PROBABILITY
AND
MATHEMATICAL STATISTICS
Vol. 43, Fasc. 1 (2023), pp. 23–39
Published online 13.4.2023
doi:10.37190/0208-4147.00091

ESTIMATION BY STABLE MOTIONS AND ITS APPLICATIONS

BY

ANIRBAN DAS (STATE COLLEGE, PA), MANFRED DENKER (GÖTTINGEN),
ANNA LEVINA (TÜBINGEN), AND LUCIA TABACU (NORFOLK)

Dedicated to the memory of Wojbor A. Woyczyński (1943–2021)

Abstract. We propose a family of confidence intervals for nonparametric moment estimators if the observations have large or infinite variances. The theoretical underpinnings which guarantee the soundness of the method are demonstrated. Extensive numerical simulations show its superiority over bootstrap and normal approximation and its wide applicability. Finally, a confidence interval to estimate the coupling strength in neuronal networks is proposed.

2020 Mathematics Subject Classification: Primary 62G05; Secondary 62E17, 62G09, 62G15, 60H05.

Key words and phrases: stochastic integral, stable motion, moment estimator, U-statistics, neural avalanches.

1. INTRODUCTION

Resampling by using stable distributions was introduced in [10], but no application and no simulation of its performance was provided at that time. The purpose of this note is to fill this gap, to provide an effective algorithm for this resampling procedure and to apply the method for nonparametric moment estimators.

Most existing methods for establishing asymptotically consistent confidence intervals of parameters rely on exact distributions or on normal approximation (the Central Limit Theorem (CLT)), provided the underlying statistic has finite variance. However, when the variance of the underlying distribution is infinite, the CLT does not hold, no numerically efficient method seems to be known, and again when the variance is large, the confidence intervals become too wide to permit reliable conclusions. Thus the natural question here is to find new statistics not changing the parameter, but allowing for calculations of asymptotic confidence intervals for moments. A class of such new statistics is called resampling using stable motions

© Probability and Mathematical Statistics, 2023

in [10], here abbreviated as *stable resampling*.

The limiting distribution for our resampling method depends on stochastic integrals with respect to stable motions. Here we need only simple integrals of the form $\int f(x)\, M(dx)$. These integrals have a long history, going back to [3, 23, 24, 15, 16] to name a few. The method here should also work for multiple integrals as introduced in [22] by Rosiński and Woyczyński. The quantiles of the limiting distribution are not directly calculable. It is possible to use almost sure versions for the convergence to the limiting distribution (abbreviated as ASLT in analogy to the normal case) which are obtained in [14]. This last work only establishes the limit of simple integrals, though the general case is likely to hold as well.

We begin in Section 2 with a description of the program to be used in the simulations. The corresponding R-code for Table 1, line 1, Table 2, line 1 and Table 4, line 6 is given in [9]; the other entries in the tables are produced using that code with obvious modifications of parameter settings. The program splits into two subroutines; the first is the estimation of confidence intervals, once quantiles are estimated. This is a standard formality, but relies on the limit theorem for stochastic integrals. The other novelty lies in the second subroutine which consists of an estimation of the limiting distribution function using the ASLT method. In Section 3 the theoretical background is briefly sketched leading to the specific form used to calculate confidence intervals and to the formula which permits estimating the unknown distribution from the data directly. This vaguely resembles a bootstrap method, though it is an almost sure and different resampling method.

Simulations are collected in Section 4. It splits into several subsections. First, using a simple form of the ASLT algorithm, simulations for the performance of the estimate of the unknown distribution function are presented.

The second and major part of the section builds the core of this note. Its applications are rather wide, also due to the fact that it works for nondegenerate U-statistics in general. In Section 4, one specific class of U-statistics is used, to estimate confidence intervals for the mean of a distribution which has large or infinite variance. Its algorithm uses the full strength of the resampling algorithm. Clearly, other moment estimators can be handled in the same way. The purpose of this subsection is to present simulation results of various types. First of all we use distributions with heavy tails (power-like distributions) since we shall make use of such distributions in the last subsection. For these models (varying power law and perturbations thereof) we examine different statistics (parametrized by the order of the stable motion), different sample sizes and different confidence levels. In addition we compare the new method with the bootstrap method (established in [10]) and approximation by normal distribution.

Many processes observed in biology and neural science are subject to heavy tail distributions or those where estimation of the mean is affected by large variances even if sample sizes are large. The last subsection provides such an example. We apply our algorithm, called *Stable Resampling for Moments* (SRM), to estimate the connection strength in a complete neuronal network [12] from the expectation

$E(X)$ of its avalanche size distribution. It is shown that the method has advantages over the classical confidence interval estimation for asymptotically normal observables. This is due to the fact that the variance of the distribution is about $(E(X))^3$ and $E(X) \to \infty$ as the number of neurons increases.

2. STABLE RESAMPLING: THE ALGORITHM

This section contains the description of the algorithm on which our estimation procedure for moments is based upon. It will be called *Stable Resampling* and has a more formal description which easily enables the transformation into a code; it consists of two subroutines, where the first one relies on the second one.

2.1. Stable resampling for moments. This subroutine will be called *stable resampling for moments* and is abbreviated as SRM$(q, q', p, r_l, r_u, \delta)$.

Let $q > p > 1$, $q' < q/p$, $r_u, r_l \in \mathbb{N}$, $n_0 < n \in \mathbb{N}$ and $\delta \in \{0, 1\}$. For each choice of these parameters the SRM-algorithm to estimate the q'th moment based on a sample of size n proceeds as follows:[1]

Data: $X_1, \dots X_n$, iid sample following the distribution of X, with $E[|X|^q] < \infty$.
Result: Given $\alpha \in [0, 1]$, the output is a one- or two-sided α-level confidence interval for $E[X^{q'}]$.
Sequential steps:[2]

1. Calculate
$$\hat{\mu} := \frac{1}{n} \sum_{i=1}^{n} X_i^{q'}.$$
2. Generate an iid sample $\{Y_i\}_{i=1}^n$ (independent of the sample $X_1, \dots, X_n$) from a stable distribution with location parameter 1, skewing parameter 0, stability parameter p, and scale parameter 0.5.[3]
3. For $i = 1, \dots, n$ calculate
$$W_i = X_i^{q'} Y_i - \hat{\mu} Y_i. \tag{2.1}$$
Pass the sample $\{W_i\}_{i=1}^n$ as input to the ASLT(p, r_l, r_u) in Section 2.2 to obtain two estimated distribution functions Edf_l and Edf_u. From Edf_l calculate the lower $\alpha/2$ quantile L and from Edf_u calculate the upper $\alpha/2$ quantile U.

