Sequential Implementation of Monte Carlo Tests with Uniformly Bounded Resampling Risk

Axel Gandy

Axel Gandy is Lecturer in Statistics, Department of Mathematics, Imperial College London, UK (E-mail: a.gandy@imperial.ac.uk). Part of the work was carried out at the Centre of Advanced Study at the Norwegian Academy of Science and Letters in Oslo. The author thanks the referees, an associate editor and two editors for helpful comments.

Abstract

This paper introduces an open-ended sequential algorithm for computing the p-value of a test using Monte Carlo simulation. It guarantees that the resampling risk, the probability of a different decision than the one based on the theoretical p-value, is uniformly bounded by an arbitrarily small constant. Previously suggested sequential or non-sequential algorithms, using a bounded sample size, do not have this property. Although the algorithm is open-ended, the expected number of steps is finite, except when the p-value is on the threshold between rejecting and not rejecting.

The algorithm is suitable as standard for implementing tests that require (re-)sampling. It can also be used in other situations: to check whether a test is conservative, iteratively to implement double bootstrap tests, and to determine the sample size required for a certain power. An R-package implementing the sequential algorithm is available online.

Key words: Monte Carlo testing; p-value; Sequential estimation; Sequential test; Significance test.

1 Introduction

Consider a statistical test that rejects the null hypothesis for large values of a test statistic T. Having observed a realization t, one usually wants to compute the p-value

$$p = \mathrm{P}(T \geq t)$$

where, ideally, P is the true probability measure under the null hypothesis. Of course, when the null hypothesis is composite, P is often estimated (parametrically or non-parametrically).

In many cases, e.g. for bootstrap tests, the p-value p cannot be evaluated explicitly. The usual remedy is a Monte Carlo test that essentially replaces p by

$$\hat{p}_{\mathrm{naive}} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\left\{T_i \geq t\right\},$$

where $T_1, \ldots, T_n$ are independent replicates of the test statistic T under P and $\mathbb{I}\{\}$ denotes the indicator function.

A reasonable requirement for a statistical method is what Gleser (1996)

called the *first law of applied statistics*: "Two individuals using the same statistical method on the same data should arrive at the same conclusion." For a test, "conclusion" is whether it rejects or not, i.e. whether p is above or below a given threshold α (often $\alpha = 0.05$).

Monte Carlo tests do not satisfy this law: For an estimator $\hat{p}$, let the *resampling risk* $\mathrm{RR}_p(\hat{p})$ be the probability that $\hat{p}$ and the true p are on different sides of the threshold α. More precisely,

$$\mathrm{RR}_p(\hat{p}) \equiv \begin{cases} \mathrm{P}_p(\hat{p} > \alpha) & \text{if } p \leq \alpha, \\ \mathrm{P}_p(\hat{p} \leq \alpha) & \text{if } p > \alpha. \end{cases}$$

The resampling risk $\mathrm{RR}_p(\hat{p}_{naive})$ of the naive estimator $\hat{p}_{\mathrm{naive}}$ can be substantial, e.g. $\mathrm{RR}_p(\hat{p}_{naive}) = 0.146$ for $n = 999$, $\alpha = 0.1$, and $p = 0.11$ (Jöckel, 1984, Table 1). Furthermore, no matter how large n is chosen,

$$\sup_{p \in [0,1]} \mathrm{RR}_p(\hat{p}_{naive}) \geq 0.5.$$

In the present article, we introduce a recursively defined sequential algorithm which gives an estimator $\hat{p}$ of p that uniformly bounds the resampling risk, i.e.

$$\sup_{p \in [0,1]} \mathrm{RR}_p(\hat{p}) \leq \epsilon$$

for some arbitrary (small) $\epsilon > 0$. Although the algorithm is open-ended, i.e. the number of steps is not bounded, the expected number of steps is finite for $p \neq \alpha$. In particular, if p is far away from the threshold α then the algorithm usually stops quickly.

Having reached step n without stopping, there exists an interval (with length going to 0 as $n \to \infty$) that contains the not yet available estimate $\hat{p}$. This interval can be used as an interim result.

It is well known, that Monte Carlo tests may lose power compared to the theoretical test, see e.g. Hope (1968) or Davison and Hinkley (1997, p. 155,156). Our algorithm bounds this loss of power by the arbitrarily small constant ϵ.

The proposed sequential algorithm can be used as standard implementation for (re-)sampling based tests in statistical software. Essentially, one only has to set ϵ to a suitably small default value (e.g. 10^{-3} or 10^{-5}), and ensure that the algorithm reports intermediate results until it finishes. An R-package is available, see the supplemental material.

Other sequential procedures to compute p-values have been suggested previously. The suggestion of Davidson and MacKinnon (2000) is relatively close to our algorithm. They also use a uniform bound on the resampling risk as motivation. However, their algorithm does not really guarantee this bound since, when deciding whether to stop, they do not take into account the problem of multiple testing. Furthermore, whereas we allow stopping after each step, they only allow stopping after $2^k B$ steps for $k = 0, \ldots, n$, where B is some constant.

Besag and Clifford (1991) suggest a sequential procedure which stops if the partial sum $\sum_{i=1}^{n} \mathbb{I}\{T_i \geq t\}$ reaches a given threshold or if a given number of samples n is reached. The motivation is that if p is high, i.e. if the test result is far away from being significant, fewer replicates are needed than in the naive approach. More recently, Fay et al. (2007) suggested using a truncated sequential probability ratio test.

Andrews and Buchinsky (2000, 2001) suggest to use as criterion for the number of bootstrap repetitions the relative difference between the p-value from the finite-sample bootstrap and the 'ideal' bootstrap using infinite sample

size. This method involves drawing some fixed number of bootstrap samples and then using asymptotic arguments to determine the number of repetitions needed. Once the number of repetitions has been chosen, the test can be performed by drawing the remaining bootstrap repetitions without further sequential considerations.

Bayesian approaches have been suggested in Lai (1988) and in Fay and Follmann (2002). By putting a (prior) distribution on p, the average resampling risk $\mathrm{E}(\mathrm{RR}_p(\hat{p}))$ can be bounded. This bound is much weaker than our uniform bound on $\mathrm{RR}_p(\hat{p})$.

The present article is structured as follows: In Section 2, we precisely define the sequential algorithm and describe the key results of the paper. In Section 3, we comment on several aspects of our algorithm such as the expected number of steps, choice of tuning parameters, and details of the implementation. In Section 4, we demonstrate the wide applicability of our sequential algorithm in a simple practical example. Proofs are relegated to the appendix.

