Auto-regressive Adjusted Batch Means Estimator

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 from π $X = \{X_1, X_2, X_3..\}$ then,

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

But in order to judge the quality of our markov chain we have to employ statistical metrics. One such important metric is Monte Carlo Standard Error (MCSE). It is usually based on the estimate of the variance of the asymptotic normal distribution for the markov chain. Therefore it is important to estimate this variance in the CLT of the markov chain, as this variance provided us information about the MCSE whcih inturn informs us about the quality of our simulation. The approximate sampling distribution of the Monte Carlo Error, $\hat{g}_n - E_\pi g$, is available via a Markov Chain CLT if there exists a constant $\sigma_g^2 \in (0, \infty)$ such that

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

What we want to estimate is σ_g^2. 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 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 Monte Carlo standard error (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 $\text{Var}_\pi g$, 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 AR(1) processes. To explain the process briefly, once we have drawn the samples, we batch the samples together, and then fit the batch means of these batches to an AR(1) process. Once we have fit the AR(1) process, we use the formula for the CLT variance of an AR(1) process to calculate $\hat{\sigma}_g^2$.

2. Batch Means Estimator

In order to assess the estimates made by our estimator we use batch means method to simultaneously calculate the value of σ_g^2. The methodology of the batch means estimator, is to first divide the samples into batches, calculate individual means of each of the batches, referred to as batch means and then estimate the variance in CLT, by calculating the variance in these batch means, and suitably scaling this 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 the variance in the CLT for MCSE, i.e. the statistic we are trying to estimate.

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 :

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

where $\hat{\theta} = \frac{1}{n}\sum_{i\ =\ 1}^{n} X_i$. Also $\hat{\Sigma}_{BM}$ is the co-variance matrix in the multivariate CLT for MCSE, i.e. the statistic we are trying to estimate in the multivariate case.

3. Proposed Method

To understand how our method helps, we would first have to learn about AR(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 ϵ_n are IID and drawn from $N(0, \alpha^2)$. In such a process, the distribution of the first sample X_1 is important. We assume it to be from any distribution having finite variance. 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

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

and since variances are non-negative, we must have $\rho^2 < 1$.
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, n be the number of samples, where $X_1, X_2, X_3 ... X_n$ form an AR(1) process, and thus form a Markov Chain, with $\bar{g}_n = \frac{1}{n} \sum_{i=1}^{n} X_i$.

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

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 samples, which in this case are vectors, are generated. Let

$$y_t = \Phi y_{t-1} + \epsilon_t$$

where $y_t \in R^p$ for all t, Φ is a pp matrix, $\epsilon_t \sim N_p(0, W)$, where y_0 is a zero vector and W is the covariance matrix to be used for the normal distribution generating the randomness in every term.
There is an assumption that the largest eigenvalue of Φ is less than 1 in the absolute value in which the case the stationary distribution for the process is $F = N_p(0, V)$ where $vec(V) = I_{p^2} - (\Phi \otimes \Phi)^{-1} vec(W)$. 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

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

3.3. Motivation and Method

Motivation

It is known that given some data, the batch means estimator always under estimates the value of the variance in the MCSE for that data considerably. Furthermore, batch means estimator doesnt work well when the batch size is kept small. Our motivation to develop a new estimator for estimating the variance in the MCSE was to take advantage of the correlation present in the data, as well as develop an estimator that works well with a low batch size.

Method

We have a set of samples : $X_1, X_2, X_3, ...X_n$. For such a sample our goal is to estimate the variance in the CLT for the MCSE. That is σ_g^2 as defined below, along with $\bar{g}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$:

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

Our first step is to calculate the batch means for the given samples, where in we have sample size of b, and number of samples to be a. So we define $Y_i = X_i - \bar{g}_n$ and then define $\overline{Y_i} = \sum_{k=\ 1}^{b} Y_{ib+k}$.
Now we assume these batch means to form a stationary AR(1) process, where in $\overline{Y_1}$ is the first sample, say having sampled from a distribution with finite variance. Also say this AR(1) process is 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, we have the following where $\bar{Y} = \frac{1}{a}\sum_{i=1}^{a} \bar{Y}_i = \bar{g}_n$ and $\mu = E(Y_i)$:

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

Looking at the CLT for the samples obtained :

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

Combining the variance expressions :

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

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 use the mle estimation method for the same. There is another option of using the Akaike Information Criterion while fitting the AR(1) process. This criterion if set to true, uses the AIC criterion to choose the order of the model to which the data best confirms to, otherwise the model of order.max is fitted. 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 remains the same, only the expression for the variance changes to account for the batch size which is b. For multivariate data we use the vars package to fit the VAR(1) process to the data available. Suppose $\hat{\Phi}$ and $\hat{W}$ are the estimated values of Φ and W respectively for the data present. Then :

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

4. Examples

How will we judge our estimator's performance?