[1]The integer n_0 is a separate parameter which has a purely numerical purpose. It prevents small number of observations having a big influence on the estimation of the p-resampled distribution in the second subroutine. In the simulations below in Section 4 we use $n_0 = 9$.

[2]Below, the symbol $a \leftarrow b$ means that a variable a is assigned a value b.

[3]We use here the classical parametrization of a stable distribution! Again, the scale parameter may be chosen differently for variants of the subroutine. Also, repeat this procedure until the average of the Y sample is between 0.7 and 1.3, say. This is no essential restriction and necessary since for the confidence interval one needs to divide by this average.

4. $\overline{XY} \leftarrow \frac{\sum_{i=1}^{n} X_i{}^{q'} Y_i}{n}$, $\overline{Y} \leftarrow \frac{\sum_{i=1}^{n} Y_i}{n}$.

5. If $\delta = 1$ then $C_U \leftarrow \frac{\overline{XY}}{\overline{Y}} - \frac{n^{1/p-1}L}{\overline{Y}}$, $C_L \leftarrow \frac{\overline{XY}}{\overline{Y}} - \frac{n^{1/p-1}U}{\overline{Y}}$. If $\delta = 0$ then $C_U \leftarrow \frac{\hat{\mu}}{\overline{Y}} - \frac{n^{1/p-1}L}{\overline{Y}}$, $C_L \leftarrow \frac{\hat{\mu}}{\overline{Y}} - \frac{n^{1/p-1}U}{\overline{Y}}$.

6. Output the α-level confidence interval $C(q, p, r_l, r_u, \delta) = (C_L, C_U)$ and the $\alpha/2$-level confidence intervals $(-\infty, C_U)$ and (C_L, ∞).

2.2. Estimating the p-resampled distribution function. The two distribution functions needed for the SRM subroutine are called *p-resampled distribution functions* and are calculated according to the following subroutine ASLT(p, r_l, r_u).

Data: $W_1, \ldots, W_n$, iid sample following the distribution of W, $n > n_0$.
Result: A pair of distribution functions Edf_u and Edf_l.
Initialization: $CL_{\text{vec}} \leftarrow$ Vector(size: r_l), $CU_{\text{vec}} \leftarrow$ Vector(size: r_u),
Sequential steps: $r_m \leftarrow \max(r_l, r_u)$,
For $perm \leftarrow 1$ **to** r_m **by** 1 **do**

1. Randomly permute $W_1, \ldots, W_n$ to obtain $W'_1, \ldots, W'_n$.

2. Define $T_1, \ldots, T_{n-n_0}$ as

$$T_i = (i + n_0)^{-1/p} \sum_{i'=1}^{i+n_0} W'_{i'}. \tag{2.2}$$

3. Discretize the argument of T: Define $m = \min\{T_i : i = 1, \ldots, n - n_0, \text{all } perm\}$ and $M = \max\{T_i : i = 1, \ldots, n - n_0, \text{all } perm\}$. Define a suitable step size $h > 0$.

4. For any $t = m + uh \leqslant M$, $u \in \mathbb{Z}_+$, define

$$C(t) = \left(\sum_{i'=n_0+1}^{n} \frac{1}{i'} \right)^{-1} \left(\sum_{i'=n_0+1}^{n} \frac{1}{i'} \mathbb{1}_{(-\infty,t]}(T_{i'-n_0}) \right). \tag{2.3}$$

   If $perm \leqslant r_l$ then $CL_{\text{vec}}[perm] \leftarrow C$.
   If $perm \leqslant r_u$ then $CU_{\text{vec}}[perm] \leftarrow C$.

5. Output

$$Edf_l(t) = \frac{1}{r_l} \sum_{c \in CL_{\text{vec}}} c(t), \qquad Edf_u(t) = \frac{1}{r_u} \sum_{c \in CU_{\text{vec}}} c(t).$$

REMARK. The reason to use a different number of permutations to calculate the p-resampled distribution function is motivated by the lack of support points for the distribution function in tail regions. A permutation changes the values of T_i, so the tails have different supporting points, thus also their average.

3. THEORETICAL JUSTIFICATIONS FOR STABLE RESAMPLING FOR MOMENTS

We first demonstrate the theoretical underpinnings which guarantee the consistency of the method SRM$(q, q', p, r_l, r_u, \delta)$ described in Section 2.1, where q' is the order of the moment being estimated. From [10, Theorem 3.3] we have:

THEOREM 3.1. *Given an iid sample $X_1, X_2, \ldots$ following the distribution of X, any function h satisfying $E[|h(X)|^r] < \infty$, and an iid sample $Y_1, Y_2, \ldots$ following a centered p-stable distribution with p satisfying $r > p > 1$, and also with Y_i independent of $X_{i'}$ for all i, i', we have*

$$\frac{1}{n^{1/p}} \sum_{i \leqslant n} (h(X_i) - E_{x \sim X}[h(x)]) Y_i \xrightarrow{\text{weakly}} G_{X,h,p} \tag{3.1}$$

for some random variable $G_{X,h,p}$ whose distribution depends on X, h, and p.

The setting considered in the algorithm SRM$(q, q', p, r_l, r_u, \delta)$ is recovered if we apply the above theorem with $h(x) = x^{q'}$ such that $E[|X|^q] < \infty$ and $p < q/q'$.

Next we demonstrate that in step 3 of the algorithm of Section 2.1, the call to the algorithm of Section 2.2 yields two distribution functions approximating $G_{X,h,p}$. This in conjunction with (3.1) will establish the veracity of the algorithm of Section 2.1.

DEFINITION 3.1 (p-resampled distribution). For the setting considered in the algorithm SRM$(q, q', p, r_l, r_u, \delta)$, the *$p$-resampled distribution* $G_{X,q',p}$ is defined as the unique limit (in the sense of convergence in law) of $n^{1-1/p} \sum_{i=1}^n W_i$ as $n \to \infty$, with W_i defined in (2.1).

