Auto-regressive Adjusted Batch Means Estimator

BSE399A - Undergraduate Project (UGP)

Submitted By
Lakshay Rastogi
180378, BSBE

Submitted To
Dr. Dootika Vats (MTH)
Dr. Appu Singh (BSBE)

Introduction

Suppose we have a probability distribution π with support X and we want to calculate $E_\pi g := \int_X g(x)\pi(dx)$ where g is real-valued, π integrable function. In a lot of situations, the π is sufficiently complex, that such an integration is inefficient to compute directly. In these cases we employ Markov Chain Monte Carlo (MCMC) methods to estimate Eπ g. So suppose we generate a markov chain drawing samples X from π where X = {X1, X2, X3..} where Xi $\in$ R then

$$\bar{g}_n = \frac{1}{n}\sum_{i=1}^{n} g(X_i) \to E_\pi g \text{ as } n \to \infty.$$

if we get a large number of samples then $\bar{g}_n$ is a sufficient approximation of $E_\pi g$.

However since it is an approximation we need to provide a measure of its quality for which we employ certain statistical metrics.

One such important metric is the variance of the asymptotic distribution of the Monte Carlo Standard Error (MCSE) which is $\bar{g}_n - E_\pi g$ .This variance is available via a Markov Chain Central Limit Theorem (CLT) provided there exists a constant $\sigma_g^2 \in (0, \infty)$ such that

$$\sqrt{n}(\bar{g}_n - E_\pi g) \xrightarrow{d} N(0, \sigma_g^2) \text{ as } n \to \infty.$$

In this project we propose a new estimator for evaluating σ_g^2 by incorporating the concepts from batch means and auto-regressive processes. According to this method, once we have drawn the samples, we batch the samples together and calculate their batch means. Using these batch means as samples we fit an auto regressive process of order 1. Finally we use the formulae available for the CLT variance of such auto regressive processes of order 1 to calculate σ_g^2

Batch Means Estimator

The Batch Means Estimator is very popular method used to evaluate the asymptotic Monte Carlo variance for the MCSE. This estimator in this project acts as a metric to assess our own estimator. The steps involved in calculating the value of the batch means estimator, is to first divide the samples into batches, then calculate the mean of each of the batches which are referred to as batch means and then estimate the variance in the CLT by calculating the variance in these batch means, and suitably scaling the value.

For univariate data, Let us suppose we have a Markov Chain : $X_1, X_2, X_3 \ldots, X_n$ where $X_i \in R$. Define $Y_k = \frac{1}{b} \sum_{i=1}^{b} X_{kb+i}$ for k = 0, 1, 2.., a – 1. The batch means estimator is defined as

$$\hat{\sigma}^2_{BM} = \frac{b}{a-1} \sum_{k=0}^{a-1} \left(Y_k - \hat{\mu}_n\right)^2$$

Proposed Estimator

AR(1) Process

Since our estimator is based on concepts from an auto-regressive process it is important to give some introduction about them. We will introduce an AR(1) process.

An AR(1) where AR stands for auto-regressive is characterized by a way in which the sequential samples are linked :

$$X_{n+1} = \rho X_n + \epsilon_n$$

where $\epsilon_n \sim N(0, \alpha^2)$ are IID. In such a process, the distribution of the first sample X_1 is important. We assume it to be from a distribution having finite variance, called a stationary distribution. A result that we get from the above relation is

$$\mathrm{Cov}(X_{n+k}, X_n) = \rho^k \mathrm{Var}(X_n)$$

If the process is stationary then :

$$\begin{aligned} \mathrm{Var}(X_n) &= \mathrm{Var}(X_{n+1}) \\ &= \rho^2 \mathrm{Var}(X_n) + \mathrm{Var}(\epsilon_n) \\ &= \frac{\alpha^2}{1-\rho^2} \text{ with } \rho^2 < 1. \end{aligned}$$

Using the Markov Chain CLT, and using the results stated here we get :

$$\sigma^2_{MC} = \frac{\alpha^2}{1-\rho^2} \left(\frac{1+\rho}{1-\rho}\right)$$

Motivation and Method

Although batch means is used so frequently, there are some issues with this estimator :

- It always underestimates the actual value of the asymptotic monte carlo variance, and it doesn't work well with a small batch size, which is preferred when we don't have a lot of original samples available in order to generate more batch means.
- Also in a lot of situations, there exists significant correlation among the samples for whom we have to calculate the asymptotic monte carlo variance. Using the batch means estimator in these cases would not enable us to take advantage of this correlation present among the samples.

