Bayesian Estimation Of A Probit Regression Model

An Empirical comparison of the Metropolis-Hastings and Auxiliary Gibbs sampling methodologies

Mert Tekdemir - mert.tekdemir@studbocconi.it
Davide Drago - davide.drago@studbocconi.it
Bernardo Principi - bernardo.principi@studbocconi.it

Contents

1 Introduction

In this document performance diagnostics of two different Bayesian procedures for simulating the posterior distribution of a set of coefficients in a probit regression model are compared. The simulation methods compared are the Metropolis algorithm and the auxiliary Gibbs algorithm. These procedures are compared using both a simulated dataset and a real dataset and for each dataset through different choices of priors, initializations of the coefficients and parameters for the proposal distributions.

The paper begins by introducing first the framework in Section 2 and then the two algorithms in Section 3. A basic understanding on Bayesian statistics and Markov Chain Monte Carlo is assumed. Section 4 introduces the datasets uses for the models and Section 5 compares the performance of the two algorithms for different models on the datasets.

2 Framework

The objective at hand is to study Bayesian procedures for performing probit regression. While Frequentist probit regression considers the data as random and the parameters as fixed, the Bayesian approach considers the parameters as random and the data as fixed. A particular benefit of the Bayesian approach is that the researcher's prior information, not captured within the available data, can be incorporated through the choice of a prior distribution of the parameters. Moreover, beyond simple point estimates, Bayesian inference also allows to estimate the probability distribution of the parameters of interest, which may be further summarized if desired.

To begin, recall the traditional probit regression framework. The dependent variable Y is an $n \times 1$ vector of binary values and the data are stored in an $n \times p$ covariate matrix of continuous and/or categorical data.

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} \quad \text{where} \quad Y_1, ...Y_n \overset{\text{ind.}}{\sim} Bernoulli(\pi_i)$$

$$X = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & & & \\ 1 & x_{n1} & \cdots & x_{np} \end{bmatrix}$$

By considering a vector β of p parameters for the covariates one forms the traditional linear predictor:

$$\eta_i = x_i^T \beta$$

Given the binary dependent variables, the linear predictor is related to the conditional expectation through a link function. In the probit regression setting the link function is the CDF of the standard normal distribution:

$$\eta_i = \Phi^{-1}(\pi_i) \quad \Rightarrow \quad \pi_i = \Phi(\eta_i)$$

As already previously mentioned, when using a Bayesian approach a prior is assigned to the vector of parameters β, $\pi(\beta)$, and all inferences are based on the posterior distribution $\pi(\beta|Y)$. Given that the elements of Y are independent Bernoulli random variables, the likelihood is given by:

$$\mathcal{L}(\beta, Y, X) = \prod_{i=1}^{n} \pi_i^{y_i}(1-\pi_i)^{1-y_i} = \prod_{i=1}^{n} (\Phi(\eta_i))^{y_i}(1-(\Phi(\eta_i))^{1-y_i}$$

And so the posterior distribution can be given up to a normalizing constant through the Bayes' theorem:

$$\pi(\beta|Y) \propto \pi(\beta) \prod_{i=1}^{n} (\Phi(\eta_i))^{y_i} (1 - (\Phi(\eta_i))^{1-y_i}$$

Unfortunately the expression is analytically intractable and so computational methods must be employed to successfully sample from the posterior distribution.

3 Metropolis-Hastings Algorithms

Metropolis-Hastings algorithms are a family of Markov chain Monte Carlo methods for perfect sampling from a possibly multivariate target posterior distribution. Given an initialized value for the parameter vector β_0 and a conditional density $q(\beta|\cdot)$, called a proposal density, at iteration $t+1$ a new sample β_{t+1} is added to the chain using β_t as follows:

- Draw a candidate sample β^* from $q(\beta|\beta_t)$
- Draw U from a uniform(0,1)
- Solve $\alpha(\beta_t, \beta^*) = min\left(1, \frac{\pi(\beta^*|Y)}{\pi(\beta_t|Y)} \frac{q(\beta_t|\beta^*)}{q(\beta^*|\beta_t)}\right)$
- Set $\beta_{t+1} = \beta^*$ if $U \leq \alpha(\beta_t, \beta^*)$
- Otherwise set $\beta_{t+1} = \beta_t$

3.1 Random Walk Metropolis Sampler Algorithm

A random walk Metropolis algorithm is a specific implementation of the MH class of algorithms where a candidate value β^* is proposed by perturbing the most recent value in the chain β_t by ϵ, a random variable having density g independent on the current state β_t. In this way the random walk determines a local exploration of the neighborhood of the current value.

$$\beta^* = \beta_t + \epsilon$$

For the probit regression framework, consider a simple extension of the random walk algorithm where the size of the perturbation changes between iterations. In particular, consider a Gaussian random walk where the distribution of ϵ is a multivariate normal centered at zero with variance-covariance matrix given by the inverse of the Fisher information matrix evaluated at β_t and scaled by a parameter τ.

$$\beta^* = \beta_t + \epsilon_t,\ \epsilon_t \sim N(0,\tau V)$$
$$V = (-\ell''(\beta_t))^{-1} = \mathbb{I}(\beta_t)^{-1}$$

This is equivalent to considering as the the proposal distribution for iteration t of the MH algorithm the multivariate normal centered at β_t with the same variance-covariance matrix as above.

$$q(\beta^*|\beta_t) = N(\beta_t,\tau V)$$

The Fisher information can be calculated from the data as follows:

$$\mathbb{I}(\beta_t) = \left(X^T W_t X\right)$$

Where W is an $n \times n$ diagonal matrix such that:

$$W_{ii} = \text{Var}(Y_i)^{-1} \left(\frac{\partial \eta_i}{\partial \mu_i}\right)^{-2}$$

Recall that Y_i has the Bernoulli distribution with parameter π_i so that:

$$\text{Var}(Y_i) = \pi_i(1-\pi_i) = \mu_i(1-\mu_i)$$

Also recall that η_i is the CDF of the standard normal distribution so that:

$$\frac{\partial \eta_i}{\partial \mu_i} = \frac{1}{\phi(\eta_i)}.$$

And so the the matrix W simplifies to:

$$W_{ii} = \frac{1}{\mu_i(1-\mu_i)}\phi(\eta_i)^2$$

Finally, notice that by symmetry of the proposal distribution the acceptance probability α simplifies to:

$$\alpha(\beta_t, \beta^*) = min\left(1, \frac{\pi(\beta^*|Y)}{\pi(\beta_t|Y)}\right)$$

Therefore the candidate that increases the posterior distribution is always accepted.

3.2 Auxiliary Variable Gibbs Sampler Algorithm

While the Metropolis algorithm provides one method for sampling from the posterior distribution, convergence may take long since no conjugate prior $\pi(\beta)$ exists for the parameters of the probit regression model. To overcome this problem, an alternative Metropolis-Hastings algorithm called the auxiliary Gibbs sampler may be used. The procedure uses an extension of the Gibbs sampler which introduces additional auxiliary variables in order to render the conditional distributions of the model parameters equivalent to those under a Bayesian normal linear regression model with Gaussian noise.

To begin, first the Gibbs sampler is introduced. The Gibbs sampler is an algorithm commonly used in circumstances where, given a multivariate model, it is easier to sample from the distribution of a parameter conditional on all the other parameters in the model (also called "full-conditional") rather than from its unconditional distribution.

In particular, the procedure is a one-run sampling scheme that can be used to output just one long chain in which convergence is achieved relatively quickly with only few observations discarded.

Given the set of parameters of interest $(\beta_0, \beta_1, ..., \beta_p)$ initialized at $(\beta_0^{(0)}, \beta_1^{(0)}, ..., \beta_p^{(0)})$, for t in the range $(0, T-1)$, where T is the desired length of the MCMC, do the following:

- Simulate $\beta_0^{(t+1)}$ from $\pi(\beta_0 \mid \{\beta_j^{(t)}, j \neq 0\})$
- Simulate $\beta_1^{(t+1)}$ from $\pi(\beta_1 \mid \beta_0^{(t+1)}, \{\beta_j^{(t)}, j > 1\})$

  $\vdots$
- Simulate $\beta_p^{(t+1)}$ from $\pi(\beta_p \mid \{\beta_j^{(t+1)}, j < p\})$

The procedure at each iteration performs p Metropolis-Hastings steps with an acceptance probability of 1 where the (certain) candidate values are drawn from the full-conditional distributions. Further, it can be shown that after a certain period t^*, the values of β^t with $t > t^*$ generated by the algorithm can be considered as draws from the joint posterior distribution. However, the auto-correlations for nearby draws are very high and so while the values can be used to estimate the moments of the posterior distribution, the batched-means procedure should be employed to compute standard errors.

In an auxiliary Gibbs Sampler, the goal is to obtain a representative sample of the parameter of interest by introducing a vector $(Z_1, Z_2, ..., Z_n)$ of auxiliary variables on which the parameter of

interest can be conditioned in order to be able to sample from its full-conditional. In particular, each Z_i is defined as follows:

$$\pi(Z_i \mid \beta) \overset{\text{ind.}}{\sim} N(x_i^T \beta, 1)$$

$$\mathrm{Y}_i = \begin{cases} 1 \text{ if } Z_i > 0 \\ 0 \text{ if } Z_i \leq 0 \end{cases}$$

This formulation of the auxilary variables is particularly useful as any probit model can be expressed in terms of a multivariate linear regression model of the auxiliary variables on the covariates:

$$Y_i = 1 \iff Z_i > 0$$

$$Z_i = x_i^T \beta$$

A Gibbs sampler is then applied to draw values from $\pi(\beta, Z \mid Y)$.

- At each iteration a pair of values (β_t, Z_t) is obtained, by sampling from the corresponding full-conditionals $\pi(\beta \mid Z, y)$ and $\pi(Z \mid \beta, Y)$.
- Only draws β_t are stored and used for solving the problem of interest

What remains for the implementation of the algorithm is the derivation of the full conditionals $\pi(\beta \mid Z, y)$ and $\pi(Z \mid \beta, Y)$.

3.2.1 Derivation of the full conditionals

In order to derive the full conditionals for the two parameters of interest β and Z, begin by considering the following joint *p.d.f.*:

$$\pi(\beta, Z \mid y) = C\pi(\beta) \prod_{i=1}^{n} \pi(Z_i \mid \beta, y) \pi(y_i | z_i)$$

Where

$$\pi(Z_i \mid \beta, y) = \phi(Z_i; X_i^T \beta, 1)$$

$$\pi(y_i \mid z_i) = \{\mathbb{1}(Z_i > 0)\mathbb{1}(y_i = 1) + \mathbb{1}(Z_i \leq 0)\mathbb{1}(y_i = 0)\}$$

Next, to compute the full conditional for β first note that it is conditionally independent of y given Z due to the deterministic relationship between Z and y. Therefore $\pi(\beta \mid Z, y) = \pi(\beta \mid Z)$. Moreover, by applying the Bayes' theorm:

$$\pi(\beta \mid Z, y) = \pi(\beta \mid Z) = \frac{\pi(\beta)\pi(Z \mid \beta)}{\pi(Z)} \propto \pi(\beta)\pi(Z \mid \beta)$$

where $\pi(\beta)$ is the prior distribution of the parameters β and $\pi(Z \mid \beta)$ is just a multivariate normal distribution with mean $X_i^T \beta$ and the identity matrix as variance-covariance matrix.

$$\pi(\beta \mid Z, y) \propto \pi(\beta) \prod_{i=1}^{n} N(Z_i; X_i^T \beta, 1)$$

This quantity is actually the posterior density of the normal linear regression model, and so under a (constant) non-informative prior ($\pi(\beta) \propto 1$) the standard linear model results can be applied:

$$\pi(\beta \mid Z, y) \sim N((X^T X)^{-1} Z, (X^T X)^{-1})$$

However, if instead the conjugate prior $N(\beta_0, V_0)$ is considered the full conditional is given by:

$$\pi(\beta \mid Z, y) \sim N((V_0^{-1} X^T X)^{-1} (V_0^{-1} \beta_0 + X^T Z), (V_0^{-1} X^T X)^{-1})$$

Now consider $\pi(Z \mid \beta, y)$. Independent of y the distribution would have been the one defined in section 3.2. However, given the definition of Z_i, conditional on y, the distribution of the Z_i's must be truncated by 0, at the right if $y_i = 0$ and truncated at the left if $y_i = 1$.