From [14, Theorem 4.1] we get:

THEOREM 3.2. *Let $X_1, \ldots, X_n$ be an iid sample following the distribution of X, h any real-valued function, $Y_1, \ldots, Y_n$ an iid sample following a p-stable distribution with mean 0, and set*

$$T_n(h) := \frac{1}{n^{1/p}} \sum_{i \leqslant n} (h(X_i) - E_{x \sim X}[h(x)]) Y_i.$$

If Y_i is independent of $X_{i'}$ for all i, i', and $T_n(h) \xrightarrow{\text{weakly}} G$ for some random variable G, then for any $-\infty < a < b < \infty$ such that a, b are continuity points of G, we have

$$\lim_{n \to \infty} \frac{1}{\log n} \sum_{k=1}^n \frac{1}{k} \mathbb{1}_{(a,b)}(T_k(h)) \to \mathcal{P}(G \in (a,b)) \quad a.s. \tag{3.2}$$

If Theorem 3.2 holds true under the relaxed condition that $E[Y_i] = 1$ (see the preceding paragraph), it implies that when SRM$(q, q', p, r_l, r_u, \delta)$ calls the subroutine ASLT(p, r_l, r_u) in step 3, the two distributions which are returned both converge in distribution to the p-resampled distribution $G_{X,q',p}$.

Theorem 3.2 is only applicable when $E[Y_1] = 0$. We show its veracity for nonzero values of $E[Y_1]$ when $E_{x\sim X}[|h(x)|^r] < \infty$ for some $r > p > 1$. Without loss of generality let $E_{x\sim X}[h(x)] = 0$. Then

$$\begin{aligned} T_n(h) &= \frac{1}{n^{1/p}} \sum_{i\leqslant n} h(X_i)Y_i \\ &= \frac{1}{n^{1/p}} \sum_{i\leqslant n} h(X_i)(Y_i - E[Y_i]) + \frac{E[Y_i]}{n^{1/p}} \sum_{i\leqslant n} h(X_i). \end{aligned}$$

Theorem 3.2 is applicable for the first part of the sum above; the second part goes to zero because $E_{x\sim X}[|h(x)|^r] < \infty$ and $r > p$ (see [4]).

REMARK 3.1. The sequences $\{T_i\}_{i=1}^{n-n_0}$ defined in (2.2) are different for different permutations $\{W_i'\}_{i=1}^{n}$. This enables different estimates of the quantiles to be derived by permuting the data. One can use this to reduce the variance of the final estimate by averaging over the estimates from different permutations. This is not possible for statistics which do not depend on the order of the samples.

REMARK 3.2. The results of [10] and [14] are also applicable to U-statistics in general under some additional assumption. Since we are only simulating moment estimators, we are not formulating this here.

4. SIMULATION RESULTS

As explained in the introduction, this section summarizes our simulation results for the p-resampled distribution (in Section 4.1), the stable resampling for moments (in Section 4.2) and an application to neuronal avalanches (in Section 4.3).

4.1. Simulations for estimating p-resampled distributions. The algorithm from Section 2.2 when called from step 3 of the algorithm of Section 2.1 returns estimates for the p-resampled distribution (Definition 3.1). In general such functions are called *empirical logarithmic distribution functions* in analogy to second order averaging. Here we demonstrate the robustness of the distribution functions inferred by varying the number of samples, and compare them with estimates of the p-resampled distribution obtained by bootstrapping.

We took N random samples Z_i from a Pareto distribution with shape parameter 2 and location parameter 3, and independently we took N samples Y_i from a stable distribution with order $p = 1.2$, shape $\gamma = 1$, skewness $\beta = 0$ and mean $\delta = 1$. We transformed the data of the first sample using the map

$$x \mapsto f(x) = x \max\{\log|x|, 1),$$

in order to get a distribution not in the strict domain of attraction of a stable distribution (here called *Pareto-like*). Then, for the sample

$$X_i = f(Z_i),$$

define (as in (2.1) with $q' = 1$)

$$W_i = X_i Y_i - \hat{\mu} Y_i, \quad \text{where} \quad \hat{\mu} := \frac{1}{n} \sum_{i=1}^{n} X_i. \tag{4.1}$$

The data was generated using the seed 1345 in the R-software. We compute the empirical distributions produced by passing the sequence W_i to ASLT$(1.2, 1, 1)$ (using $T_1, \ldots, T_n$ instead of $T_1, \ldots, T_{n-n_0}$ in (2.2)) and (2.3). We also estimate the p-resampled distribution by applying bootstrapping in two ways:

- bootstrapping $W_i = X_i Y_i - \hat{\mu} Y_i$, where $\hat{\mu} := \frac{1}{1100} \sum_{i=1}^{1100} X_i$,
- bootstrapping $W_i = X_i Y_i - 12.565 * Y_i$.

Note that 12.565 is the mean of X_1 calculated using 10^6 samples.

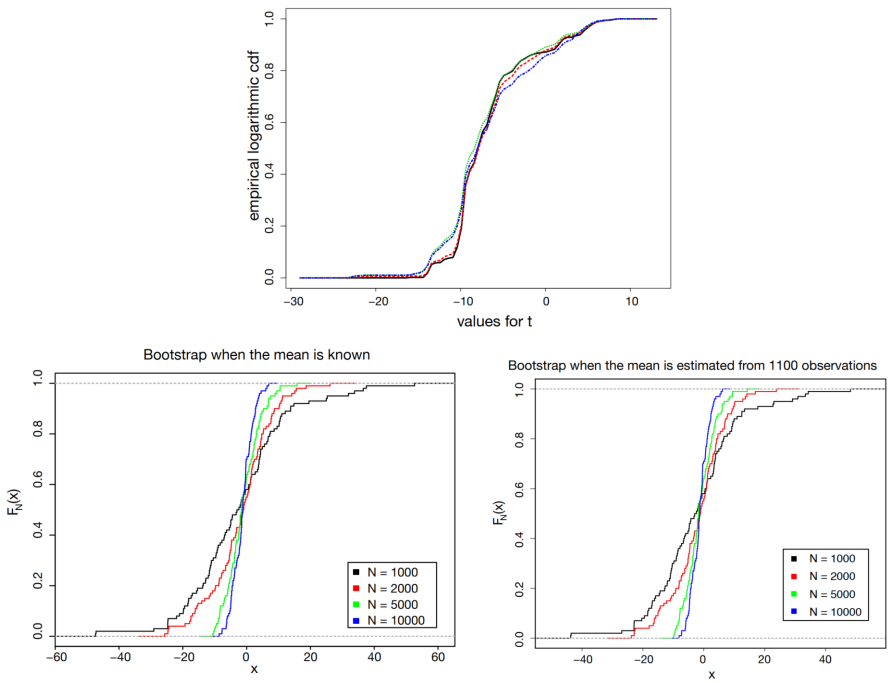

FIGURE 1. ASLT versus bootstrap

Figure 1 shows estimates of the p-resampled distribution for various sample sizes 1000 (black), 2000 (red), 5000 (green) and 10000 (blue) (for colors see the pdf file), using different methods. In Figure 1 the upper graphic shows the estimates when the ASLT algorithm is used. The bottom left graphic shows the performance for the bootstrap method with true mean and the bottom right graphic shows the bootstrap method with estimated mean.

