Undergraduate Project Report

BSE399A

Auto-regressive Adjusted Batch Means Estimator

Submitted By:
Lakshay Rastogi
180378
BSBE IIT Kanpur

Submitted to:
Dr. Dootika Vats (MTH)
Dr. Appu K. Singh (BSBE)

April 22, 2022

DECLARATION

I, Lakshay Rastogi, solemnly declare that the project report **Auto-regressive Adjusted Batch Means Estimator** is based on my own work carried out during the course of our study under the supervision of **Dr. Dootika Vats**. I assert the statements made and conclusions drawn are an outcome of my research work. I further certify that

1. the work contained in the report is original and has been done by me under the general supervision of my supervisor.

2. we have followed the guidelines provided by the university in writing the report.

3. whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them in the text of the report and giving their details in the references.

April 22, 2022

Auto-regressive Adjusted Batch Means Estimator

Contents

1 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 $E_\pi g$. So suppose we generate a markov chain drawing samples X from π where $X = \{X_1, X_2, X_3..\}$ where $X_i \in R$ then,

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

Thus 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

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

Obtaining a good estimate say $\hat{\sigma_g}^2$, is important specifically for two reasons: (1) It can be used to calculate the asymptotically valid confidence intervals for $E_\pi g$ and (2) is a key component of the rigorous rules used to decide when to terminate the simulation. In case of $\hat{\sigma_g}^2$ being a consistent estimator of σ_g^2, a valid MCSE of $\hat{g}_n$ is given by $\hat{\sigma_g}/\sqrt{n}$.
A simple way to estimate the $\hat{\sigma_g}^2$ is to calculate $\text{Var}_\pi g$ using the samples that we have obtained, but due to the inherent serial correlation in the Markov Chain, this estimate would not be correct. Rather we would have to calculate

$$\begin{aligned}\sigma_g^2 &= \sum_{k\ =\ -\infty}^{\infty} \text{Cov}(X_{n+k},\ X_n) \\ &= \text{Var}(X_n)\ +\ 2\sum_{k\ =\ 1}^{\infty} \text{Cov}(X_{n+k},\ X_n).\end{aligned}$$

Easier ways to estimate the same have been proposed including non-overlapping batch means(BM), overlapping batch means (OBM), spectral variance (SV) methods, and regenerative simulation (RS).
In this report we propose another 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 $\hat{\sigma}_g^2$.
Even though in the most of the report we refer to the concepts using notation for univariate data, in the project we have also worked on a similar estimator for multivariate data. The intuition and the motivation for such an estimator remains the same, and the notation will be described later in the report.

2 Batch Means Estimator

In order to assess the estimates made by our proposed estimator we use batch means estimator to simultaneously calculate the value of the asymptotic Monte Carlo variance. 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 CLT by calculating the variance in these batch means, and suitably scaling the value.

2.1 Batch Means Estimator for Univariate Data

Let us suppose we have a Markov Chain : $X_1, X_2, X_3..., X_n$ 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 :

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

where $\hat{\mu}_n = \frac{1}{n}\sum_{i\,=\,1}^{n} X_i$. Also $\hat{\sigma}^2_{BM}$ is an estimate of the variance for the asymptotic normal distribution $N(0, {\sigma_p}^2)$ in the CLT for MCSE.

2.2 Batch Means Estimator for Multivariate Data

Let us suppose we have a Markov Chain : $X_1, X_2, X_3, ...X_n$ where $X_i \in \mathrm{R}^p$. The definition for Y_k remains the same, wherein $Y_k = \frac{1}{b}\sum_{i\,=\,1}^{b} X_{kb\,+\,i}$ for $k = 0, 1, 2..a-1$, except now $Y_k \in \mathrm{R}^p$. The batch means estimator is defined as :

$$\hat{\Sigma}_{BM} = \frac{b}{a-1}\sum_{k=0}^{a-1}(\bar{Y_k} - \hat{\theta})(\bar{Y_k} - \hat{\theta})^T.$$

where $\hat{\theta} = \frac{1}{n}\sum_{i\,=\,1}^{n} X_i$. Also $\hat{\Sigma}_{BM}$ is the estimate of the co-variance matrix for the asymptotic multivariate normal distribution $N(0, \Sigma)$ in the CLT for MCSE.

3 Proposed Estimator

To understand how our estimator works, we will give some introduction about order 1 auto-regressive processes. When it consists of univariate data, we call it an AR(1) process, and when it consists of multivariate data we call it a VAR(1) process.

