Comparing Likelihood Free MCMC Algorithms for Matrix Normal Distribution via Gaussian Copula

Statistical Computing STAT 540: Project Report

Amal Agarwal*

Pennsylvania State University
aua257@psu.edu

Abstract

Likelihood Free MCMC algorithms play a key role in Bayesian inference for situations when likelihood model is either intractable or computationally expensive [1]. These methods suffer from curse of diensioanlity and in high dimensions a procedure to implement them via Gaussian copula was introduced [2]. In certain situations, Matrix Normal distribution can prove very useful to model high dimensional data [3]. This project compares different variants of LF-MCMC algorithms for Matrix Normal distribution via Gaussian copula.

1. Introduction

There are many situations where the likelihood function $\pi(\underset{\sim}{y}|\theta)$ is either unavailable or computationally expensive. Common examples come from stochastic processes, particularly time series models. This poses challenge to obtain a significant effective sample size within reasonable amount of time since the chain runs very slowly. The classes of algorithms that have been developed to avoid evaluating likelihood function at each iteration of Metropolis Hastings (MH) algorithm are called Likelihood Free computations [1, 4–7].

Likelihood free methods have been gaining a lot of popularity in diverse applications where Bayesian inference via usual MH algorithm becomes computationally impractical [1]. Some of these applications include wireless communications engineering, quantile distributions, population genetics, protein networks, α-stable models etc [1].

The purpose of this project is three-fold. At first, we explore the broad topic of Likelihood free Markov Chain Monte Carlo (LF-MCMC) which comes under the umbrella of Approximate Bayesian Computations (ABC). We will review the basic idea and motivation behind using LF-MCMC algorithm over regular Metropolis Hastings (MH) algorithm [1]. We then briefly go over different variants of LF-MCMC listing out advantages and disadvantages of each algorithm. Secondly we go in deep to understand how LF-MCMC can be applied to high dimensional posteriors via Gaussian Copula. Lastly we will briefly go over Matrix Normal distribution and its advantages for modelling high dimensional data. We compare different variants of LF-MCMC for Matrix Normal likelihood model based on various criteria like effective sample size per second, total run time and mean square error for the estimated parameters [2].

2. Likelihood Free MCMC

In its most trivial accept reject form, a LF algorithm consists of the following steps:

- Sample $\theta' \sim \pi(.)$ from the prior.
- Generate dataset $\underset{\sim}{x}$ from the model $\pi(.|\theta')$.
- Accept θ' if $\underset{\sim}{x} \approx \underset{\sim}{y}$, where $\underset{\sim}{y}$ is the observed dataset.

The last step is crucial. If the simulated and observed datasets are similar in some manner, we regard θ' as a plausible candidate sample from out target posterior since there is a high

[*] Graduate Student, Department of Statistics

chance this θ' could have generated the observed data from the given model. These steps form the basis of LF-MCMC algorithm and its variants to be described section 2.2. Note that here we have circumvented the problem of evaluation of likelihood function and replaced it with the problem of assessing if the simulated dataset is close to observed dataset in some manner.

To assess the closeness of two datasets, we usually use some kind of statistical distance metric like Euclidean, Mahalanobis, Minkowski etc. For continuous distributions the probability that the simulated dataset is equal to the observed dataset is exactly zero and so the acceptance rate of the above algorithm is also zero. To resolve this, we reduce the dimension of the dataset by using some sufficient statistic $T(.)$ for θ and assess if $T(\underset{\sim}{x}) \approx T(\underset{\sim}{y})$. The idea is that there can be more than one simulated datasets which obey $T(\underset{\sim}{x}) \approx T(\underset{\sim}{y})$, thus increasing aceptance probability.