Since we have established the form of our estimator, in order to judge its estimate we would have to use an already well established estimator which is used to estimate the MCSE CLT variance. We use Batch Means Estimator as described earlier to do the same.

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_is form an AR(1) process. So if in a particular scenario this is not the case, then this estimator is not expected to perform better than the batch means estimator. Rather it is expected to better than the batch means estimator in cases where there is significant correlation among the samples drawn from the Markov Chain, since that would lead to our assumption actually being valid.

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 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 way we are going to do this is,

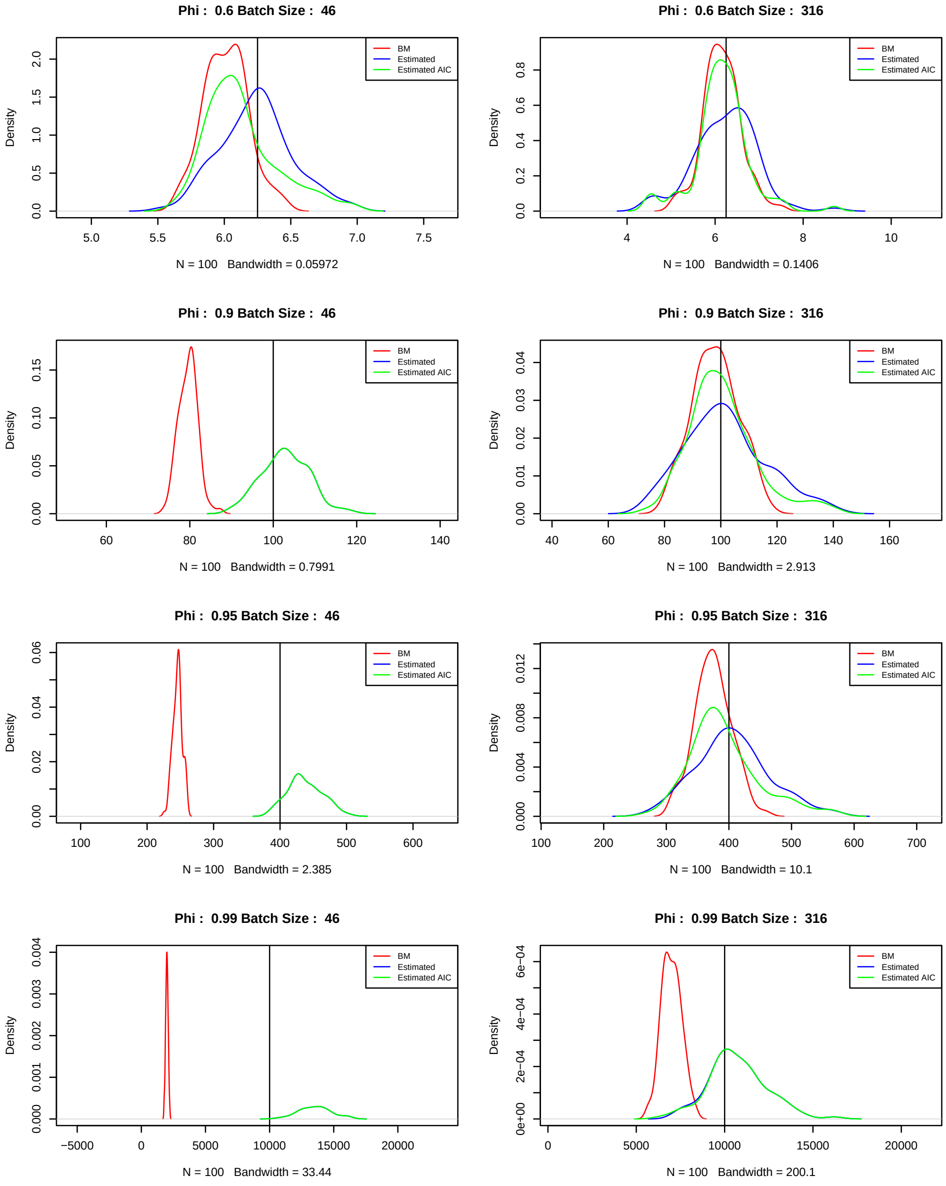
- 1. We are going to run **iter** number of iterations.
- 2. In each iteration we are going to generate **T** number of samples.
- 3. In each simulation we are going to apply our estimator and the batch means estimator with two batch sizes : $\sqrt[3]{T}$ and $\sqrt{T}$.