2 The Algorithm and Key Results

Instead of considering independent replicates of $\mathbb{1}\{T \geq t\}$ one can obviously consider replicates from a Bernoulli distribution with parameter p. From now on, let $X_1, X_2, \ldots$ be independent and identically distributed Bernoulli distributed random variables with parameter p. In the notation of the introduction, $X_i = \mathbb{1}\{T_i \geq t\}$. Expectations and probabilities are taken using the p that is indicated in a subscript ($\mathrm{P}_p(\cdot)$ and $\mathrm{E}_p(\cdot)$).

Our sequential algorithm stops once the partial sum $S_n = \sum_{i=1}^{n} X_i$ hits boundaries given by two integer sequences $(U_n)_{n\in\mathbb{N}}$ and $(L_n)_{n\in\mathbb{N}}$ with $U_n \geq L_n$,

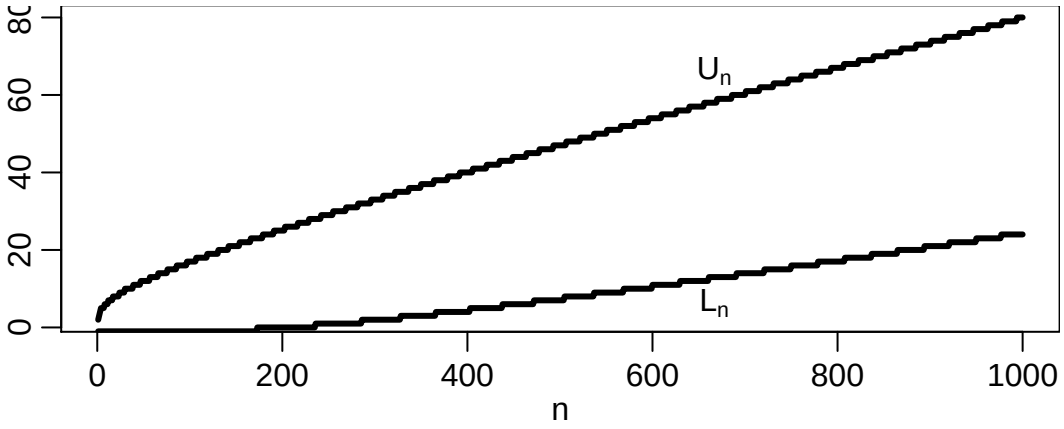

Figure 1: The algorithm stops if $S_n \geq U_n$ or $S_n \leq L_n$ The boundaries are computed for the threshold $\alpha = 0.05$ using the spending sequence $\epsilon_n = \frac{n}{1000(1000+n)}$.

i.e. we stop after

$$\tau = \inf\{k \in \mathbb{N} : S_k \geq U_k \text{ or } S_k \leq L_k\}$$

steps. In the above, $\mathbb{N} = \{1, 2, \dots\}$. Figure 1 shows an example of sequences (U_n) and (L_n) resulting from the following definition.

We construct (U_n) and (L_n) such that for $p \leq \alpha$ (resp. $p > \alpha$) the probability of hitting the upper boundary (U_n) (resp. lower boundary (L_n)) is at most ϵ, where $\epsilon > 0$ is the desired bound on the resampling risk. We will see that it suffices to ensure that for $p = \alpha$ the probability of hitting each boundary is at most ϵ. We use a recursive definition that for each n minimizes U_n (resp. maximizes L_n) conditional on

$$\mathrm{P}_\alpha(\text{hit upper boundary until } n) = \mathrm{P}_\alpha(\tau \leq n, S_\tau \geq U_\tau) \leq \epsilon_n \text{ and} \tag{1}$$

$$\mathrm{P}_\alpha(\text{hit lower boundary until } n) = \mathrm{P}_\alpha(\tau \leq n, S_\tau \leq L_\tau) \leq \epsilon_n, \tag{2}$$

where ϵ_n is a non-decreasing sequence with $\epsilon_n \to \epsilon$ and $0 \leq \epsilon_n < \epsilon$. We call ϵ_n *spending sequence*. The sequence ϵ_n is used to control how fast the allowed resampling risk ϵ is spent. In the examples of this paper, we will use $\epsilon_n = \epsilon\frac{n}{k+n}$

for some constant k. There is a close connection of our spending function ϵ_n to the α-spending function (or "use" function) of Lan and DeMets (1983).

The formal definition of the boundaries (U_n) and (L_n) is as follows: Let $U_1 = 2, L_1 = -1$ and, recursively for $n \in \{2, 3, \dots\}$, let

$$\begin{aligned} U_n &= \min\{j \in \mathbb{N} : \mathrm{P}_\alpha(\tau \geq n, S_n \geq j) + \mathrm{P}_\alpha(\tau < n, S_\tau \geq U_\tau) \leq \epsilon_n\}, \\ L_n &= \max\{j \in \mathbb{Z} : \mathrm{P}_\alpha(\tau \geq n, S_n \leq j) + \mathrm{P}_\alpha(\tau < n, S_\tau \leq L_\tau) \leq \epsilon_n\}, \end{aligned} \tag{3}$$

where $\mathbb{Z}$ is the set of integers. Note that U_n (resp. L_n) is the minimal (resp. maximal) value for which (1) (resp. (2)) holds true given $U_1, \dots, U_{n-1}$ and $L_1, \dots, L_{n-1}$. Using induction, one can see that (1) and (2) hold true for all n. This together with $\epsilon_{n-1} \leq \epsilon_n$ and $0 \leq S_n \leq n$ implies that the set over which the min (resp. max) is taken in (3) always contains $n + 1$ (resp. -1) and is thus nonempty.

The following theorem shows that the expected number of steps of the algorithm is finite for $p \neq \alpha$ and that the probability of hitting the "wrong" boundary is bounded by ϵ.

Theorem 1. *Suppose that* $\epsilon < 1/2$ *and* $\log(\epsilon_n - \epsilon_{n-1}) = o(n)$ *as* $n \to \infty$. *Then* $U_n - \alpha n = o(n)$, $\alpha n - L_n = o(n)$ *and* $\mathrm{E}_p(\tau) < \infty$ *for all* $p \neq \alpha$. *Furthermore,*

$$\sup_{p \in [0,\alpha]} \mathrm{P}_p(\tau < \infty, S_\tau \geq U_\tau) \leq \epsilon \text{ and } \sup_{p \in (\alpha,1]} \mathrm{P}_p(\tau < \infty, S_\tau \leq L_\tau) \leq \epsilon. \tag{4}$$

As estimator for p we use the maximum likelihood estimator

$$\hat{p} = \begin{cases} \dfrac{S_\tau}{\tau}, & \tau < \infty, \\ \alpha, & \tau = \infty. \end{cases}$$

Figure 2 shows how the estimator depends on when the boundary is hit.

The next theorem gives the uniform bound on the resampling risk.