3.1 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, where in such a distribution is called a stationary distribution. A result that we get from the above relation is

$$\begin{aligned}\mathrm{Cov}(X_{n+k}, X_n) &= \rho\ \mathrm{Cov}(X_{n+k-1}, X_n)\\ &= \rho^{k-1}\ \mathrm{Cov}(X_{n-1}, X_n)\\ &= \rho^k\ \mathrm{Var}(X_n)\end{aligned}$$

If the process is stationary, then

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

Now this Markov Chain obeys the CLT :

$$\begin{aligned}
\sigma^2_{MC} &= \text{Var}(X_n) \ + \ 2 \sum_{k \ = \ 1}^{\infty} \text{Cov}(X_{n+k}, \ X_n) \\
&= \text{Var}(X_n) \ + \ 2 \left(\sum_{k \ = \ 1}^{\infty} \ \rho^k \text{Var}(X_n) \right) \\
&= \ \text{Var}(X_n) \left(1 \ + \ 2 \sum_{k \ = \ 1}^{\infty} \ \rho^k \right) \\
&= \ \text{Var}(X_n) \left(1 \ + \ \frac{2\rho}{1-\rho} \right) \\
&= \ \text{Var}(X_n) \left(\frac{1+\rho}{1-\rho} \right) \\
&= \frac{\alpha^2}{1-\rho^2} \ \left(\frac{1+\rho}{1-\rho} \right)
\end{aligned}$$

So now, with n as the number of samples, $X_1, X_2, X_3 ... X_n$ form an AR(1) process, thus also form a Markov Chain, with $\bar{g}_n = \frac{1}{n} \sum_{i=1}^{n} X_i$ such that :

$$\sqrt{n}(\bar{g}_n - E_\pi g) \xrightarrow{d} N(0, \sigma^2_{MC}).$$

3.2 VAR(1) Process

A VAR(1) process stands for a vector autoregressive process of order 1 and is too characterized by a correlation among the generated samples, which in this case are vectors :

$$y_{n+1} = \Phi y_n + \epsilon_n$$

where $y_n \in R^p \ \forall n$, p being the dimension of the space in which the vectors are defined. Φ is a $p \times p$ matrix, $\epsilon_n \sim N_p(0, W)$, where y_0 is a zero vector and W is the covariance matrix to be used by the multivariate normal distribution as the covariance matrix to generate values of ϵ_n. We also require the absolute value of the largest eigenvalue of Φ to be less than 1 which case the stationary distribution for the process is $F = N_p(0, V)$ where $\text{vec}(V) = I_{p^2} - (\Phi \otimes \Phi)^{-1}\text{vec}(W)$. Here $\otimes$ represents the Kronecker product and I_{p^2} is the $p^2 \times p^2$ identity matrix. For the above set up a CLT holds with

$$\begin{aligned}
\Sigma = & \sum_{s=-\infty}^{\infty} \gamma(s) \\
&= \sum_{s=0}^{\infty} \gamma(s) + \sum_{s=-\infty}^{0} \gamma(s) - V \\
&= \sum_{s=0}^{\infty} \Phi^s V + \sum_{s=-\infty}^{0} V(\Phi^T)^s - V \\
&= (1-\Phi)^{-1}V + V(1-\Phi^T)^{-1} - V
\end{aligned}$$

An important thing to note here is that the Σ matrix is an invertible matrix, and its determinant can be computed.

3.3 Motivation and Method

Motivation

The batch means estimator, although considered to be one of the most frequently used, has been proven to always under estimate the value of the variance. Furthermore, batch means estimator doesn't work well with a small batch size. Also in a lot of situations, there exists significant correlation among the samples for whom we have to calculate the variance in the MCSE. Using the batch means estimator in these cases would mean to disregard this correlation present among the samples and treat them as IID samples. Our motivation to develop a new estimator for estimating the variance in the MCSE was two fold, (1) to take advantage of the correlation present in the data and give a better estimate than the batch means estimator for such data, (2) to develop an estimator that works with small batch sizes.

Method

We will first talk about the univariate case, and then briefly explain about the mutlivariate case.

Suppose we have a set of samples : $X_1, X_2, X_3, ...X_n$ where $X_i \in R$. For such a sample our goal is to estimate the asymptotic Monte Carlo Variance. That is σ_g^2 as defined below, where $\bar{g}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$:

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

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 to these batch means, where in $\bar{Y}_1$ is the first sample, assumed to be sampled from a distribution with finite variance. Let this AR(1) process be defined as follows :

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

where $\epsilon_t \sim N(0, \alpha^2)$.