2.1. Theoretical background

A natural question to ask is how the above formalism can be incorporated in MH algorithm and if so, why would it work. Recall that our target is $\pi(\theta|\underset{\sim}{y})$. Consider the following new augmented posterior target:

$$\pi_{\mathrm{LF}}(\theta, \underset{\sim}{x}|\underset{\sim}{y}) \propto \pi(\underset{\sim}{y}|\underset{\sim}{x}, \theta)\pi(\underset{\sim}{x}|\theta)\pi(\theta) \tag{1}$$

Here the auxilliary parameter $\underset{\sim}{x}$, is the simulated dataset from $\pi(.|\theta)$. The last two terms make up the posterior $\pi(\theta|\underset{\sim}{x})$ and the density $\pi(\underset{\sim}{y}|\underset{\sim}{x}, \theta)$ puts more weight on this posterior in regions where $\underset{\sim}{x}$ and $\underset{\sim}{y}$ are similar i.e. $T(\underset{\sim}{x}) \approx T(\underset{\sim}{y})$. For $\underset{\sim}{x} = \underset{\sim}{y}$, the density $\pi(\underset{\sim}{y}|\underset{\sim}{x}, \theta)$ is assumed to be constant, thus returning back the target posterior exactly. Note that we are interested in the marginal posterior

$$\pi_{\mathrm{LF}}(\theta|\underset{\sim}{y}) \propto \pi(\theta)\int_{\mathcal{Y}} \pi(\underset{\sim}{y}|\underset{\sim}{x}, \theta)\pi(\underset{\sim}{x}|\theta)d\underset{\sim}{x} \tag{2}$$

which acts as an approximation to $\pi(\theta|\underset{\sim}{y})$. We can simply discard the realizations of auxilliary data sets in order to perform the above integration numerically.

Now the augmented proposal directed towards an MCMC sampler for $\pi_{\mathrm{LF}}(\theta, \underset{\sim}{x}|\underset{\sim}{y})$ can be factorized as

$$q[(\theta, \underset{\sim}{x}), (\theta', \underset{\sim}{x}')] = q(\theta, \theta')\pi(\underset{\sim}{x}'|\theta') \tag{3}$$

This means that given the current state θ we first sample θ' from the proposal $q(\theta, \theta')$ and then given this θ' we simulate a data set $\underset{\sim}{x}$ from our likelihood model. Using standard arguments of detailed balance and Harris ergodicity, we can show the stationarity of the augmented proposal. The probability of going from the current state $(\theta, \underset{\sim}{x})$ to the next state $(\theta', \underset{\sim}{x}')$ within the MH framework is just $\min(1, \alpha[(\theta, \underset{\sim}{x}), (\theta', \underset{\sim}{x}')])$ where

$$\begin{aligned}
\alpha[(\theta, \underset{\sim}{x}), (\theta', \underset{\sim}{x}')] =& \frac{\pi_{\mathrm{LF}}(\theta', \underset{\sim}{x}'|\underset{\sim}{y})\ q[(\theta', \underset{\sim}{x}'), (\theta, \underset{\sim}{x})]}{\pi_{\mathrm{LF}}(\theta, \underset{\sim}{x}|\underset{\sim}{y})\ q[(\theta, \underset{\sim}{x}), (\theta', \underset{\sim}{x}')]} \\
=& \frac{\pi(\underset{\sim}{y}|\underset{\sim}{x}', \theta')\pi(\underset{\sim}{x}'|\theta')\pi(\theta')}{\pi(\underset{\sim}{y}|\underset{\sim}{x}, \theta)\pi(\underset{\sim}{x}|\theta)\pi(\theta)} \times \\
& \frac{q(\theta', \theta)\pi(\underset{\sim}{x}|\theta)}{q(\theta, \theta')\pi(\underset{\sim}{x}'|\theta')} \\
=& \frac{\pi(\underset{\sim}{y}|\underset{\sim}{x}', \theta')\pi(\theta')q(\theta', \theta)}{\pi(\underset{\sim}{y}|\underset{\sim}{x}, \theta)\pi(\theta)q(\theta, \theta')}
\end{aligned}$$

This shows that the intractable likelihoods do not need to be evaluated in the acceptance probability calculation. The density function $\pi(\underset{\sim}{y}|\underset{\sim}{x}, \theta)$ is usually approximated by kernel density estimation methods as

$$\pi_\epsilon(\underset{\sim}{y}|\underset{\sim}{x}, \theta) \propto \frac{1}{\epsilon}K\left(\frac{d(T(\underset{\sim}{y}), T(\underset{\sim}{x}))}{\epsilon}\right) \tag{4}$$

Here $d(.)$ is some statistical distance metric, K is some standard kernel density function like uniform, gaussian, epanechnikov etc. and ϵ is the scale parameter.