The distribution functions from applying the algorithm of Section 2.2 clearly show the heavy tail behavior of distributions: Large simulated values occur rarely, so are seen only in larger sample size simulations. The mean differs considerably from its median, hence the distribution is not symmetric around 0. The distribution does not have second moments but is in the domain of attraction of a normal. The graphics show that the estimation of the distribution function stabilizes quite well as $n \to \infty$, as expected from Theorem 3.2.

We did not put the graphics in Figure 1 into a single picture since the differences are better seen when keeping them apart. Comparing the approximations in Figure 1, it is first observed that the ASLT convergence is faster than the bootstrap convergence. While the distribution functions in both cases are close for a sample size of 10,000, the convergence for bootstrap is much slower and not at all accurate in the tails for lower sample sizes. Moreover, for a sample size of 10,000 the tails in the bootstrap cases seem to be underestimated. Table 5 supports that observation on the tail behavior.

Therefore it can be said that the ASLT approach is at least as good as – if not better than – the bootstrap approximation from the point of view of speed of convergence. It should also be noticed that the bootstrap distribution seems to become symmetric around 0. It is known from other simulations and the discussion in Section 4.2 that the ASLT approach can even be improved by using some permutations of the data and deleting some initial terms in the summation procedure (cf. the discussion in footnote 1 in Section 2.1 and the Remark in Section 2.2). This has been incorporated in the algorithms of Section 2.

Figure 1 also shows the same behavior using bootstrap when the true mean μ is replaced by an estimated $\hat{\mu}$. The graphics show the same type of approximation as in the bottom left graphic of Figure 1, slightly shifted to the left, an effect due to the underestimation of μ in the example.

4.2. Confidence intervals for the mean of power-like distributions. Here we will investigate the performance of the SRM method for synthetic data sets. First we introduce some metrics which will be used to evaluate the performance.

Recall that the *coverage probability* is the proportion p_c of runs where the unknown parameter θ^* lies in the α-confidence interval divined by the estimation. If $p_c > \alpha$, then the method is called *conservative*. If $p_c < \alpha$, then the method is termed *permissive*.

Also the length of a confidence interval (*CI length*) is used to evaluate the accuracy of an estimation method. We take the average of the lengths of confidence intervals over all runs to obtain the CI length.

We will explicitly restrict ourselves to examples for estimating the first moment. Except otherwise stated, the data for the examples studied in this section are generated from a random variable $X \sim Z \log Z$, where Z follows a Pareto distribution with scale parameter 1, and shape parameter q_{P} (q_{P} varies for various examples); these distributions are loosely called "nearly power laws". Given these

settings, $E[|X|^q]$ is finite for all $q < q_{\mathrm{P}}$, and is ∞ for all $q \geqslant q_{\mathrm{P}}$. Whenever we apply SRM$(q, 1, p, r_l, r_u, \delta)$, we are careful to choose a value of p satisfying $p < q_{\mathrm{P}}$.

SRM$(q, 1, 1.2, r, r, \delta)$ versus normal approximation. We compare the performance of SRM$(q, 1, 1.2, r, r, \delta)$, for both $\delta = 0$ and $\delta = 1$, to a standard CLT based method (which uses normal approximation) in a series of examples where we vary $q < q_{\mathrm{P}}$. We use the coverage probability and CI length to make comparisons. 500 runs of both methods were used; in each run 1000 samples were used, and a two-sided 95 percentile interval was constructed. The exact results for the case $\delta = 0$ are recorded in Table 1; the case $\delta = 1$ showed similar results. For smaller values of q_{P} (underlying distribution is more heavily tailed) the coverage probability of normal approximations is very poor. At $q_{\mathrm{P}} = 4$, the performance of the normal method is excellent (Table 1). The coverage probability produced by SRM$(q, 1, 1.2, r, r, 0)$ is near the required .95 mark for all values of q_{P}; similar trends hold for SRM$(q, 1, 1.2, r, r, 1)$.

TABLE 1. SRM versus normal distribution. P and L are the coverage probability and CI length of SRM$(q, 1, 1.2, r, r, 0)$ for the nearly power laws; P_{nor} and L_{nor} are the coverage probability and CI length for methods using normal approximations.

	P	L	P_{nor}	L_{nor}
$q_{\mathrm{P}} = 1.5, r = 5$	0.948	72.85	0.582	11.04
$q_{\mathrm{P}} = 1.6, r = 5$	0.968	38.06	0.648	4.72
$q_{\mathrm{P}} = 1.7, r = 4$	0.97	27.4	0.738	2.86
$q_{\mathrm{P}} = 2.1, r = 2$	0.946	8.42	0.804	0.95
$q_{\mathrm{P}} = 3.1, r = 2$	0.96	2.28	0.904	0.21
$q_{\mathrm{P}} = 4, r = 2$	0.97	1.33	0.942	0.11

Variation of the nearly power law. We investigate the performance of SRM$(q, 1, 1.2, r_l, r_u, 0)$ for different kinds of "nearly power laws". In particular, we generate a Pareto distribution with scale parameter 1 and shape parameter 1.5 ($q < 1.5$), and pass it through a function f to generate the ground truth data. The performance of the method is evaluated for various choices of f; see Table 2. The method gives acceptable performance for deriving both one-sided and two-sided confidence intervals for a variety of choices for f, which shows a type of robustness of the method with respect to perturbations.

Using a single random permutation. As noted in Remark 3.1, the ASLT part of the algorithm has the ability to reduce variance by taking permutations; here we demonstrate that taking only a single permutation in the algorithm of Section 2.1 results in a rather poor final confidence interval (Table 3).