First of all, recall that the *p.d.f.* of a normal distribution centered in μ, having standard deviation σ, truncated at the left at a and at the right at b is:

$$f(x \mid \mu, \sigma, a, b) = \frac{1}{\sigma} \frac{\phi(\frac{x-\mu}{\sigma})}{\Phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{a-\mu}{\sigma})}$$

From this result, it is enough to plug-in the values for the mean and the variance of the different Z_i and the points at which each distribution is truncated to derive the following:

$$f(Z_i \mid \beta, y = 1) = \frac{\phi(Z_i - X_i^T \beta)}{1 - \Phi(0)} \mathbb{1}(Z_i > 0) = 2\phi(Z_i - X_i^T \beta) \mathbb{1}(Z_i > 0)$$

$$f(Z_i \mid \beta, y = 0)) = \frac{\phi(Z_i - X_i^T \beta)}{\Phi(0)} \mathbb{1}(Z_i \leq 0) = 2\phi(Z_i - X_i^T \beta) \mathbb{1}(Z_i \leq 0)$$

Thus conditionally on β and on Y, $(Z_1, ..., Zn)$ are independent variables having Normal density, with expected value $X_i^T \beta$ and variance 1, truncated by 0. The truncation is at the left if $y_i = 0$ and at the right if $y_i = 0$

Finally, note that the distributions are not centered around 0 and the *p.d.f.* of Z_i (without conditioning on y_i) evaluated at two points having the same absolute value but different signs may differ significantly. For instance, suppose $X_i^T \beta = 5$: then for $Z_i = 5$:

$$\phi(5 - 5) = \phi(0) \sim 0.4$$

While for $Z_i = -5$:

$$\phi(-5 - 5) = \phi(-10) \sim 0$$

As a consequence it may sometimes happen that $X_i^T \beta$ evaluates to a value relatively far from 0 and the full conditional is truncated in such a way that the parameter space does not include $X_i^T \beta$ (in the example above, if $y_i = 0$ then $Z_i \in (-\infty; 0]$ while $X_i^T \beta \notin (-\infty; 0]$).

This is an important consideration to handle in the implementation of the algorithm. One solution to this problem is to set as the new candidate $Z_i = 0$ if the ordinary sampling procedure from a truncated normal does not complete successfully.

4 Data

In order to assess the performance of the proposed procedures two different datasets will be considered.

Simulated Data The first dataset has been artificially generated to have three covariates, the first being a constant. The covariates were generated from independent draws from a $N(0, 1)$ and the dependent variable was generated by applying the linear predictor where the parameters were given by $(\beta_0 = 2, \beta_1 = -5, \beta_2 = 1)$. The output of the linear predictor was perturbed by a $N(0, \sigma = 2.5)$ and a threshold value of "2" was chosen for deriving a binary dependent variable.

Real Data The second dataset has instead been derived from the *"UCI ML Wine Data Set"*. To create a binary dependent variable the category "0" has been removed. Because this category corresponds to "bad" quality wine, removing this category also results in a more difficult classification task as the two remaining categories are more related. Additionally, for the sake of simplifying the demonstration only a subset of all the covariates has been selected, "volatile acidity", "citric acid", "residual sugar", and "chlorides". An additional constant term has also been included in the data.

5 Parameter Estimation and Model Performance

5.1 Introduction

5.1.1 Introduction to the evaluation criteria

To evaluate the performances of the two sampling algorithms several different diagnostics are considered. In particular, a focus is placed on exploring the convergence of the algorithms, whether they reach a stationary distribution, the independence of the data produced and the efficiency through which these data are sampled.

Trace Plots An intuition on the convergence of the sampler can be obtained by looking at the Trace Plots. In particular, as shown in the Trace Plot Example [1], two functions are plotted: the sampled values at each iteration in blue, and the moving average (with lag 50) in orange. To assess convergence one expects to see perturbations around the moving average. In general, earlier iterations of the chain will explore the parameter space before stabilizing, the quicker the samples convergence the fewer the number of observation that must be discarded from the beginning of the chain (burn-in).

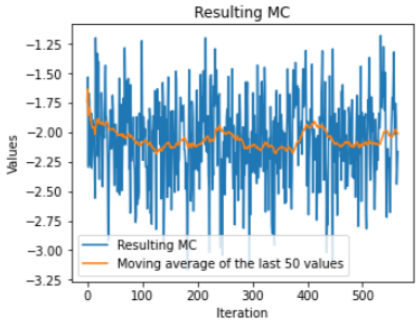

Figure 1: Traceplot example

Autocorrelations To explore the chains ability to generate independent samples from the target distribution autocorrelation plots are used which, for given lag values, plots the average autocorrelation of samples drawn lag steps apart [2]. As evident in the algorithms each iteration of the chain depends on the latest sample, therefore nearby samples are expected to have higher autocorrelation than further samples. To combat this, a sample from the chain may be drawn at a cycle of every k steps. The autocorrelation plot of such a chain should resemble stochastic changes in the plotted values.

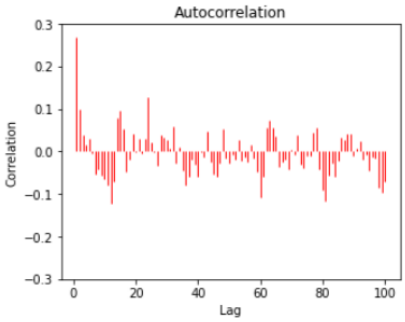

Figure 2: Autocorrelation example

Kernel Densities Intuitively, for a converged chain the distribution of samples drawn at different intervals of the chain must differ only due to chance. Thus as a diagnostic for the convergence of the chain the kernel density estimates of equal batches of a chain may be plotted and compared [3].

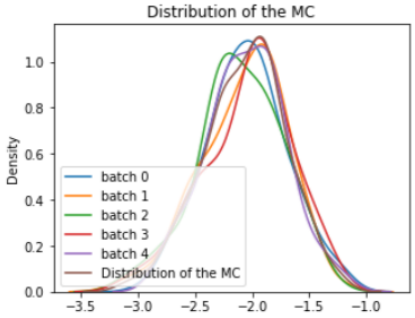

Figure 3: Kernel Densities example

Acceptance Rate An additional diagnostic for convergence is the acceptance rate. As the chain explores the parameter space pertubations in the acceptance rate is expected, however as the chain converges the acceptance rate should settle around a mean value. The moving average of the acceptance rate for the most recent 50 iterations is plotted and used to assess both the convergence and the speed of convergence [4]. A downside of this diagnostic is that it cannot be used for the Gibbs sampler which has a constant acceptance rate of 1.

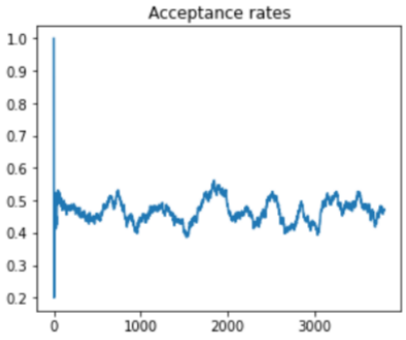

Figure 4: Acceptance Rate example

Multivariate ESS As mentioned, the procedures used are dependent samplers and so if one desires *i.i.d* samples the outputs of the chain should be selected at cycles of k steps. A metric to help select k is the effective sample size (ESS). The ESS aims to calculate the equivalent number of independent samples needed to produce the same amount of information as contained in the dependent samples.

If the ratio between the ESS of a chain and the actual sample size is near to one the sample can be considered as good as independent. Thus, after convergence diagnostics are preformed, the ESS of

converged chains can be compared to identify which of the chains produces samples that can be considered closer to independent.

By splitting a single long Markov chain across different candidate choices for k one effectively creates different chains that can be compared. Through this comparison between the different chains created one should look to select the smallest possible candidate k that produces a chain with an ESS to real sample size ratio near 1. As large values for k will result in discarding a significant portion of the chain one should aim to strike a balance between the ESS and the size of the cycle.

When it comes to estimating the ESS one can either use a univarite or multivariate estimator. A univariate ESS will provide an effective sample size for each parameter separately and conservative methods dictate to consider the smallest estimate. This method ignores all cross-correlations across components. The multivariate ESS (mESS) returns one number for the effective sample size of each parameter and it does so by accounting for all the cross-correlations in the process and is the metric considered in this paper:

$$mESS = n \left(\frac{|(\Lambda)|}{|\Sigma|} \right)^{\frac{1}{p}}.$$

Where:

Λ is the covariance structure of the posterior (also the asymptotic covariance in the CLT if you had independent samples)

Σ is the asymptotic covariance matrix in the Markov chain CLT (different from Λ since samples are correlated).

p is number of quantities being estimated (or in this case, the dimension of the posterior).

$|\cdot|$ is the determinant.

n is the sample size

Finally, the mESS can be estimated by using the sample covariance matrix to estimate Λ and the batch means covariance matrix to estimate Σ.

5.1.2 Introduction to the parameters used

The two samplers are compared across different configurations to asses the efficiency of each procedure in different scenarios.

Initialized Parameters The intialized parameters influence the convergence time but are not expected to affect the estimates of the coefficients. By initalizing the model parameters with varying degrees of prior information and examining the number of iterations required for convergence, the ability of the algorithim to recover from poor starting values and perfom under agnostic situations can be assessed.

Priors The prior distribution assigned to the parameters allows the researcher to incorporate information on the problem not included in the data. For a given informative prior, a lower dispersion corresponds to stronger prior believes, thus leading towards weighting less the information gained through the available observations when computing the posterior distribution with respect to the case of a prior with a greater dispersion. Both improper and proper priors are explored to study their impact on convergence, convergence rates, autocorrelation and parameter estimation.

Taus In particular, the proposal distribution for the Metropolis algorithm has been defined as:

$$q(\beta^*|\beta_t) = N(\beta_t, \tau V)$$

The parameter tau scales the variance-covariance matrix, influencing the result of the models: In fact, for a given variance-covariance matrix the parameter τ controls the acceptance rate of a chain, as higher variance implies more exploration on the data and hence lower acceptance rates.