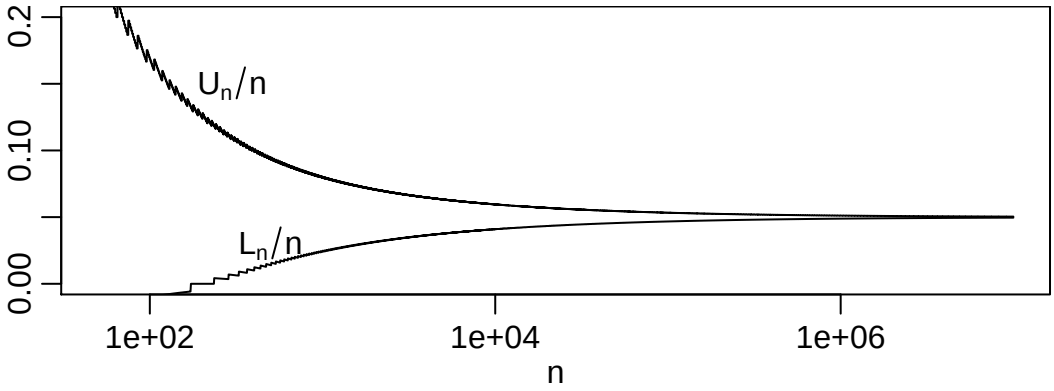

Figure 2: If the upper boundary is hit after τ steps then $\hat{p} = U_\tau/\tau$, if the lower boundary is hit then $\hat{p} = L_\tau/\tau$. We used $\alpha = 0.05$ and $\epsilon_n = \frac{n}{1000(1000+n)}$. Note the log-scale on the horizontal axis.

Theorem 2. *Suppose* $\epsilon \leq 1/4$ *and* $\log(\epsilon_n - \epsilon_{n-1}) = o(n)$ *as* $n \to \infty$. *Then*

$$\sup_{p\in[0,1]} \mathrm{RR}_p(\hat{p}) \leq \epsilon.$$

Note that our default spending sequence $\epsilon_n = \epsilon\frac{n}{k+n}$ satisfies $\log(\epsilon_n - \epsilon_{n-1}) = o(n)$ and thus Theorems 1 and 2 apply to it. Furthermore, an initial period of not stopping at all can be accommodated by using a spending sequence ϵ_n with $\epsilon_k = 0$ for $k \leq k_0$ for some $k_0 \in \mathbb{N}$.

The proofs of Theorems 1 and 2 can be modified to accommodate certain other spending sequences ϵ_n that do not satisfy $\log(\epsilon_n - \epsilon_{n-1}) = o(n)$, including ϵ_n that only change every ν steps, for some $\nu \in \mathbb{N}$, and thus essentially induce checking for stopping only every ν steps.

3 Remarks

3.1 Lower Bound on the Expected Number of Steps

In this section we derive a lower bound on the expected number of steps $\mathrm{E}_p(\tau)$ and show that in a Bayesian setup the expected number of steps is infinite. The bound is not only valid for our algorithm, it is valid for any algorithm with uniformly bounded resampling risk.

For $p_0 > \alpha$ and $\alpha > \delta > 0$ consider the hypotheses $H_0 : p = \alpha - \delta$ and $H_1 : p = p_0$. We can construct a test of H_0 against H_1 by rejecting H_0 iff $\hat{p} > \alpha$. The probability of both the type I and the type II error is ϵ. This is a sequential test that terminates with probability one, therefore, by the lower bound on the expected number of steps given by Wald (1945, (4.81)),

$$\mathrm{E}_{p_0}(\tau) \geq \frac{\epsilon \log\left(\frac{\epsilon}{1-\epsilon}\right) + (1-\epsilon)\log\left(\frac{1-\epsilon}{\epsilon}\right)}{p_0 \log(\frac{p_0}{\alpha-\delta}) + (1-p_0)\log\left(\frac{1-p_0}{1-(\alpha-\delta)}\right)}.$$

Letting $\delta \to 0$, this gives

$$\mathrm{E}_{p_0}(\tau) \geq \frac{\epsilon \log\left(\frac{\epsilon}{1-\epsilon}\right) + (1-\epsilon)\log\left(\frac{1-\epsilon}{\epsilon}\right)}{p_0 \log(\frac{p_0}{\alpha}) + (1-p_0)\log\left(\frac{1-p_0}{1-\alpha}\right)} \equiv \mu_{p_0}. \tag{5}$$

The same lower bound can also be derived for $p_0 < \alpha$.

Suppose, in a Bayesian sense, that p is random, having distribution function F with derivative $F'(\alpha) > 0$. Then, for some $c > 0$ and some $\gamma > 0$,

$$\mathrm{E}(\tau) = \int_0^1 \mathrm{E}_p(\tau) dF(p) \geq c \int_\alpha^{\alpha+\gamma} \left(p \log\left(\frac{p}{\alpha}\right) + (1-p)\log\left(\frac{1-p}{1-\alpha}\right)\right)^{-1} dp = \infty$$

The last equality holds since the integrand is proportional to $(p-\alpha)^{-2}$ as $p \to \alpha$ (by e.g. l'Hospital's rule).

In Section 3.6 we will suggest iterative uses of our algorithm, e.g. to compute the power or the level of a test. Then p is indeed random and $\mathrm{E}(\tau) = \infty$.

Thus in this case we have to truncate our stopping time τ, e.g. by some deterministic constant.

3.2 Error Bound and Spending Sequence

Figure 3 illustrates the dependence of the expected number of steps $\mathrm{E}_p(\tau)$ on the true p and on the error bound ϵ. For most p, the algorithm stops quite quickly. Furthermore, the dependence on the bound ϵ of the resampling risk is only slight, so ϵ can be chosen small in practice.

The lower part of Figure 3 shows that our algorithm with the default spending sequence is not too far away from the theoretical boundary. Can this be improved by choosing a different ϵ_n? What should "improved" mean? There is no obvious optimality criterion. As Section 3.1 shows, a criterion like $\int_0^1 E_p(\tau)dp$ cannot be used since it is always infinite. An option would be to minimize the area under the functions plotted in the lower part of Figure 3, i.e. minimize $\int_0^1 \mathrm{E}_p(\tau)/\mu_p dp$, where μ_p denotes the lower bound in (5). However, pursuing this further is beyond the scope of this article.

In a small study, not reported here, we have looked at other choices besides our default $\epsilon_n = \epsilon\frac{n}{k+n}$. The main conclusion is that the choice of ϵ_n does not seem to have a big influence - as long as the allowed error is spent at a rate satisfying the conditions in Theorems 1 and 2.

3.3 Bounds on the p-value Before Stopping

In practice, the algorithm should report back after a fixed number of steps, even if it has not stopped yet. After this intermediate stop one can continue the algorithm.