Once these simulations are done, we are going to calculate the mean MCSE variance, the variance in the MCSE variance and in some cases the mean squared error (MSE) in the MCSE variance.

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 $\alpha = 1$ where ϕ is the correlation coefficient for the AR(1) process and α is the variance of the normal distribution that introduces a little variability in each of the generated samples. For this simulation $T = 1e5$, $iter = 100$ and the batch sizes as described above. The AR model is fit twice to the data generated in one iteration, 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 variance in the MCSE and compare it to the one estimated by the different estimators.

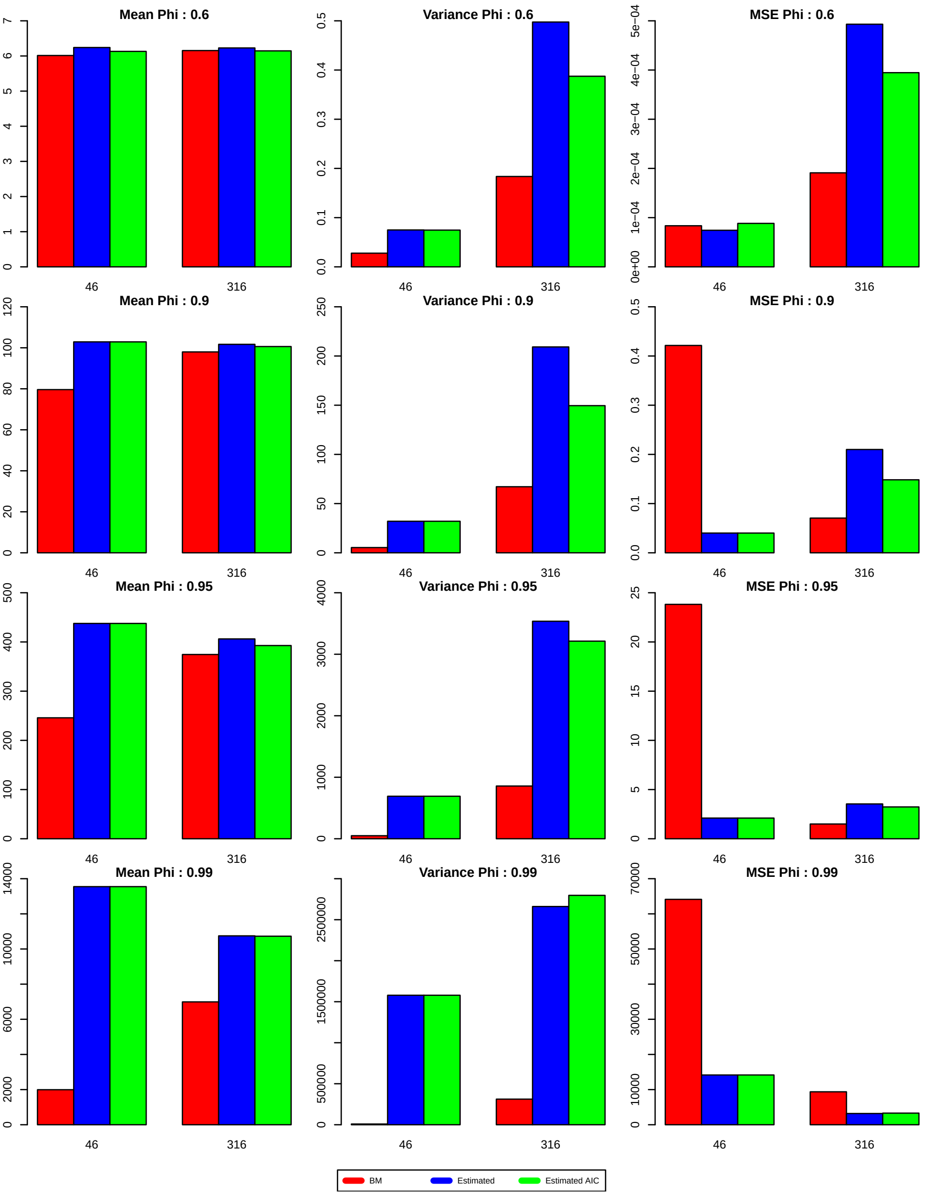


There are certain trends that can be inferred from the graphs that are shown above :