According to the Markov Chain CLT, with $\bar{Y} = \frac{1}{a}\sum_{i=1}^{a} \bar{Y}_i = \bar{g}_n$ and $\mu = E(Y_i)$ we have:

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

Applying the CLT to the original samples:

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

Using the two expression above, we get:

$$\begin{aligned}
\sigma_g^2 &\approx \frac{n}{a} \cdot \sigma_{MC}^2\\
&\approx b \cdot \sigma_{MC}^2\\
&\approx b \cdot \frac{\alpha^2}{(1-\rho)^2}.
\end{aligned}$$

This is the expression for our estimator for univariate data. Here b is known, however α, ρ have to be estimated. They are estimated using the **ar** function from the **stats** package. 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. In the latter case, we may get some obscure values for ρ since there may arise a case where we try to fit an AR(1) process to nearly iid samples. Suppose the estimates obtained by fitting the AR(1) process to the given data are $\hat{\alpha}$ and $\hat{\rho}$, then :

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

For multivariate data, the calculation of the batch means follows the same procedure, only the expression for the covariance matrix changes to account for the batch size which is b. For multivariate data we use the **vars** package to fit a VAR(1) process to the data available. Suppose $\hat{\Phi}$ and $\hat{W}$ are the estimated values of Φ and W respectively. Then :

$$\begin{aligned}\text{vec}(\hat{V}) &= I_{p^2} - (\hat{\Phi} \otimes \hat{\Phi})^{-1}\text{vec}(\hat{W}) \\ \hat{\Sigma} &= b * \left((1-\hat{\Phi})^{-1}\hat{V} + \hat{V}(1-\hat{\Phi}^T)^{-1} - \hat{V}\right).\end{aligned}$$

4 Examples

How will we judge our estimator's performance?

In order to judge the estimates made by our estimator we will compare its estimates with the estimates generated by the batch means estimator. Furthermore in some cases we will testing 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.

What results are we expecting and why?

A major assumption that our estimator builds upon is the assumption that the batch means i.e. the Y_i's form an AR(1)/VAR(1) process. So if in a particular scenario this is not the case, then we don't expect the proposed estimator to perform better than the batch means estimator. Rather it is expected to be better than the batch means estimator in cases where there is significant correlation among the samples and when we work with small batch sizes, since that would lead provide some basis to our assumption.

What sort of examples are we going to see?

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 the analysis is done.
The general strategy that is going to be followed in most of the examples below 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 : $\sqrt[3]{T}$ and $\sqrt{T}$. In the univariate case we are going to make a density plot of the estimated obtained by both the estimators and in the multivariate case we are going to use the determinant of the Σ matrix obtained by both the estimator to make the density plots.

4.1 Generating samples from an AR(1) process.

Here we are going to generate AR(1) samples on which we are going to compare the performance of our proposed estimator and the batch means estimator.

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 as seen in the expression for an AR(1) process.

For this simulation T = $1e5$, iter = 100 and the batch sizes as described above. 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.

