Resampling Technique for Simulation Output Analysis

Yun-Bae Kim*

Abstract

To estimate the probability of long delay in a queuing system using discrete-event simulation is studied. We contrast the coverage, half-width, and stability of confidence intervals constructed using two methods: batch means and new resampling technique; binary bootstrap. The binary bootstrap is an extension of the conventional bootstrap that resamples runs rather than data values. Empirical comparisons using known results for the M/M/1 and D/M/10 queues show the binary bootstrap superior to batch means for this problem.

(DISCRETE-EVENT SIMULATION; BOOTSTRAP; SIMULATION OUTPUT ANALYSIS; BINARY TIME SERIES; BATCH MEANS)

1. Introduction

We introduce the "binary bootstrap", a new approach to inference about the probability of long delay in a queuing system based on a single run of a discrete event simulation. The binary bootstrap has certain advantages over the conventional batch means method for constructing a confidence interval from a single simulation run.

Most simulation analyses of queuing-type systems focus on the mean delay (or time in system). However, system performance standards are often expressed in terms of tail probabilities rather than first moments. For instance, the L. L. Bean Company's service standard requires 85 percent of incoming calls to be answered in 20 seconds [16]. Thus this process is viewed as binary, taking the value 1 if the delay exceeds 20 seconds and 0 otherwise; as the performance evaluation requires estimation of a proportion, not a mean.

If successive customers' delays are 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.

A standard approach to inference is the method of batch

* Department of Mathematics, New Mexico Tech.

means [7, 14]. This approach divides the observations in a single long run into "batches"; if the batches are sufficiently large, their using classical 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, since the degree of correlation between successive batch means must be varied for various batch sizes.

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

2. The Binary Bootstrap

The binary bootstrap was inspired by seminal work in three fields: in statistics, Efron's [4, 5] invention of the bootstrap method of inference; in probability, Kedem's [9] analysis of binary time series; in simulation, Fishman and Moore's [8] attention to inference when simulation outputs are binary.

Briefly, the bootstrap is a "resampling" technique that works by creating artificial replicates from an 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 [6].

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. Thoms and Schucany [20] applied the bootstrap to time series using an ARIMA with residual. Künsch [13] and Liu and Singh [15] resampled partially overlapping blocks of observations. Politis, Romano, and Lai [17] generalized the moving blocks procedure by resampling "blocks of blocks" of observations. Politis and Romano [18] developed the "stationary" bootstrap method, which resamples blocks whose starting points are uniformly distributed on $\{1,\cdots,n\}$ and whose lengths are geometrically distributed, where n is the total number of data points. Politis and Romano [19] also developed a "circular block-resampling" technique that amounts to "wrapping" the data around in a circle before blocking. 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.

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 diverge from the mean tend to be self-reversing, so that large excursions away 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:

1: Clip the time series to binary form.

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

3: a. Create B 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

no more than the original 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.

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

Let π = Prob[long delay]. To compute a 90 percent confidence interval for π, simply sort the B bootstrap estimates $\{\pi_i^*\}$, identify (or interpolate) the 5th and 95th percentiles, and use these values as the lower and upper limits of the confidence interval. The length of the interval,

$$\textit{Estimated half-width} = (\pi_{.95B} - \pi_{.05B})/2.0 \qquad (1)$$

Where $\{\pi_{.95B}\}$ = $(0.95*B)^{th}$ value of the ordered $\{\pi_i^*\}$

$\{\pi_{.05B}\}$ = $(0.95*B)^{th}$ value of the ordered $\{\pi_i^*\}$,

is a dispersion functional of the empirical distribution function. This confidence interval procedure does not require any assumptions about the distribution of data values, such as Normality or even symmetry.

3. Empirical Results

First, 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. Second, we demonstrate that the binary bootstrap performs better than batch means in drawing inferences from single runs simulating M/M/1 and D/M/10 queues.

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. The assumption has been verified for the M/M/1 case Kim [11].

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 queue, for which analytical results are well known, and the D/M/10 queue, where results can be obtained by the numerical procedures, Kleinrock [12]. 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 selected a difficult case for analysis: with 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 therefore few 1-runs. We took three values for run length: n=5,000, n=20,000, and n=100,000. In the M/M/1 model, we deleted the first 3,000 customers to allow the system to reach steady state. In the D/M/10 model, we limited the sample collection to the steady-state. Our analyses were based on 50 trials at each of the three run lengths.

Exhibit 1 compares the measured coverage of nominal 90 percent confidence intervals for the proportion of long delays in the M/M/1 model. 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 observa-

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

Run Length[a]	Estimated Coverage	
	Batch Means	Binary Bootstrap[b]
5,000	50% ± 12%[c]	72% ± 11%[c]
20,000	72% ± 11%	86% ± 8%
100,000	84% ± 9%	88% ± 8%

[a] Includes the first 3,000 transient observations, which were deleted.

[b] B=500 bootstrap replications

[c] Sampling uncertainties expressed as 90% confidence intervals for coverage probabilities, based on 50 simulation runs.

tions.

Exhibit 2 compares the half-widths of the confidence intervals for n=100,000 customers in an M/M/1 queue, 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 2 also shows that the sbability of the half-widths were approximately equal.

Exhibit 3 is the analog of Exhibit 1 for the D/M/10 model. Again the binary bootstrap provided better coverage than batch means. For n=5,000 customers, neither method yielded acceptable coverage; for n=20,000 and n=100,000 customers, both methods provided adequate coverage. For the n=100,000 customers, the binary bootstrap performed slightly better than the batch means by all three criteria: coverage, accuracy, and stability.