Hence we want to work on an estimator that takes advantage of this correlation among the samples present, works well with low batch sizes, and also gives closer estimates to the actual value when compares to the batch means estimates.

Suppose we have a set of samples : $X_1, X_2, X_3, \ldots X_n$. The first step is to calculate the batch means for the given samples. The batch size is denoted by b, and number of batches by a. We define $Y_i = X_i - \bar{g}_n$ and then define $\overline{Y_i} = \sum_{k=1}^{b} Y_{ib+k}$ to be the batch means for the samples. Now we fit a stationary AR(1) process, where $\epsilon_t \sim N(0, \alpha^2)$, to these batch means such that

$$\overline{Y_t} = \rho \cdot \overline{Y}_{t-1} + \epsilon_t$$

Motivation and Method

According to the Markov Chain CLT with the batch means obtained:

$$
\begin{aligned}
\sqrt{a}(\bar{Y} - \mu) &\xrightarrow{d} N(0, \sigma^2_{MC}) \\
\sqrt{a}(\bar{g}_n - E_\pi g) &\xrightarrow{d} N(0, \sigma^2_{MC}) \\
\mathrm{Var}(\bar{g}_n - E_\pi g) &\approx \frac{\sigma^2_{MC}}{a}
\end{aligned}
$$

According to the Markov Chain CLT on the original samples obtained :

$$
\begin{aligned}
\sqrt{n}(\bar{g}_n - E_\pi g) &\xrightarrow{d} N(0, \sigma^2_g) \\
\mathrm{Var}(\bar{g}_n - E_\pi g) &\approx \frac{\sigma^2_{MC}}{n}.
\end{aligned}
$$

Combining the above two expression we have

$$
\hat{\sigma}^2_g = b \cdot \frac{\hat{\alpha}^2}{(1-\hat{\rho})^2}
$$

Where we obtain the estimates for α and ρ by using the **ar** function from the **stats** package in R. This function defaults to the yule-walker method of estimating α and ρ, however we could also choose to generate the mle estimates for the same. This function while fitting an AR(1) process also provides us with another option of using the Akaike Information Criterion(AIC). This criterion if set to true, chooses the order of the model to which the data best confirms to (although this order is always less than the order to which we want the function to fit the samples to), otherwise the model of provided order is fitted.

Examples

In order to study the performance of the estimator in difference scenarios, we are going compare the asymptotic Monte Carlo variance estimated by our estimator and the batch means estimator under different scenarios wherein we will change the way in which we generate the samples on which this analysis is done.

The general strategy to be followed for most examples is

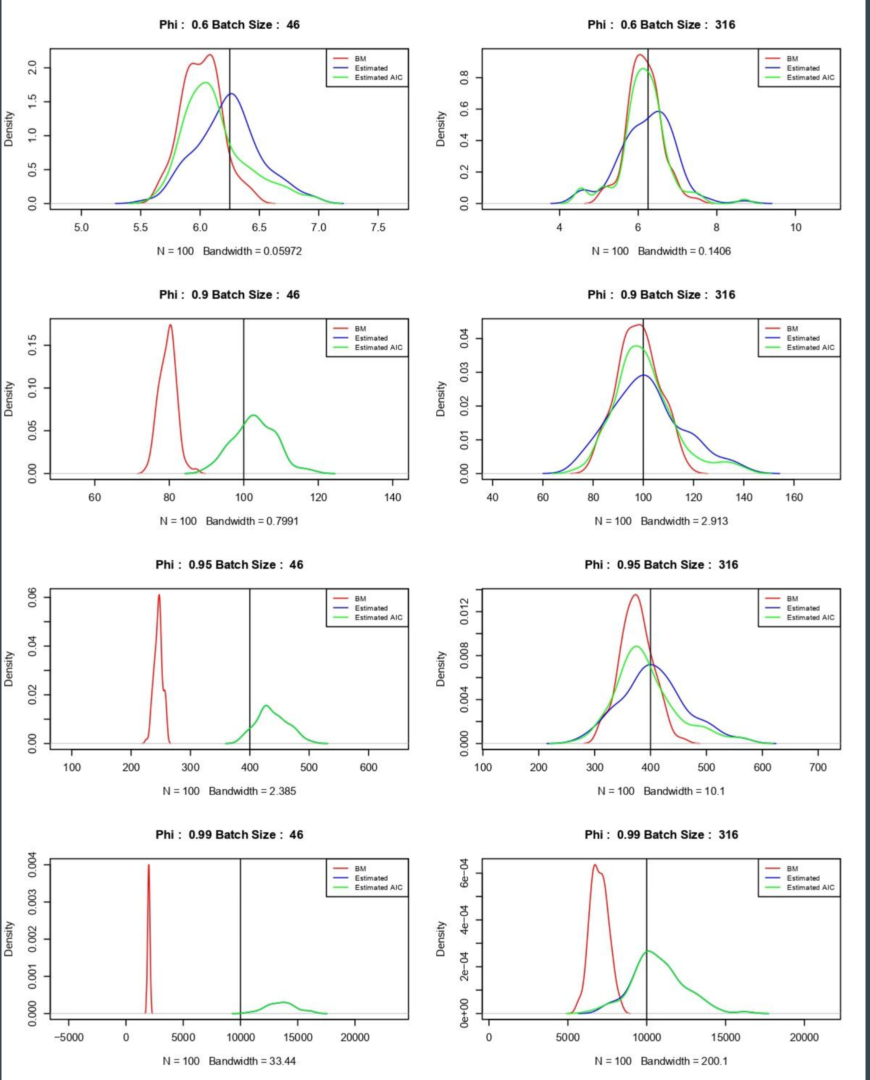
1. We are going to run **iter** number of iterations.
2. In each iteration we are going to generate **T** number of samples.
3. In each simulation we are going to apply our estimator and the batch means estimator with two batch sizes : $3\sqrt{T}$ and $\sqrt{T}$. In the univariate case we are going to make a density plot of the estimated obtained by both the estimators and then compare their means as well their variances. In these examples we could vary the batch size in cases where we need to assess the behavior of our estimator on changing batch sizes.