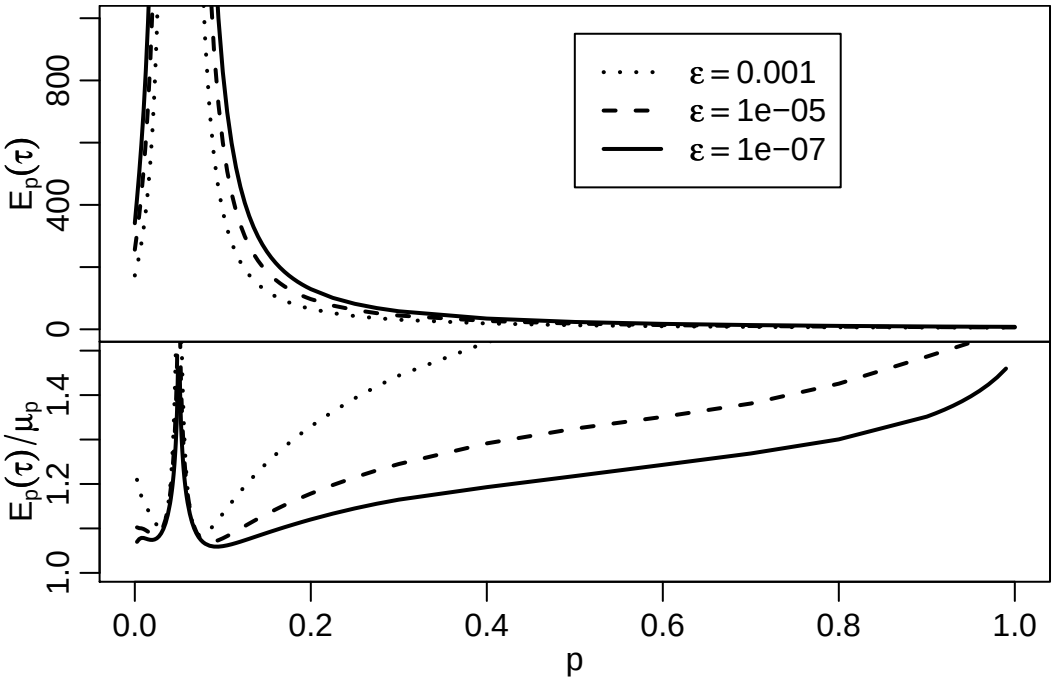

Figure 3: Top: Expected number of steps $\mathrm{E}_p(\tau)$ of the algorithm against the true parameter p. Bottom: $\mathrm{E}_p(\tau)$ divided by the theoretical lower limit in (5). The threshold $\alpha = 0.05$ and the spending sequence $\epsilon_n = \epsilon \frac{n}{1000+n}$ were used.

In the case of such an intermediate stop one can compute an interval in which $\hat{p}$ will eventually lie. In this subsection we will henceforth assume that we have not stopped after n steps.

We start with three observations. The first time the lower/upper boundary can be hit are

$$k_L = \inf\{\nu > n : L_\nu \geq S_n\} \quad \text{and} \quad k_U = \inf\{\nu > n : U_\nu \leq S_n + (\nu - n)\}.$$

The lower boundary can only be hit at step ν if $L_\nu > L_{\nu-1}$ and the upper boundary can only be hit if $U_\nu \leq U_{\nu-1}$. The minimal/maximal values $\hat{p}$ can take when hitting the lower/upper boundary at step ν are

$$\frac{L_{\nu-1}+1}{\nu} \quad \text{and} \quad \frac{U_{\nu-1}}{\nu}.$$

Putting these three observations together we get

$$\hat{p}_{\min} \leq \hat{p} \leq \hat{p}_{\max} \tag{6}$$

where

$$\hat{p}_{\min} = \min\{\frac{L_{\nu-1}+1}{\nu} : \nu \geq k_L, L_\nu > L_{\nu-1}\},$$
$$\hat{p}_{\max} = \max\{\frac{U_{\nu-1}}{\nu} : \nu \geq k_U, U_\nu \leq U_{\nu-1}\}.$$

In the sequel we only discuss the upper bound $\hat{p}_{\max}$ since the lower bound $\hat{p}_{\min}$ has similar properties.

First note that $\hat{p}_{\max} \to \alpha$ as $n \to \infty$. This follows from

$$\frac{U_{\nu-1}}{\nu} \leq \frac{U_{\nu-1}}{\nu-1} \leq \alpha + \gamma_{\nu-1}, \tag{7}$$

and $\gamma_\nu \to 0$ as $\nu \to \infty$, where the last inequality is from (11) in the appendix with $\gamma_\nu = (\sqrt{-\nu \log(\epsilon_\nu - \epsilon_{\nu-1})/2}+1)/\nu$. When the default spending sequence $\epsilon_n = \epsilon\frac{n}{k+n}$ is used then γ_ν is decreasing and $\gamma_\nu \sim \sqrt{\log(\nu)/\nu}$.

The bound $\hat{p}_{\max}$ can be computed in finitely many steps as follows. Assuming that γ_ν is decreasing let M be the smallest integer solution of

$$\alpha + \gamma_M \leq \frac{U_{k_U-1}}{k_U}. \tag{8}$$

Using (7) and (8) we get for $\nu > M$ that

$$\frac{U_{\nu-1}}{\nu} \leq \alpha + \gamma_{\nu-1} \leq \alpha + \gamma_M \leq \frac{U_{k_U-1}}{k_U} \leq \hat{p}_{\max}.$$

Hence,

$$\hat{p}_{\max} = \max\{\frac{U_{\nu-1}}{\nu} : M \geq \nu \geq k_U, U_\nu \leq U_{\nu-1}\}. \tag{9}$$

Since the bound in (7) is not very tight, M may be much larger than k_U and thus evaluating the bounds in (9) can still be computationally expensive. For example with $\alpha = 0.05$, $\epsilon_n = \frac{n}{1000(1000+n)}$ and $k_U = 1000$ (resp. $n = 10000$), we get $M = 10451$ (resp. 117733).

However, (9) usually holds for smaller M than the M obtained through (8). By pre-computations one can obtain M that can be used for specific thresholds α and spending sequences ϵ_n. For example, for $\alpha = 0.05$ and $\epsilon_n = \frac{n}{1000(1000+n)}$, equation (9) holds for $M = k_U + 2000$ if $k_U \leq 10^6$.