Exhibit 3: Coverage of Nominal 90% Confidence Intervals for Probability of Long Delay in D/M/10 Queue

Run Length	Estimated Coverage	
	Batch Means	Binary Bootstrap[a]
5,000	68% ± 11%[b]	78% ± 10%[b]
20,000	80% ± 8%	88% ± 7%
100,000	86% ± 8%	88% ± 7%

[a] B=500 bootstrap replications

[b] Sampling uncertainties expressed as 90% confidence intervals for coverage probabilities, based on 50 simulation runs.

Exhibit 4 is the analog of Exhibit 2 for the D/M/10 model. Both methods succeeded in providing adequate coverage when run length n=

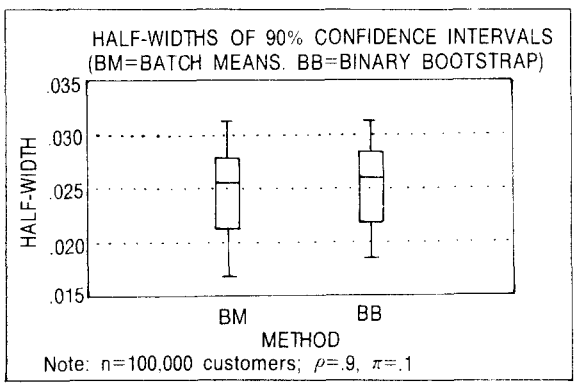

Exhibit 2: Half-widths of Nominal 90% Confidence Intervals for M/M/1 queue

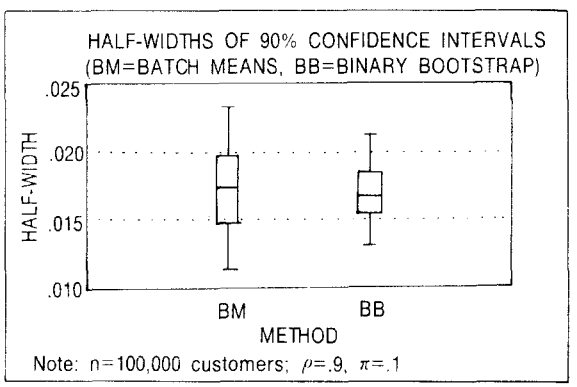

Exhibit 4: Half-widths of Nominal 90% Confidence Intervals for D/M/10 queue

100,000 customers. The mean half-widths were not significantly different (paired t=.706, df=49, p=.487; a Normal probability plot verified the Normality assumption underlying t-test). Inspection of Exhibit 4 shows that the half-widths produced by the binary bootstrap were more stable than those produced by batch means. The variance of half-widths from the binary bootstrap was 33% smaller than those from the batch means.

4. 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 compared confidence intervals produced by the binary bootstrap and batch means for M/M/1 and D/M/10 queues.

The conventional bootstrap resamples individual data values and thereby destroys the autocorrelation structure in the data. In contrast, the binary bootstrap resamples runs and thereby preserves the autocorrelation.

Our main goal was to compare the binary bootstrap with the method of batch means. 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. For run lengths of 100,000 customers, both methods generated valid intervals with essentially equal mean half-widths and stability of half-widths. In the D/M/10 case, for run length n=20,000, both methods generated valid intervals, with slightly better performance by the binary bootstrap. For run lengths of 100,000, both methods produced valid interval, but the binary bootstrap was superior to batch means in stability. For coverage and accuracy, both methods performed almost equally.

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 advoated single replication methods of output analysis on the basis of their computational efficiency relative to the method of independent replications. In single-run analysis, the binary bootstrap has two computational advantages. First, the steps in the binary bootstrap are mechanical and do not require the determination of optimal batch size involves trial and error and repeated calculations of the lag one autocorrelation between batch means. Second, the binary bootstrap algorithm is inherently suited to parallel computation, since multiple processors can each handle single simulation runs simultaneously. 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. First, one would like to explore 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. Second, one would like to extend the principles of the binary bootstrap to non-binary data, thus allowing for inference on conventional performance measures, such as means.

References

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

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

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

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

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

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

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

[8] 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.

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

[10] Kim, Y. B., J. Haddock, and T. R. Willemain, "The Binary Bootstrap from a Single Simulation Run", DSES Technical Report No. 37-92-304, RPI, Troy, New York, February, 1992.

[11] Kim, Y. B. "Single Replication Methods for Simulation Experiments", *Ph. D. dissertation*, Dept. of Engineering Science, Rensselaer Polytechnic Institute, Troy, New York, 1992.

[12] Kleinrock, L., *Queeuing Systems Volume 1: Theory*, John Wiley and Sons, New York, 1975.

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

[14] 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.

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

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

[17] 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.

[18] Politis, D. N. and J. P. Romano, "The Stationary Bootstrap", Technical Report #91-03, Department of Statistics, Purdue University, January, 1991a.

[19] 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.

[20] Thoms, L. A. and W. R. Schucany, "Bootstrap Prediction Intervals for Autoregression", *J. Amer. Statist. Assoc.*, 1990, 486-492.

저자소개

*

Yun-Bae Kim

Yun Bae Kim is Assistant Professor of Mathematics at New Mexico Tech., Socorro, NM 87801. He received Ph. D. from Rensselaer Polytechnic Institute in 1992. His current research interests are simulation output analysis, resampling techniques for correlated data, and modeling and analysis of manufacturing systems.