5.2 Metropolis algorithm

	Chosen Parameters				β_0		β_1		β_2	
	Starting β	*Prior Distr.*	*Tau*	*Burn-in*	*Mean*	*Std. Error*	*Mean*	*Std. Error*	*Mean*	*Std. Error*
MLE	*Maximum Likelihood Estimates*				-0.168	0.200	-2.617	0.542	1.044	0.307
Model 1	β_start ~ N(0, 4)	~ N(β_start, 1)	0.45	300	-0.013	0.171	-1.630	0.275	0.744	0.210
Model 2	β_start ~ N(0, 4)	Non-informative	0.45	300	-0.142	0.184	-2.522	0.494	1.010	0.294
Model 3	β_MLE	Non-informative	0.45	200	-0.159	0.194	-2.505	0.521	0.990	0.289
Model 4	β_MLE	~ N(β_MLE, 1)	0.45	200	-0.142	0.196	-2.498	0.414	1.010	0.257
Model 5	β_MLE	~ N(β_MLE, 5)	0.45	200	-0.172	0.195	-2.483	0.509	0.991	0.298
Model 6	β_MLE	~ N(β_MLE, 1)	0.30	400	-0.178	0.209	-2.504	0.442	1.003	0.271

Figure 5: Metropolis models comparison on simulated data

	Chosen Parameters				β_0		β_1		β_2		β_3		β_4	
	Starting β	*Prior Distr.*	*Tau*	*Burn-in*	*Mean*	*Std. Error*	*Mean*	*Std. Error*	*Mean*	*Std. Error*	*Mean*	*Std. Error*	*Mean*	*Std. Error*
MLE	*Maximum Likelihood Estimates*				-3.008	1.634	2.344	0.808	0.006	0.069	0.009	0.010	-1.946	0.357
Model 1	β_start ~ N(0, 4)	~ N(β_start, 1)	0.45	400	0.160	0.833	1.656	0.574	-0.027	0.059	0.000	0.008	-1.913	0.329
Model 2	β_start ~ N(0, 4)	Non-informative	0.45	400	-2.971	1.618	2.183	0.746	0.015	0.065	0.009	0.010	-1.876	0.329
Model 3	β_MLE	Non-informative	0.45	200	-2.749	1.565	2.094	0.823	0.012	0.068	0.009	0.010	-1.835	0.351
Model 4	β_MLE	~ N(β_MLE, 1)	0.45	200	-2.976	0.855	2.189	0.581	0.017	0.059	0.009	0.009	-1.908	0.312
Model 5	β_MLE	~ N(β_MLE, 5)	0.45	200	-2.816	1.378	2.137	0.741	0.012	0.071	0.010	0.010	-1.882	0.356
Model 6	β_MLE	~ N(β_MLE, 1)	0.30	200	-2.951	0.845	2.270	0.565	0.003	0.055	0.009	0.008	-1.833	0.310

Figure 6: Metropolis models comparison on real data

5.2.1 Initalized Parameters

The ergodic theorm suggest that the itialized parameters should not have an impact on the eventual convergence of the chain. Thus, for two runs of the algorithim with different inital parameters, all else held equal both output chains are expected to converge to similar values however the chain with the poorer selection for the initalized values is expected to take longer in such a convergence and therefore have a large burn-in.

Supporting this point is the comparison between Model 2 and Model 3 [5, 6]. Both models are equal in all parameters except the initalized values, where Model 2 is initalized at random and Model 3 is initalized using the probit regression maximum likelihood estimates of the parameters. The posterior distributions and their summaries are very similar between the two models both in simulated data [12, 13] and in real data [22, 23]. However the informed initialization is more efficient requiring 100 fewer estimates to be burned from the start of the chain before convergence.

5.2.2 Priors

Considering priors, all else held equal a more assertive prior distribution will more heavily weight the researcher's prior information as opposed to the data. As a result, unless the researcher has a strong intuition or strong support for their prior belief it is often more effective to remain agnostic.

This is emphasized in the comparison of Model 1 and Model 2. Model 1 places a strong prior distribution on randomly initialized parameters while Model 2 uses a non-informative improper prior [5, 6]. It is seen that while both chains converged the posterior distributions are very different both in simulated data [11, 12] and in real data [21, 22]. In fact, the distribution summaries for Model 1 are far from those of any other model and in particular from those of the benchmark ones obtained through the MLE estimation.

On the other hand, when using the same informed starting point such as the β_{MLE}, between two models with a more assertive and less assertive prior not much difference is seen. For example, consider Model 4 and Model 5 whose only difference is the dispersion of the prior distribution. The estimated posterior distributions both in simulated data [14, 15] and in real data [24, 25] are similar and the burn-in rates are the same.

5.2.3 Taus

As previously discussed, the parameter τ influences the range of the parameter space which the chain is likely to explore at each iteration. In particular, a higher τ implies a higher variance of the proposal distribution and thus results in successive drafts which are on average farther apart from each other. Depending on the actual distribution of the posterior and on the starting value

of the chain, this parameter may also eventually impact the average acceptance rate of the candidates throughout the chain as well as the speed at which the chain converges towards a stationary distribution.

As for the deployment of the model on the considered data, it is possible to observe that decreasing τ from 0.45 to 0.30 in Model 6 seems to lead to an higher acceptance rate on the real data [10], while on the simulated ones [9] the main impact appears to be an increased number of iterations needed to achieve convergence, thus resulting in an higher burn-in and a less efficient sampling procedure.

5.3 Auxiliary Gibbs Sampling

	Chosen Parameters			β_0		β_1		β_2	
	Starting β	*Prior Distr.*	*Burn-in*	*Mean*	*Std. Error*	*Mean*	*Std. Error*	*Mean*	*Std. Error*
MLE	*Maximum Likelihood Estimates*			-0.168	0.200	-2.617	0.542	1.044	0.307
Model 1	β_start ~ N(0, 4)	~ N(β_start, 1)	50	-0.010	0.181	-1.804	0.334	0.832	0.238
Model 2	β_start ~ N(0, 4)	Non-informative	100	-0.169	0.188	-2.734	0.517	1.093	0.314
Model 3	β_MLE	Non-informative	50	-0.180	0.196	-2.798	0.496	1.112	0.304
Model 4	β_MLE	~ N(β_MLE, 5)	100	-0.188	0.208	-2.829	0.546	1.128	0.309

Figure 7: Auxiliary Gibbs models comparison on simulated data

	Chosen Parameters			β_0		β_1		β_2		β_3		β_4	
	Starting β	*Prior Distr.*	*Burn-in*	*Mean*	*Std. Error*	*Mean*	*Std. Error*	*Mean*	*Std. Error*	*Mean*	*Std. Error*	*Mean*	*Std. Error*
MLE	*Maximum Likelihood Estimates*			-3.008	1.634	2.344	0.808	0.006	0.069	0.009	0.010	-1.946	0.357
Model 1	β_start ~ N(0, 4)	~ N(β_start, 1)	100	-3.008	0.838	2.383	0.603	0.008	0.060	0.008	0.009	-1.995	0.329
Model 2	β_start ~ N(0, 4)	Non-informative	150	-3.004	1.651	2.493	0.811	0.002	0.070	0.008	0.010	-2.043	0.372
Model 3	β_MLE	Non-informative	50	-2.916	1.613	2.460	0.782	0.004	0.067	0.008	0.010	-2.064	0.349
Model 4	β_MLE	~ N(β_MLE, 5)	50	-3.017	1.273	2.456	0.741	0.007	0.067	0.008	0.010	-2.059	0.345

Figure 8: Auxiliary Gibbs models comparison on real data

5.3.1 Initalized parameters

As seen in the Metropolis sampler above, the choice of the starting points should not have an impact on the estimation of the posterior distribution, but it may influence the speed at which the model convergences and the efficiency of the sample drawn. Looking at the results obtained from comparing Model 1 [17] [27] with Model 4 [20] [30] and Model 2 [18] [28] with Model 3 [19] [29], both in simulated data and real data, it is seen that the algorithm is able to converge faster if initialized from points closer to the actual mean of the distribution of the parameter of interest without significant penalization on the estimation itself.

5.3.2 Priors

Regarding the choice of priors the conclusions drawn from the Gibbs sampler is again similar to the one of the Metropolis. Employing a prior heavily centered on the initialized parameter values may penalize the model and lead to a poor estimation of the posterior distribution when the initalization is done randomly. This is seen from the results of Model 1 on the simulated data [17] (even though the results on the real data [27] seems less alarming).

Moreover, as one would expect, the initialization matters much less if a non-informative prior is used as the data is heavily favored in converging to the posterior distribution. In particular, the informative prior centered in the estimates obtained through maximum likelihood in Model 4 seems not to grant any further improvement in terms of performance to the non-informative prior used in Model 3. This suggests that in the case of the Gibbs sampler it may be better to remain conservative when it comes to the choice of a prior distribution as the convergence of the Gibbs sampler occurs quickly and accurately in both cases.

Finally, from the results of the mESS procedure described in paragraph 5.1.1, there is no clear indication of what parameters may influence the efficiency of the models in terms of cycle size. On the other hand, the optimal step size decreases notably from simulated to real data. In fact, 22.5 is the average size - across models - in simulated data, while it goes down to 8.5 for real data models.

6 Model Comparisons and Concluding Remarks

Both the Metropolis and Gibbs Sampler algorithms are well studied and proven to converge, resulting in a perfect sampling from the target distribution. In fact, results in this paper between the two samplers are seen to be very similar in terms of the distribution of the posteriors and their summaries. Thus, the primary basis for comparison must be their efficiency.

In terms of efficiency a comment can be made on two aspects, the number of discarded samples before the chain converges, the length of the cycle size needed to create an effective independent sample and the time needed to run the algorithm. Note that, both the first two are a measure of the number of additional iterations required from a chain before it serves its purpose.

Regarding the burn-in, in all examples the Auxillary Gibbs sampler is seen to converge quicker than the Metropolis sampler and can be considered more efficient. This was indeed the expected result as the introduction of the auxiliary variables allows to leverage the conjugate prior, however this is also seen to be the case when a non-informative improper prior is used.

Considering the cycle size used in the models and the running time, the Auxiliary Gibbs Sampler performs better than the Metropolis algorithm. As explained in paragraph 5.1.1, the size has been chosen by calculating Multivariate mESS for different candidates. To visualize the results the mESS to actual sample size ratio was plotted for each at each candidate step, shown in Figure 31. When comparing the cycle size, the Auxillary Gibbs sampler requires significantly fewer steps across all models run on the real dataset. Contrary to expectations, for the sample data, the latter does not differentiate from Metropolis in simulated data. On the other hand, in both real and simulated data, Auxiliary Gibbs Sampling has a much lower running time.

Finally, while frequentest methodologies for regression are widely studied the Bayesian framework of similar problems are less ubiquitous. This is indeed a shame as a Bayesian approach allows researchers to study more than point estimates and can fortify an experiment by assigning a prior distribution according to the literature on the topic. However, while the use of priors represents a strong tool in the researcher's arsenal, this paper has also shown that it should be used with care, as misguided priors can strongly influence a model's final result. One suggestion is for researchers to study a dual problem, running both a null model using an agnostic initialization and prior and comparing it to a full model. This can serve to add insight on the significance of the theory guiding a researcher's choice of prior distribution and initializing values.