Performance under different stable distributions and sample sizes. We demonstrate the performance of SRM$(q, 1, p, 5, 5, \delta)$ in two simulations as p varies and

TABLE 2. Nearly power laws. P(0) is the coverage probability when a 95% symmetric two-sided confidence interval is built for the mean; Pu(0) is the coverage probability when a 97.5% one sided upper confidence interval is built; Pl(0) is the coverage probability when a 97.5% one sided lower confidence interval is built; n is the number of samples; r_l and r_u are the parameters of SRM$(q, 1, p, r_l, r_u, 0)$, $q \sim 1.5$.

	Function	r_l	r_u	n	P(0)	Pl(0)	Pu(0)	L
$p = 1.2$	$x \log \lvert x \rvert$	5	5	1000	0.958	0.978	0.98	43.82
$p = 1.2$	$x(1 + \log x)^2$	2	2	1000	0.93	0.992	0.938	0.30
$p = 1.2$	x	5	5	1000	0.976	0.978	0.998	10.64
$p = 1.2$	x	4	2	1000	0.958	0.98	0.978	9.06
$p = 1.2$	$x(1 + \frac{1}{2} \cos x)$	5	5	1000	0.964	0.964	1	10.32
$p = 1.2$	$x(1 + \frac{1}{2} \cos x)$	5	2	1000	0.968	0.97	0.998	12.93
$p = 1.2$	$x \cos x$	5	5	1000	0.994	0.998	0.996	8.02
$p = 1.2$	$x \cos x$	2	2	1000	0.954	0.976	0.978	5.78

TABLE 3. Single permutation. Testing the performance of SRM$(1.5, 1, p, r, r, \delta)$ for making 90% confidence intervals. $P(\delta)$ is the coverage probability when a 90% symmetric two-sided confidence interval is built for the mean; $Pu(\delta)$ ($Pl(\delta)$) is the coverage probability when a 95% one-sided upper (lower) confidence interval is built; n is the number of samples. The data is created by generating a Pareto distribution with scale parameter 1 and shape parameter 1.5, and passing the result through the function $f(x) = x \log x$. 500 runs are made to derive the coverage probabilities, and for each run n samples are taken.

	r	n	P(0)	Pl(0)	Pu(0)	P(1)	Pl(1)	Pu(1)	L
$p = 1.1$	1	1000	0.804	0.894	0.91	0.816	0.958	0.858	38.92
$p = 1.2$	1	1000	0.74	0.872	0.868	0.76	0.956	0.804	24.7
$p = 1.3$	1	1000	0.702	0.882	0.82	0.698	0.957	0.74	35.03
$p = 1.4$	1	1000	0.676	0.874	0.802	0.662	0.964	0.698	14.27

$q \sim 1.5$. The data is generated by the same process as in Table 3. 200 runs are made (with n samples used in every run); for run m, the output from SRM is a confidence interval $(u(m), l(m))$. To visualize the performance, we observe the mean and 2.5-th percentile of the upper limit $u(m)$ (calculated across the runs) in various controlled simulations where the value of p and n is varied. Similarly we also observe the mean and 97.5-th percentile of the lower limit $l(m)$ (see Figure 2). It is observed that for $p = 1.2$, the 97.5-th percentile of the lower limit and the 2.5-th percentile of the upper limit are close to the actual value of the parameter for all values of n. This clearly is not the case when $p = 1.7$. The reason for the poor performance at $p = 1.7$ is that the condition $p < q_{\mathrm{P}}$ is violated for this example.

200 runs of SRM$(1.5, 1, p, 5, 5, \delta)$ are made, with n samples used in every run. For each value of n, p, δ, the mean (orange •) and 2.5-th percentile (orange △) of the upper limit of confidence intervals are indicated. Also the mean (blue •) and 97.5-th percentile (blue △) of the lower limit of confidence intervals are given.

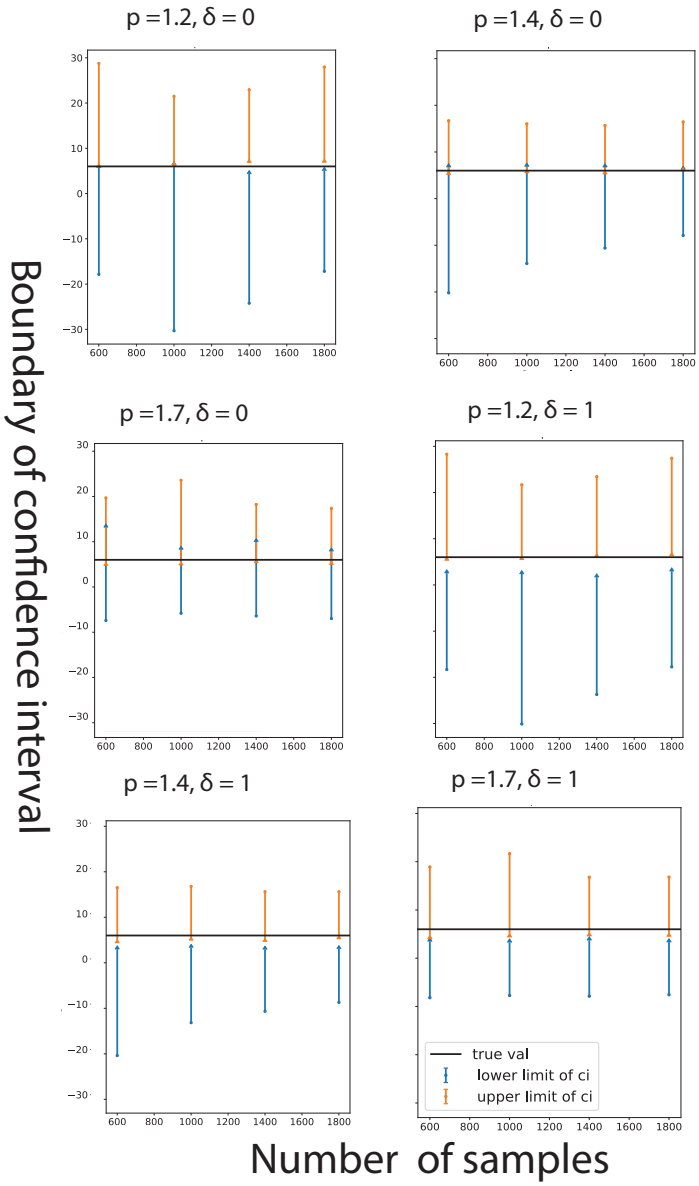

FIGURE 2

In the second simulation we give the results for SRM under various choices of parameters when the underlying data is generated from the same nearly power laws as the data for Table 1 (with $q_{\mathrm{P}} = 1.5$). The results are summarized in Table 4. It is notable that when p, r_l and r_u are fixed, the coverage probability increases with the sample size, so the method gets more conservative.