3.4 Confidence Intervals

Confidence intervals for p can be constructed similarly to Armitage (1958). Suppose the algorithm stops and returns p^{obs} as result. Then a $1-\beta$ confidence interval for p is given by $[\underline{p}, \overline{p}]$, where $\underline{p} = 0$ for $p^{obs} = 0$, $\overline{p} = 1$ for $p^{obs} = 1$, and otherwise

$$\mathrm{P}_{\underline{p}}(\hat{p} \geq p^{obs}) = \beta/2, \quad \mathrm{P}_{\overline{p}}(\hat{p} \leq p^{obs}) = 1 - \beta/2.$$

Armitage (1958) showed that the probabilities on the left hand sides are strictly monotonic in $\underline{p}$ and $\overline{p}$ and thus $\underline{p}$ and $\overline{p}$ are well-defined.

One can compute $\underline{p}$ and $\overline{p}$ numerically. If the computation of the above probabilities involves an infinite sum we consider the complementary event instead, e.g. we replace $\mathrm{P}_{\underline{p}}(\hat{p} \geq p^{obs})$ by $1 - \mathrm{P}_{\underline{p}}(\hat{p} < p^{obs})$.

Suppose the algorithm has not stopped, i.e. $\tau > n$. Then by the arguments of the previous subsection we get $\hat{p}_{\min}$, $\hat{p}_{\max}$ such that $\hat{p} \in [\hat{p}_{\min}, \hat{p}_{\max}]$. Replacing p^{obs} in the definition of $\underline{p}$ (resp. $\overline{p}$) by $\hat{p}_{\min}$ (resp. $\hat{p}_{\max}$) produces an interval that includes the confidence interval one gets once the algorithm has finished. Thus this is a confidence interval itself, with a (slightly) increased coverage probability.

3.5 Implementation Details

To compute U_n and L_n via (3), one needs the distribution of S_n given $\tau \geq n$ as well as $\mathrm{P}_\alpha(\tau < n, S_\tau \geq U_\tau)$ and $\mathrm{P}_\alpha(\tau < n, S_\tau \leq L_\tau)$. These quantities can be updated recursively. Furthermore, the amount of memory required to store these quantities is proportional to $U_n - L_n$.

What is the additional computational effort for the sequential procedure? The main effort at each step is to compute the distribution of S_n given $\tau \geq n$ from the distribution of S_{n-1} given $\tau \geq n-1$. This effort is proportional to $U_n - L_n$. Hence, if the sequential procedure stops after n steps the computational effort is roughly proportional to $\sum_{i=1}^{n} |U_i - L_i|$. In order to get an idea of how big $U_n - L_n$ is we considered a specific example in Figure 4. In this example it seems as if $U_n - L_n \sim \sqrt{n \log n}$. The overhead of the algorithm can be removed through pre-computation of L_n and U_n.

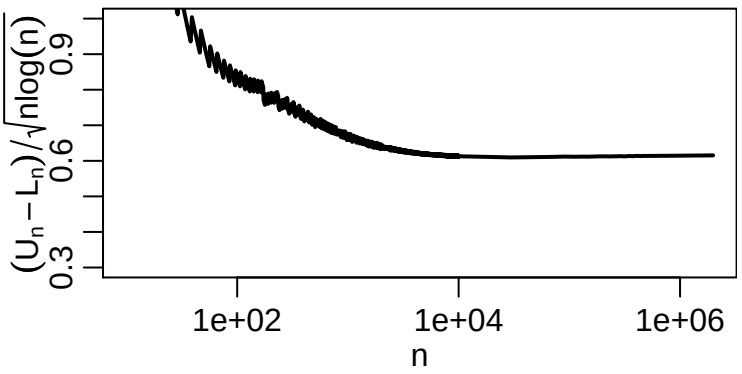

Figure 4: $U_n - L_n$ seems to be roughly proportional to $\sqrt{n \log n}$. We used $\alpha = 0.05$ and $\epsilon_n = \frac{n}{1000(1000+n)}$. Note the log-scale on the horizontal axis.

3.6 Iterated use of the Algorithm

Suppose we want to estimate the power of a resampling based test, e.g. of a bootstrap test. The naive approach would be to n times generate a data set to which the resampling based test is applied with a fixed number m of resamples. Thus the naive approach runs two nested loops, an "outer loop" of length n and an "inner loop" of length m.

We can use our algorithm in the "inner loop". Now the true p-value p is random and to avoid an expected infinite number of steps (see Section 3.1), we need to introduce a fixed upper bound M on the number of resamples. Of course this truncated algorithm no longer satisfies the uniform bound on the resampling risk. The main advantage over the naive approach in the "inner loop" with $m = M$ is a reduction in the number of resamples needed, while (almost) yielding the same answer.

Furthermore, we can use our algorithm in the "outer loop" as well, in particular if we are interested to see if the power of the resampling based test is above a given threshold.

Thus we can use our algorithm in both loops, effectively iterating the algorithm. Similar iterative uses can can be found in Section 4, even going up to a three-times nested loop for checking the level of a double bootstrap tests.

For the problem of computing the power of a bootstrap test, some dedicated algorithms have been suggested. Boos and Zhang (2000) suggest to use an extrapolation method based on a small number of repetitions in the inner loop. Their algorithm can be combined with ours by using our sequential procedure in the inner loop. Jennison (1992) suggest a sequential procedure for the "inner" loop. Jennison (1992) use an approximation to bound the probability of deciding differently than the bootstrap that uses only a fixed number of samples. In contrast to that, the present article bounds the probability of deciding differently than the "ideal" bootstrap based on an infinite sample size.

4 Applications

This section demonstrates the wide applicability of our algorithm in a simple example, already used by Mehta and Patel (1983), by Newton and Geyer (1994), and by Davison and Hinkley (1997, Example 4.22). Suppose 39 observations have been categorized according to two categorical variables resulting in counts given by the following two-way sparse contingency table:

1	2	2	1	1	0	1
2	0	0	2	3	0	0
0	1	1	1	2	7	3
1	1	2	0	0	0	1
0	1	1	1	1	0	0

Let $\boldsymbol{A} = (a_{ij})$ denote this matrix.

Consider the test of the null hypothesis that the two variables are independent which rejects for large values of the likelihood ratio test statistic

$$T(\boldsymbol{A}) = 2 \sum_{i,j} a_{ij} \log(a_{ij}/h_{ij}),$$

where $h_{ij} = \sum_\nu a_{\nu j} \sum_\mu a_{i\mu} / \sum_{\nu\mu} a_{\nu\mu}$. It is well known that under the null hypothesis, as the sample sizes increases, the distribution of $T(\boldsymbol{A})$ converges to a χ^2-distribution with $(7-1)(5-1) = 24$ degrees of freedom. Applying this test to the above matrix leads to a p-value of 0.031.

4.1 Parametric Bootstrap

Since the contingency table $\boldsymbol{A}$ is sparse, the asymptotic approximation may be poor. To remedy this, Davison and Hinkley (1997) suggested a parametric bootstrap that simulates under the null hypothesis based on the row and column sums of $\boldsymbol{A}$.