7 Appendix

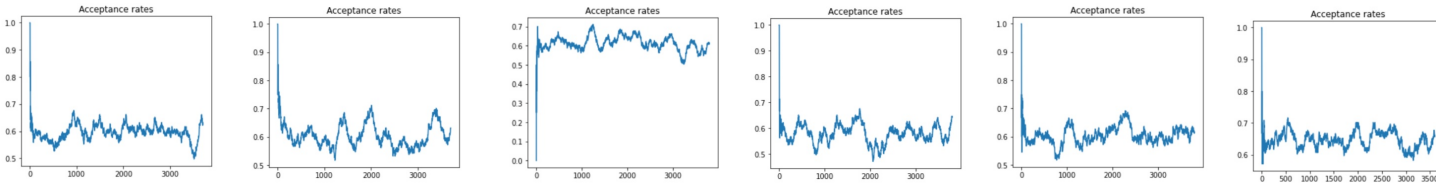

Figure 9: Moving average of the proportion of accepted candidates over time for the six models on **simulated data** discussed in the report.

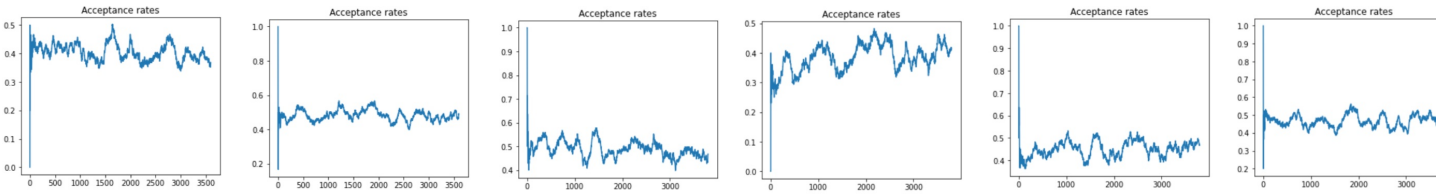

Figure 10: Moving average of the proportion of accepted candidates over time for the six models on **realized data** discussed in the report.

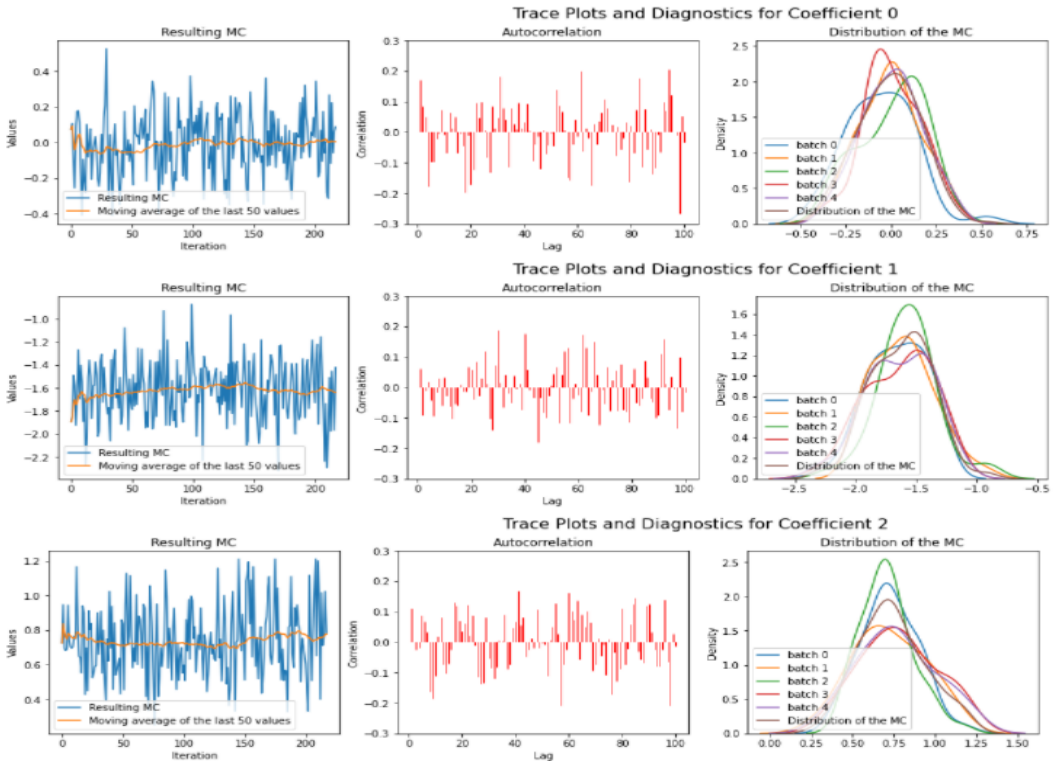

Figure 11: Traceplots, autocorrelations and densities for the coefficients β_0, β_1 and β_2 in the **Metropolis model** on **simulated data** with **randomly initialized coefficients** and **noninformative prior**. The variance of the proposal distribution is scaled by $\tau = 0.45$.

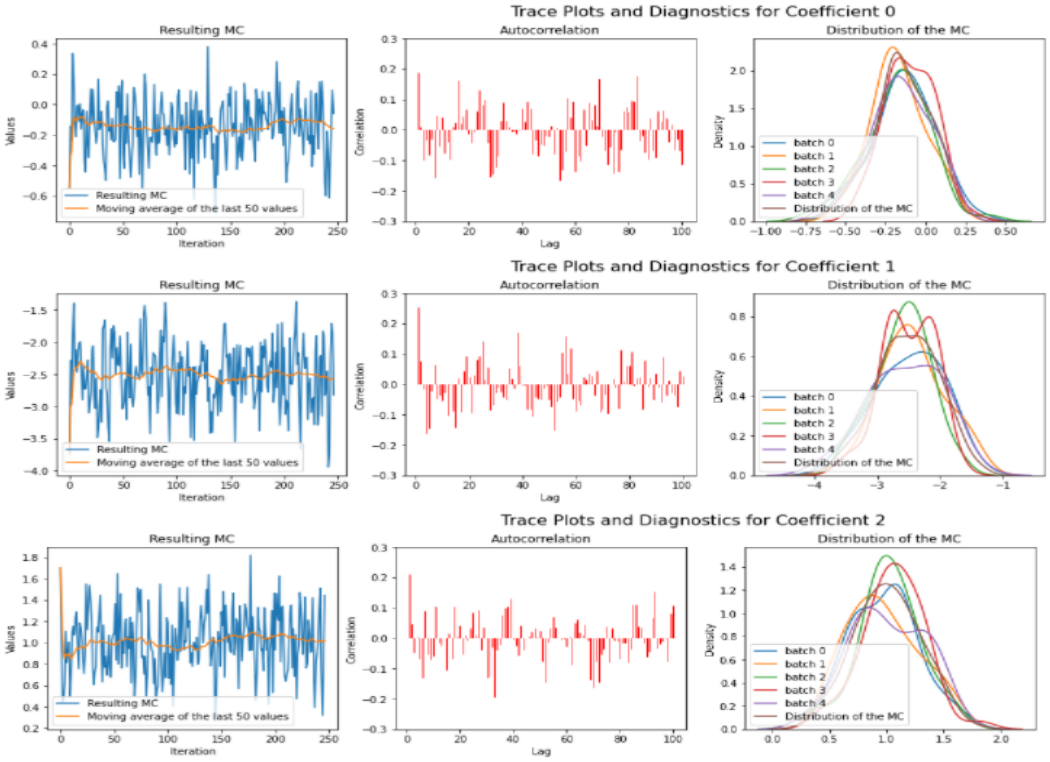

Figure 12: Traceplots, autocorrelations and densities for the coefficients β_0, β_1 and β_2 in the **Metropolis model** on **simulated data** with **randomly initialized coefficients** and as **prior a Normal centered in such values with the identity matrix as variance-covariance matrix**. The variance of the proposal distribution is scaled by $\tau = 0.45$.

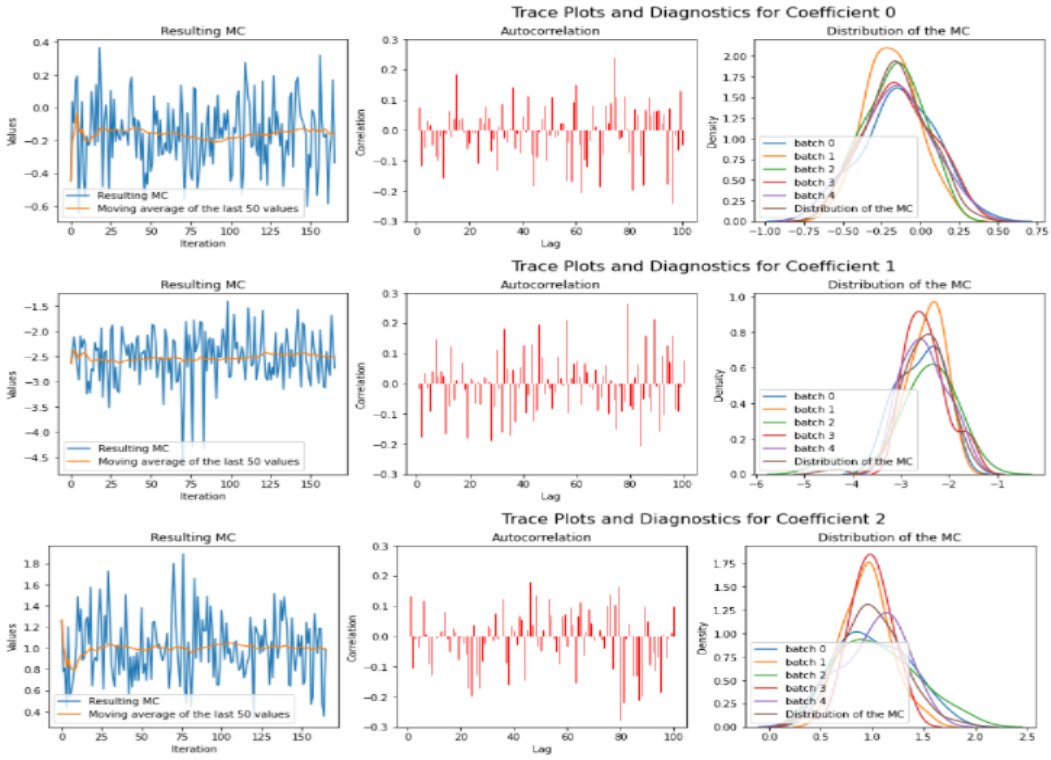

Figure 13: Traceplots, autocorrelations and densities for the coefficients β_0, β_1 and β_2 in the **Metropolis model** on **simulated data** with coefficient initialized at the **maximum likelihood estimates** and **noninformative prior**. The variance of the proposal distribution is scaled by $\tau = 0.45$.

