Linear Regression: Bayesian Monte Carlo or Traditional Least Squares?

Dasen Ye, Alon Shapiro, Amal Koodoruth, Xun Lu

November 13, 2023

I. Introduction

Linear regression is a widely used statistical technique for modeling a linear relationship between a dependent continuous variable and one or more independent variables. Given its popularity and simplicity, various statistical approaches and algorithms have been developed to fit a linear regression model, each with its advantages and disadvantages. Here, we seek to investigate how the maximum likelihood estimation (MLE) approach using the ordinary least squares (OLS) criterion compares to a Bayesian approach.

As the MLE approach seeks to maximize the likelihood function for the model parameters given the observed data, the parameter estimations are provided as a single value. This leaves out context that may be important in order to provide deeper insights regarding the model and its fit to the data.

On the other hand, the Bayesian approach constructs a posterior distribution which represents the estimated parameter distribution (W. K. Hastings 1970, N. Metropolis 1953). The posterior distribution is constructed of the likelihood function which is the probability of observing the data that was observed given a specific set of model parameters, as well as the prior distribution - that is, our best guess of what the model parameters should be given our previous theoretical knowledge of the system and previous findings. This approach has multiple advantages over the MLE approach. First, it allows us to bias our model towards our prior understanding of the given process from which the data was generated. This challenges the observed data to provide strong evidence in order to influence our previously-established knowledge. Second, this approach constructs a posterior distribution for each parameter, which provides more information about the estimated distribution of each parameter and may provide further insights, depending on the specific question being investigated. This follows from the shift in mentality between the MLE approach which views the true parameters as a given unknown value, against the Bayesian approach, which views the true parameters as random variables which cannot be adequately estimated by just one value.

As there are many different non-Bayesian approaches to linear regression parameter estimation, so there are many different Bayesian approaches. Here, the goal is to explore the stated research question through the employment of Monte Carlo (MC) simulations. Thus, the Bayesian modeling will be done with a Markov Chain Monte Carlo (MCMC) method, implemented through a manual implementation 'from scratch' of the Metropolis-Hastings (MH) algorithm. This approach leverages rejection sampling from the posterior distribution by repeatedly sampling from a proposal distribution centered at the most recent Markov chain step, which always accepts higher-probability steps, but only sometimes accepts lower-probability steps (by calculating the ratio of probabilities of the next step and the current step in the chain). This allows the algorithm to explore regions of high probability density while occasionally jumping to lower density areas, thus reconstructing the parameter distribution, which would otherwise be difficult or impossible to calculate.

II. Research Question

In this work, we seek to compare the robustness of the traditional MLE parameter estimation approach to the Bayesian MCMC parameter estimation approach under the context of linear regression. The central research question guiding this investigation is as follows: How do these contrasting methodologies fare concerning two crucial criteria - precision of parameter estimations and overall goodness of fit?

Precision of parameter estimations will be compared in two ways. First, by comparing the closeness to and the spread around the true parameter value for the estimated parameter distributions for each method. Second, by comparing the coefficients of determination produced by each method. Further details are provided in the Measures of Performance sub-section in the Methods section.

Overall goodness of fit will be compared for both methods by using the Root Mean Square Error (RMSE).

III. Methods

Data Generation and simulation steps

In order to leverage the full potential of an MC simulation, the data generation and model fitting followed the following general flow:

1. Randomly generate n_data values from a uniform distribution in the interval $[x_{\min}, x_{\max}]$ to serve as the independent variable (x).
2. Generate corresponding n_data values for the dependent variable (y) value for each of the independent variable values using a given model.
3. Use the generated data set to fit an MLE linear regression model using the built-in function $lm()$ available in R.
4. Store the estimated parameters for the MLE.
5. Use the generated data to fit a Bayesian linear regression model using the function we developed called MCMC() (with the Metropolis-Hastings algorithm).
6. Store the estimated parameters, which are obtained by taking the mean of estimated values for all walkers and all iteration steps.
7. Repeat steps 1-6 n_rep times, storing the estimated parameters of the model at each iteration. This is done to capture the variability of the parameter estimations.

Note that a new data set (x and y values) was generated at each repetition, for a total of n_rep repetitions, using the same true model parameters (variability in the data set was inherent as x values were generated randomly from a uniform distribution in each iteration). This was done to produce variability in the OLS parameter estimations, which was necessary for quantifying the overall performance and variability of parameter estimations with OLS.

