The Binary Bootstrap for Single Simulation Output Analysis

김 윤 배

Department of Decision Science and Engineering Science
Rensselaer Polytechnic Institute
Troy, NY 12180
U. S. A.

요 약

이 논문에서는 discrete-event 모의 실험을 사용해서 대기행렬 모형에서 대기시간이 길어지는 확률을 추정하는 문제를 연구했읍니다. 단 한번의 모의 실험에서 확률의 신뢰구간을 구할수있는 방법인, binary bootstrap을 개발 했읍니다. Bernoulli trial과 first-order Markov processes에 적용하여 본 결과 이론치에 별 차이없이 추정하였읍니다. 또한 M/M/1 대기행렬모형에서 대기시간이 길 확률을 추정했을때 batch means 방법 보다 binary bootstrap이 월등히 우수한 결과를 보였읍니다.

I. Introduction

We present a new method for inference about the probability of long delay in a queueing system based on a single run of a discrete event simulation. We call the method "binary bootstrap". The binary bootstrap has certain advantages over the conventional batch means method for creating a confidence interval from a single simulation run.

Most simulation analyses of queueing-type systems focus on the mean delay (or time in system). However, system performance standards are often expressed in terms of tail probabilities rather the first moments. For L.L. Bean, a American telemarketing company, has a service standard that requires 85% of incoming calls to be answered in 20 seconds [Quinn, et al 1991]. Viewed from this perspective, the simulation output can be thought of as binary, with a given customer's delay expressed as a 1 if the delay exceeds the performance threshold and 0 otherwise. Thus, the performance evaluation requires estimation of proportion , not a mean.

If successive customers' delays were independent, it would be a simple matter to construct a confidence interval for the proportion with long delays. However, inference in queuing systems is complicated by autocorrelation, which usually inflates the variance of the performance estimate in a way that is difficult to measure.

One response to the problem of autocorrelation in simulation output is the method of independent replications, which yields a single summary of system performance for each simulation run and requires multiple runs. For non- terminating (steady-state) systems, independent replications are computationally inefficient, since one must discard the transient phase of each run before accumulating statistics. A more efficient approach to inference is the method of batch means [Fishman 1978, Law and Carson 1979]. This approach divides the observations in a single long run into "batches"; if the batches are sufficiently large, their means can be considered approximately independent, and inference proceeds using conventional methods. By discarding only a single transient period, the method of batch means saves computation, but there is significant additional effort involved in determining how large to make the batches. This effort involves testing the degree of correlation between successive batch means for various batch sizes. (See Whitt [1991] for a full treatment of the relative efficiencies of one long run versus independent replications in non-terminating simulations.)

Generally, there is more computation involved in simulating a customer than in analyzing that customer's performance datum, especially when simulating complex systems. As a result, we focus on output analysis using a single simulation run. Ultimately, with faster computers, it will become feasible to use single-run simulation to provide real-time or near real-time advice on system management. More efficient output analysis methods will hasten that day.

II. The Binary Bootstrap

The binary bootstrap was inspired by seminal work in three fields: in statistics, Efron's [1979, 1982] invention of the bootstrap method of inference; in probability, Kedem's [1980] analysis of time series "clipped" at some threshold to convert them to binary form; in simulation, Fishman and Moore's [1979] attention to inference when simulation outputs are binary.

Briefly, the bootstrap is a "resampling" technique that works by creating artificial replicates from a single original data set. Given a set of iid data values, one creates artificial replications by sampling the original data values with replacement. Given each "bootstrap replication", one computes a statistic of interest. Repeating this process generates the sampling distribution of the statistic. A thorough survey of bootstrap procedures for confidence intervals is provided by DiCiccio and Romano [1988]. Efron and Tibshirani [1991] explain the bootstrap procedure in descriptive way without mathematical development.

The conventional bootstrap method does not apply to time series data, such as the output of a simulation of a queuing system, because successive data values are not independent. Applications of the conventional bootstrap by Thoms and Schucany [1990] to time series data require that the data be modeled in such a way as to produce a set of iid residuals, as by creating an ARIMA model; the residuals are then independent and can be resampled. Other approaches to dependent data involve Künsch's [1989] and, independently, Liu and Singh's [1988] "moving blocks" bootstrap method, which constructs replicates by resampling partially overlapping blocks of observations. Politis and Romano [1989,1990] and Politis, Romano, and Lai [1990] generalized the moving blocks procedure by resampling "blocks of blocks" of observations. Politis and Romano [1991a] developed a new resampling technique, the "stationary" bootstrap method, which resamples blocks whose starting points are uniformly distributed on $\{1,\ldots,n\}$ and whose lengths are geometrically distributed, where n is the total number of data points. Our binary bootstrap differs from the moving block approach in that we let the data divide itself into "blocks" of random length, consisting of runs of 0's and 1's. Politis and Romano [1991b] also developed a "circular block-resampling" technique that amounts to "wrapping" the data around in a circle before blocking. They showed the consistency and asymptotic accuracy of the circular

block method by comparing it with the moving blocks method.