- 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, to enable significant correlation to exist between 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, variance and the MSE for the MCSE CLT variance being estimated :



- It is apparent that the variance for the value generated by our estimator is higher than the variance in the value generated by Batch Means Estimator.
- When we look at the MSE for the batch means and our own estimator, we see a trend that depicts why our estimator is better than the batch means estimator when there exists significant correlation among the samples that are drawn. For a small batch size, one which preserves correlation among the batch means, we see that the MSE for the values generated by Batch Means is significantly higher than the MSE for the values generated by our estimator.

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 Φ taking different values as described below and $W = 0.3 * diag(p)$. For this simulation we set $T = 1e5$, $iter = 100$ and the batch sizes as described earlier.

We generate a Φ matrix using the formula for an SVD decomposition. According to the SVD decomposition : $M = U\Sigma V^T$ where M is real, U and V are orthogonal matrices consisting of the eigenvector and Σ is a diagonal matrix with eigenvalues as the diagonal elements. We take a special case of the SVD decomposition wherein we set Φ to be a PSD matrix with non-negative real eigen values. We also set the diagonal values of the Φ matrix to be values less than 1, as that is necessary for producing a Φ matrix with real non-negative eigen values that are less than 1.

Φ with large eigen values

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 well both with small and large batch sizes.

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

From the above graph we can draw some considerable inferences :

- 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 with a larger batch size. This is in sync with the results seen in AR(1) case as well.
- looking at the error plots it is seen that our estimator generates estimates that are closer to the true value for both batch sizes, when the samples have significant correlation among them.