All the model fits in the above procedure were fitted to a simple quadratic linear regression model of the form:

$$Y = ax^2 + bx + c + \epsilon,\ \epsilon \sim Normal(0, \sigma^2) \tag{1}$$

Where Y is the dependent variable, X is the independent variable, ϵ is the error term which is normally distributed with mean zero and variance σ^2. The scalars a, b, and c (along with σ), are the parameters that were estimated in each model fit. They were fixed in order to allow for data generation to be as follows:
$a = 2, b = 5, c = 10, \sigma = 6$.

Note that this choice of model respects all the underlying assumptions for linear regression. The error term ϵ as well as Y are normally distributed with a constant variance, and observations were independently generated from a uniform distribution (for X, which was then used to calculate Y). As no assumptions are made on the independent variable X, randomly generating it from a uniform distribution is acceptable.

MLE Parameter Estimation

The parameters for the model were estimated using the built-in $lm()$ function in R. By default, the OLS criterion is used.

Bayesian Parameter Estimation

The Bayesian framework required a more involved set up, as the algorithm was custom made for this simulation, as well as the inherent flexibility involved in the framework involving definition of likelihood and prior distributions and hyperparameter tuning. Note that all probability distributions were coded on the log scale in order to avoid running into errors related to NaN values.

The Likelihood
The likelihood function represents the probability of observing the given data given a specific set of model parameters. Here, we assumed a Normal distribution which directly follows from the underlying assumption of the linear regression model - that the dependent variable Y is normally distributed (which in turn follows from the same assumption imposed on the error terms).

The Prior
Before observing any data, we often have some initial beliefs or knowledge about the likely values of the model parameters based on already existing knowledge of the data-generating process being studied. Here, as the true model parameters (which are often unknown) were chosen prior to data generation, a fair prior distribution can be chosen to represent prior belief (note that as data was fit for a quadratic model, the parameters could also be estimated very roughly by examining the scatter plot of the data). A reasonably wide normal distribution was chosen for all parameters, centered at the initial guess of parameters with a standard deviation of 0.1 times the initial guess for each parameter.

The Posterior
After observing the data, our beliefs about the model parameters are updated. The posterior probability function represents the updated beliefs about the parameters after considering both the observed data and the prior information. It is calculated using Bayes' theorem:

$$P(\theta|\text{data}) \propto P(\text{data}|\theta) \times P(\theta)$$

Here, $P(\theta|\text{data})$ is the posterior probability, $P(\text{data}|\theta)$ is the likelihood function, $P(\theta)$ is the prior probability. Note that in order for equality to hold, the RHS has to be normalized by $P(\text{data})$, the marginal likelihood. However, As we are performing sampling and are not interested in estimating any probabilities, only the shape of the posterior is of interest.

Metropolis-Hastings (MH) Algorithm
The MH algorithm was chosen to implement this Bayesian MCMC parameter estimation due to its robustness and simplicity. The implemented algorithm works as follows:

1. Start with an initial guess for the parameter values.
2. Generate a proposed next step from a normal proposal distribution, centered at the current step. 3. This is done separately for each parameter.
3. Calculate the ratio of probabilities of the next and current steps using the posterior distribution. Note that the effect of lack of normalization disappears as a division is involved.
4. Randomly draw a value r from a uniform distribution on [0,1].
5. If the ratio of probabilities is higher than r, accept the next step. If the ratio is smaller than r, reject it and set the next step to be the current step.
6. Repeat steps 1-5 for a predetermined number of iterations num_of_steps.

(Note: we use the term iteration to refer to the number of steps taken within one chain, whereas the term repetition is used to refer to the number of times both the MCMC and the OLS models were re-fit, i.e. the procedure described in the data generation section.)

This is the basic MH algorithm implemented in this work. In order to further improve the exploration of the posterior distribution, steps 1-6 were repeated 5 times, with each repetition called a 'walk' - as the MH algorithm is a random walk which explores the posterior. Further, the initial guess was offset by a small random noise term (with standard deviation roughly 10% of the initial guess) for each walker in order to allow for further variability of outcome among the 5 walkers.

The MH algorithm allows each walker to construct a Markov chain - that is, a chain of steps consisting of plausible parameter estimates taken along the random walk around the posterior distribution.