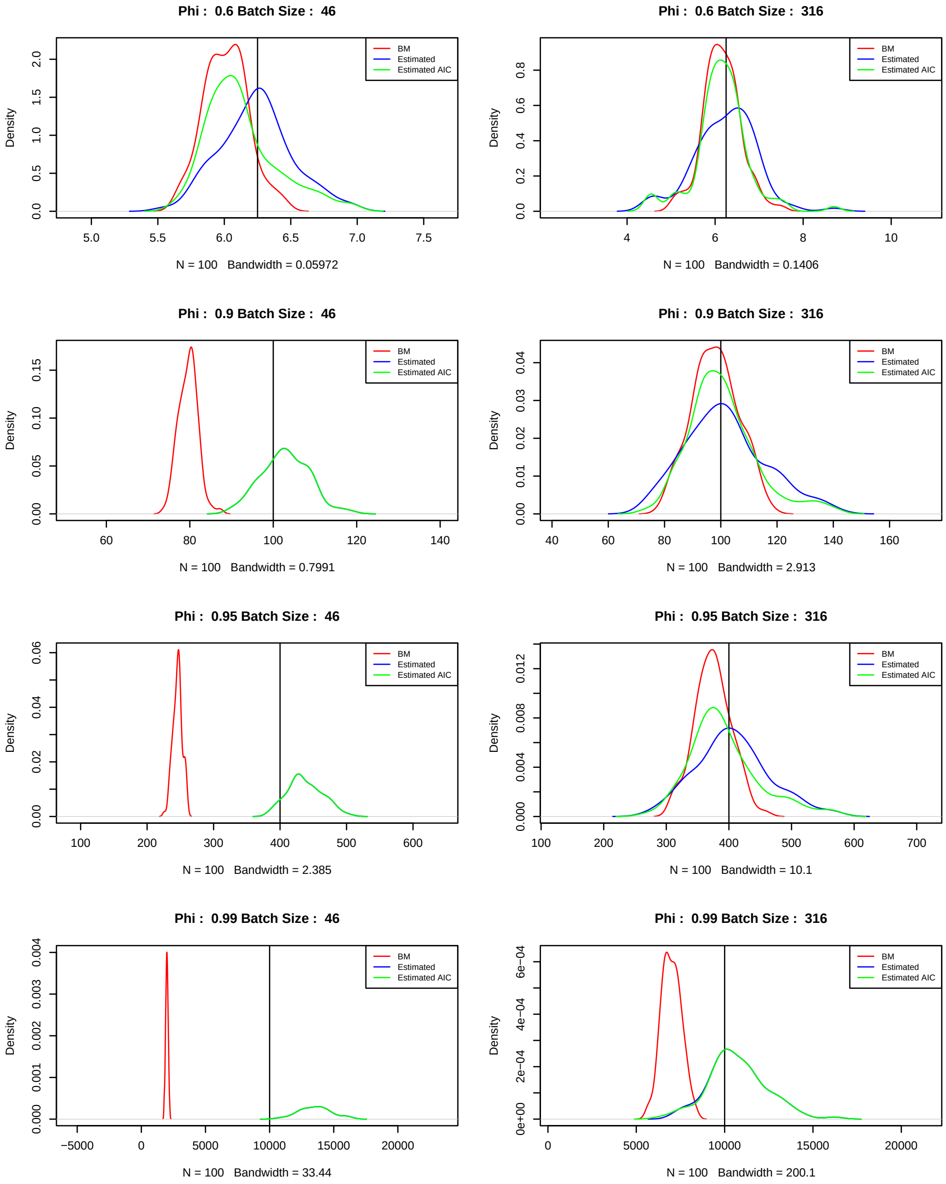

Figure 1: Density Plot for the estimates by Batch Means Estimator, our estimator (with AIC = True, and AIC = False).

There are certain trends that can be inferred from Figure 1:

- An increase in the batch size leads to our estimator behaving like the batch means estimator when the ϕ is not very high. The reason behind this trend could be the fact that 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 and 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, and the batch size is low which enables good correlation to exist among the batch means as well, our estimator behaves considerably better than the batch means estimator, although it over estimates the variance in some cases.

If we look at the values of the mean and variance for the asymptotic Monte Carlo variance being estimated in Figure 1, we can say that :

- The variance for the values generated by our estimator is higher than the variance for the values generated by 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 over estimate the value of the variance.

4.2 Generating samples from a VAR(1) process

In this example, we are going to generate VAR(1) samples on which we are going to compare the performance of batch means estimator and our proposed estimator. We generate VAR(1) samples, say x_i where $x_i \in R^p$ where $p = 5$, with different values of Φ and $W = 0.3 * diag(p)$. In order to judge the difference in the estimated value and the original value we also plot the relative frobenius error obtained by using the estimated Σ matrix and the generated Σ matrix.

For this simulation we set $T = 1e5$, $iter = 100$ and the batch sizes as described earlier. We generate a Φ matrix using an Singular Value Decomposition (SVD).

According to the SVD : $M = UAV^T$ where M is real, U and V are orthogonal matrices consisting of the eigenvectors corresponding to the eigen values that make up the A which is a diagonal matrix with eigenvalues as the diagonal elements. We use a special case of the SVD decomposition wherein we set A to be a Positive Semi Definite(PSD) matrix with non-negative real eigen values being less than 1. (A condition that was described when describing the estimator for the VAR(1) process).

Φ with large eigen values

$$\begin{bmatrix} 0.99 & 0 & 0 & 0 & 0 \\ 0 & 0.95 & 0 & 0 & 0 \\ 0 & 0 & 0.93 & 0 & 0 \\ 0 & 0 & 0 & 0.92 & 0 \\ 0 & 0 & 0 & 0 & 0.9 \end{bmatrix}$$

With large eigen values, while generating the markov chain we will obtain samples that show really high correlation with each other. With samples with high correlation our estimator should perform reasonably better than batch means estimator for both with small and large batch sizes.

From Figure 2 we can draw some considerable inferences :