Φ with small eigen values

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

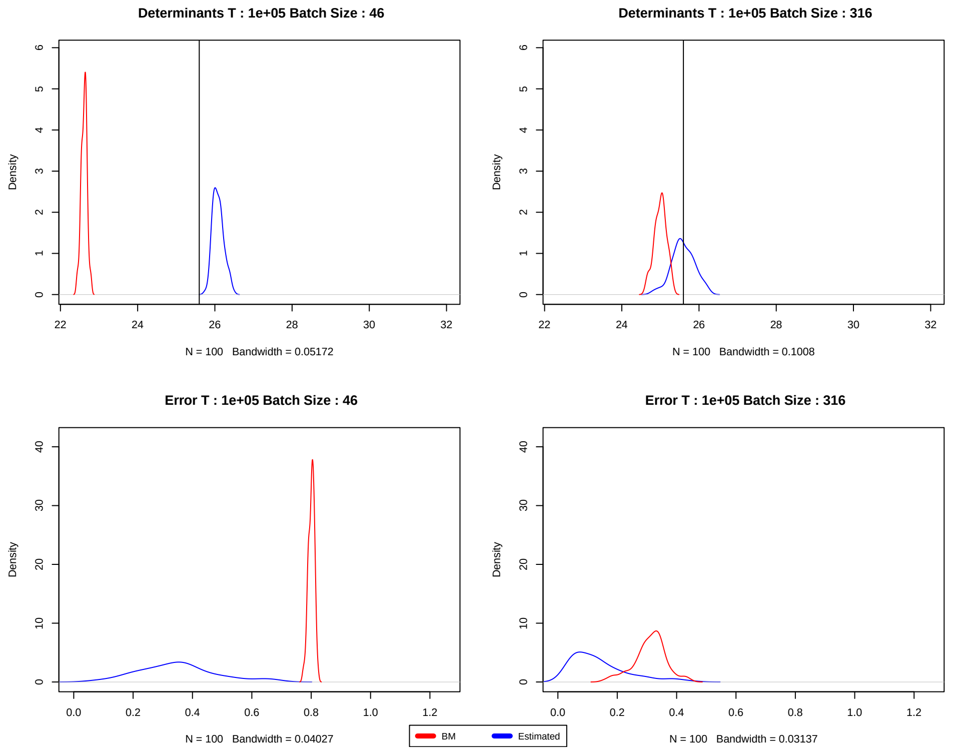


Figura 1: Determinant of Σ for chosen Φ

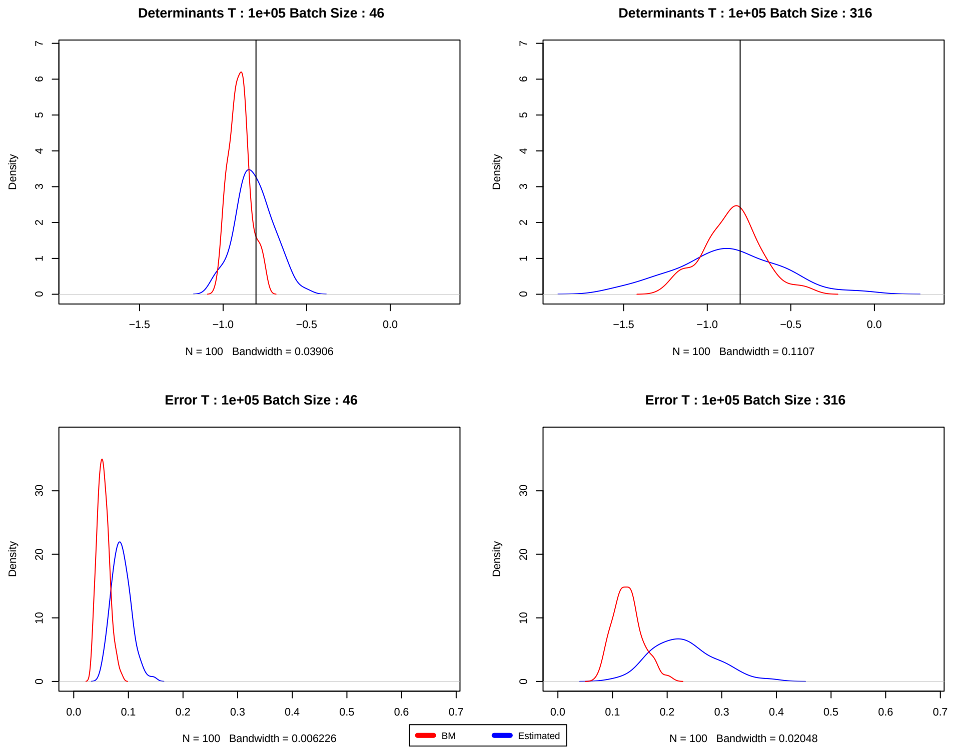


Figura 2: Determinant of Σ for chosen Φ

4.3. Sampling from a Bayesian Logistic Regression Model

Bayesian Logistic Regression Model

In order to validate the working of our proposed estimator, we need to test its performance on MCMC data. Now in order to generate data that does not inherently have any bias in its generation 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 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 in 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

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

For using MH algorithm to generate a Markov Chain, 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. Using the above set up we generate the MCMC data on which we then apply our own estimator and the batch means estimator.

It is apparent that in this case there is no way to evaluate the correct value of the the CLT variance in MCSE, 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 exploit 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 $\hat{\sigma}^2_{BM} \rightarrow \hat{\sigma}^2_g$ with probability one as $n \rightarrow \infty$. Which essentially means that the value of $\hat{\sigma}^2_{BM}$ moves towards the actual value $\hat{\sigma}^2_g$ as the number of samples increase. Hence if on increasing the sample size we see batch means estimator and our own estimator moving towards to the same direction then it would be reasonable to assume that our estimator is giving good estimates of the variance under consideration.

Working with univariate sample data

In this case we use a single covariate of the samples generated to test our estimator on. On those sample we apply our estimator to judge the value of the variance in the MCMC error.
In we look at the graphs for the higher batch size, we notice a trend. The graph for the BM estimates, increases in its spread as the the sample size increases, and this also leads to a shift in the mean of the variance in the direction of the estimate that our own estimator predicts.
Thus it can be concluded that our estimator is over estimates the value of the variance of the MCMC error. From the previous example it can also be concluded that for better correlated samples our estimator gives a much better value than the batch means estimator, and from this example it can be seen that the pattern our estimator follows is similar to the pattern the batch means estimator also follow as the sample size increases. Thus it's estimate can be considered quite reliable. It is apparent that our estimator estimates variance in a much more varied away. The estimate too is shifted a lot right that to the actual value, which is sync with the results that we obtained above. Looking at the mean of the variance generated for each of the sample sizes :