Lastly, in order to improve the parameter estimation, the burn in phase (initial 2000 steps) were removed, as the walkers are typically still far away from the peak of the posterior at that phase, and the final chains were thinned by a factor of 10, which helps reduce the frequency of less likely parameter values, and improve the precision of the estimated parameter distribution.

Measures of performance
Two measures of performance will be employed in the comparison of the two methods. Recall that in order to produce a proper distribution for the parameter estimations using the OLS method, the model fitting procedure is repeated n_rep times, fitting one MCMC model and one OLS model at each repetition. All the parameter estimations from all the repetitions, and for MCMC all the walkers and iterations (after burn in removal and thinning) were then aggregated to form one distribution for each parameter.

RMSE and RMSE Ratio

From the calculated distributions, the means were taken for each parameter and for each method (MCMC and OLS) as the best and final estimation. One final data set was generated to serve as the validation set, and an RMSE was generated for each of the methods.

A ratio of the RMSE values was then calculated to summarize the relative performance of both models:

$$RMSE\ Ratio = \frac{RMSE_{MCMC}}{RMSE_{OLS}}$$

$RMSE_{OLS}$ therefore served as a reference to which $RMSE_{MCMC}$ was compared to. When the ratio is close to 1, there is not much difference in goodness of fit between both methods. However, when the ratio is smaller than 1, this means the model generated by the MCMC fit the validation set better than the OLS, and vice versa for ratios bigger than 1 (this follows from the fact that smaller RMSE is better).

Coefficient of Variation (CV)
From the calculated distributions, the means and standard deviations for each parameter were used to calculate a CV measure defined as:

$$CV = \frac{\text{std. dev.}}{\text{mean}}$$

This measure was chosen in order to quantify the uncertainty in the parameter estimation. Once a CV was calculated for each parameter estimation of each one of the methods, a ratio of CV was taken:

$$CV\ Ratio = \frac{CV_{MCMC}}{CV_{OLS}}$$

Thus, each parameter was assigned a CV ratio, which quantified the comparison of the parameter estimation precision between both methods. Similarly to the RMSE Ratio, a ratio close to 1 means both methods yielded similarly precise estimations. Ratio smaller than 1 means MCMC yielded more precise estimations compared to OLS.

IV. Perturbations to the Model

Due to the long runtime of the algorithm, it was decided to restrict the scope of this study to investigate the effect of training data set size and number of repetitions of the algorithm. This decision was made based on

prior understanding of both MCMC and OLS. That is, as an MCMC algorithm runs for longer repetitions (and in conjunction with appropriately adjusting the burn-in and thinning hyperparameters), it would tend to perform better (up to some point). Similarly, one of the properties of MLE estimators which are provided by OLS is that they are asymptotically unbiased, that is, as the data set size increases, the estimations approach the true parameter.

Therefore, the joint effect of both dataset size and number of repetitions on the relative performance between MCMC and OLS parameter estimation was investigated. The dataset was varied between 10, 100, and 300 points, while the number of repetitions varied between 5, 20, and 50. All possible combinations for these values were run, for a total of 9 different combinations.

Heat maps were produced for both the RMSE ratio and the CV ratio to summarize these findings below.

V. Results

In order to confirm the MCMC algorithm runs properly, the trace plots for all the parameters are presented in fig. 1 below (before removal of initial burn-in or thinning). Each color represents a different walker, with each having taken 5000 steps. First step for each trace represents the initial guess provided, along with a small random error value to ensure random progression of the walk. By step 1000, all the traces seem to stabilize around the area of high probability density for each parameter's posterior distribution.