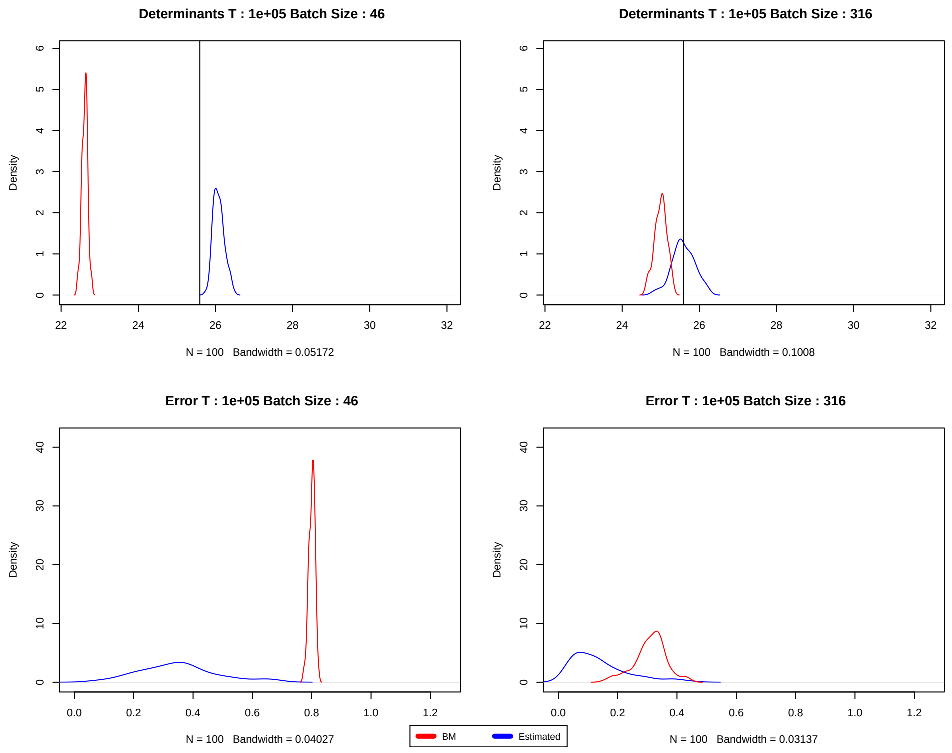

Figure 2: Density Plots for the Determinant of Σ for Φ with large eigen values, and the Relative Frobenius Error.

- it is clear that the batch means estimator under estimates the value of the variance, while the estimates made by our estimator are significantly larger than the batch means estimate and closer to the original value.
- a larger batch size generates batch means with low correlation among them and thus the estimates of the batch means estimator and our own estimator show considerable similarity. This is in sync with the results seen in AR(1) case as well.
- the error plots indicate that our estimator generates estimates that are closer to the true value for both batch sizes when compared to the batch means estimates, when the samples have significant correlation among them.

Φ with small eigen values

$$\begin{bmatrix} 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0.4 & 0 & 0 & 0 \\ 0 & 0 & 0.3 & 0 & 0 \\ 0 & 0 & 0 & 0.2 & 0 \\ 0 & 0 & 0 & 0 & 0.1 \end{bmatrix}$$

With low eigen values, while generating the markov chain we will obtain samples that do not show much correlation with each other, and can be thought to be almost iid. With samples with such low correlation we don't expect our estimator to perform better than the batch means with a higher batch size, however a better performance can be expected in case of a small batch size.

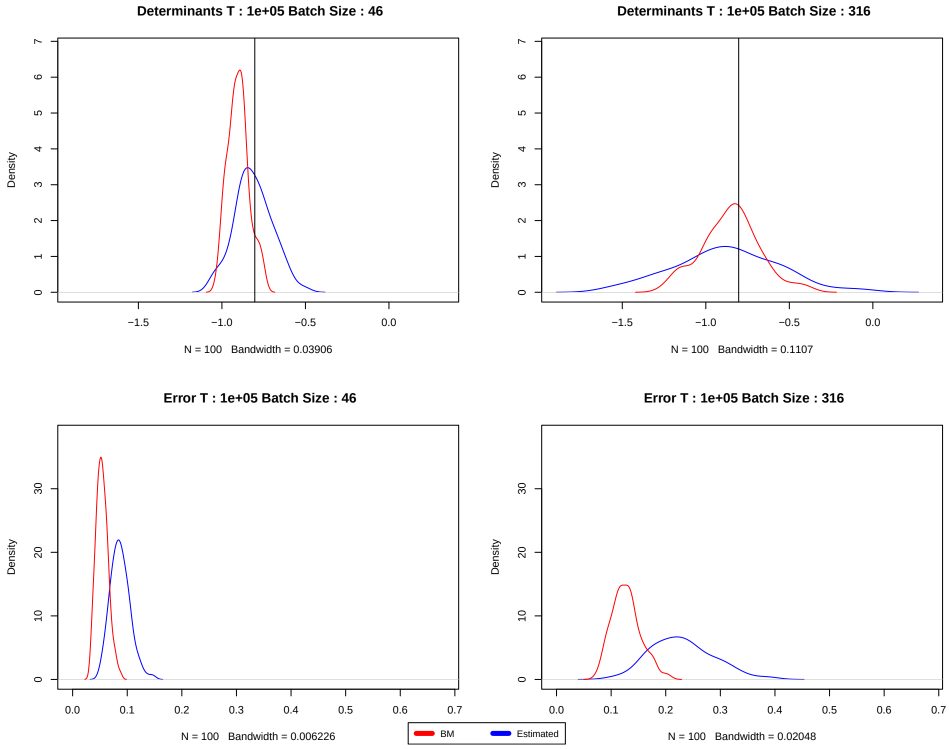

Figure 3: Density Plots for the Determinant of Σ for Φ with small eigen values, and the Relative Frobenius Error.

From Figure 3 we can draw some considerable inferences :

- with low correlation the density plots of the estimate made by both the estimators appear considerably similar in terms of their position. However the variance of the estimates made by our estimator is significantly higher than those made by the batch means estimator.
- when comparing the two batch sizes, our estimator performs better than the batch means estimator when using a low batch size, which is sync with what was expected.