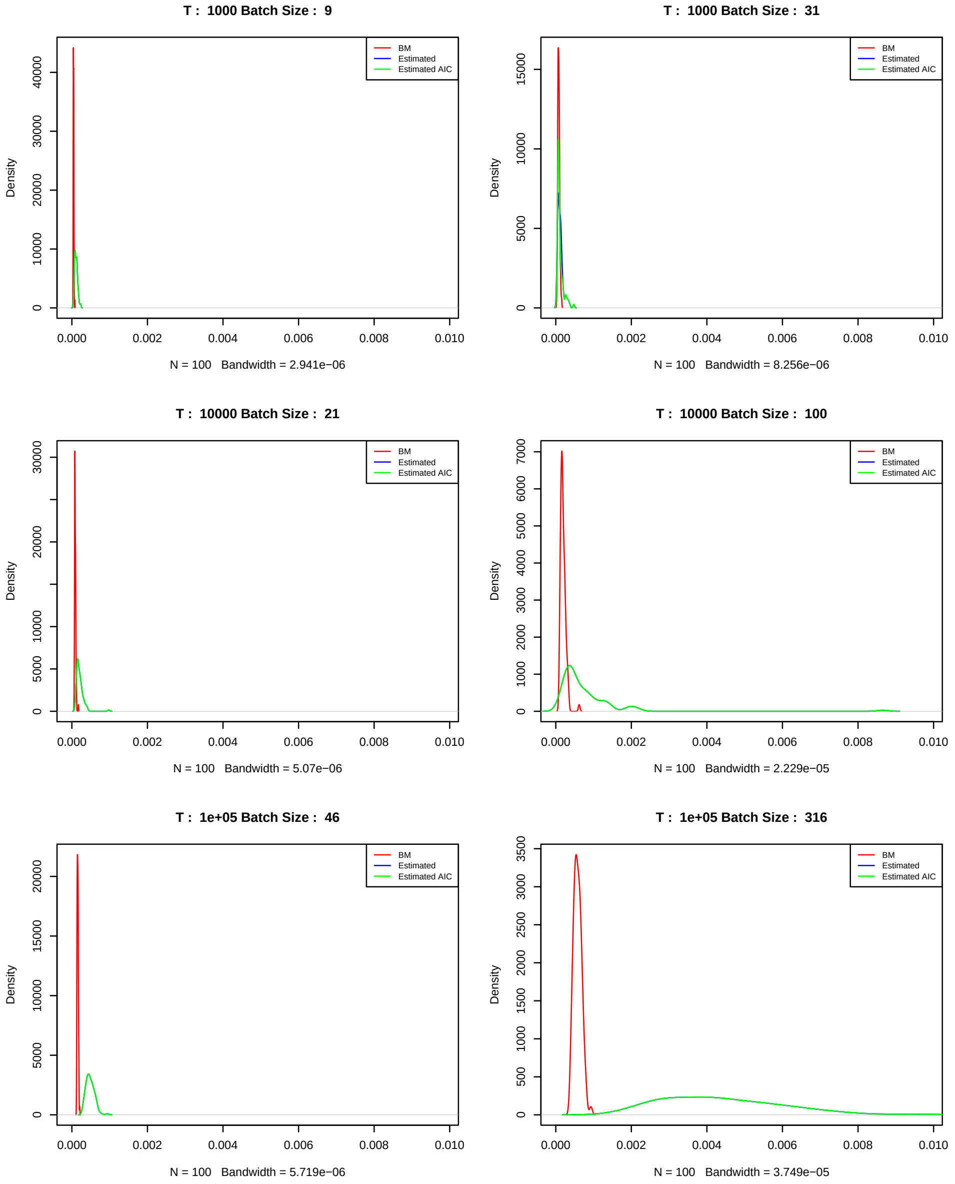


Figura 3: What's the haps here

- with an increase in the sample size the value of the mean also increases for both the estimators.
- the value of the mean is greater for a larger batch size, which can be attributed to just there being additional terms to sum for the value of the variance.
- for a particular sample size, for a batch size dramatically increases. Thus indicating their might be an optimal value of the batch size which helps produce good results with increasing 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.

Working with multivariate sample data

In this example we use all the covariates of the generated samples and we fit a VAR(1) model to this data and estimate the variance in the MCSE of these samples using the batch means estimator and our own estimator. While interpreting the above data it is necessary to note that the axis is in negative since we are working with the log of the determinants for co variance matrix obtained by both the estimators. Again since in this case there is no way to actually evaluate the correct value of the MCSE variance for the samples generated, we follow the same approach that we did for the single co-variate case, i.e., we vary the samples size and judge the performance of the batch means estimator on the increasing sample size.