If we take limit $\epsilon \to 0$ on the RHS, it would become a point mass on $T(\underset{\sim}{x}) = T(\underset{\sim}{y})$, in which case the target posterior is exactly recovered. On the other hand if $\epsilon > 0$ or if $T(.)$ is not sufficient statistics, then the LF approximation to $\pi(\theta|\underset{\sim}{y})$ is $\pi_{\text{LF}}(\theta|\underset{\sim}{y})$. Pseudocode for the corresponding LF-MCMC algorithm is given below:

```
1. Initialize $(\theta_0, \underset{\sim}{x}_0)$ and $\epsilon$. Set the iteration $t = 0$.
2. At iteration t
   (a) Generate $\theta' \sim q(\theta_t, \theta)$ from the proposal distribution of $\theta$.
   (b) Generate $\underset{\sim}{x}' \sim \pi(\underset{\sim}{x}|\theta')$ from the model given $\theta'$.
   (c) With probability $\min(1, \alpha[(\theta, \underset{\sim}{x}), (\theta', \underset{\sim}{x}')])$,
       set $(\theta_{t+1}, \underset{\sim}{x}_{t+1}) = (\theta', \underset{\sim}{x}')$, otherwise
       set $(\theta_{t+1}, \underset{\sim}{x}_{t+1}) = (\theta_t, \underset{\sim}{x}_t)$.
   (d) Set $t \leftarrow (t+1)$ and repeat step 2 until convergence.
```

2.2. Variants of LF-MCMC algorithm

2.2.1 Error augmented sampler

In all LF-MCMC algorithms, low values for the scale parameter ϵ results in high accuracy as suggested by taking the limit $\epsilon \to 0$. The disadvantage is slow mixing of chains through low acceptance rates. Increasing the ϵ value improves the chain mixing but accuracy is compromised. Optimizing over ϵ could be critical and the following error augmented sampler [8] takes this into account by treating ϵ as a tempering parameter.

$$\pi_{\text{LF}}(\theta, \underset{\sim}{x}, \epsilon|\underset{\sim}{y}) \propto \pi_\epsilon(\underset{\sim}{y}|\underset{\sim}{x}, \theta)\pi(\underset{\sim}{x}|\theta)\pi(\theta)\pi(\epsilon) \tag{5}$$

The approximation to the target is then given by

$$\pi^{\mathcal{E}}_{\text{LF}}(\theta|\underset{\sim}{y}) \propto \int_{\mathcal{E}}\int_{\mathcal{Y}} \pi_{\text{LF}}(\theta, \underset{\sim}{x}, \epsilon|\underset{\sim}{y}) d\underset{\sim}{x} d\epsilon \tag{6}$$

where $\epsilon \in \mathcal{E} \subseteq \mathcal{R}^+$

2.2.2 Multiple augmented sampler

To improve the sampler performance by reducing the variability in the acceptance probability, one approach is to use multiple augmented sampler [9], where instead of simulating one dataset at each iterate, we simulate S auxiliary datasets $\underset{\sim}{x}_{1:S} = (\underset{\sim}{x}^1, ..., \underset{\sim}{x}^S)$ where $\underset{\sim}{x}^s \sim \pi(\underset{\sim}{x}, \theta)$. The augmented target posterior becomes

$$\pi_{\text{LF}}(\theta, \underset{\sim}{x}_{1:S}|\underset{\sim}{y}) \propto \frac{1}{S}\sum_{s=1}^{S} \pi_\epsilon(\underset{\sim}{y}|\underset{\sim}{x}^s, \theta) \prod_{s=1}^{S} \pi(\underset{\sim}{x}^s|\theta)\pi(\theta) \tag{7}$$