To give insight into the binary bootstrap, consider the nature of serially-correlated time series data. One way to explain why conventional inference does not apply is that the serial correlation changes the structure of runs in the data. This is seen most easily when the data are clipped to binary form. If successive binary data values were iid (i.e., Bernoulli trials), then there would be geometric distributions for the lengths of runs of 0's ("0- runs") and runs of 1's ("1-runs"). With positive autocorrelation, run lengths increase, thereby increasing the size of the stochastic excursions taken by the series away from its mean. The increase in run lengths results in a wider dispersion of sample realizations about the mean ("variance inflation"). Conversely, with negative autocorrelation, runs that bias the estimate of the mean tend to be self-reversing, so that large excursions from the mean are rare. The decrease in run length results in a narrower distribution about the mean ("variance deflation"). The binary bootstrap modifies the conventional bootstrap by regarding runs, rather than individual data values, as the sampling units. This preserves the correlation structure in the bootstrap replicates.

The steps in the binary bootstrap are:

Step 1: Clip the time series to binary form.

Step 2: Break the binary data into alternating sequences of 0-runs and
1-runs.

Step 3: a. Create bootstrap replicates by alternately sampling with replacement from the pools of 0-runs and 1-runs. Truncate the final run to insure that there are n data values in the replicate. Our empirical work suggests that, as with the conventional bootstrap, 500 replicates is an
appropriate number.

b. For each bootstrap replicate, compute the estimated probability of long delay.

Step 4: Analyze the set of bootstrap estimates as if they were independent replications.

To compute a 90 percent confidence interval for π, simply sort

the B values of $\{\hat{\pi}_i\}$, identify (or interpolate) the 5th and 95th percentiles, and use these values as the lower and upper limits of the confidence interval. This confidence interval procedure does not require any assumptions about the distribution of data values, such as Normality or even symmetry.

III. Empirical Results

We present empirical evidence for the value of the binary bootstrap. First, we show that the binary bootstrap works well for Bernoulli trials, which have no autocorrelation, and for binary first-order Markov processes. Second, we present evidence that sampling alternately from the pools of 0-runs and 1-runs, as described in Step 3a above, is justified even when the data are known to have a high positive autocorrelation. Finally, we demonstrate that the binary bootstrap performs better than batch means in drawing inferences from a single run simulating an M/M/1 queue.

1. Binary Bootstrap for Bernoulli Trials

To develop confidence in the binary bootstrap, and to determine a choice for B, the number of bootstrap replications, we first applied the binary bootstrap to Bernoulli trials, which are iid. If π = Prob[X_t=1], the sample average of n data values has mean π and standard deviation $[\pi(1-\pi)/n]^{.5}$.

We generated 50 binary sequences, each consisting of n=2,000 Bernoulli trials. Exhibit 1 shows that the binary bootstrap successfully estimated both the mean and the standard deviation of the distribution of the sample mean, even with as few as B=100 bootstrap replications. Thus, the binary bootstrap is consistent with classical methods in the special case of no serial correlation. In other words, in the case of Bernoulli trials, one has the choice of bootstrapping either the individual binary data values or the runs.

2. Binary Bootstrap for the First-order Markov Processes

To compare the performance of the binary bootstrap with asymptotic theoretical values for correlated data, we next considered binary data generated by a first-order Markov process. Suppose the data values $\{X_i\}$ are a realization of a first-order binary Markov process.

Define the two Markov parameters as $p \equiv Pr[X_i=1|X_{i-1}=0]$ and $q \equiv Pr[X_i=0|X_{i-1}=1]$. In this case,

$$\pi = \frac{p}{p+q}. \qquad (1)$$

Billingsley [1961] and Bedrick and Aragon [1989] developed the asymptotic variance of the estimator $\hat{\pi}$,

$$\lim_{n\to\infty} n\, Var(\hat{\pi}) = \pi(1-\pi)[1+2\frac{(1-p-q)}{p+q}] \qquad (2)$$

where the term in brackets represents the variance inflation factor (VIF) caused by the positive serial correlation.

To illustrate the application of the binary bootstrap to binary Markov data, we present the analysis of sample series in Exhibit 2. The series consisted of 100,000 observations, with values of π equal to .1. Note that the π's of the simulated data were not exactly equal to the nominal π values. The nominal values are calculated using equations (1)-(2) based on the specified p and q values from each data set. Exhibit 2 shows that the binary bootstrap estimates π and its deviation well.

3. Independence of Successive Run Lengths

Implicit in the binary bootstrap is the assumption that successive run lengths are independent. If this were not so, we would have to modify Step 3a, which alternately samples from the observed pools of 0-runs and 1-runs without regard to the value last sampled from the other pool.