Using the naive test statistic $\hat{p}_{naive}$ with $n = 1{,}000$ replicates results in a p-value of 0.041. This is below the usual threshold of 5% and thus the test would be interpreted as significant. However, as further computations show, the probability of reporting a p-value larger than 5% with this procedure is roughly 0.08.

Next, we applied our algorithm using $\alpha = 0.05$, $\epsilon = 10^{-3}$ and $\epsilon_n = \epsilon \frac{n}{1000+n}$. We shall use this ϵ and ϵ_n in all other examples of Section 4. Assume that we decide to let our algorithm run for at most 1,000 steps initially. Not having reached a decision, the algorithm tells us that the final estimate will be in the interval $[\hat{p}_{\min}, \hat{p}_{\max}] = [0.0286, 0.0797]$. Our algorithm finally stops after 8,574 samples, reporting a p-value of 0.040. The advantage of our algorithm is that

we can be (almost) certain that the ideal bootstrap would also return a p-value below 0.05.

4.2 Some Notation

To describe further uses of our algorithm we introduce the following notation. Let h_α be the function that applies our algorithm with the threshold α to a sequence with elements in $\{0, 1\}$ and returns the resulting estimate $\hat{p}$. If the sequence is finite, say of length n, and the algorithm has not stopped after n steps then h_α simply returns the current estimate S_n/n.

With this, the above use of our algorithm for the parametric bootstrap can be written as

$$h_{0.05}\left(\mathbb{1}\left\{T(\boldsymbol{A}_i) \geq T(\boldsymbol{A})\right\}_{i\in\mathbb{N}}\right),$$

where $\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots$ denote independent samples under the null hypothesis estimated from the row and column sums of the matrix $\boldsymbol{A}$.

4.3 Checking the Level of Tests

As mentioned earlier, the asymptotic χ^2-distribution may not be a good approximation because the observed matrix $\boldsymbol{A}$ is relatively sparse. We can use our algorithm to check whether the test is conservative or liberal. To do this at the 5% level we estimate the rejection rate by

$$h_\alpha\left(\mathbb{1}\left\{T(\boldsymbol{A}_i) \geq \chi^2_{24,0.95}\right\}_{i\in\mathbb{N}}\right),$$

where $\chi^2_{24,0.95}$ denotes the 0.95 quantile of the χ^2-distribution with 24 degrees of freedom. We start our procedure with a threshold of $\alpha = 0.05$. It stops after 1,437 steps and reports a p-value of 0.074. Hence, the test based on the asymptotic distribution seems to be liberal.

How liberal is it? To find out whether the rejection rate is above 0.07 we start our procedure with a threshold of $\alpha = 0.07$. After 66,736 samples, the estimated rejection rate is 0.075. Thus it is (almost) certain that the test at the nominal level 5% has indeed a level of a least 7%.

Next, we check whether the parametric bootstrap test of Section 4.1 does any better. For this we use the sequential procedure iteratively and compute the rejection rate by

$$h_\alpha\left(\mathbb{1}\left\{h_{0.05}\left(\mathbb{1}\left\{T(\boldsymbol{A}_{ij}) \geq T(\boldsymbol{A}_i)\right\}_{j=1,\ldots,M}\right) \leq 0.05\right\}_{i\in\mathbb{N}}\right),$$

where for each $i \in \mathbb{N}$, $\boldsymbol{A}_{i1}, \boldsymbol{A}_{i2}, \ldots$ denote independent samples under the null hypothesis estimated from the matrix $\boldsymbol{A}_i$.

As explained in Section 3.6, in the "inner" use of $h_{0.05}$ we need to stop after a finite number M of steps. For the following we use $M = 250$. Setting $\alpha = 0.05$, the outer algorithm stops after 603 steps yielding a p-value of 0.090. Using $\alpha = 0.07$ after 1,460 steps we get a p-value of 0.098. Hence, the bootstrap test seems to be quite liberal as well.

For $\alpha = 0.05$ (resp. $\alpha = 0.07$) we generated a total of 44,672 (resp. 156,399) samples in the inner loop. M steps were reached in the inner $h_{0.05}$ in 16.9% (resp. 17.4%) of the cases.

What is the influence of the choice of M? We got the same conclusion about the bootstrap test being liberal when we used $M = 50$, $M = 1000$ and $M = 5000$. Of course, the percentage of times M steps are used in the inner loop and the total number of samples changes. For example, for $\alpha = 0.05$ and $M = 1000$ the outer loop stops after 557 steps, the total number of samples generates in the inner loop is 88,461 and the inner loop reaches $M = 1000$ steps in 8.8% of the cases.

A naive implementation to check the level of the parametric bootstrap test consists of just two nested loops, e.g. using M steps in the inner loop and 1,000 steps in the outer loop. For this, $(M + 1) \times 1000$ samples need to be generated, far more than in our nested sequential algorithm.

4.4 Double Bootstrap

Davison and Hinkley (1997, Example 4.22) suggest that the parametric bootstrap could be improved by using a double bootstrap. The double bootstrap employs two loops that are nested within one another. Davison and Hinkley (1997) suggest that a sensible choice would be to use roughly 1,000 steps in the outer loop and 250 steps in the inner loop. As the classical double bootstrap needs to resample once before starting the inner loop, it needs 251,000 resampling steps.

To reduce the number of steps, we can use our algorithm iteratively: First, compute the p-value from the parametric bootstrap using, say, 10,000 samples by

$$p = \frac{1}{10{,}000} \sum_{i=1}^{10{,}000} \mathbb{1}\left\{T(\boldsymbol{A}_i) \geq T(\boldsymbol{A})\right\}.$$

After that, we compute the p-value of the double bootstrap by

$$h_{0.05}\left(\mathbb{1}\left\{h_p\left(\mathbb{1}\left\{T(\boldsymbol{A}_{ij}) \geq T(\boldsymbol{A}_i)\right\}_{j=1,\ldots,M}\right) \leq p\right\}_{i\in\mathbb{N}}\right).$$

Applying this with $M = 250$, the outer algorithm stops after 1,117 samples in the outer loop and returns a p-value of 0.078, which, in contrast to the previous tests, is not significant at the 5% level. The maximal number of steps $M = 250$ is reached in 15.2% of the cases. In the inner loop we used only 77,405 samples. Adding the 10,000 samples needed to compute p and the 1,117 samples from the null model fitted to $\boldsymbol{A}$ the total number of samples

generated is 88,522. This compares favorably to the 251,000 for the classical double bootstrap (which gave a p-value of 0.081).

Using $M = 1000$, the algorithm stops after 1,432 samples, the p-value is 0.075 and the total number of samples is 224,463. The maximal number of steps $M = 1000$ is reached in 7.6% of the cases.