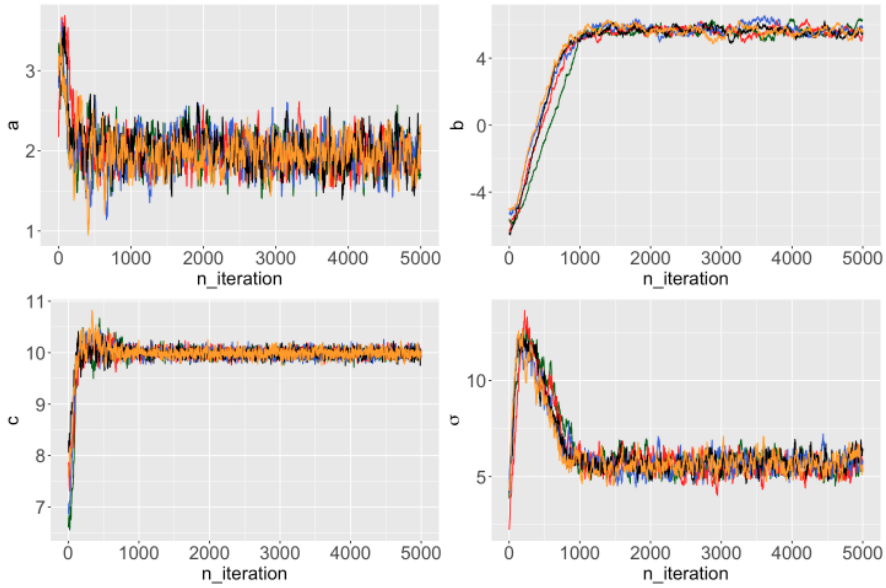

Figure 1: **Trace plots from one repetition of the MCMC algorithm are presented for all parameters.** Traces start at the initial guess which is far from the true parameter value, and move toward the true parameter value as the simulation progresses. These plots also confirm the MCMC algorithm is properly converging on the correct area of the posterior function.

Next, the generated data set should be inspected through a scatter plot. This typically gives a visual confirmation prior to model fitting that the model is correctly specified. Here, one of the many data sets that were generated is visualized to confirm a quadratic model correctly captures the relationship in the data. Then, the various fitted models are visualized to provide a visual which confirms the model fits the data well, with no overfitting in *Fig. 2"*. In pink, multiple fits were generated using the MCMC parameter estimates generated in one MCMC repetition. The few fits that are very far off and very faded were generated by the parameters estimated during the initial burn-in. As the algorithm approaches the area of high probability density for each of the parameter posteriors, the model fits get closer together and the lines become less transparent due to more overlap, which indicates the MCMC starting to produce estimates very close to the

true parameter values (as the model fits the data better). In black, the single best-fit line that is calculated with the OLS algorithm is shown.

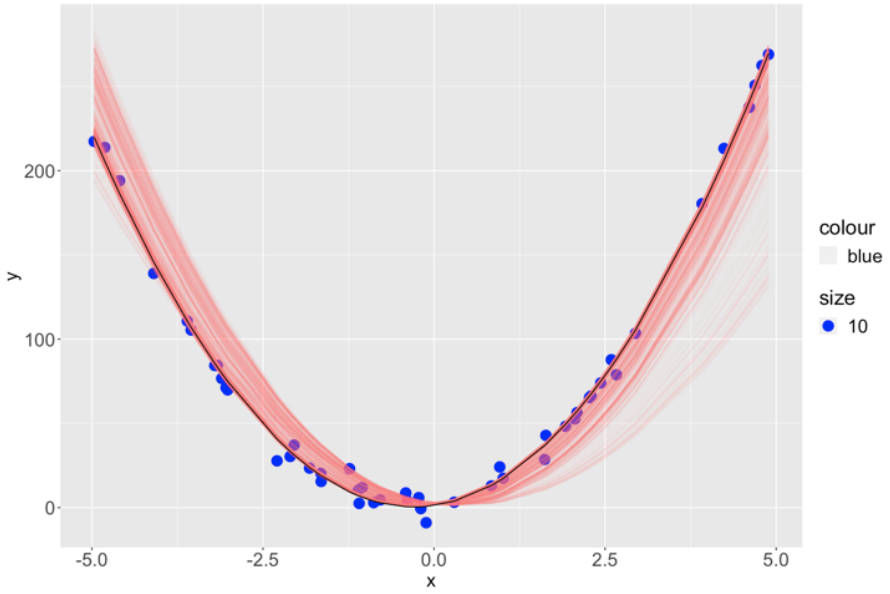

Figure 2: **Scatter plot of one of the randomly generated training data sets used for model fitting.** In pink, all the parameter estimations from the MCMC linear regression were used to fit multiple best-fit lines. The initial burn-in steps can be observed in the low line density area on the right. As MCMC algorithm moves toward the true parameter value, the lines converge and variation is reduced. In black, the single best-fit line from the OLS linear regression is plotted for comparison.

The parameter distributions generated by both methods for 50 data points and 100 repetitions are shown in *Fig. 3* below, with the true parameter value indicated by a green vertical line. All the distributions generated by both methods appear to be centered around the true parameter values. However, the spread of all the distributions generated by the MCMC algorithm is significantly smaller than that generated by OLS.