To test the assumption of independence, we generated 50 samples of clipped delays (π=.1) from an M/M/1 queuing system with high utilization (ρ=.9). Each sample contained 100,000 steady-state values known to have high positive autocorrelation. Pooling the results of these 5,000,000 customers yielded a total of 25,464 pairs of successive runs (either a 0-run followed by a 1-run or vice versa). Despite the large sample size, analysis of contingency tables of successive run lengths confirmed independence, whether starting from a 0-run (Chi-square=42.9, df=35, p=.17) or a 1-run (Chi-square=30.0, df=28, p=.36).

4. Empirical Comparison of Binary Bootstrap and Batch Means

To demonstrate the value of the binary bootstrap in simulation output analysis, we compared it to the method of batch means in simulations of the M/M/1 queuing system, for which analytical results are known. Our primary concern was the actual coverage achieved by nominal 90 percent confidence intervals. A secondary concern was the half-width of correct confidence intervals. A tertiary concern was the stability of the half-widths.

We chose a difficult case to analyze: high server utilization (ρ=.9), so that the delay autocorrelation dissipated very slowly, and low probability of exceeding the delay threshold (π=.1), so that excedences were infrequent events and there were few 1-runs. We varied the number of simulated customers, using n=5,000, n=20,000, and n=100,000. In all cases, we deleted first 3,000 customers to allow the system to reach steady state. Our analyses are based on 50 trials at each of the three run lengths.

Exhibit 3 compares the measured coverage of nominal 90 percent confidence intervals for the proportion of long delays. The binary bootstrap provided better coverage than the method of batch means; as expected, coverage improved with run length for both methods. For run length n=5,000, neither method yielded adequate coverage; for n=20,000, only the binary bootstrap succeeded; for n=100,000, both methods provided adequate coverage, with a slight advantage to the binary bootstrap. Since 100,000 observations would be considered a short run for batch means analysis, it is noteworthy that the binary bootstrap performed well for as few as n=20,000 observations.

Exhibit 4 compares the half-widths of the confidence intervals for n=100,000, at which run length both methods provided valid intervals. The mean half-widths were approximately the same for both methods (paired t= -1.49, df=48, p=.14; a Normal probability plot verified the Normality assumption underlying the t-test). Inspection of Exhibit 4 also shows that the stability of the half-widths were approximately equal.

IV. Summary and Conclusions

We considered the problem of estimating the probability of long delay in a queuing system using a single run from a

discrete-event simulation. We introduced the binary bootstrap as a method for constructing valid confidence intervals for the probability.

The conventional bootstrap resamples individual data values and thereby destroys the autocorrelation structure in the data. In contrast, the binary bootstrap resamples 0-runs and 1-runs and thereby preserves the autocorrelation. We presented empirical evidence that the binary bootstrap performs well both for iid data (Bernoulli trial) and for data from a first-order Markov process. We also demonstrated the validity of the implicit assumption of independence between successive run lengths using data with a high degree of positive autocorrelation.

Our main goal was to compare the binary bootstrap with the method of batch means for estimating the probability of long delay in discrete-event simulation of queuing systems. Using data from a heavily-loaded M/M/1 queuing system, the binary bootstrap produced valid 90 percent confidence intervals with run lengths as small as 20,000 customers, for which the batch means method failed to generate valid intervals. At run lengths of 100,000 customers, both methods generated valid intervals with essentially equal mean half-widths and stability of half-widths. These results suggest the value of the binary bootstrap for simulation output analysis.

One of the motivations for our work was the problem of reducing the computational burden of discrete event simulation. While computer speed grows rapidly, it appears that the complexity of the systems we wish to simulate keeps pace, so that the problem of computational cost does not diminish. We advocated single replication methods of output analysis on the basis of their computational efficiency relative to the method of independent replications. Of the two methods of single-run analysis, the binary bootstrap has computational advantages worth mentioning. First, the steps in the binary bootstrap are quite simple and mechanical, whereas the determination of optimal batch size involves trial and error and repeated calculations of the lag one autocorrelation between batch means. Second, looking to the future, the binary bootstrap appears to be an algorithm inherently suited to parallel computation, since multiple processors can each work to generate and analyze a bootstrap replication of the single simulation run. This simplicity and suitability for parallel processing add to the appeal of the binary bootstrap.

There is much to do to follow up our initial empirical results. One would like to have: first, the kind of asymptotic theoretical results that are available for the conventional bootstrap; second, experience with queuing systems besides the M/M/1; third, exploration of the value of the binary bootstrap in other types of time-series inference, such as statistical process control, where one might create p-charts that allow for serial correlation between failures.