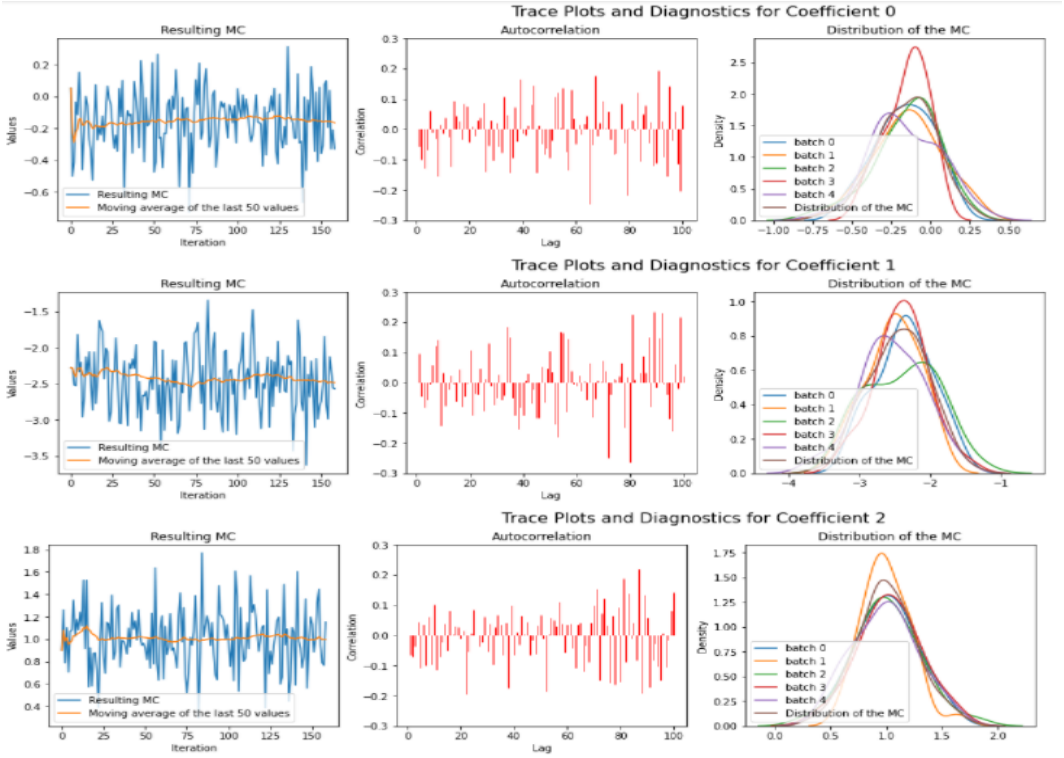

Figure 14: Traceplots, autocorrelations and densities for the coefficients β_0, β_1 and β_2 in the **Metropolis model** on **simulated data** with **coefficient initialized at the maximum likelihood estimates** and as **prior a Normal centered in such values with the identity matrix as variance-covariance matrix**. The variance of the proposal distribution is scaled by $\tau = 0.45$.

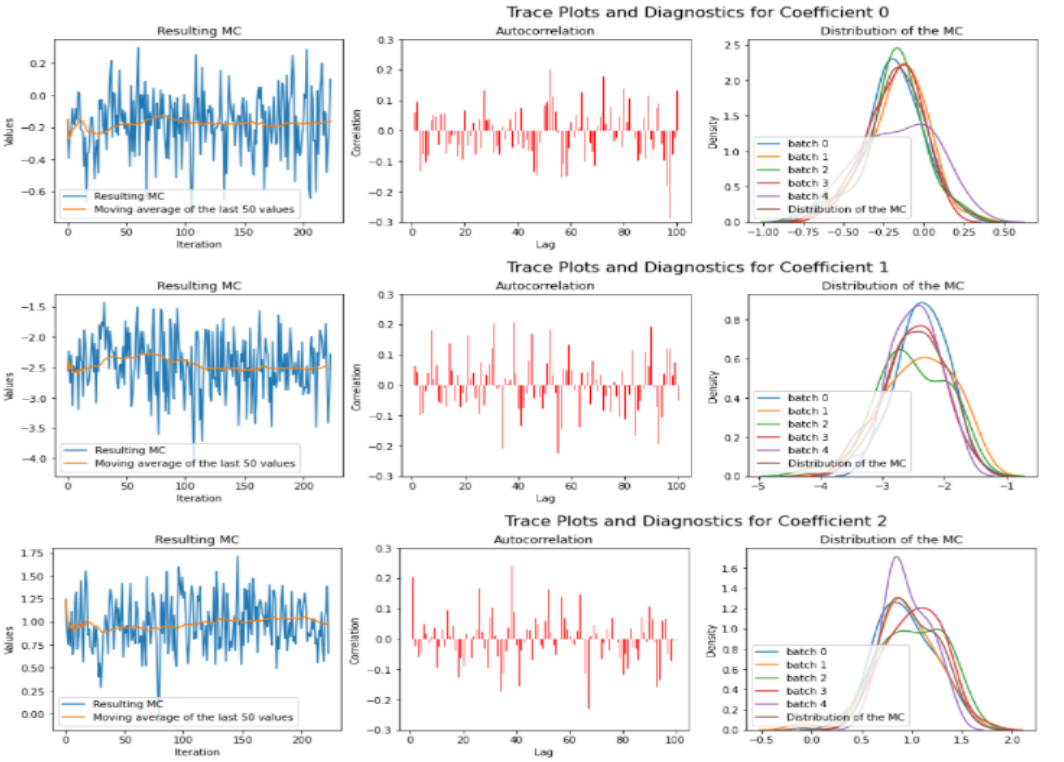

Figure 15: Traceplots, autocorrelations and densities for the coefficients β_0, β_1 and β_2 in the **Metropolis model** on **simulated data** with coefficient initialized at the **maximum likelihood estimates** and as **prior a Normal centered in such values with the identity matrix as variance-covariance matrix**. The variance of the proposal distribution is scaled by $\tau = 0.30$.

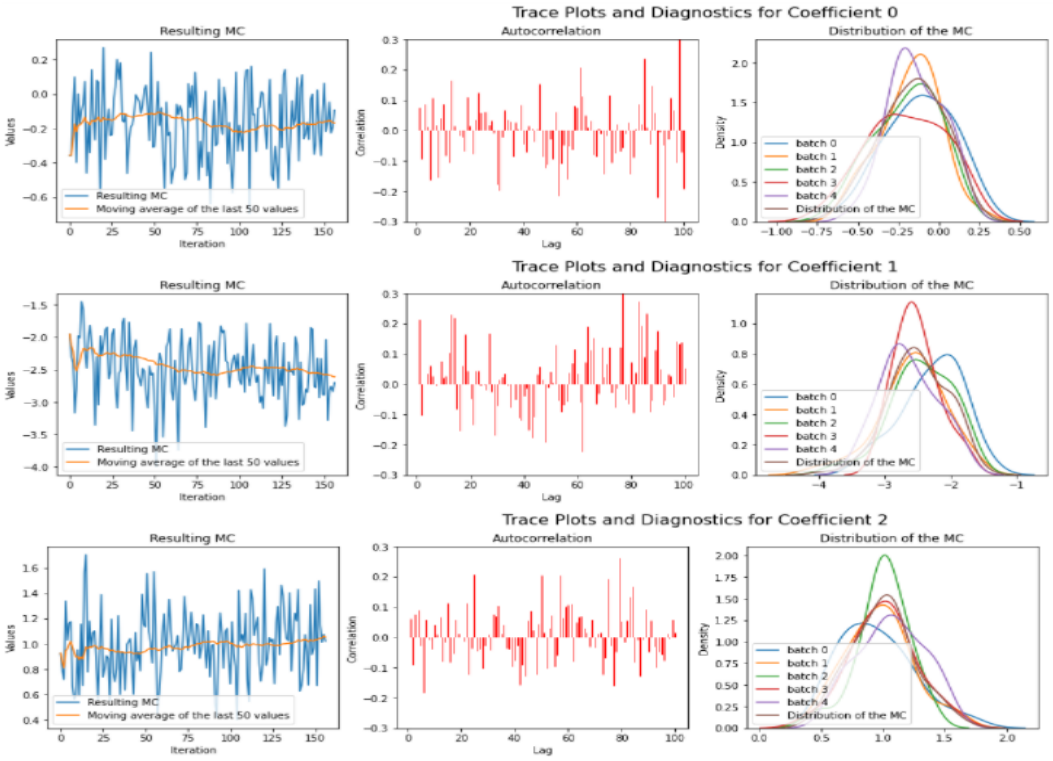

Figure 16: Traceplots, autocorrelations and densities for the coefficients β_0, β_1 and β_2 in the **Metropolis model** on **simulated data** with **coefficient initialized at the maximum likelihood estimates** and as **prior a Normal centered in such values with the identity matrix as variance-covariance matrix**. The variance of the proposal distribution is scaled by $\tau = 0.45$.

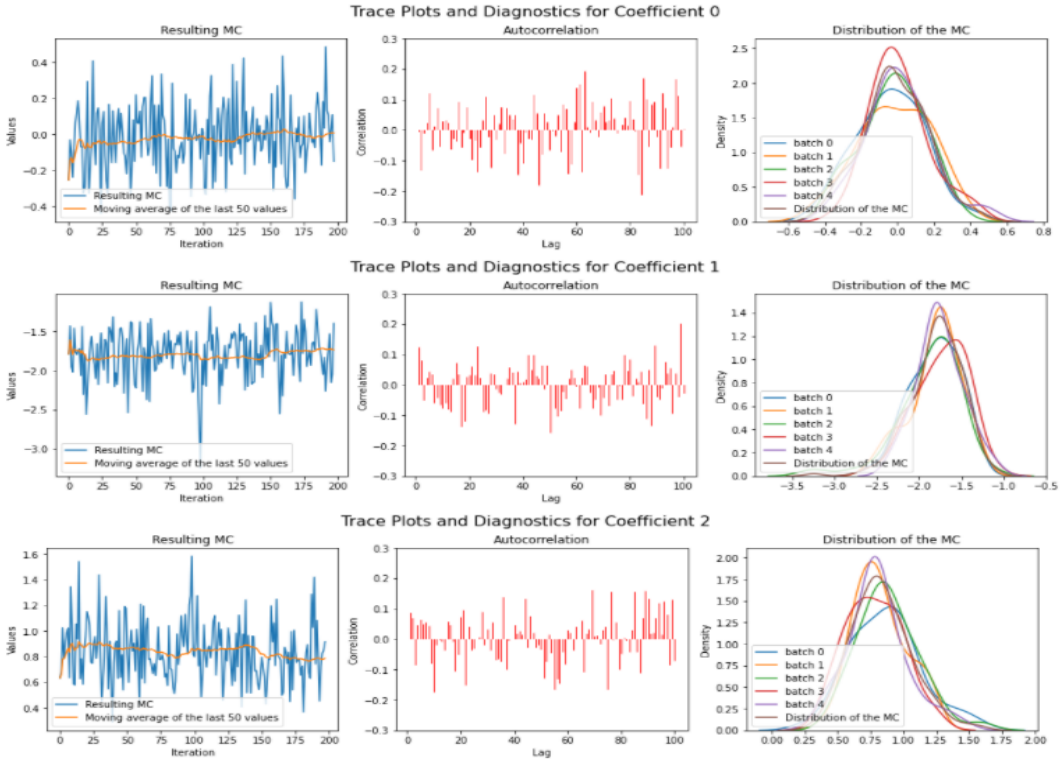

Figure 17: Traceplots, autocorrelations and densities for the coefficients β_0, β_1 and β_2 in the **Auxiliary Gibbs model** on **simulated data** with **randomly initialized coefficients** and as **prior a Normal ccentered in such values with the identity matrix as variance-covariance matrix**.