For all 9 different combinations of data set size and number of repetitions, the RMSE ratio remained very close to 1, indicating the goodness of fit of both methods was almost identical across all scenarios. This can be seen in *Fig. 4* below. The ratio varied from a minimum of 0.91, to a maximum of 1.11, indicating at most 11% difference between the two methods.

Lastly, the CV ratios were plotted for all scenarios and all estimated parameters in *Fig. 5* below. For parameters a and c, the CV ratio always remained far below 1, indicating that the MCMC parameter estimation provided far more precision compared to the OLS method. For parameter b, the MCMC parameter estimation provided a much smaller standard deviation in all scenarios except for the scenario with 300 data points and 5 repetitions, where the CV ratio was 1.5, and the scenario with 100 data points and 50 repetitions, where the CV ratio was 1.07. For σ, the MCMC method also provided more precise estimations except for two scenarios, with 300 data points and 20 repetitions the ratio was 1.28, and with 300 data points and 5 repetitions the ratio was 1.04.

VI. Discussion

When choosing which statistical technique should be used to investigate a given research question, there's more to consider than just the potential of the analysis outcome. Linear regression with MLE (in particular, OLS) is so popular due to its simplicity, versatility, and good quality of fit (in most cases). OLS parameter estimation provides the optimal estimates for the observed data, with very minimal modifications required to standard algorithms available in various statistical software options. A more sophisticated approach which

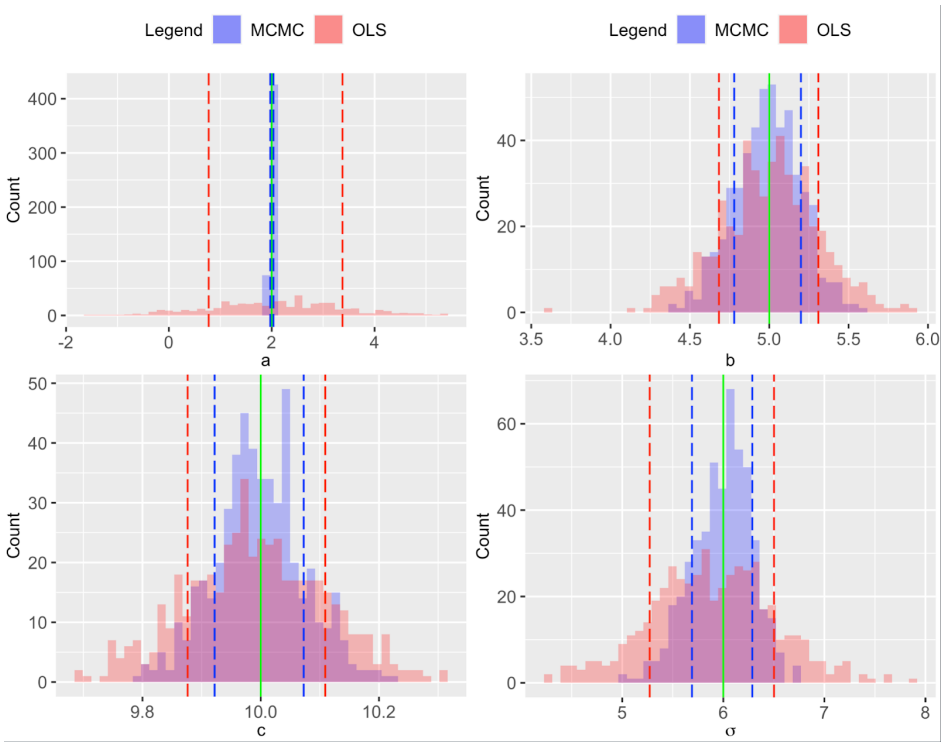

Figure 3: **Parameter estimation distributions from the MCMC algorithm and the aggregated OLS estimations are presented for each of the parameters.** Blue distribution was produced by the MCMC algorithm, along with dashed blue lines representing one standard deviation away from the mean value. Red distribution was produced by the OLS estimations, along with red dashed lines for one standard deviation away from the OLS mean estimation. Green line marks the true parameter value.