Φ with mixed eigen values

$$\begin{bmatrix} 0.95 & 0 & 0 & 0 & 0 \\ 0 & 0.90 & 0 & 0 & 0 \\ 0 & 0 & 0.50 & 0 & 0 \\ 0 & 0 & 0 & 0.30 & 0 \\ 0 & 0 & 0 & 0 & 0.10 \end{bmatrix}$$

With mixed eigen values, while generating the markov chain we will obtain samples wherein some components would show high correlation with each other and others would show low correlation with each other.

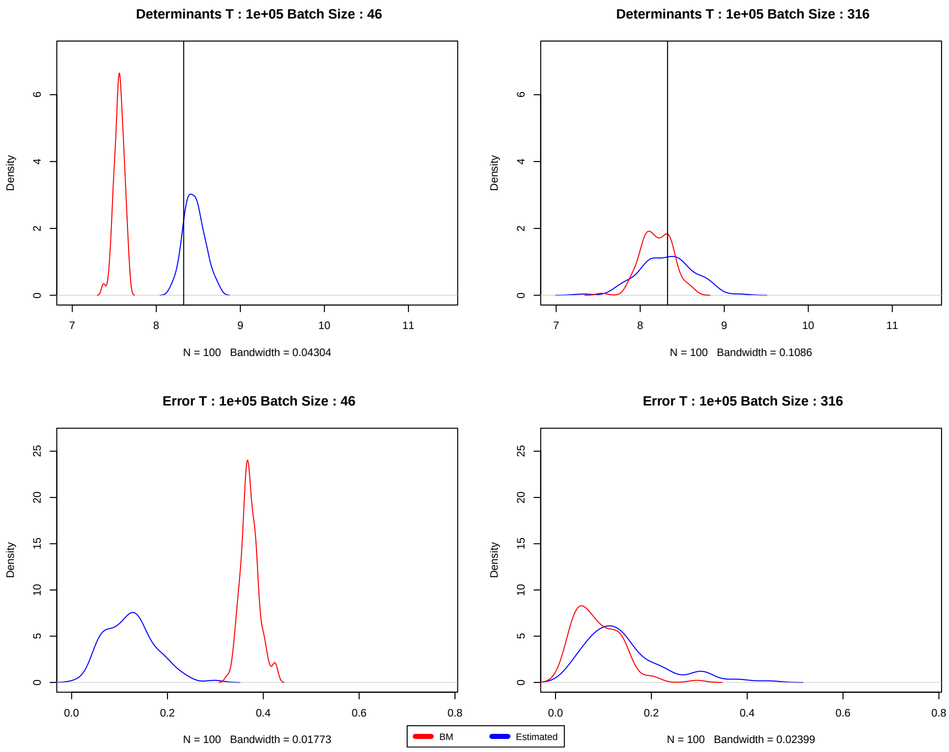

Figure 4: Density Plots for the Determinant of Σ for Φ with mixed eigen values, and the Relative Frobenius Error.

From Figure 4 we can draw some considerable inferences :

- with a low batch size the batch means show significant correlation among them, which leads to our estimator performing better than the batch means estimator. However with a large batch size, the batch means are not able to capture the correlation that the samples have and this leads to our estimator performing pretty similar to the batch means estimator.

4.3 Sampling 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

$$x_i = (1, x_{i2}, x_{i3}, x_{i4}, ...x_{i(p-1)})^T$$

be the vector of covariates for the ith observation such that $x_i \in R^p$and for this model let β 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

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

To generate the distribution we will draw samples from the 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 being the step size, and the mean to be the previous accepted value of β. Using the above set up we generate the MCMC data on which we then apply our own estimator and the batch means estimator.

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 $\sigma^2_{BM} \to \sigma^2_g$ with probability one as $n \to \infty$. Which essentially means that the value of σ^2_{BM} moves towards the actual value σ^2_g 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.

Hence we run 3 rounds of iterations, each with iter = 100 and with a different value of $T = 1e3, 1e4, 1e5$ with step size for the proposal distribution to be 0.005.

Working with univariate sample data

In this case we use a single component of the samples generated to work with on our estimator.
From Figure 5 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 means 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 which helps produce good results which might be a function of the sample size.

The variance in the graph too helps provide evidence for the fact that an optimal batch size for batch means as well as our estimator must exist because the variance in the MCMC variance jumps drastically, when the sample size increases, and the batch size remains constant.