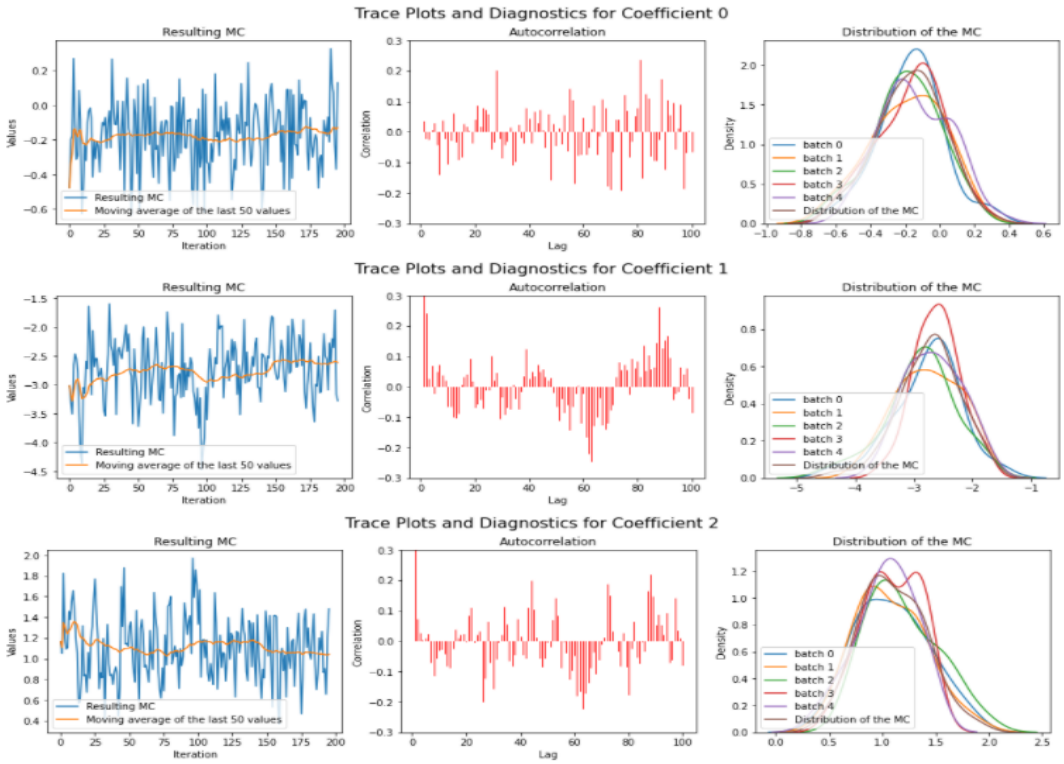

Figure 18: Traceplots, autocorrelations and densities for the coefficients β_0, β_1 and β_2 in the **Auxiliary Gibbs mode** on **simulated data** with **randomly initialized coefficients** and **noninformative prior**.

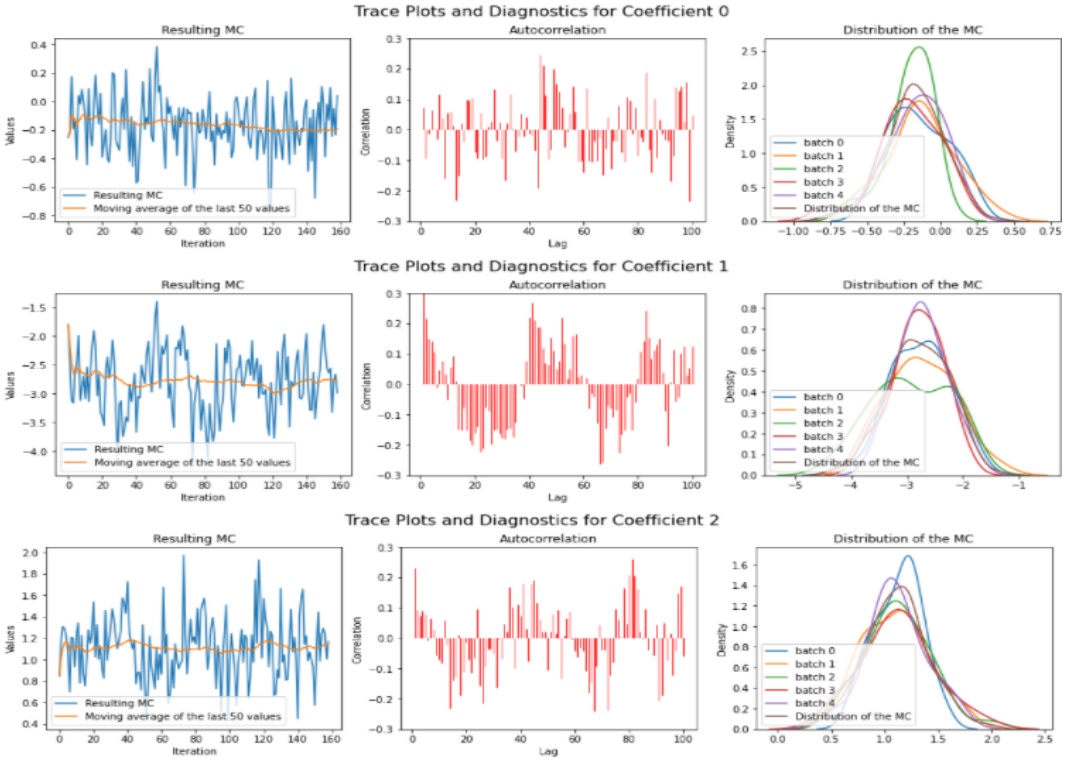

Figure 19: Traceplots, autocorrelations and densities for the coefficients β_0, β_1 and β_2 in the **Auxiliary Gibbs model** on **simulated data** with **coefficient initialized at the maximum likelihood estimates** and **noninformative prior**.

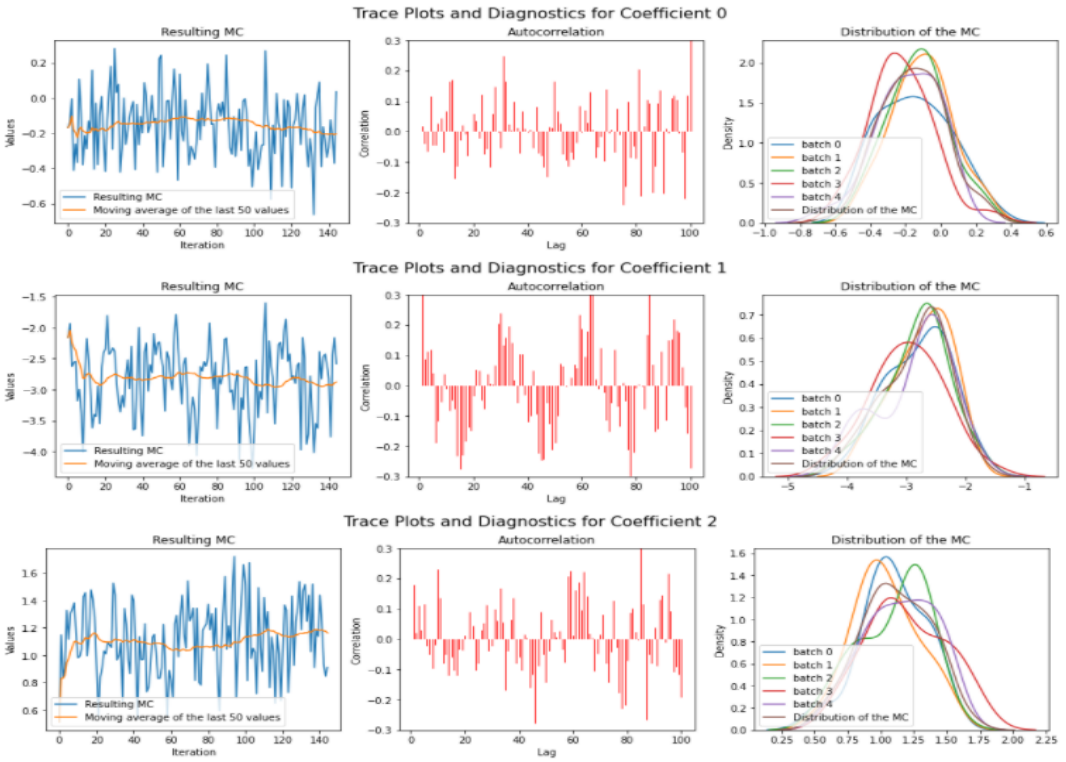

Figure 20: Traceplots, autocorrelations and densities for the coefficients β_0, β_1 and β_2 in the **Auxiliary Gibbs mode** on **simulated data** with **coefficient initialized at the maximum likelihood estimates** and as **prior a Normal centered in such values with an identity matrix scaled by a factor of 5 as variance-covariance matrix**.