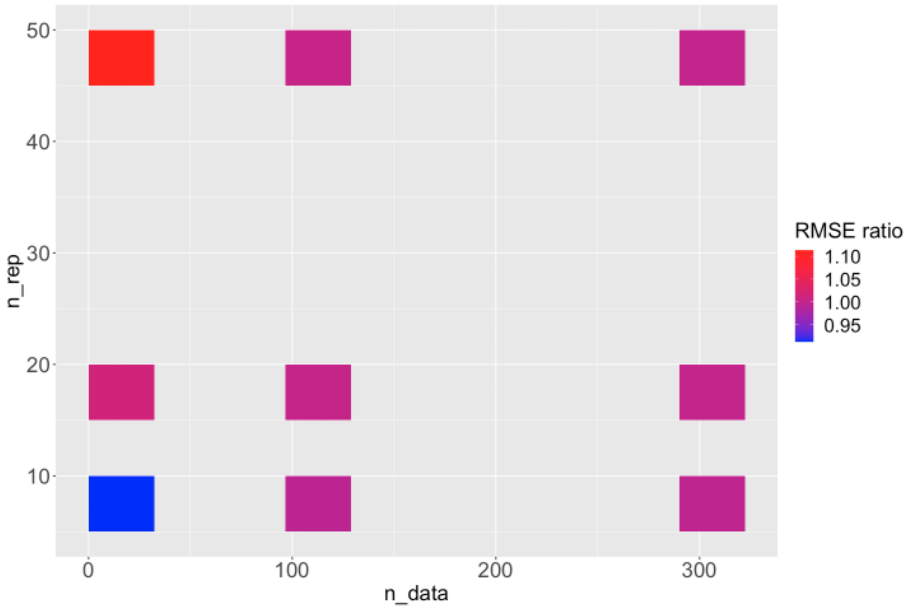

Figure 4: **Low-density heat map of all the RMSE ratio values for all the scenarios.**

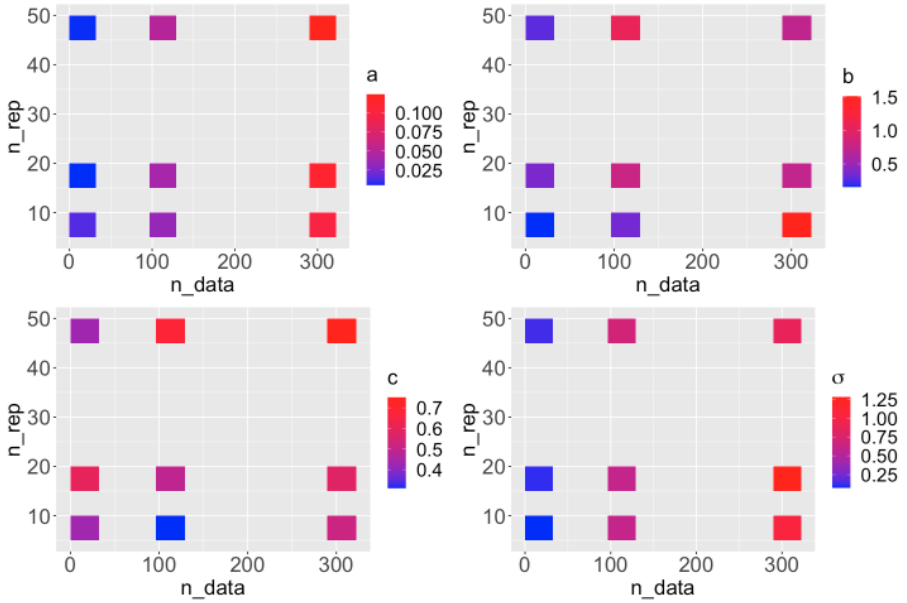

Figure 5: **Low-density heat map of all the CV ratios for each scenario and estimated parameter.**

provides more nuanced analysis such as Bayesian MCMC linear regression has the potential to provide more insightful analysis, however at a price. Setting up an MCMC simulation requires multiple steps, at each, decisions regarding the set up have to be made, which depends in big part on empirical knowledge. Proper definition of the likelihood and the prior distributions is crucial for correct definition of underlying assumptions regarding the data, and could lead to misleading results, or to a failure of the algorithm.

