CS177 Supplementary Lecture Notes
Confidence Intervals via the Adaptive Batch Means Method

Mart Molle

March 2, 2016

1 Basic Batch Means Method

Recall from lecture that if $X_1, X_2, \ldots X_n$ is a sequence of n i.i.d. samples, then we can for a confidence inteval for the mean μ from *sample mean*, $\bar{X}(n)$, and *sample variance*, $s^2(n)$, in the usual way:

$$\bar{X}(n) = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{1}$$

$$s_X^2(n) = \frac{1}{n-1}\sum_{i=1}^{n}\left(X_i - \bar{X}(n)\right)^2 \tag{2}$$

$$\bar{X}(n) - t_{n-1,1-\alpha/2}\sqrt{s_X^2(n)/n} \leq \mu \leq \bar{X}(n) + t_{n-1,1-\alpha/2}\sqrt{s_X^2(n)/n} \leq \tag{3}$$

In practice, the sequence $\{X_i\}$ is generated by a programming technique called *batch means*, in which we partition one long simulaton experiment into a sequence of "batches" representing the measured behavior of the model over disjoint time periods. Thus, we might define X_1 to be result of measurments obtained from the *first interval* (from time 0 to time T, say), X_2 to be result of measurments obtained from the *second interval* (from time T to time $2T$, say), and so on. The problem with this approach is that the sequence $\{X_i\}$ will be identically distributed (assuming each batch represents the same interval length) but they will *not* be mutually independent because the initial state of the model for batch i is *defined* to be the same as the final state of the model for batch $i-1$. If the individual batch sizes are "large" compared with the time required for the model to reach "steady-state" (aka its long-term average behavior – think about how the value of knowing the initial conditions for the "trained flea" example decay over time) then this correlation at the boundary is not a problem. Thus the real question is: *how do you know that your chosen batch size is large enough?*

2 Adaptive Batch Means

Consider the following thought experiment.

> How would your results change if you modified your program to collect data for $n/2$ batches $Y_1, \ldots Y_{n/2}$, each representing the model behavior during disjoint intervals of length $2T$, instead of n batches representing disjoint intervals of length T?

Notice that this change in data collection has *zero* effect on the behavior of the model. All that has changed is the frequency with which you output a sample and reset the associated statistical counters. Thus, if each sample represents the *average* of some quantity from the model (e.g., average queue size, number of cars balking per hour) rather than a *total* value, then we see immediately that:

$$Y_i = \frac{X_{2i-1} + X_{2i}}{2} \tag{4}$$

and hence that the two sample means are identical:

$$\bar{Y}(n/2) = \frac{1}{n/2}\sum_{i=1}^{n/2} Y_i = \frac{1}{n/2}\sum_{i=1}^{n/2} \frac{X_{2i-1}+X_{2i}}{2} = \frac{1}{n}\sum_{i=1}^{n} X_i = \bar{X}(n) \tag{5}$$

The calculation of the sample variance is a lot messier, but eventually leads to an interesting result. We have:

$$s_Y^2(n/2) = \frac{1}{n/2-1}\sum_{i=1}^{n/2}\big(Y_i - \bar{Y}(n/2)\big)^2 = \frac{1}{n/2-1}\sum_{i=1}^{n/2}\big(Y_i - \bar{X}(n)\big)^2 \tag{6}$$

The trick for continuing the derivation is to expand the square inside the rightmost sum from Eq.6:

$$\big(Y_i - \bar{X}(n)\big)^2 = \left(\frac{X_{2i-1}+X_{2i}}{2} - \bar{X}(n)\right)^2 = \frac{1}{4}\Big(\big(X_{2i-1} - \bar{X}(n)\big) + \big(X_{2i} - \bar{X}(n)\big)\Big)^2 \tag{7}$$

Using the identity that $(A+B)^2 \equiv A^2 + 2AB + B^2$, Eq.7 can be rewritten into the form:

$$\big(Y_i - \bar{X}(n)\big)^2 = \frac{1}{4}\Big(\big(X_{2i-1} - \bar{X}(n)\big)^2 + 2\big(X_{2i-1} - \bar{X}(n)\big)\big(X_{2i} - \bar{X}(n)\big) + \big(X_{2i} - \bar{X}(n)\big)^2\Big) \tag{8}$$

Substituting Eq.8 into Eq.6 we obtain after some simplifications:

$$s_Y^2(n/2) = \frac{1}{2(n-2)}\sum_{i=1}^{n}\big(X_i - \bar{X}(n)\big)^2 \; + \; \frac{1}{n-2}\sum_{i=1}^{n/2}\big(X_{2i-1} - \bar{X}(n)\big)\big(X_{2i} - \bar{X}(n)\big) \tag{9}$$

It is interesting to compare Eq.9 with Eq.2. First, let us ignore the "off-by-one" discrepancy in the denominator terms of $n-1$ versus $n-2$, since the difference is insignificant as long as n is not too small. Next, we see that the first term in Eq.9 is essentially the same as $s_X^2(n)/2$. Finally, we recognize that the second term in Eq.9 represents the *covariance* between X_{2i-1} and X_{2i}. We consider two (extreme) cases.

- If the batch sizes are large enough, then X_{2i-1} and X_{2i} will be independent, and the covariance term drops to zero. In this case, if we form a confidence interval from $\{Y_i\}$ using Eq.3, then (i) the sample variance drops by half and (ii) the number of samples also drops by half, so the expression inside the square root sign is unchanged, and the final answer only differs by the amount by which the t distribution changes due to the reduction in the number of degrees of freedom. (If n is large enough for us to not care about the difference between $n-1$ and $n-2$, then surely t won't change much either!) Thus to summarize: *if the batch size is large enough, pairwise aggregation of the data shouldn't change the confidence interval.*

- If the batch size is too small, then X_{2i-1} and X_{2i} will be positively correlated. In the most extreme case, they will be indentical, in which case, the right-hand sum in Eq.9 looks like Eq.2 with half the terms missing. Since in this extreme situation, the missing half of the terms are identical to the terms we did include, the value of the right-hand sum in Eq.9 would become $s_X^2(n)/2$ – which is the same as the first term, so that the final result becomes $s_Y^2(n/2) \approx x_X^2(n)$. Substituting this result for the sample variance into Eq.3 shows that the confidence interval formed from $\{Y_i\}$ should be $\sqrt{2}$ time larger than the confidence interval formed from $\{X_i\}$. To summarize: *if the batch size is too small, then pairwise aggregation of the data will cause the size of the confidence interval to increase signficantly.*

Based on these results, we are led to the following *adaptive batch means algorithm*:

1. Choose an initial interval size, T, and number of batches, n
2. Run your program long enough to generate n batches of size T for sample values $X_1, ..., X_n$, then pause to calculate some confidence intervals.
3. Aggregate the n samples of X to produce $n/2$ samples of Y, and calculate the confidence intervals for both sequences.
4. If CI_X is small to meet your requirements, and $CI_X \approx CI_Y$ then stop.
5. Relabel samples $Y_1, \ldots, Y_{n/2}$ to become samples $X_1, \ldots, X_{n/2}$ and double the batch size T. Resume execution of your model for $n/2$ additional batches of the new size, and loop back to step 3.