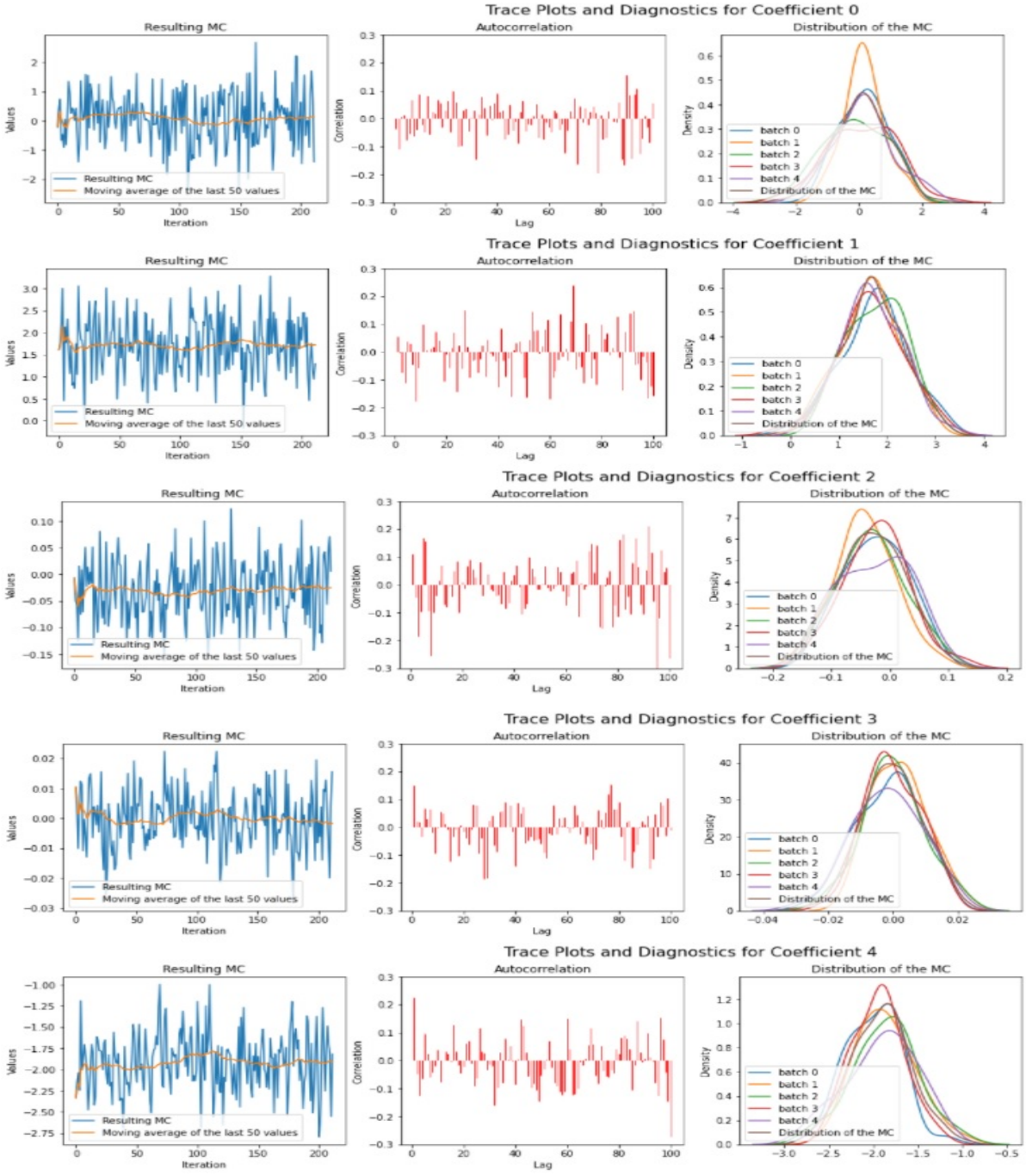

Figure 21: Traceplots, autocorrelations and densities for the coefficients β_0, β_1 and β_2 in the **Metropolis model** on **real data** with **randomly initialized coefficients** and **noninformative prior**. The variance of the proposal distribution is scaled by $\tau = 0.45$.

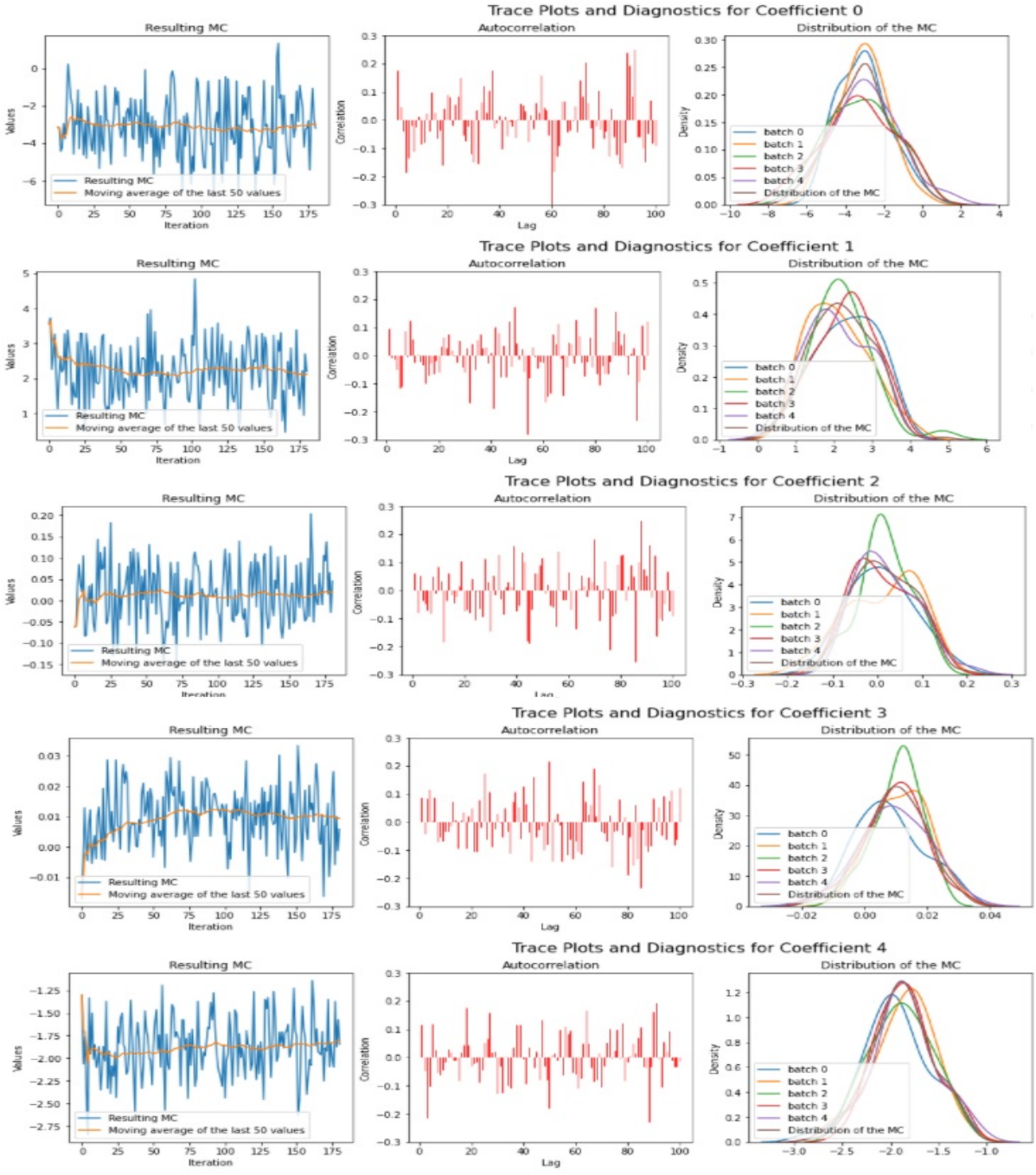

Figure 22: Traceplots, autocorrelations and densities for the coefficients β_0, β_1 and β_2 in the **Metropolis model** on **real data** with **randomly initialized coefficients** and as **prior a Normal centered in such values with the identity matrix as variance-covariance matrix**. The variance of the proposal distribution is scaled by $\tau = 0.45$.

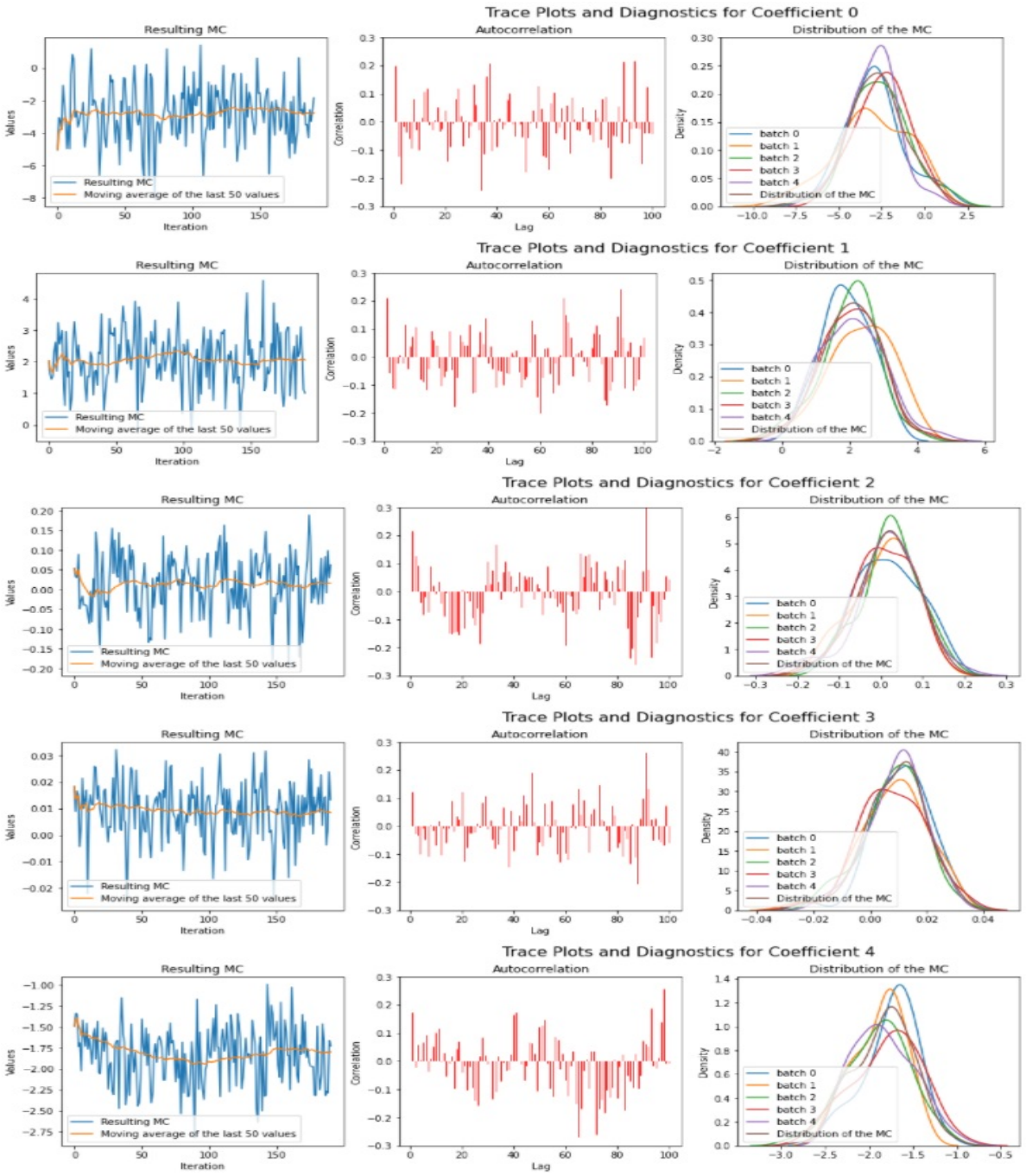

Figure 23: Traceplots, autocorrelations and densities for the coefficients β_0, β_1 and β_2 in the **Metropolis model** on **real data** with coefficient initialized at the **maximum likelihood estimates** and **noninformative prior**. The variance of the proposal distribution is scaled by $\tau = 0.45$.