TABLE 4. Performance of SRM$(q, 1, p, r_l, r_u, \delta)$ when 500 runs of the method are used, each having n samples. P(δ) denotes the coverage probability of the two-sided CI for the particular value of δ; Pl(δ) (Pu(δ)) is the coverage probability of the one-sided CI $[z_\alpha, \infty)$ (CI $(-\infty, z_\alpha]$) where $\alpha = 0.025$ ($\alpha = .975$).

	r_l	r_u	n	P(0)	Pl(0)	Pu(0)	P(1)	Pl(1)	Pu(1)	L
$p = 1.1$	5	5	500	0.962	0.976	0.986	0.978	1	0.978	74.77
$p = 1.1$	5	5	1000	0.986	0.988	0.998	0.99	1	0.99	89.69
$p = 1.1$	5	5	2000	0.976	0.98	0.996	0.996	1	0.996	71.7
$p = 1.1$	2	5	500	0.914	0.932	0.982	0.952	0.99	0.962	74.34
$p = 1.2$	5	5	500	0.944	0.98	0.964	0.948	1	0.948	48.68
$p = 1.2$	5	5	1000	0.958	0.978	0.98	0.958	1	0.958	43.82
$p = 1.2$	5	5	2000	0.974	0.984	0.99	0.966	1	0.966	48.67
$p = 1.2$	3	6	500	0.932	0.946	0.964	0.946	0.996	0.95	24.69
$p = 1.2$	6	7	500	0.934	0.97	0.968	0.97	1	0.97	50.45
$p = 1.2$	3	6	1000	0.93	0.952	0.978	0.96	0.992	0.968	52.52
$p = 1.2$	6	7	1000	0.968	0.982	0.986	0.97	0.998	0.972	168.52
$p = 1.2$	3	2	2000	0.93	0.972	0.958	0.938	0.998	0.94	236.21
$p = 1.2$	2	4	2000	0.946	0.956	0.99	0.946	0.986	0.96	41.4
$p = 1.3$	5	5	500	0.91	0.974	0.936	0.888	1	0.888	44.88
$p = 1.3$	5	5	1000	0.936	0.964	0.972	0.928	0.998	0.93	37.03
$p = 1.3$	5	5	2000	0.954	0.972	0.982	0.942	0.998	0.944	35.56
$p = 1.3$	5	7	500	0.89	0.964	0.934	0.91	0.998	0.912	37.24
$p = 1.4$	5	5	500	0.858	0.952	0.906	0.864	0.998	0.866	36.86
$p = 1.4$	5	5	1000	0.898	0.96	0.938	0.888	0.994	0.894	43.51
$p = 1.4$	5	5	2000	0.926	0.964	0.962	0.94	0.996	0.944	29.27
$p = 1.4$	6	7	500	0.88	0.964	0.916	0.87	0.998	0.972	31.86
$p = 1.4$	5	8	1000	0.928	0.96	0.968	0.92	0.998	0.922	34.02

Bootstrapping versus ASCLT algorithm. In the final set of simulations we see what happens if we pair the resampling approach with bootstrapping (see [11]), instead of the ASLT algorithm. Briefly, the bootstrap method BRM(q', p, m) follows the lines of the algorithm of Section 2.1, with the only difference being that the two estimated distribution functions Edf_l and Edf_u are not estimated using the algorithm of Section 2.2 but by using a bootstrap sample of size m (see [13] for details). The results are given in Table 5; the underlying data is generated from the same nearly power laws as for Table 1 (with $q_{\mathrm{P}} = 1.5$). For all the methods studied, two-sided symmetric 95% confidence intervals for the mean are derived by making 500 runs, each using 1000 samples. The coverage probabilities for the SRM methods remain relatively closer to the desired 95% for all values of parameters and sample size (see Table 4, section pertaining to $p = 1.2$ for more details) when

compared to the BRM setting. Table 5 shows that as m grows, BRM$(1, 1.2, m)$ becomes more conservative; similar trends were observed for SRM$(q, 1, 1.2, 5, 5, \delta)$ as the size of samples was increased, but the results become far less conservative for the SRM$(q, 1, 1.2, \cdot, \cdot, \cdot)$ methods (see Table 4, section pertaining to $p = 1.2$ for more details).

TABLE 5. SRM versus bootstrap. Comparison of different estimators for a 95% CI for two-sided symmetric 95% confidence intervals for the mean are derived by making 500 runs, each using 1000 samples.

Estimator	Coverage probability	Length of CI
SRM(1.5,1,1.2,5,5,0)	0.96	44.27
BRM(1,1.2,50)	0.984	33.72
BRM(1,1.2,200)	0.988	35.96
BRM(1,1.2,500)	0.994	34.87
Normal	0.576	6.26

Comments on the choice of parameters. In order to apply the previous estimations to a given data set $(X_i)_{i \leqslant n}$, one needs to have some pre-information on the existing moments q. This can be done by obtaining a rough numerical or theoretical estimate of the decay of the distribution on the X-variables. Since q' is given as the moment to be estimated, the choice of p is determined by $q' < q/p$ but the quotient should not be close to q' (see Table 4). The parameters r_u, r_l and n_0 are chosen to make the estimation more precise. One needs $\min\{r_u, r_l\} \geqslant 2$ (see Table 3) but should not have too many of those permutations since otherwise limiting procedures for permutation statistics may interfere. As mentioned before, there is an intuitive reason for choosing these parameters $\geqslant 2$: Since we need to estimate tail probabilities of distributions which have widely spread out tails, one needs to have many points in those tail regions. This is accomplished by the permutation procedure. Certainly, for very large sample sizes n one may not need this trick. The role of n_0 was explained in footnote 1 in Section 2.1. The notation $\delta = 0$ or 1 is used to denote the way how the confidence interval is centered, either by $\frac{\overline{XY}}{\overline{Y}}$ or by $\frac{\overline{X}}{\overline{Y}}$. The simulation shows that there is not much difference choosing either of them. The R-code of the program is available in [9].