Furthermore, we will also test our estimator on samples that have been generated from an autoregressive process, thus in these cases we will be able to compare our estimates against the true value of the variance of the asymptotic normal distribution of the MCSE.

Samples from an AR(1) Process

We generate AR(1) samples with $\phi = \{0.60, 0.90, 0.95, 0.99\}$, and $\alpha^2 = 1$ where ϕ is the correlation coefficient for the AR(1) process and α is the variance of the normal distribution that introduces some randomness in each of the generated samples.

For this simulation, T = 1e5, iter = 100 and the batch sizes as described in the previous slide. In one iteration the AR model is fit twice, once while keeping the AIC criterion to be true and once while keeping the AIC criterion to be false. Since in this case we are fitting the AR(1) model to the data generated from an AR(1) process, we are able to calculate the true value of the asymptomatic Monte Carlo variance and compare it with the ones estimated by the different estimators.



Inference

- An increase in the batch size leads to our estimator behaving like the batch means estimator when the ϕ is not very high. This could be because as we increase the batch size, the averaging effect caused by the increase in the samples used to calculate the batch means leads to a reduction in the correlation among the batch means. Such a reduction invalidates the assumption that we use to build our estimator (correlation present among samples), therefore it behaves very similar to the batch means estimator.
- This reduction in correlation is further verified by the fact that the graphs for when AIC = TRUE and AIC = FALSE, also differ, indicating that there are some cases where in when we try to fit the batch means to an AR(1) process we get a negative ρ, which means no correlation exists between the batch means.
- In cases where there exists significant correlation among the samples, with low batch size our estimator behaves considerably better than the batch means estimator, although it over estimates in some cases, however still it seen to be closer to the actual value than the batch means estimator. As the correlation increases, our estimator better estimates the asymptotic Monte Carlo variance, however in case of really high correlation it tends to overestimate the value of the variance.
- The variance for the values generated by our estimator is higher than the variance for the values generated by the Batch Means Estimator.

Samples from a Bayesian Logistic Regression Model

Bayesian Logistic Regression Model

In the above examples we saw the behaviour of our estimator with samples with significant to low correlation. Now we would like to test our estimator with MCMC data. In order to generate data that does not inherently have any bias we use a typical example which is used as introduction to MCMC sampling, that is Bayesian Logistic Regression. Consider a Bayesian Logistic Regression model where for i = 1,2..,n, where n is the number of samples, we have $x_i = (1, x_{i2}, ...x_{i(p-1)})^T$ be the vector of covariates for the ith observation such that x_i in R^p and for this model let □ in R^p be the corresponding vector of regression coefficients. A realization of a response Y_i would then be

$$Y_i|x_i, \beta \sim \text{Bern}(p_i) \text{ where } p_i = \frac{\exp(x_i^T\beta)}{1+\exp(x_i^T\beta)}$$