There are a considerable number of inference that can be drawn from the above graphs :

- As the batch size increases, the graphs for the BM estimate 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 could be seen, and thus the estimates of our own estimator and the batch means estimator appear similar.
- 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 is always able to capture a value greater than the batch means estimator for most sample sizes.
- The distribution of the estimates generated by our estimator though is pretty similar for both batch sizes for a particular sample size.

An interesting inference seen from this graph is that with an increasing sample size, the variance in the estimates made by our estimator shows a decreasing trend.

5. Conclusion/Discussion

What all to write?

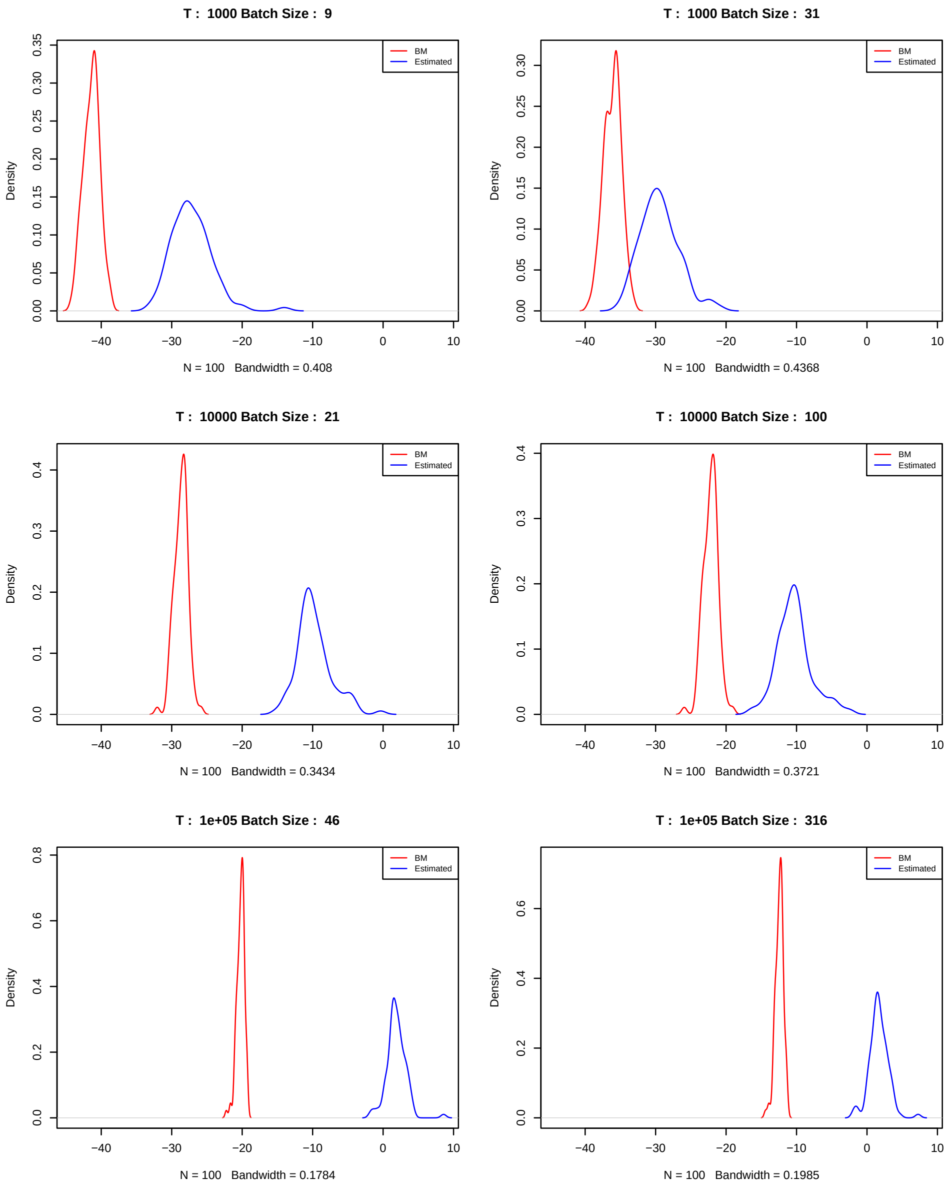


Figura 4: What's the haps here