4.3. An application to neural avalanches. The Abelian distribution is important in models studying neural avalanches (see [12, 17, 18]), and it belongs to the class of Quasi Binomial II distributions [5]. Neural avalanches were observed in field studies, for example by Beggs et al. [1, 2]. Cultured slices from the brain were attached to multielectrode ensembles, and LFP (Local Field Potential) signals were recorded. The data retrieved showed brief intervals of activity, when electrodes detected LFPs above the threshold. The period between these short bursts of activity was marked by idleness. A sequence of such sustained activity was called

an *avalanche*. There are models [12] where the avalanche size (number of neurons firing during an avalanche) statistic follows an Abelian distribution. Recall that this distribution is a probability distribution on $\{1, \ldots, N\}$ defined by the probability density

$$P(Z_{N,p} = b) = C_{N,p} \binom{N-1}{b-1} p^{b-1}(1-bp)^{N-b-1} b^{b-2},$$

where $C_{N,p}$ is the normalization constant defined by $C_{N,p} = \frac{1-Np}{1-(N-1)p}$ with $N \in \mathbb{N}$ and $p \in (0, 1/N)$ ([17], see also [18]). The p in the Abelian distribution is often taken as α/N, where $0 < \alpha < 1$. It is known [17, 18] that

$$E(Z_{N,\alpha/N}) = \frac{N}{N-(N-1)\alpha}, \quad \text{hence} \quad \lim_{N\to\infty} E(Z_{N,\alpha/N}) = \frac{1}{1-\alpha}, \tag{4.2}$$

and (see [7])

$$\lim_{N\to\infty} V(Z_{N,\alpha/N}) = \frac{\alpha}{(1-\alpha)^3}. \tag{4.3}$$

We note that (4.3) can be proved by borrowing results about Quasi Binomial II distributions [6], and asymptotic properties of incomplete gamma integrals. However, an elementary simple proof is given in [7].

The parameter N represents the number of neurons; in practice it is a large number. Also avalanches have been observed for collections of neurons of various sizes [1, 19, 21, 27, 25, 20, 8]; as such, they are not presumed to be a phenomenon dependent on N. For a healthy brain the parameter α is hypothesized to be close to 1 (see [12]). At $\alpha = 1$ it is easy to show that (see [17])

$$\lim_{\alpha\to 1} \lim_{k\to\infty} \lim_{N\to\infty} \frac{P(Z_{N,\alpha/N} = k)}{Ck^{-1.5}} = 1.$$

All of this has three main consequences:

1. For neural avalanche data the ratio of the underlying variance and mean will be very large.

2. The distribution follows a nearly power law with critical exponent 1.5. This is in agreement with experimental observations where avalanche size distributions have been found to follow power-law statistics, possibly with exponential cut-off [26, 1].

3. The quantity α is a useful parameter to be estimated from the data, since the extent of its closeness to 1 is thought to be a measure of the health of the brain. The quantity α can be estimated by estimating the mean.

So this motivates us to estimate the mean of neural avalanche data (using (4.2), one can estimate the confidence interval for α using the confidence intervals for the mean) using the SRM algorithm.

Outline of simulations

Data: We use synthetic data. Our data is generated from a 1.5 exponent power law with upper cut-off at x_m (we will analyze several data sets with different values of x_m). We will generate $n = 1000$ iid instances of the data for each experiment, denoted by $X_1, \ldots, X_n$.

Results and discussion: CLT and SRM methods (p value used is 1.7) for calculating confidence intervals for α for three different values of x_m are shown in Figure 3.

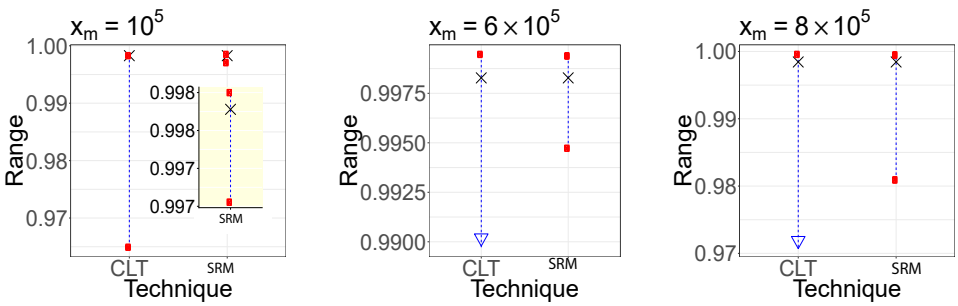

FIGURE 3

On the x-axes we indicate the method used to obtain confidence intervals for α. On the y-axes is shown the range of the 4% confidence interval obtained for each method. Red dots indicate the ends of the confidence intervals. The blue $\bigtriangledown$ symbol indicates that a lower bound for the confidence interval cannot be calculated using the method in question. To calculate confidence intervals we use 1000 instances of synthetic data. The points indicated by $\times$ show the sample mean calculated from 900000 instances of synthetic data. The leftmost inset is to show the SRM results for this case more prominently.

Our simulation studies throw up some features worth noting:

1. To check if our results are accurate we derive the sample mean from a much larger amount of synthetic data than what is used for establishing confidence intervals. This estimate will be called the *precise sample mean* and is marked by a $\times$ in Figure 3.

2. When $x_m = \infty$, the data is generated from a 1.5 exponent power law over all of the positive integers. This distribution has both infinite first and second moments. In such a setting both the SRM and CLT methods will fail. As x_m grows larger, the accuracy of both methods deteriorates. However, because the CLT method depends on higher moment conditions, its accuracy deteriorates faster. Note, however, that for $x_m = 10^5$ the precise sample mean is near the center of the confidence interval calculated by the SRM method. But for $x_m = 8 \times 10^5$ the lower bound of the confidence interval is quite far away from the precise sample mean.

3. It is interesting to note that the methods can sometimes fail to give any lower bound for the confidence interval. The reason for this is as follows: We derive the confidence interval for α from the confidence interval for the mean μ using the understanding $\mu = \frac{1}{1-\alpha}$. For this one requires that both upper and lower confidence bound estimates for the mean be positive; absence of such conditions can result in lack of bounds. This happens in the case of the CLT method for $x_m = 8 \times 10^5$ and $x_m = 6 \times 10^5$. Although the underlying data is non-negative valued, the variance is so large that the lower confidence bound obtained for the mean using CLT becomes negative.