Now for this model we need to generate the distribution of β. Given the prior distribution to be $N_p(0, I_p)$, the posterior comes out to be

$$\pi(\beta|y) \quad \propto \pi(\beta)\prod_{i=1}^{n} f(y_i, \beta) \quad \propto e^{\frac{-\beta^T\beta}{2}}\prod_{i=1}^{n}(p_i)^{y_i}(1-p_i)^{1-y_i}$$

To generate the distribution we will draw samples from this posterior, and for that we use the MH algorithm. Here, we use a multivariate normal distribution as the proposal distribution where in the covariance matrix is set to be a diagonal matrix with a single value on all the diagonals, this value here being the step size, and the mean of the distribution to be the previous accepted value of □.

The data obtained for the simulation is from the titanic data set that is present online. The titanic data frame describe the survival status of individual passengers on the Titanic. The principal source for data about Titanic passengers is the Encyclopedia Titanica. The variables on this extracted dataset are pclass, survived, name, age, siblings, parch, room, ticket, fare, cabin, sex and embarkment. Out of these we use survived, sex(male), age, siblings, parch and fare for our analysis.

We don't have the actual value of the asymptotic Monte Carlo variance in this case, as was the case in the examples where we generated samples from an auto regressive process, thus in order to judge the performance of our estimator we would use the property of consistency of the Batch Means Estimator.

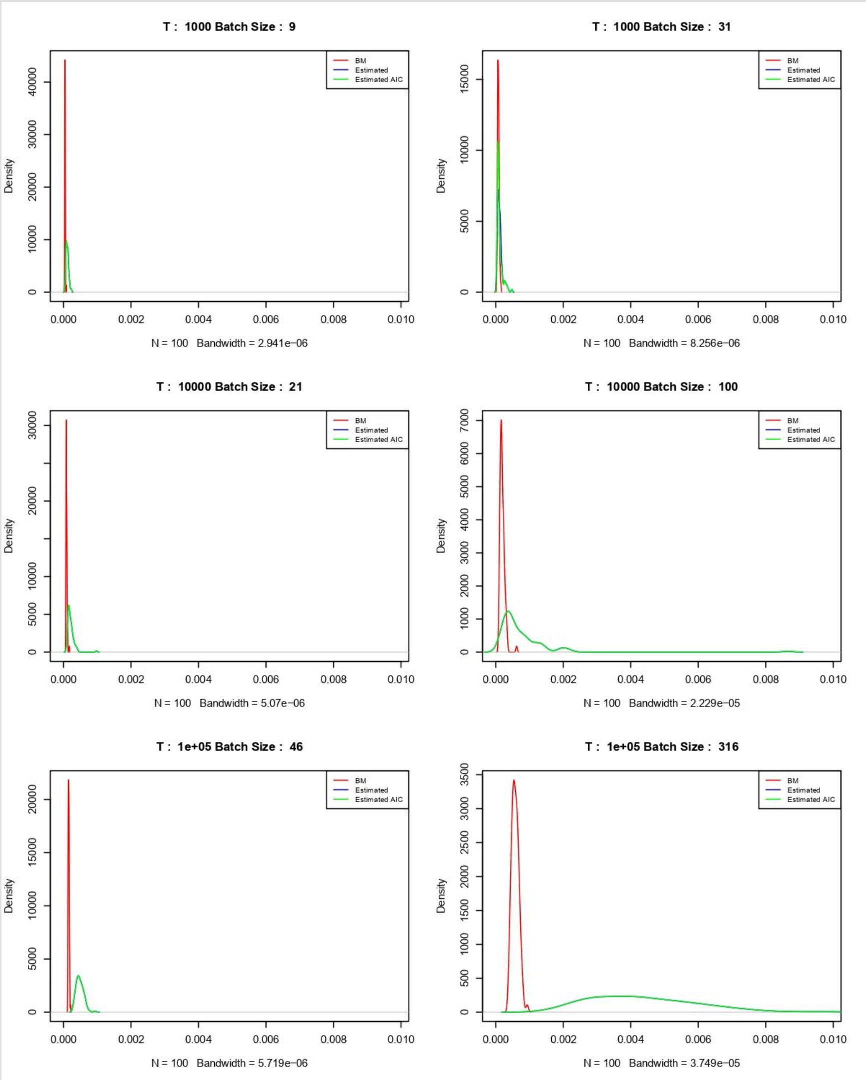
According to this property, if the batch size, and the number of batches increase with the number of samples (e.g. by setting $a = b = n^{1/2}$) then $\sigma_{BM}^2 \rightarrow \sigma_g^2$ with probability one as $n \rightarrow \infty$. Which essentially means that the value of σ_{BM}^2 moves towards the actual value σ_g^2 as the number of samples increase. Thus on increasing the sample size we should see the batch means estimator and our own estimator moving in the same direction.