Working with multivariate sample data

In this example we use all the components of the generated samples ($p = 6$) and we fit a VAR(1) model to the data to estimate the asymptotic monte carlo variance in the MCSE using the batch means estimator and our own estimator. While interpreting the above data it is necessary to note that we plot the

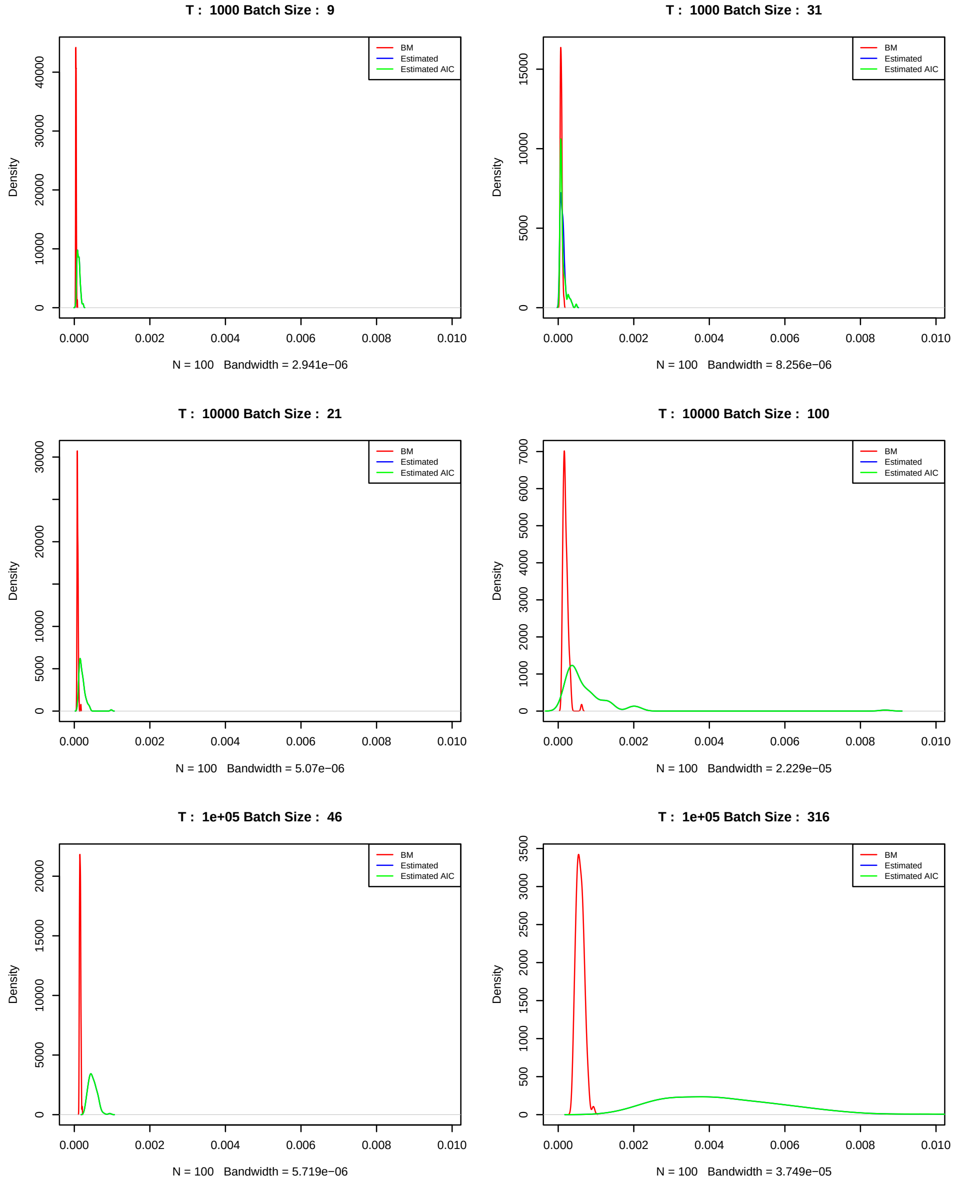

Figure 5: Density Plot of the logarithm of the determinant of the Σ matrix obtained by the batch means estimator, our estimator (AIC = FALSE and AIC = TRUE)

logarithm of the determinant of the Σ matrix on the x-axis and thus it might result in negative values. Again since in this case there is no way to actually evaluate the correct value of the MCSE variance for