The approximation to the target is then given by

$$\pi^{S}_{\text{LF}}(\theta|\underset{\sim}{y}) \propto \int_{\mathcal{Y}^S} \pi_{\text{LF}}(\theta, \underset{\sim}{x}_{1:S}|\underset{\sim}{y}) d\underset{\sim}{x}_{1:S} \tag{8}$$

2.2.3 Marginal space sampler

Simulating S datasets at each iterate similar to multiple augmented sampler, the integration in (2), can be approximated by a Monte Carlo summation as

$$\pi_{\text{LF}}(\theta|\underset{\sim}{y}) \approx \frac{\pi(\theta)}{S}\sum_{s=1}^{S} \pi_\epsilon(\underset{\sim}{y}|\underset{\sim}{x}^s, \theta) \tag{9}$$

This permits to directly construct a LF-MCMC sampler targeting $\pi_{\text{LF}}(\theta|\underset{\sim}{y})$. With increase in S, the Monte Carlo approximation becomes more accurate. However the sampler performance is very poor in this case since it only approximates $\pi_{\text{LF}}(\theta|\underset{\sim}{y})$ which approximates $\pi(\theta|\underset{\sim}{y})$. An interesting thing to note is that the accepting probability of both Multiple augmented sampler and Marginal space sampler turns out to be exactly same. Thus both the samplers possess identical mixing and efficiency properties.

3. LF-MCMC in high dimensions

One of the primary restrictions to implement LF-MCMC algorithms is that they suffer from the curse of dimensionality [10]. Kernel density estimation methods are reliable only in low dimensions. Also for parameter identifiability the dimension of $T(.)$ must be greater than or equal to dimension of $\underset{\sim}{\theta}$. These two constraints imply that the LF methods would perform poorly even with a moderate number of parameters. In specific situations like when the untractable likelihood is factorizable, this problem can be solved. However a general solution for constructing a LF approximation to the target in high dimensions is to construct a Gaussian copula that can approximate the dependence structure of $\pi(\underset{\sim}{\theta}|\underset{\sim}{y})$ [2].

This approach relies on the fact that under standard regularity conditions, the posterior distribution $\pi(\underset{\sim}{\theta}|\underset{\sim}{y})$ is asymptotically normal [11]. Now Meta Gaussian family of distributions has a peculiar property that the p-dimensional joint density can be reconstructed from all bivariate marginal densities. Thus using LF-MCMC methods, the bivariate densities can be estimated and then meta-Gaussian approximations of these densities can be combined together to form the neta-Gaussian approximation of the full posterior distribution. Let us first define a few things and set the notations:

Consider the random vector $\underset{\sim}{\theta} = (\theta_1, ..., \theta_p)^T$ has continuous multivariate density $g(.)$. The marginal distribution functions and densities for θ_i are $G_i(.)$ and $g_i(.)$, $i = 1, ..., p$ respectively. The copula $C(\underset{\sim}{\theta})$ of is defined as the joint distribution of $\underset{\sim}{U} = (U_1, ..., U_p)^T = (G_1(\theta_1), ..., G_p(\theta_p))$ and contains the full dependence structure among all the components of $\underset{\sim}{\theta}$. Using Sklar's theorem theorem [12], we can write down $g(\underset{\sim}{\theta}) = C(G_1(\theta_1), ..., G_p(\theta_p))$. Now define $\underset{\sim}{\eta} = (\eta_1, ..., \eta_p)^T$ with $\eta_i = \Phi^{-1}(G_i(\theta_i))$ for $i = 1, ..., p$ where Φ is the standard normal cumulative distribution function. If $\eta \sim N(0, \Lambda)$, then $C(\underset{\sim}{\theta})$ is called a Gaussian copula and $\underset{\sim}{\theta}$ has meta-Gaussian distribution function with density function:

$$g(\underset{\sim}{\theta}) = \frac{1}{|\Lambda|^{1/2}} \exp\left[\frac{1}{2}\eta^T(I - \Lambda^{-1})\eta\right] \prod_{i=1}^{p} g_i(\theta_i) \tag{10}$$

Also it is can be easily shown that multivariate normal family is embedded within the family of meta-Gaussian family [2]. Using this fact and assuming the approximate normality of the target, we can implement the following algorithm to approximate $g(\underset{\sim}{\theta})$:

1. For each pair (i, j) with $i = 1, ..., p-1$ and $j = i+1, ..., p$,
   (a) Identify the sufficient statistics $T_{(i,j)}$ for (θ_i, θ_j).
   (b) Draw approximate samples $(\underset{\sim}{\theta}^{(1)}, ..., \underset{\sim}{\theta}^{(n)})$ using LF-MCMC algorithm (or one of its variant). Separate out the $(i, j)^{\text{th}}$ components as $(\theta_i^{(1)}, \theta_j^{(1)}), ..., (\theta_i^{(n)}, \theta_j^{(n)})$.
   (c) Let $r_i^{(1)}, ..., r_i^{(n)}$ be the ranks of $\theta_i^{(1)}, ..., \theta_i^{(n)}$ and $q_j^{(1)}, ..., q_j^{(n)}$ be the ranks of $\theta_j^{(1)}, ..., \theta_j^{(n)}$. Put $\eta_i^{(l)} = \Phi^{-1}\left(\frac{r_i^{(l)}}{n+1}\right)$ and $\eta_j^{(l)} = \Phi^{-1}\left(\frac{q_j^{(l)}}{n+1}\right)$ for $l = 1, ..., n$.
   (d) Calculate the sample correlation $(\eta_i^{(1)}, \eta_j^{(1)}), ..., (\eta_i^{(n)}, \eta_j^{(n)})$ and denote it as $\hat{\Lambda}_{i,j} = \hat{\Lambda}_{j,i}$.
2. For $i = 1, ..., p$
   (a) Identify the sufficient statistics $T_{(i)}$ for (θ_i).
   (b) Draw approximate samples $(\underset{\sim}{\theta}^{(1)}, ..., \underset{\sim}{\theta}^{(n')})$ using LF-MCMC algorithm (or one of tits variant). Separate the i^{th} component as $\theta_i^{(1)}, ..., \theta_i^{(n')}$.
   (c) Approximate the marginal density $g_i(\theta_i)$ based on the above samples for the i^{th} component using density estimation methods.
3. Combine all the $\hat{\Lambda}'_{\mathrm{i,j}}s$ from step 1 to form the p dimensional correlation matrix $\hat{\Lambda}$. Using $\hat{\Lambda}$ and $\hat{g}_i(\theta_i)$, obtain an estimate of the approximation of the target using (10).

The appoximation of the estimate of the target can be used in the Sampling Importance Resampling routine to generate samples from the this target. The steps are: For $i = 1, ..., N$,

- Sample the vector from the importance function $\underset{\sim}{\theta}_i \sim f(\underset{\sim}{\theta})$.
- Calculate the weights $w^{(i)} \propto \dfrac{g(\underset{\sim}{\theta}_i)}{f(\underset{\sim}{\theta}_i)}$
- Calculate the normalized weights.

- Re-sample $\underset{\sim}{\theta}_1, ..., \underset{\sim}{\theta}_N$ according to the normalized weights.

4. Matrix Normal Distribution

Undirected Gaussian graph models provide a powerful tool to model conditional independencies in high dimensional data [13]. In particular for a Gaussian Markov Random Field (GMRF) [14,15], we are generally interested in estimating the concentration matrix $C = \Sigma^{-1}$ where Σ is the corresponding variance-covariance matrix. This is usually aimed at exploiting the sparsity of the concentration matrix which is common in many situations. There have been many algorithms developed to deal with this problem. The famous Graphical Lasso algorithm [15] provides a sparse and shrinkage estimator. Other algorithms like BIGQUIC [16] can solve for upto one million variables.

