Tags: Probability *Written By: Nimitt*

Bernoulli Factories

This Daily Computer gives a short introduction to *Bernoulli factories*: randomized algorithms that simulate a Bernoulli random variable with parameter $f(p)$ using only samples from a Bernoulli distribution with parameter p.

A Fair Coin. First, let's see a simple algorithm for simulating a Bernoulli sample with parameter p, given access to a Bernoulli distribution with parameter p.

Algorithm 1 Fair coin using unbiased coin (von Neumann trick)

Require: Sample access to a Bernoulli distribution B_p with parameter $p \in (0, 1)$

```
while true do
    Draw independent samples X_1, X_2 ~ B_p
    if (X_1, X_2) = (0, 1) then
        return 0
    else if (X_1, X_2) = (1, 0) then
        return 1
    end if
end while
```

Theorem 1. *Given sample access to a Bernoulli distribution with parameter p, Algorithm 1 returns a Bernoulli sample with parameter $1/2$. Moreover, it draws*

$$\frac{1}{p(1-p)}$$

samples in expectation.

Proof. In a single repetition,

$$\Pr[\text{output } 0] = (1-p)p = p(1-p)$$
$$\Pr[\text{output } 1] = p(1-p)$$

Therefore,

$$\Pr[\text{algorithm outputs}] = 2p(1-p).$$

Conditioned on the event that the algorithm outputs, we obtain

$$\Pr[\text{output } 1 \mid \text{algorithm outputs}] = \frac{p(1-p)}{2p(1-p)} = \frac{1}{2}.$$

Similarly,

$$\Pr[\text{output } 0 \mid \text{algorithm outputs}] = \frac{1}{2}.$$

Thus, the algorithm returns a Bernoulli sample with parameter $1/2$. Now, each repetition succeeds with probability

$$2p(1-p).$$

Hence, the number of repetitions until termination follows a geometric distribution with expectation

$$\frac{1}{2p(1-p)}.$$

Since each repetition uses exactly 2 samples from B_p, the expected number of samples drawn is

$$2 \cdot \frac{1}{2p(1-p)} = \frac{1}{p(1-p)}.$$

□

Bernoulli Factories. A *Bernoulli Factory* is a randomized algorithm that, given sample access to a Bernoulli distribution with parameter p returns a Bernoulli random variable with parameter $f(p)$ for some function f. The above algorithm implements constant function

$$f(p) = \frac{1}{2}$$

A central question in Bernoulli factories is determining which functions f can be simulated using only samples from B_p.

Bernstein polynomial. It is easy to see that using exactly n samples, one can simulate any function of the form

$$f(p) = \sum_{k=0}^{n} a_k \binom{n}{k} p^k (1-p)^{n-k},$$

where $a_k \in \{0, 1\}$ for all k (since probability of k successes equals $\binom{n}{k} p^k (1-p)^{n-k}$). Equivalently, f is a linear combination of the degree-n Bernstein basis polynomials

$$\binom{n}{k} p^k (1-p)^{n-k}.$$

Alternatively, one may write

$$f(p) = \sum_{P \in \mathcal{P}} a_P P,$$

where $\mathcal{P}$ denotes the set of degree-n Bernstein polynomials and $a_P \in \{0, 1\}$.

The above observation suggests that Bernoulli factories are inherently connected to polynomial structure. Indeed, probabilities generated using finitely many samples are combinations of terms of the form $p^k(1-p)^{(n-k)}$. Surprisingly, the following theorem shows that this viewpoint characterizes all Bernoulli factories.

Theorem 2 (Keane and O'Brien (1994))**.** *A function $f : S \subseteq [0, 1] \to [0, 1]$ admits a Bernoulli factory if and only if one of the following holds:*
f is constant, or
f is continuous and polynomially bounded on S.

A function f is polynomially bounded if there exists $n \in \mathbb{N}$ such that $f(p),\ 1 - f(p) \geq \min\left(p^n, (1-p)^n\right)$ for all $p \in S$.