stems that

$$\mathrm{E}[N] \geq \frac{2C'f(p)(1-p)}{p(1-f(p))} > \frac{C'f(p)}{p} \quad \text{for } p \in S \cap (0, \delta'), \tag{69}$$

which establishes the result. $\qquad\square$

7.8. Proof of Theorem 5

The following result about geometric random variables will be used.

Lemma 3. *Geometric random variables have exponential tails.*

Proof. Given $\theta \in (0,1)$, consider a random variable W with $\Pr[W = w] = \theta(1-\theta)^{w-1}$, $w \in \mathbb{N}$. Computing $\Pr[W \geq w+1] = (1-\theta)^w$ shows that (1) is satisfied with $A = 1$, $\beta = 1 - \theta < 1$. $\qquad\square$

Steps 2–4 of Algorithm 2 (which are the same as in Algorithm 1 except step 3) form an (outer) loop on i which is repeated until the exit condition in step 4 is met. For each iteration of this loop, Algorithm 1 uses one input from X; whereas Algorithm 2 uses that input plus additional ones that are needed for generating V_i, as specified by steps 3.1–3.3. For a given i, the number of iterations of the (inner) loop on j formed by steps 3.2 and 3.3 is a geometric random variable K_i with parameter $1/2$, and thus $\mathrm{E}[K_i] = 2$. For each j, the number of blocks of 2 inputs required within step 3.2 (which can be regarded as a third-level, innermost loop) is a geometric random variable $L_{i,j}$ with parameter $2p(1-p)$, and thus $\mathrm{E}[L_{i,j}] = 1/(2p(1-p))$. Let

$$L_i = L_{i,1} + \cdots + L_{i,K_i}. \tag{70}$$

The variables $L_{i,j}$ are i.i.d. and independent from K_i, and therefore [18, page 194] $\mathrm{E}[L_i] = \mathrm{E}[K_i]\,\mathrm{E}[L_{i,1}] = 1/(p(1-p))$. Thus, for each i Algorithm 2 uses one input from X in step 2 (like Algorithm 1 does), plus $1/(p(1-p))$ 2-input blocks on average in steps 3.1–3.3. Consequently, Algorithm 2 uses on average $1 + 2/(p(1-p))$ as many inputs as Algorithm 1 does. This proves (21).

By Lemma 3, the variables $L_{i,j}$ as well as K_i have exponential tails; and then [3, proposition 12] guarantees that L_i has an exponential tail. For each i, the iteration formed by steps 2–4 of Algorithm 2 requires $1 + 2L_i$ inputs. The total number of iterations of the outer loop on i coincides with the number of inputs of Algorithm 1, which has an exponential tail as established by Theorem 2. Since $1 + 2L_i$ also has an exponential tail, applying [3, proposition 12] again shows that the total number of inputs used by Algorithm 2 has an exponential tail, that is, satisfies (1). $\qquad\square$

7.9. Proof of Proposition 4

Consider functions $f_1(p) = 1 - \sum_{i=1}^{\infty} c_{1,i}(1-p)^i$ and $f_2(p) = 1 - \sum_{j=1}^{\infty} c_{2,j}(1-p)^j$ that satisfy (2)–(4), and let f, g and h be defined as in (22)–(24).

1. Function $f(p) = f_2(f_1(p))$. Identifying coefficients in

$$f(p) = f_2(f_1(p)) = 1 - \sum_{j=1}^{\infty} c_{2,j} \left(\sum_{i=1}^{\infty} c_{1,i}(1-p)^i \right)^j = 1 - \sum_{k=1}^{\infty} c_k (1-p)^k \qquad (71)$$

it is seen that $c_k \geq 0$ for $c_{1,i}, c_{2,j} \geq 0$. Also, $\lim_{p \to 0} f(p) = 0$ because $\lim_{p \to 0} f_1(p) = \lim_{p \to 0} f_2(p) = 0$; and thus $\sum_{k=1}^{\infty} c_k = 1$.

Assume that f_1 and f_2 satisfy (11). Conditions (2)–(4) imply that f_1 has an inverse function, such that $p = f_1^{-1}(s)$, and that $p \to 0$ if and only if $s \to 0$. Therefore, condition (11) for f_1 can be written as

$$\lim_{p \to 0} \frac{f_1(p)}{p} = \lim_{s \to 0} \frac{s}{f_1^{-1}(s)} = \infty. \qquad (72)$$

Thus

$$\lim_{p \to 0} \frac{f(p)}{p} = \lim_{p \to 0} \frac{f_2(f_1(p))}{p} = \lim_{s \to 0} \frac{f_2(s)}{f_1^{-1}(s)} = \lim_{s \to 0} \frac{f_2(s)}{s} \frac{s}{f_1^{-1}(s)}. \qquad (73)$$

Since f_2 also satisfies (11), it follows that $\lim_{p \to 0} f(p)/p = \infty$, which establishes condition (11) for f.