To check the level of the double bootstrap test we can combine the approach of Section 4.3 with the approach of the current subsection. This results in iterating the procedure three-times. For the double bootstrap we set $M = 250$ and stop the outer algorithm after 500 steps. If we check whether the true level of our algorithm at the asymptotic level 5% is above or below 0.07 the outermost loop stops after 2,386 steps and reports a p-value of 0.050. Hence the double bootstrap seems to be less liberal (if it is liberal at all) than the asymptotic test or the simple parametric bootstrap. In the innermost application of our algorithm we needed 12,688,117 resampling steps. The naive approach with a similar maximal number of steps for the inner loops and 1,000 steps for the outer loop would have used $1{,}000 \cdot 500 \cdot 250 = 1.25 \cdot 10^8$ steps in the innermost loop, more than 9 times the number of samples our iterated algorithm needed.

4.5 Determining Sample Size

In Section 4.3, we have seen how to use our algorithm to check the level of a test. Similarly, with the obvious modification of generating $\boldsymbol{A}_i$ from the given alternative, one can check whether a test achieves a desired power for a given sample size. Furthermore, the minimal sample size that achieves a certain power can be found by combining our algorithm with e.g. a bisectioning

algorithm.

5 Conclusions

We presented a sequential procedure to compute p-values by sampling. When the algorithm stops one has the "peace of mind" that, up to a small error probability, the p-value reported by the procedure is on the same side of some threshold as the theoretical p-value. In other words, the resampling risk is uniformly bounded by a small constant. If the algorithm has not stopped then one can give an interval in which the final estimate will be.

The basic algorithm can also be used in several other situations. It can be used to check whether a test is conservative or liberal, it can be used iteratively for double bootstrap test, and it can be used to determine the sample size needed to achieve a certain power.

A Proofs

The following lemma is needed in the proof of Theorem 1.

Lemma 3. *For* $0 \le p \le q \le 1$,

$$\mathrm{P}_p(\tau < \infty, S_\tau \ge U_\tau) \le \mathrm{P}_q(\tau < \infty, S_\tau \ge U_\tau),$$

$$\mathrm{P}_p(\tau < \infty, S_\tau \le L_\tau) \ge \mathrm{P}_q(\tau < \infty, S_\tau \le L_\tau).$$

Proof. Let $X_1, X_2, \ldots$ be independent Bernoulli(p) random variables and let $Y_i = X_i + (1 - X_i)V_i$, $i = 1, 2, \ldots$, where $V_i \sim$ Bernoulli($(q - p)/(1 - p)$) and $V_1, X_1, V_2, X_2, \ldots$ are independent. Then $Y_i \sim$ Bernoulli(q). On the event $\{\tau < \infty, S_\tau \ge U_\tau\}$, the trajectory (n, S_n) avoids the lower boundary up to τ

and hits the upper boundary at τ. Since $(n, \sum_{k\le n} Y_k)$ lies above (n, S_n), it also avoids the lower boundary up to τ and hits the upper boundary no later than τ. Thus, $\mathrm{P}_p(\tau < \infty, S_\tau \ge U_\tau) \le \mathrm{P}_q(\tau < \infty, S_\tau \ge U_\tau)$.
The second inequality can be shown similarly. □

Proof of Theorem 1. Suppose $L_n \ge U_n$ for some n. Then, by the definition of U_n and L_n,

$$1 = \mathrm{P}_\alpha(\tau \le n) \le \mathrm{P}_\alpha(S_\tau \ge U_\tau, \tau \le n) + \mathrm{P}_\alpha(S_\tau \le L_\tau, \tau \le n) \le 2\epsilon_n < 2\epsilon < 1,$$

which is a contradiction. Hence, $U_n > L_n$.

Let $j_n = \lceil \Delta_n + n\alpha \rceil$, where $\Delta_n = \sqrt{-n \log(\epsilon_n - \epsilon_{n-1})/2}$ and $\lceil x \rceil$ denotes the smallest integer greater than x. We show $U_n \le j_n$. By a special case of Hoeffding's inequality, see Okamoto (1958, Theorem 1) or Hoeffding (1963, Theorem 1),

$$\begin{aligned} \mathrm{P}_\alpha(S_n \ge j_n, \tau \ge n) &\le \mathrm{P}_\alpha(S_n \ge j_n) = \mathrm{P}_\alpha(S_n/n - \alpha \ge j_n/n - \alpha) \\ &\le \exp\left(-2n(j_n/n - \alpha)^2\right) \\ &\le \exp\left(-2n\left(\frac{\Delta_n}{n}\right)^2\right) = \epsilon_n - \epsilon_{n-1}. \end{aligned} \tag{10}$$

By the definition of U_{n-1},

$$\mathrm{P}_\alpha(S_\tau \ge U_\tau, \tau < n) = \mathrm{P}_\alpha(S_{n-1} \ge U_{n-1}, \tau = n-1) + \mathrm{P}_\alpha(S_\tau \ge U_\tau, \tau < n-1) \le \epsilon_{n-1}.$$

This together with the definition of U_n and (10) yields $U_n \le j_n$. Thus,

$$\frac{U_n - n\alpha}{n} \le \frac{\Delta_n + 1}{n} \to 0 \quad (n \to \infty) \tag{11}$$

Similarly, one can show

$$\frac{L_n - n\alpha}{n} \ge -\frac{\Delta_n + 1}{n} \to 0 \quad (n \to \infty). \tag{12}$$

Together with $U_n \geq L_n$ we get $U_n - n\alpha = o(n)$ and $n\alpha - L_n = o(n)$.

Next, we show $\mathrm{E}_p(\tau) < \infty$ for $p > \alpha$. For large n (say $n \geq n_0$),

$$p - \frac{U_n}{n} = (p - \alpha) - (\frac{U_n}{n} - \alpha) = (p - \alpha) + o(1) \geq (p - \alpha)/2.$$

For $n \geq n_0$, since $U_n/n - p \leq 0$, Hoeffding's inequality shows

$$\mathrm{P}_p(\tau > n) \leq \mathrm{P}_p(S_n < U_n) \leq \mathrm{P}_p\left(\frac{S_n}{n} - p \leq \frac{U_n}{n} - p\right) \leq \exp\left(-2n\left(\frac{U_n}{n} - p\right)^2\right).$$

Thus $\mathrm{P}_p(\tau > n) \leq \exp(-n(p-\alpha)^2/2)$ for $n \geq n_0$. Hence,

$$\begin{aligned}\mathrm{E}_p(\tau) =& \int_0^\infty \mathrm{P}_p(\tau > n)dn \leq n_0 + \int_{n_o}^\infty \exp(-n(p-\alpha)^2/2)dn \\ =& n_0 + \frac{2}{(p-\alpha)^2}\exp(-n_0(p-\alpha)^2/2) < \infty.\end{aligned}$$

Similarly, one can show $\mathrm{E}_p(\tau) < \infty$ for $p < \alpha$.