Acknowledgments. This work is dedicated to the memory of Wojbor A. Woyczyński, whose work laid the basis for the present investigation. The second named author was fortunate to discuss early results with WAW in 2018/9 and we are thankful for his comments and advice. We also thank an anonymous referee for various helpful comments.

REFERENCES

[1] J. Beggs and D. Plenz, *Neuronal avalanches in neocortical circuits*, J. Neurosci. 23 (2003), 11167–11177.

[2] J. Beggs and D. Plenz, *Neuronal avalanches are diverse and precise activity patterns that are stable for many hours in cortical slice cultures*, J. Neurosci. 24 (2004), 5216–5229.

[3] J. Bretagnolle, D. Dacunha-Castelle and J.-L. Krivine, *Lois stables et espaces L^p*, Ann. Inst. Henri Poincaré Sect. B 2 (1966), 231–259.

[4] B. D. Choi and S. H. Sung, *On convergence of $(S_n - ES_n)/n^{1/r}$, $1 < r < 2$, for pairwise independent random variables*, Bull. Korean Math. Soc. 22 (1985), 79–82.

[5] P. C. Consul and S. P. Mittal, *A new urn model with predetermined strategy*, Biometrische Z. 17 (1975), 67–75.

[6] P. C. Consul and F. Famoye, *Lagrangian Probability Distributions*, Springer, 2006.

[7] A. Das, *An analytic derivation of the variance for the Abelian distribution*, arXiv:1602.04887 (2016).

[8] A. Das and A. Levina, *Critical neuronal models with relaxed timescale separation*, Phys. Rev. X 9 (2019), art. 021062, 11 pp.

[9] A. Das, M. Denker, A. Levina and L. Tabacu, *R-code for stable resampling*, https://github.com/luciatabacu/Stable-Resampling-/blob/main/sample_code_moments_1JAN2023.Rmd.

[10] H. Dehling, M. Denker and W. A. Woyczyński, *Resampling U-statistics using p-stable laws*, J. Multivariate Anal. 34 (1990), 1–13.

[11] B. Efron, *Bootstrap methods: another look at the jackknife*, Ann. Statist. 7 (1979), 1–26; see also in: Breakthroughs in Statistics, Vol. II, Springer, 1992, 569–593.

[12] C. W. Eurich, J. M. Herrmann and U. A. Ernst, *Finite size effects of avalanche dynamics*, Phys. Rev. E 66 (2002), art. 066137, 15 pp.

[13] P. Hall, *Methodology and theory for the bootstrap*, in: Handbook of Econometrics 4, Elsevier, 1994, 2341–2381.

[14] H. Holzmann, S. Koch and A. Min, *Almost sure limit theorems for U-statistics*, Statist. Probab. Lett. 69 (2004), 261–269.

[15] M. Kanter, *Linear sample spaces and stable processes*, J. Funct. Anal. 9 (1972), 441–459.

[16] J. Kuelbs, *A representation theorem for symmetric stable processes and stable measures on H*, Z. Wahrsch. Verw. Gebiete 26 (1973), 259–271.

[17] A. Levina, *A mathematical approach to self-organized criticality in neural networks*, PhD Diss., Univ. of Göttingen, 2008.

[18] A. Levina and J. M. Herrmann, *The Abelian distribution*, Stoch. Dynam. 14 (2014), art. 1450001, 7 pp.
[19] A. Levina and V. Priesemann, *Subsampling scaling*, Nature Comm. 8 (2017), art. 15140.
[20] T. Petermann, T. C. Thiagarajan, M. A. Lebedev, M. A. L. Nicolelis, D. R. Chialvo and D. Plenz, *Spontaneous cortical activity in awake monkeys composed of neuronal avalanches*, Proc. Nat. Acad. Sci. USA 106 (2009), 15921–15926.
[21] V. Priesemann, M. Valderrama, M. Wibral and M. Le Van Quyen, *Neuronal avalanches differ from wakefulness to deep sleep–evidence from intracranial depth recordings in humans*, PLOS Comput. Biol. 9 (2013), e1002985.
[22] J. Rosiński and W. A. Woyczyński, *On Ito stochastic integrals with respect to p-stable motion: Inner clock, integrability of sample paths, double and multiple integrals*, Ann. Probab. 14 (1986), 271–286.
[23] M. Schilder, *Some structure theorems for the symmetric stable laws*, Ann. Math. Statist. 41 (1970), 412–421.
[24] M. Schreiber, *Quelques remarques sur les caractéristiques des espaces L^p,* $0 \leqslant p < 1$, Ann. Inst. Henri Poincaré 8 (1972), 83–92.
[25] O. Shriki, J. Alstott, F. Carver, T. Holroyd, R. N. A. Henson, M. L. Smith, R. Coppola, E. Bullmore and D. Plenz, *Neuronal avalanches in the resting MEG of the human brain*, J. Neurosci. 33 (2013), 7079–7090.
[26] D. Sornette, *Critical Phenomena in Natural Sciences: Chaos, Fractals, Selforganization and Disorder: Concepts and Tools*, Springer, 2006.
[27] E. Tagliazucchi, P. Balenzuela, D. Fraiman and D. R. Chialvo, *Criticality in large-scale brain fMRI dynamics unveiled by a novel point process analysis*, Front. Physiol. 3 (2012), 15 pp.

Anirban Das
Department of Mathematics
The Pennsylvania State University
University Park, State College, PA 16802, USA
E-mail: anirban.das.psu.math@gmail.com

Manfred Denker
University of Göttingen
Institut für Mathematische Stochastik
Goldschmidtstr. 7
37077 Göttingen, Germany
E-mail: manfred.denker@mathematik.uni-goettingen.de

Anna Levina
Computer Science Department
University of Tübingen
Maria-von-Linden 6
72076 Tübingen, Germany
E-mail: anna.levina@uni-tuebingen.de

Lucia Tabacu
Department of Mathematics and Statistics
Old Dominion University
Engineering and Computational Sciences Building
Norfolk, VA 23529, USA
E-mail: ltabacu@odu.edu

Received 14.7.2022;
accepted 4.3.2023