One of the key assumptions in all these type of algorithms is that the samples are independent and identically distributed from a multivariate Gaussian distribution. However in certain applications like gene expression data, there is often a correlation between p genes collected over q different tissues from the same subject. To better understand this, consider $\mathbf{X}$ to be the $p \times q$ matrix of the expression data, where the j^{th} column corresponds to the expression data of p genes measured in the j^{th} tissue, and the i^{th} row corresponds to gene expressions of the i^{th} gene over q different tissues [3]. Instead of assuming that the columns or rows are independent, we assume that the matrix variate random variable $\mathbf{X}$ follows a matrix normal distribution [17], where both row and column precision matrices can be specified. These precision matrices of the matrix normal distribution provide the conditional independence structures of the row and column variables, where the non-zero off-diagonal elements of the precision matrices correspond to conditional dependencies among the elements in row or column of the matrix normal distribution [3].

For a $n \times p$ random matrix $\mathbf{X}$, the pdf of the Matrix Normal distribution $\mathcal{MN}_{n\times p}(\mathbf{M}, \mathbf{U}, \mathbf{V})$ is given as:

$$p(\mathbf{X}|\mathbf{M}, \mathbf{U}, \mathbf{V}) \propto \frac{1}{|\mathbf{V}|^{n/2}|\mathbf{U}|^{p/2}} \times \exp\left(-\frac{1}{2}tr(\mathbf{V}^{-1}(\mathbf{X}-\mathbf{M})^T\mathbf{U}^{-1}(\mathbf{X}-\mathbf{M}))\right) \tag{11}$$

For a non-sparse setting and assuming $n > p$, the time compexity of evaluating this likelihood is $O(n^3)$. Also the number of parameters involved are again of the order of $O(n^3)$. Thus to do Bayesian inference even for small matrix sizes, it becomes imperative that we use LF-MCMC methods tailored for high dimensions like the one described in previous section.

5. Simulation

This project focusses on comparing different variants of LF-MCMC algorithm described in (2.2) for obtaining samples from the posterior comprising of parameters $(\mathbf{M}, \mathbf{U}, \mathbf{V})$ from the Matrix Normal distribution via Gaussian copula approach for high dimensions. Here our likelihood model is same as described in (11) and priors over each element of $(\mathbf{M}, \mathbf{U}, \mathbf{V})$ are set as improper uniform.

Firstly, with arbitrary parameter settings, observed data $\mathbf{Y}$ was simulated using (11). This was done using the fact that $\mathbf{X} \sim \mathcal{MN}_{n\times p}(\mathbf{M}, \mathbf{U}, \mathbf{V})$ if and only if $\text{vec}(\mathbf{X}) \sim \mathcal{N}_{np}(\text{vec}(\mathbf{M}), \mathbf{V} \otimes \mathbf{U})$ where $\otimes$ denotes the Kronecker product and $\text{vec}(\mathbf{M})$ denotes the vectorization of $\mathbf{M}$. Now given this observed data, the correlation matrix Λ and the marginal densities $g_i(\theta_i)$, where θ_i are different elements of $\mathbf{M}, \mathbf{U}, \mathbf{V}$, were estimated using the the algortihm described in section 3. Three variants of LF-MCMC viz. Basic LFMCMC, Error augmented LF-MCMC and Multiple augmented MCMC were implemented in steps 1(b) and 2(b) of this algorithm. For each variant, three different choices of distance metrics viz. Euclidean, Minkowski and Mahalanobis and three different Kernel functions viz. Uniform, Gaussian and Epanechnikov were considered. Lastly, using the sampling impotance resampling procedure the samples from the actual target $\pi(\mathbf{M}, \mathbf{U}, \mathbf{V}|\mathbf{Y})$ were obtained.

6. Results and Conclusion

The main evaluating criteria were chosen as Effective Sample Size per second (ESS/s) and Total Mean Square Error (TMSE) for the parameter estimates.

In one simulation, each chain over the pairs and individual parameter components was run 10 times for $10,000$ iterates adaptively i.e. after each time, the starting value and the tuning parameter in the proposal were updated. The mean ESS/s and MSE for all the chains over different pairs was calculated. To account for the variability in the estimates of ESS/s in one simulation, 50 simulations were used to calculate the grand mean ESS/s. TMSE can be calculated by adding the mean squared errors for each element of the parameter matrices $\mathbf{M}, \mathbf{U}$ and $\mathbf{V}$.