To see (4): Let $p \leq \alpha$. By the previous lemma we have $\mathrm{P}_p(\tau < \infty, S_\tau \geq U_\tau) \leq \mathrm{P}_\alpha(\tau < \infty, S_\tau \geq U_\tau)$. Since $\epsilon_n < \epsilon$, equation (1) implies $\mathrm{P}_\alpha(\tau < \infty, S_\tau \geq U_\tau) \leq \epsilon$. Thus $\mathrm{P}_p(\tau < \infty, S_\tau \geq U_\tau) \leq \epsilon$. The second part of (4) can be shown similarly. □

Proof of Theorem 2. First, consider $p \leq \alpha$. By (4) we know $\mathrm{P}_p(\tau < \infty, S_\tau \geq U_\tau) \leq \epsilon$. Thus it suffices to show that on the complement of $\{\tau < \infty, S_\tau \geq U_\tau\}$, which is $\{\tau = \infty\} \cup \{\tau < \infty, S_\tau \leq L_\tau\}$, we correctly get $\hat{p} \leq \alpha$. If $\tau = \infty$, then by definition $\hat{p} = \alpha$. On $\{\tau < \infty, S_\tau \leq L_\tau\}$ we have $\hat{p} \leq L_\tau/\tau$. We will show $L_n/n < \alpha$ for all n. Suppose $L_n \geq n\alpha$. Let $c = \lceil n\alpha \rceil$. By (Uhlmann, 1966, Satz 6), if $c \leq \frac{n-1}{2}$,

$$\frac{1}{2} \leq \mathrm{P}_{\frac{c}{n-1}}(S_n \leq c) \leq \mathrm{P}_\alpha(S_n \leq c),$$

since $\frac{c}{n-1} \geq \frac{n\alpha}{n-1} = \alpha + \frac{\alpha}{n-1} > \alpha$; and if $c \geq \frac{n-1}{2}$,

$$\frac{1}{2} \leq \mathrm{P}_{\frac{c+1}{n+1}}(S_n \leq c) \leq \mathrm{P}_\alpha(S_n \leq c).$$

since $\frac{c+1}{n+1} \geq \frac{n\alpha+1}{n+1} = \alpha + \frac{1-\alpha}{n+1} > \alpha$. Hence,

$$\frac{1}{2} \leq \mathrm{P}_\alpha(S_n \leq c) \leq \mathrm{P}_\alpha(S_n \leq L_n) \leq \mathrm{P}_\alpha(\tau \leq n) \leq 2\epsilon_n < \frac{1}{2}$$

which is a contradiction.

Now consider $p > \alpha$. Theorem 1 shows $\mathrm{P}_p(\tau < \infty, S_\tau \leq L_\tau) \leq \epsilon$ and $\mathrm{P}_p(\tau = \infty) = 0$. Thus it suffices to show that on $\{\tau < \infty, S_\tau \geq U_\tau\}$ we correctly get $\hat{p} > \alpha$. This can be shown similarly to the first part of the proof. □

Supplemental Materials

R-package for the sequential algorithm: R-package "simctest" containing code to use the sequential algorithm with a spending sequence of type $\epsilon_n = \epsilon\frac{n}{k+n}$ for some $\epsilon > 0$ and $k > 0$. (GNU zipped tar file)

References

Andrews, D. W. K. and Buchinsky, M. (2000). A three-step method for choosing the number of bootstrap repetitions. *Econometrica*, 68(1):23–51.

Andrews, D. W. K. and Buchinsky, M. (2001). Evaluation of a three-step method for choosing the number of bootstrap repetitions. *Journal of Econometrics*, 103(1-2):345–386.

Armitage, P. (1958). Numerical studies in the sequential estimation of a binomial parameter. *Biometrika*, 45(1/2):1–15.

Besag, J. and Clifford, P. (1991). Sequential Monte Carlo p-values. *Biometrika*, 78(2):301–304.

Boos, D. D. and Zhang, J. (2000). Monte Carlo evaluation of resampling-based hypothesis tests. *Journal of the American Statistical Association*, 95(450):486–492.

Davidson, R. and MacKinnon, J. G. (2000). Bootstrap tests: How many bootstraps? *Econometric Reviews*, 19(1):55–68.

Davison, A. and Hinkley, D. (1997). *Bootstrap methods and their application*. Cambridge University Press.

Fay, M. P. and Follmann, D. A. (2002). Designing Monte Carlo implementations of permutation or bootstrap hypothesis tests. *American Statistician*, 56(1):63–70.

Fay, M. P., Kim, H.-J., and Hachey, M. (2007). On using truncated sequential probability ratio test boundaries for Monte Carlo implementation of hypothesis tests. *Journal of Computational & Graphical Statistics*, 16:946 – 967.

Gleser, L. J. (1996). Comment on *Bootstrap Confidence Intervals* by T. J. DiCiccio and B. Efron. *Statistical Science*, 11:219–221.

Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30.

Hope, A. C. A. (1968). A simplified Monte Carlo significance test procedure. *Journal of the Royal Statistical Society. Series B (Methodological)*, 30(3):582–598.

Jennison, C. (1992). Bootstrap tests and confidence intervals for a hazard ratio when the number of observed failures is small, with applications to group

sequential survival studies. In Page, C. and LePage, R., editors, *Computing Science and Statistics*, Proc. 22nd Symposium Interface, pages 89–97, New York. Springer.

Jöckel, K. H. (1984). Computational aspects of Monte Carlo tests. In *Proceedings of COMPSTAT 84*, pages 185–188, Vienna. International Association for Statistical Computing, Physica-Verlag.

Lai, T. L. (1988). Nearly optimal sequential tests of composite hypotheses. *The Annals of Statistics*, 16:856–886.

Lan, K. K. G. and DeMets, D. L. (1983). Discrete sequential boundaries for clinical trials. *Biometrika*, 70:659–663.

Mehta, C. R. and Patel, N. R. (1983). A network algorithm for performing fisher's exact test in $r \times c$ contingency tables. *Journal of the American Statistical Association*, 78(382):427–434.

Newton, M. A. and Geyer, C. J. (1994). Bootstrap recycling: A Monte Carlo alternative to the nested bootstrap. *Journal of the American Statistical Association*, 89(427):905–912.

Okamoto, M. (1958). Some inequalities relating to the partial sum of binomial probabilities. *Annals of the Institute of Statistical Mathematics*, 10:29–35.

Uhlmann, W. (1966). Vergleich der hypergeometrischen mit der binomialverteilung. *Metrika*, 10(1):145–158.

Wald, A. (1945). Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, 16(2):117–186.