There was some difficulty in the initial tuning of the hyperparameters involved in the MCMC simulation. Namely, proper choice of variance for the proposal distribution, number of walkers, number of steps per walker, number of burn-in steps to discard, how much thinning should be performed, and proper initial guesses for all parameters were all initially not tuned well, which produced erroneous results. As the algorithm was coded from scratch and was not previously verified to be coded correctly, these erroneous results raised the question of whether there are logic mistakes in the code, or improper tuning of hyperparameters. After multiple failed attempts, proper values for all hyperparameters were found, which finally allowed the algorithm to yield valid results. Therefore, such difficulties should always be considered before committing to a statistical technique, and not just the analysis outcome, when undertaking a research question.

Despite the initial difficulties faced with setting up the MCMC algorithm, the superiority of the algorithm when compared to OLS was clear in the results. As can be seen in the estimated parameter distributions, MCMC provides a much more precise estimate when compared to OLS. Despite this, it is important to note that all parameter estimations in all scenarios captured the true parameter value within 1 standard deviation from the estimated mean. Depending on the research question, estimation precision might be much more important than computation runtime, in which case the MCMC approach might be more appropriate rather than OLS. Looking at the CV ratios, for a total of 9 scenarios for the 4 parameters estimated in the simulation (total of 36 CV ratios), MCMC outperformed OLS in all but 4 instances, and even then, the ratio was not far from one (indicating equal CV values). As parameter estimations for MCMC and OLS were very similar across all scenarios, the CV ratio simplifies (approximately) to be the ratio of standard deviations in the parameter estimation between the MCMC to OLS. Considering that most of the CV ratios have a magnitude between 0.1 and 0.7, the superiority of MCMC in estimation precision as compared to OLS is again proven to be significant.

Despite the superior estimation precision, the MCMC approach fails to outperform OLS in terms of the goodness of fit, measured by the RMSE value. Looking at the RMSE ratios in Fig. 4, the magnitude varies from 0.91 to approximately 1.11. This means that at best, MCMC had an RMSE value 9% smaller than that

of OLS for the same scenario (with 10 data points and 5 repetitions), and at worse, MCMC had an RMSE value 11% bigger than that of OLS for the same scenario (with 10 data points and 50 repetitions), indicating that OLS had a better goodness of fit. This is not surprising, considering that in general, despite having a bigger variance, the estimation means of the OLS tend to be very close to the true parameter. Therefore, these variations of the RMSE ratio around one could just be due to random chance. Such details cannot be confirmed with the current work and would require further study.

Due to the initial difficulties faced and the unoptimized code, the scope of this work had to be restricted. If it were possible, rather than only running 9 comparative scenarios with big gaps in simulation parameters, running a bigger number of scenarios with smaller simulation parameter gaps could strengthen the results and potentially unveil a better defined dependency between the scenario parameters (size of data set and number of repetitions) and the effect on the MCMC and the OLS parameter estimations. Likewise, a more diverse set of perturbations could be explored, in order to provide a fuller comparison of the two methods.

VII. Conclusion

In this work, a Bayesian MCMC approach to linear regression parameter estimation was compared to the traditional MLE with OLS criterion. Whereas a Bayesian MCMC approach provides a much better precision to the estimations of the parameters, it also requires a more involved set up prior to model fitting, which can prove challenging. However, both approaches provide a similar overall goodness of fit. Therefore, the right statistical approach largely depends on the research goal at hand. In future work, the algorithm developed here can be optimized to reduce runtime, in order to allow for a wider array of simulation perturbations and collection of simulation results. This in turn would improve the comparative analysis of both methods and provide more guidance regarding the preferred approach to be chosen for different research goals and under different suboptimal conditions, such as noisy data or presence of outliers.

References

W. K. Hastings, Monte Carlo sampling methods using Markov chains and their applications, Biometrika, Volume 57, Issue 1, April 1970, Pages 97–109.

Nicholas. Metropolis et. al., Equation of State Calculations by Fast Computing Machines, J. Chem. Phys. 21, 1087–1092 (1953).

Bayesian perspectives for epidemiological research: I. Foundations and basic methods, International Journal of Epidemiology, Volume 35, Issue 3, 1 June 2006, Pages 765–775 (which is available at https://academic.oup.com/ije/article/35/3/765/735529)

Wickham H (2016). ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York. ISBN 978-3-319-24277-4, https://ggplot2.tidyverse.org.

R Core Team (2021). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/

Hammersley, J.M. and Handscomb, D.C. (1964). Monte-Carlo Methods. Springer Netherlands.

R. Y. Rubinstein. (1981). Simulation and the Monte Carlo Method. John Wiley & Sons, Inc.