References

Bedrick, E. J. and J. Aragon, "Approximate Confidence Intervals for the Parameters of a Stationary Binary Markov Chain", *Technometrics* Vol 31:4, 1989, 437-448.

Billingsley, P., *Statistical Inference for Markov Processes*, University of Chicago Press, Chicago, 1961.

Billingsley, P., *Probability and Measure*, Wiley, New York, 1986.

Efron, B., "Bootstrap methods:another look at the Jackknife", *Ann. Statist.*, 7, 1979, 1-26.

Efron, B., *The Jackknife, the Bootstrap and Other Resampling Plans*, CBMS:NSF, Philadelphia,1982.

Efron, B. and R. Tibshirani, "Statistical Data Analysis in the Computer Age", *Science*, Vol 253, 1991, 390-395.

DiCiccio, T. J. and J. P. Romano, "A Review of Bootstrap Confidence Intervals", *J. R. Statist. Soc.* **B** Vol 50:3, 1988, 338-354.

Fishman, G. S., *Principles of Discrete Event Simulation*, John Wiley, New York, 1978.

Fishman, G. S. and L. R. Moore, "Estimating the Mean of a Correlated Binary Sequence with an Application to Discrete Event Simulation", *J. of ACM*, Vol 26, 1979, 82-94.

Kedem, B., *Binary Time Series*, Marcel Dekker, New York, 1980.

Künsch, H. R.,"The Jackknife and the Bootstrap for General Stationary Observations", *Ann. Statist.*, 17, 1989, 1217-1241.

Law, A. and J. S. Carson, "A Sequential Procedure for Determining the Length of Steady-state Simulation", *Oper. Res.*, Vol 29:6, 1979, 1011-1925.

Liu, R. Y., and K. Singh, "Moving Blocks Jackknife and Bootstrap Capture Weak Dependence", Unpublished manuscript, Department of Statistics, Rutgers University, 1988.

Quinn, P., B. Andrews, and H. Parsons, "Allocating Telecommunications Resources at L. L. Bean, Inc.", *Interfaces*, Vol 21:1 1991, 75-91.

Politis, D. N., J. P. Romano, and T. L. Lai, "Bootstrap Confidence Bands for Spectra and Cross-Spectra", Technical

Report #342, Department of Statistics, Stanford University, February, 1990.
Politis, D. N. and J. P. Romano, "The Stationary Bootstrap", Technical Report #91-03, Department of Statistics, Purdue University, January, 1991a.
Politis, D. N. and J. P. Romano, "A Circular Block-resampling Procedure for the Stationary Bootstrap", Technical Report #91-07, Department of Statistics, Purdue University, February, 1991b.
Thoms, L. A. and W. R. Schucany, "Bootstrap Prediction Intervals for Autoregression", *J. Amer. Statist. Assoc.*, 1990, 486-492.
Whitt, W., "The Efficiency of One Long Run versus Independent Replications in Steady-state Simulation", *Management Sci.*, Vol 37:6 1991, 645-666.

Exhibit 1: Performance of Binary Bootstrap on Bernoulli Trials

# of Replications	Estimate of Mean	Estimate of Std. Dev.[a]
100	0.09880 ± 0.01557[b]	0.00663 ± 0.00138[b]
500	0.09885 ± 0.01585	0.00673 ± 0.00104
1000	0.09889 ± 0.01599	0.00674 ± 0.00105
Theoretical	$0.1000 = \pi$	$0.00671 = \sqrt{\frac{\pi(1-\pi)}{2000}}$

[a] Based on averages over 50 series, each of length n=2,000
[b] 95% confidence intervals

Exhibit 2: Performance of Binary Bootstrap on First-order Markov Process

Statistics	Nominal Values[a]	Average Estimate	
		Empirical[b]	Binary Bootstrap[c]
π	0.1	0.097 ± 0.012	0.111 ± 0.008
SD(π)	0.041	0.045	0.040 ± 0.002
VIF	39.0	46.0	33.9 ± 3.2

[a]) Based on p = 0.005, q = 0.045, n = 2,000 observations
[b]) Based on 50 realizations with n = 2,000 observations
[c]) Based on B = 500 bootstrap replications of each realization; uncertainties are 95% confidence intervals

Exhibit 3: Coverage of Nominal 90% Confidence Intervals for Probability of Long Delay in M/M/1 Queue

	Estimated Coverage	
Run Length[a]	Batch Means	Binary Bootstrap[b]
5,000	50% ± 12%[c]	72% ± 11%[c]
20,000	72% ± 11%	86% ± 8%
100,000	84% ± 9%	86% ± 8%

[a] Include the first 3,000 transient obs., which were deleted.
[b] B=500 bootstrap replication
[c] Sampling uncertainties expressed as 90% confidence intervals for coverage probabilities, based on 50 simulation runs.

Exhibit 4: Half-widths of Nominal 90% Confidence Intervals