Assuming that (12) holds for f_1 and f_2, to prove that it holds for f it suffices to find $C, \delta > 0$ such that $f'(p) > Cf(p)/p$ for all $p < \delta$. The derivative of f is

$$f'(p) = f_2'(f_1(p)) f_1'(p). \qquad (74)$$

Since f_1 and f_2 satisfy (12), there exist $C_1, C_2, \delta_1, \delta_2 > 0$ such that

$$f'(p) \geq C_2 \frac{f_2(f_1(p))}{f_1(p)} C_1 \frac{f_1(p)}{p} = C_1 C_2 \frac{f_2(f_1(p))}{p}. \qquad (75)$$

for all $p < \min\{\delta_1, f_1^{-1}(\delta_2)\}$. Taking $C = C_1 C_2$ and $\delta = \min\{\delta_1, f_1^{-1}(\delta_2)\}$ establishes the desired result.

2. Function $g(p) = 1 - (1 - f_1(p))(1 - f_2(p))$. Writing

$$g(p) = 1 - (1 - f_1(p))(1 - f_2(p)) = 1 - \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} c_{1,i} c_{2,j} (1-p)^{i+j} = 1 - \sum_{k=1}^{\infty} c_k (1-p)^k \qquad (76)$$

and identifying coefficients makes it clear that $c_k \geq 0$. On the other hand, it is easily seen that $\lim_{p \to 0} f(p) = 0$, which implies that $\sum_{k=1}^{\infty} c_k = 1$.

The function g can be expressed as

$$g(p) = f_1(p) + f_2(p) - f_1(p) f_2(p) = f_1(p) + f_2(p)(1 - f_1(p)). \qquad (77)$$

Taking into account that $\lim_{p \to 0} f_1(p) = 0$, it follows from (77) that (11) holds for g if it holds for f_1 and f_2.

From (77), the derivative of g is computed as

$$g'(p) = f_1'(p)(1 - f_2(p)) + f_2'(p)(1 - f_1(p)). \qquad (78)$$

Assuming that f_1 and f_2 satisfy (12) and taking into account that they are monotone increasing, it stems that there exist $C_1, C_2 > 0$, $\delta \in (0, 1/2)$ such that for all $p < \delta$

$$
\begin{aligned}
g'(p) &\geq C_1 \frac{f_1(p)}{p}(1 - f_2(p)) + C_2 \frac{f_2(p)}{p}(1 - f_1(p)) \\
&> C_1(1 - f_2(1/2))\frac{f_1(p)}{p} + C_2(1 - f_1(1/2))\frac{f_2(p)}{p}.
\end{aligned}
\tag{79}
$$

Defining $C = \min\{C_1(1 - f_2(1/2)), C_2(1 - f_1(1/2))\}$ and making use of (77) again, it stems from (79) that

$$
g'(p) > C\frac{f_1(p) + f_2(p)}{p} > C\frac{g(p)}{p}
\tag{80}
$$

for all $p < \delta$, which establishes (12) for g.

3. Function $h(p) = \alpha f_1(p) + (1 - \alpha)f_2(p)$. The proof is straightforward. $\qquad \square$

Acknowledgments

References

[1] J. von Neumann, Various techniques used in connection with random digits, National Bureau of Standards Applied Mathematics 12 (1951) 36–38.

[2] M. Keane, G. L. O'Brien, A Bernoulli factory, ACM Transactions on Modeling and Computer Simulation 4 (2) (1994) 213–219.

[3] Ş. Nacu, Y. Peres, Fast simulation of new coins from old, Annals of Applied Probability 15 (1A) (2005) 93–115.

[4] K. Łatuszyński, I. Kosmidis, O. Papaspiliopoulos, G. O. Roberts, Simulating events of unknown probabilities via reverse time martingales, Random Structures and Algorithms 38 (4) (2011) 441–452.

[5] M. Huber, Nearly optimal Bernoulli factories for linear functions, Combinatorics, Probability and Computing 25 (4) (2016) 577–591.

[6] M. Huber, Optimal linear Bernoulli factories for small mean problems, Methodology and Computing in Applied Probability 19 (2) (2017) 631–645.

[7] J. Wästlund, Functions arising by coin flipping, http://www.math.chalmers.se/~wastlund/coinFlip.pdf, Chalmers University of Technology (1999).

[8] J. Wolfowitz, The efficiency of sequential estimates and Wald's equation for sequential processes, Annals of Mathematical Statistics 18 (2) (1947) 215–230.

[9] P. Elias, The efficient construction of an unbiased random sequence, Annals of Mathematical Statistics 43 (3).

[10] Y. Peres, Iterating Von Neumann's procedure for extracting random bits, Annals of Statistics 20 (1) (1992) 590–597.

[11] Q. F. Stout, B. Warren, Tree algorithms for unbiased coin tossing with a biased coin, Annals of Probability 12 (1).

[12] W. Fleming, Functions of Several Variables, 2nd Edition, Springer-Verlag, 1977.

[13] F. Hirsch, G. Lacombe, Elements of Functional Analysis, Springer, 1999.

[14] D. Bressoud, A Radical Approach to Real Analysis, 2nd Edition, The Mathematical Association of America, 2007.

[15] M. Abramowitz, I. A. Stegun (Eds.), Handbook of Mathematical Functions, ninth Edition, Dover, 1970.

[16] E. Grosswald, Bessel Polynomials, Springer-Verlag, 1978.

[17] N. J. A. Sloane, The on-line encyclopedia of integer sequences. A144301, http://oeis.org/, 2018.

[18] A. Papoulis, Probability, Random Variables, and Stochastic Processes, 3rd Edition, McGraw-Hill, 1991.