It was found that the Euclidean distance metric and Uniform kernel density function performed best among all the variants. For these choices, the grand mean ESS/s varies with different matrix sizes as follows:

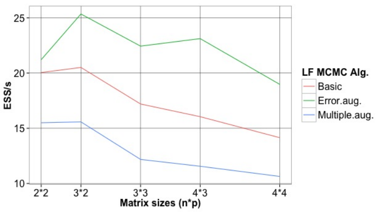

Figure 1: *ESS/s vs. matrix sizes*

The above plot clearly shows that error augmented sampler works better in terms of effective sample size per second. This can be attributed to the fact that while running the chains, ϵ is optimized for better chain mixing and accuracy. The variation of TMSE with increasing matrix sizes is given as:

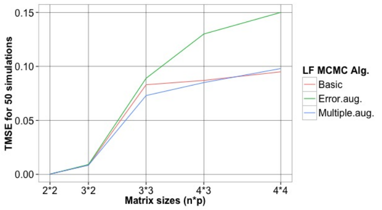

Figure 2: *TMSE vs. matrix sizes*

The above plot shows that the total mean square error for error augmented sampler is the highest. Running the chains longer solves this problem to some extent. But this clearly shows that the high effective sample size of this sampler is compromised by high TMSE. Also the multiple augmented sampler appears to be working better in terms of TMSE. This can be attributed to the fact that generating multiple datasets at each iterate help in reduction of the variance and thus faster convergence. The variation of mean total run time per simulation is given as follows:

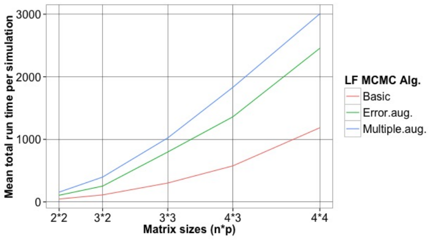

Figure 3: *Mean total run time per simulation vs. matrix sizes*

This plot clearly shows that multiple augmented sampler takes the most run time followed by error augmented sampler.

To assess the validity of the best performing algorithm, the diagnostic plots for the error augmented LF-MCMC sampler with Uniform kernel density function and Euclidean distance metric for one of the pairs, are given below:

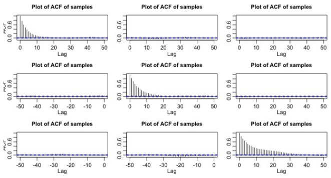

Figure 4: *Plot of ACF of samples* $(\theta_i, \theta_j, \epsilon)$

The above plot suggests a reasonable tuning parameter and good chain mixing.

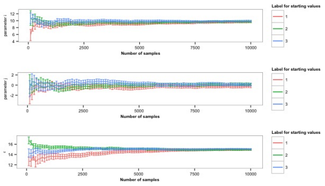

Figure 5: *Plot of estimates vs. sample size with error bars for* $(\theta_i, \theta_j, \epsilon)$

The above plot shows that for $(\theta_i, \theta_j, \epsilon)$ and for different starting values, the estimates converge to the same value.

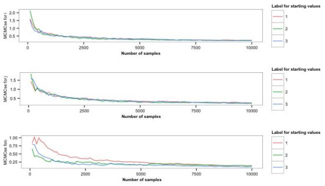

Figure 6: *Plot of MCMC se vs. sample size for* $(\theta_i, \theta_j, \epsilon)$

The above plot shows that for $(\theta_i, \theta_j, \epsilon)$ and for different starting values, MCMC standard errors converge to 0. These two plots verifies the error augmented LF-MCMC sampler to some extent.

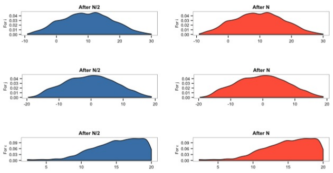

Figure 7: *Plot of estimated density for* $(\theta_i, \theta_j, \epsilon)$ *after* $N/2$ *and* N

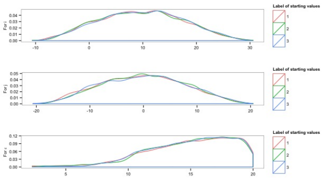

Figure 8: *Plot of marginal density for* $(\theta_i, \theta_j, \epsilon)$

From the above plots, the estimated densities after $n/2$ and after n look reasonably identical for all the 3 components. Also the estimated density for different starting values also overlap to a good extent. This further verifies the robustness and convergence of the algorithm.