Analysis with univariate data

In this case we use a single component of the samples generated to work with on our estimator. Here we run 100 iteration for different values of T = {1000, 10000, 100000} and step size of 0.005.

It can be seen that the pattern our estimator follows is similar to the pattern the batch means estimator also follows as the sample size increases. . Therefore the fact that our estimator produces estimates of increasing magnitude with increasing sample size is in sync with the consistency property of the batch means estimator. Looking at the mean and the variance we can also say that :

- with an increase in the sample size the value of the mean increases for both the estimators.
- for a particular sample size, for a high batch size the variance in the estimates made by our estimator dramatically increases. Thus indicating their might be an optimal value of the batch size (being a function of the sample size) which helps produce good results



Remark - Estimator for multivariate data

Even though in the presentation, we have solely focused on the estimator that works on univariate data. Our project also involved developing a similar estimator for multivariate data. The intuition and the derivation for such an estimator is pretty similar to the univariate case, and is detailed in the report. I will briefly describe the problem set up as well as the inferences drawn in the case of multivariate data.

For multivariate data we fit a VAR(1) - Vector Autoregressive process to the the data, $y_{n+1} = \Phi y_n + \epsilon_n$

where Φ is a $p \times p$ matrix, $\epsilon_n \sim N_p(0, W)$, where y_0 is a zero vector. Then for estimating the Σ (covariance matrix) in the asymptotic multivariate normal distribution in the CLT for the MCSE we have :

$$\Sigma = (1 - \Phi)^{-1} V + V(1 - \Phi^T)^{-1} - V$$

Where $vec(V) = I_{p2} - (\Phi \otimes \Phi)^{-1} vec(W)$. Here $\otimes$ denotes the kronecker product. For fitting this we use the **vars** package in R. An important thing to note here is that the Σ matrix is an invertible matrix, and its determinant can be computed. In the multivariate case in order to compare the value we use the determinant of the covariance matrices estimated. Some basic inferences are :

- The batch means underestimates the value of the variance, our estimator over estimates but is closer to the actual value when there is significant correlation. When working on data with no correlation both perform similar.
- It shows results which are in sync with the consistency property of the batch means estimator.

Conclusion

We with our proposed estimator aim to overcome these drawbacks of the batch means estimator. To do so we draw inspiration from the autoregressive processes of order 1, AR(1) in case of univariate data and VAR(1) in case of multivariate data. With the examples shown above we are able highlight certain aspects of out estimator :

- our estimator works very similar to the batch means estimator in case of the data being not very highly correlated, therefore in no way is the batch means estimator performing better than our estimator.
- our estimator overestimates the value of asymptotic monte carlo variance in case of highly correlated data with a low batch size, which is still closer to the actual value of the variance when compared to the batch means estimate. When we use a higher batch size in the same case, the estimated value appears very close to the actual value.
- our estimator show behaviour similar to the batch means estimator (according to the consistency property of the batch means estimator) when it shifts towards the right (increases in magnitude) with an increase in the number of samples generated.
- in the specific case of working with 2multi-variate data our estimator produces an estimate of Σ which is invertible and thus it's determinant can be computed, this is not the case for other estimator that work with multivariate data.

To conclude in this project we have tried to design a new estimator for estimating the variance of the asymptotic normal distribution of the MCSE for a collection of samples by taking into consideration the correlation that may be present among those samples, in order to give an estimate which is considerably better that the batch means estimate and can be computed as easily.

Acknowledgement

I would like to express my gratitude to Professor Dr. Dootika Vats (MTH Department) for providing me with this opportunity to work on an undergraduate project and being patient with me while working on this project. It was a very valuable learning experience. I learnt about how research happens in the mathematical and statistical community, as well as what sort of problem do they work on and went in depth to learn more about the topic of my UGP.

I would also like to thank Dr. Appu Singh (BSBE Department) for agreeing to be the co-mentor from the BSBE department for this project.

Best,

Lakshay Rastogi

BSBE, Double Major in CSE

180378