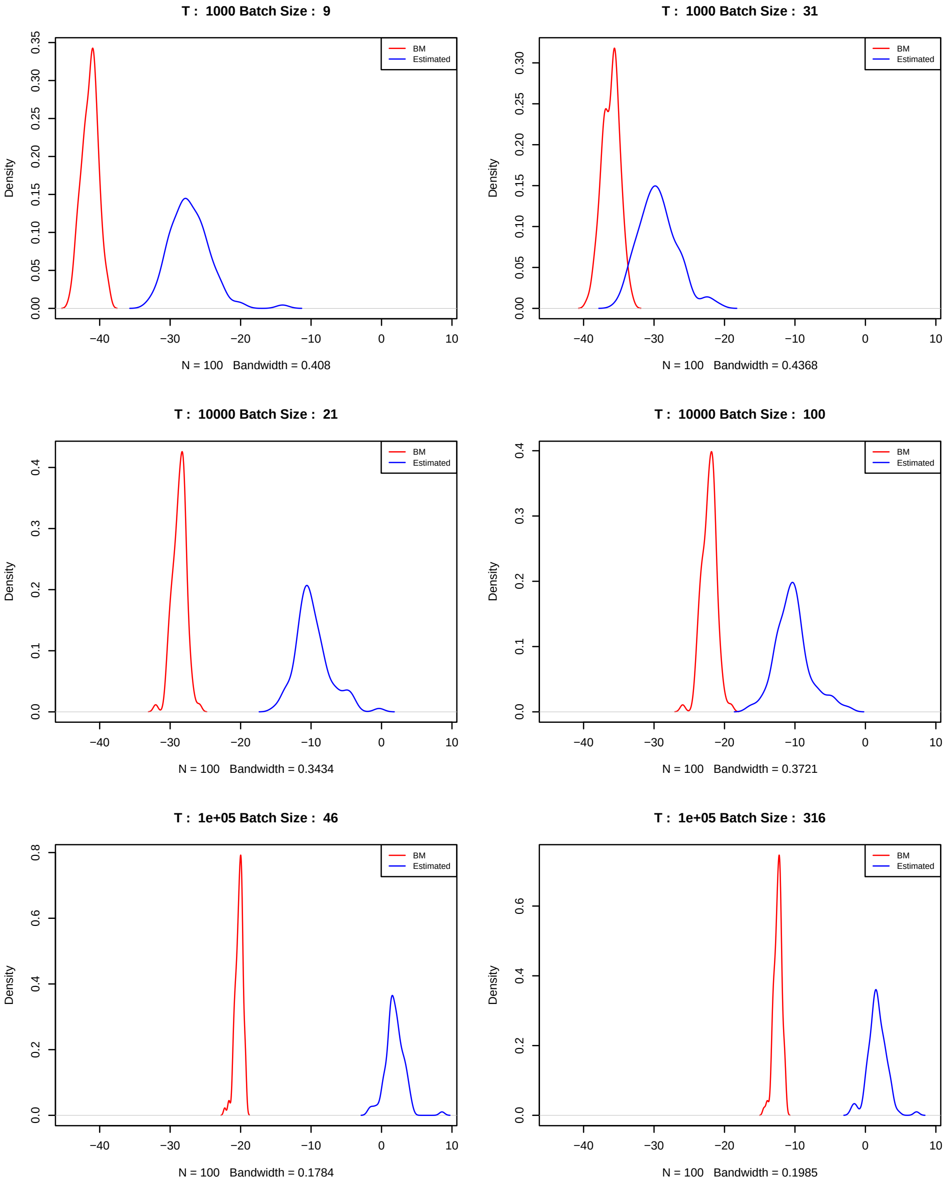

Figure 6: Density Plot of the logarithm of the determinant of the Σ matrix obtained by the batch means estimator, our estimator (AIC = FALSE and AIC = TRUE)

the samples generated, we follow the same approach that we did for the univariate case, i.e., we vary the samples size $= T = 1e3, 1e4, 1e5$. There are a considerable number of inference that can be drawn from the above graphs :

- As the batch size increases, the graphs for the bathc means estimates and estimates made by our estimator move closer to each other. This can be explained by the averaging out of the correlation among the samples on creating the batch means. The larger the batch size, the more loss of correlation occurs, and thus the estimates of our own estimator and the batch means estimator appear to come closer.
- As we increase the sample size, the batch means estimates move forward, which is in sync with the

property of consistency of the batch means estimator. But what is interesting is that the estimates made by our estimator also move in the same direction as the batch means estimate thus indicating that our estimator always captures a value greater than the batch means estimator for most sample sizes, and in most cases this would result in an over estimation of the actual variance.

- The distribution of the estimates generated by our estimator though is pretty similar for both batch sizes for a particular sample size.

5 Conclusion

With the examples above, we have highlighted the problems with the batch means estimator: that it underestimates the value of the asymptotic monte carlo variance and that it doesn't work well with correlated data, or a low batch size. We with our proposed estimator aim to overcome these drawbacks of the batch means estimator. To do so we draw inspiration from the auto-regressive 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 our estimator :

- our estimator works very similar to the batch means estimator in case of the data not being very highly correlated, therefore our estimator performs similar to the batch means in the common case.
- 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 shows 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 multivariate data our estimator produces an estimate of Σ say $\hat{\Sigma}$ which is invertible and thus it's determinant can be computed, this is not the case with 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.

6 References

1. James M. Flegal. Galin L. Jones. "Batch means and spectral variance estimators in Markov chain Monte Carlo." Ann. Statist. 38 (2) 1034 - 1070, April 2010. https://doi.org/10.1214/09-AOS735

2. Introduction to Markov Chain Monte Carlo : Chapter 1, By Charles J. Geyer

3. Dootika Vats, James M. Flegal, Galin L. Jones , "Strong Consistency of Multivariate Spectral Variance Estimators". https://doi.org/10.48550/arXiv.1507.08266