7. Discussion

Although the basic sampler is suitable in terms of total run time, it is not recommended for smaller matrix sizes. However for larger matrix sizes it is the most optimum choice in terms of both ESS/s and TMSE. Error augmented samplers can be used in situations where precision is not required and we need fast mixing and high ESS/s. However with increasing matrix sizes the problem of TMSE appears to be increasing rapidly and longer chains may be required thus increasing the total run time. The multiple augmented sampler seems to be performing very bad in terms of ESS/s and total run time and therefore not recommended in general.

8. Future work

- It might be interesting to develop LF-MCMC algorithms that exploit the sparsity structure in the precision matrices **U** and **V**
- Tensor normal distributions of higher order can be studied under LF framework.
- I was not able to run another variant of LF-MCMC that aims for diagnosing misspecification. If this works out, it might be helpful in model selection.

References

[1] S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, Handbook of Markov Chain Monte Carlo. CRC press, 2011.

[2] J. Li, D. J. Nott, Y. Fan, and S. A. Sisson, "Extending approximate bayesian computation methods to high dimensions via gaussian copula," arXiv preprint arXiv:1504.04093, 2015.

[3] J. Yin and H. Li, "Model selection and estimation in the matrix normal graphical model," Journal of multivariate analysis, vol. 107, pp. 119–140, 2012.

[4] M. A. Beaumont, W. Zhang, and D. J. Balding, "Approximate bayesian computation in population genetics," Genetics, vol. 162, no. 4, pp. 2025–2035, 2002.

[5] P. Marjoram, J. Molitor, V. Plagnol, and S. Tavaré, "Markov chain monte carlo without likelihoods," Proceedings of the National Academy of Sciences, vol. 100, no. 26, pp. 15324–15328, 2003.

[6] O. Ratmann, C. Andrieu, C. Wiuf, and S. Richardson, "Model criticism based on likelihood-free inference, with an application to protein network evolution," Proceedings of the National Academy of Sciences, vol. 106, no. 26, pp. 10576–10581, 2009.

[7] S. A. Sisson, Y. Fan, and M. M. Tanaka, "Sequential monte carlo without likelihoods," Proceedings of the National Academy of Sciences, vol. 104, no. 6, pp. 1760–1765, 2007.

[8] P. Bortot, S. G. Coles, and S. A. Sisson, "Inference for stereological extremes," Journal of the American Statistical Association, vol. 102, no. 477, pp. 84–92, 2007.

[9] C. Andrieu, A. Doucet, and G. Roberts, "The expected auxiliary variable method for monte carlo simulation," Unpublished paper, 2007.

[10] M. G. Blum and O. François, "Non-linear regression models for approximate bayesian computation," Statistics and Computing, vol. 20, no. 1, pp. 63–73, 2010.

[11] A. W. Van der Vaart, Asymptotic statistics, vol. 3. Cambridge university press, 2000.

[12] M. Sklar, Fonctions de répartition à n dimensions et leurs marges. Université Paris 8, 1959.

[13] M. Yuan and Y. Lin, "Model selection and estimation in the gaussian graphical model," Biometrika, vol. 94, no. 1, pp. 19–35, 2007.

[14] J. Friedman, T. Hastie, and R. Tibshirani, The elements of statistical learning, vol. 1. Springer series in statistics Springer, Berlin, 2001.

[15] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," Biostatistics, vol. 9, no. 3, pp. 432–441, 2008.

[16] C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, P. K. Ravikumar, and R. Poldrack, "Big & quic: Sparse inverse covariance estimation for a million variables," in Advances in Neural Information Processing Systems, pp. 3165–3173, 2013.

[17] A. P. Dawid, "Some matrix-variate distribution theory: notational considerations and a bayesian application," Biometrika, vol. 68, no. 1, pp. 265–274, 1981.