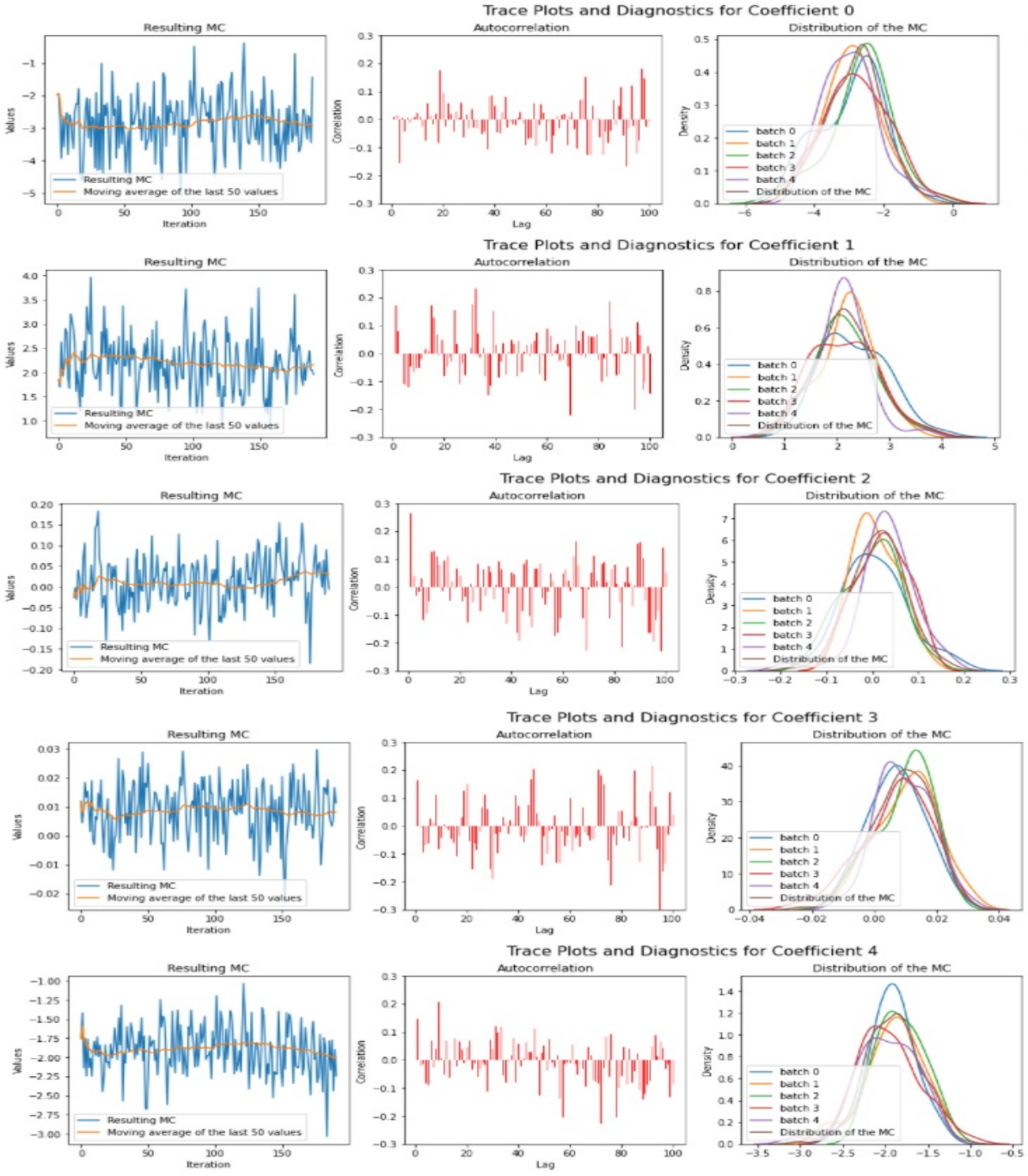

Figure 24: Traceplots, autocorrelations and densities for the coefficients β_0, β_1 and β_2 in the **Metropolis model** on **real data** with **coefficient initialized at the maximum likelihood estimates** and as **prior a Normal centered in such values with the identity matrix as variance-covariance matrix**. The variance of the proposal distribution is scaled by $\tau = 0.45$.

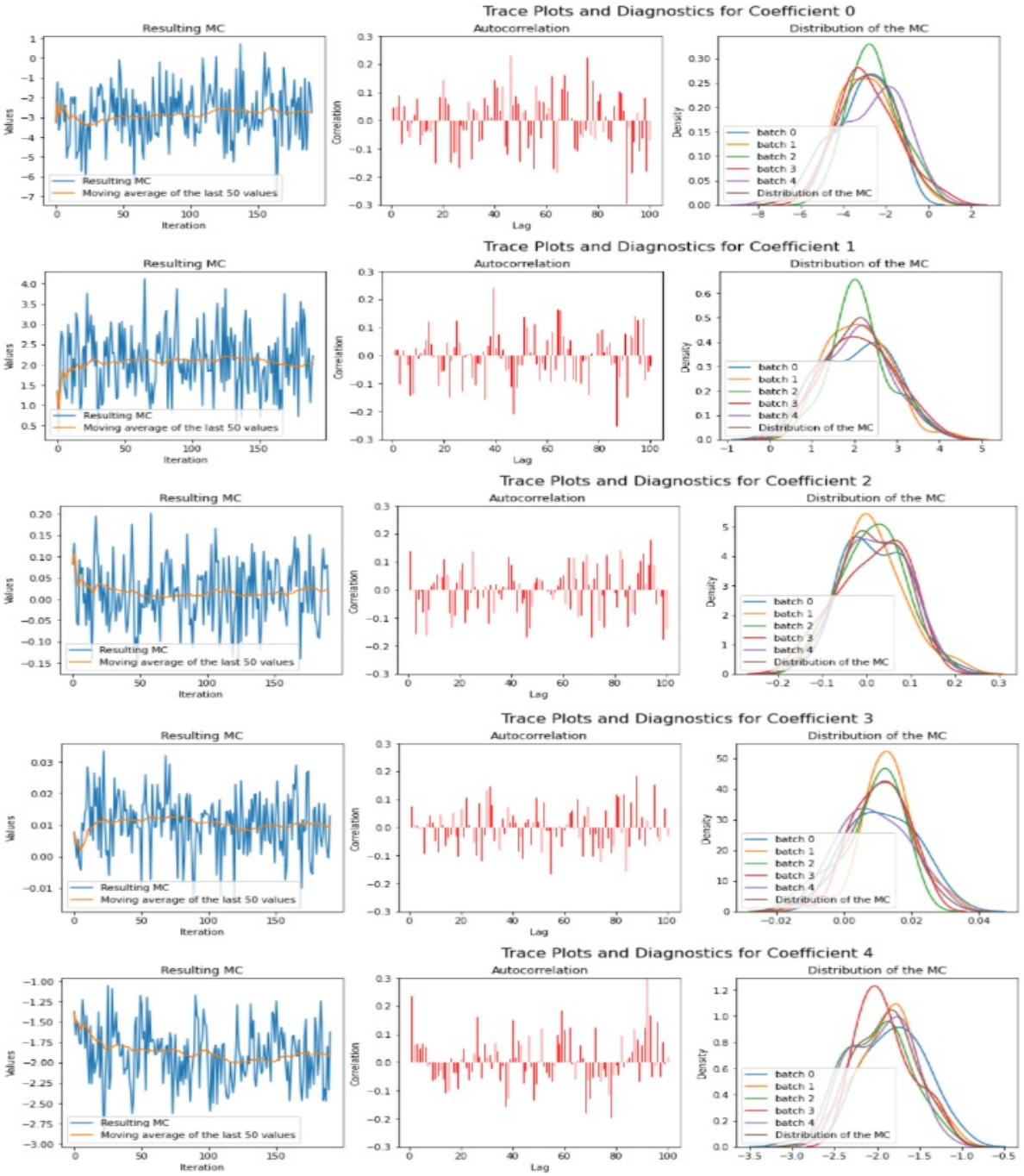

Figure 25: Traceplots, autocorrelations and densities for the coefficients β_0, β_1 and β_2 in the **Metropolis model** on **real data** with coefficient initialized at the **maximum likelihood estimates** and as **prior a Normal centered in such values with the identity matrix scaled by a factor of 5 as variance-covariance matrix**. The variance of the proposal distribution is scaled by $\tau = 0.30$.

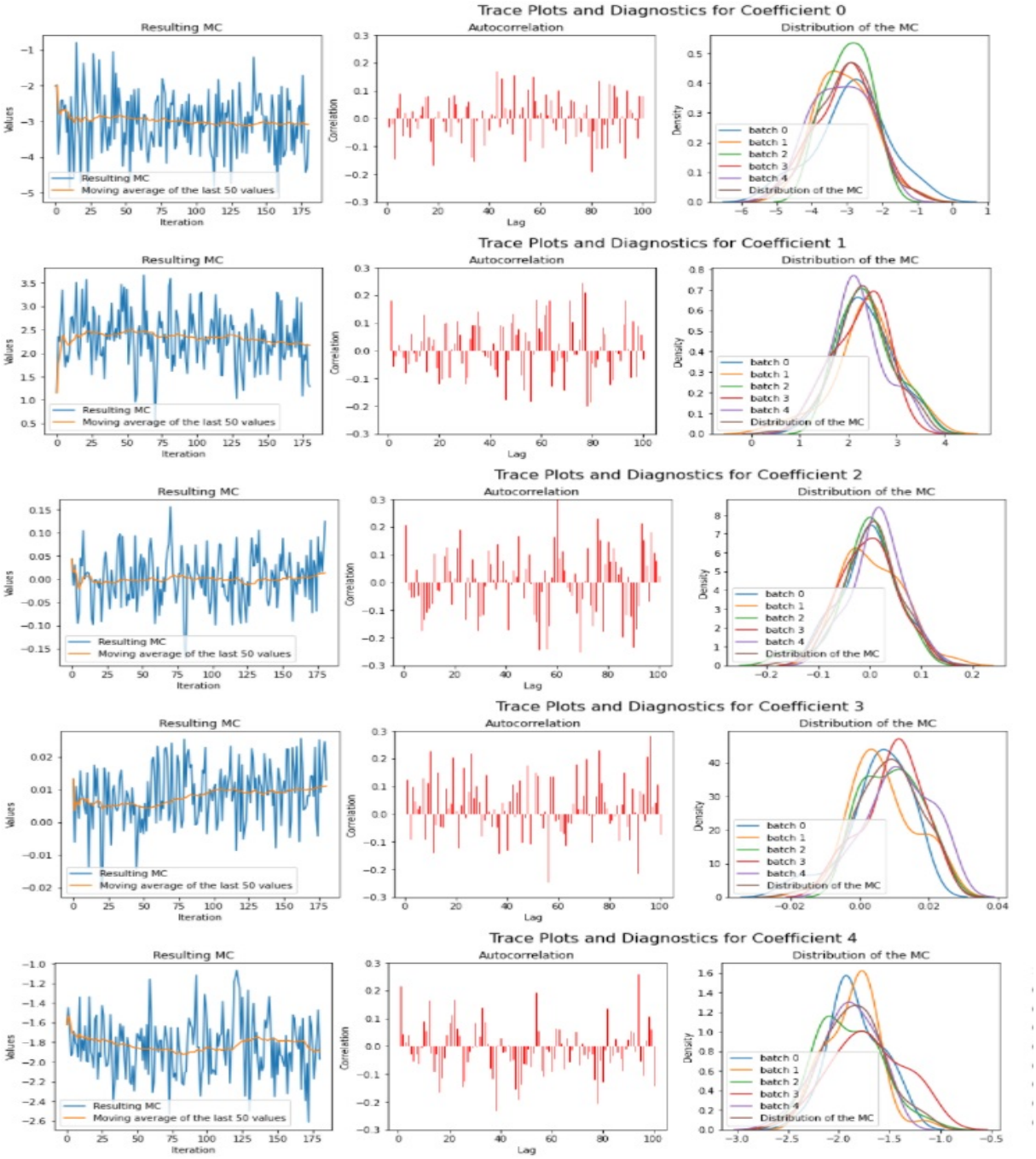

Figure 26: Traceplots, autocorrelations and densities for the coefficients β_0, β_1 and β_2 in the **Metropolis model** on **real data** with **coefficient initialized at the maximum likelihood estimates** and as **prior a Normal centered in such values with the identity matrix as variance-covariance matrix**. The variance of the proposal distribution is scaled by $\tau = 0.45$.

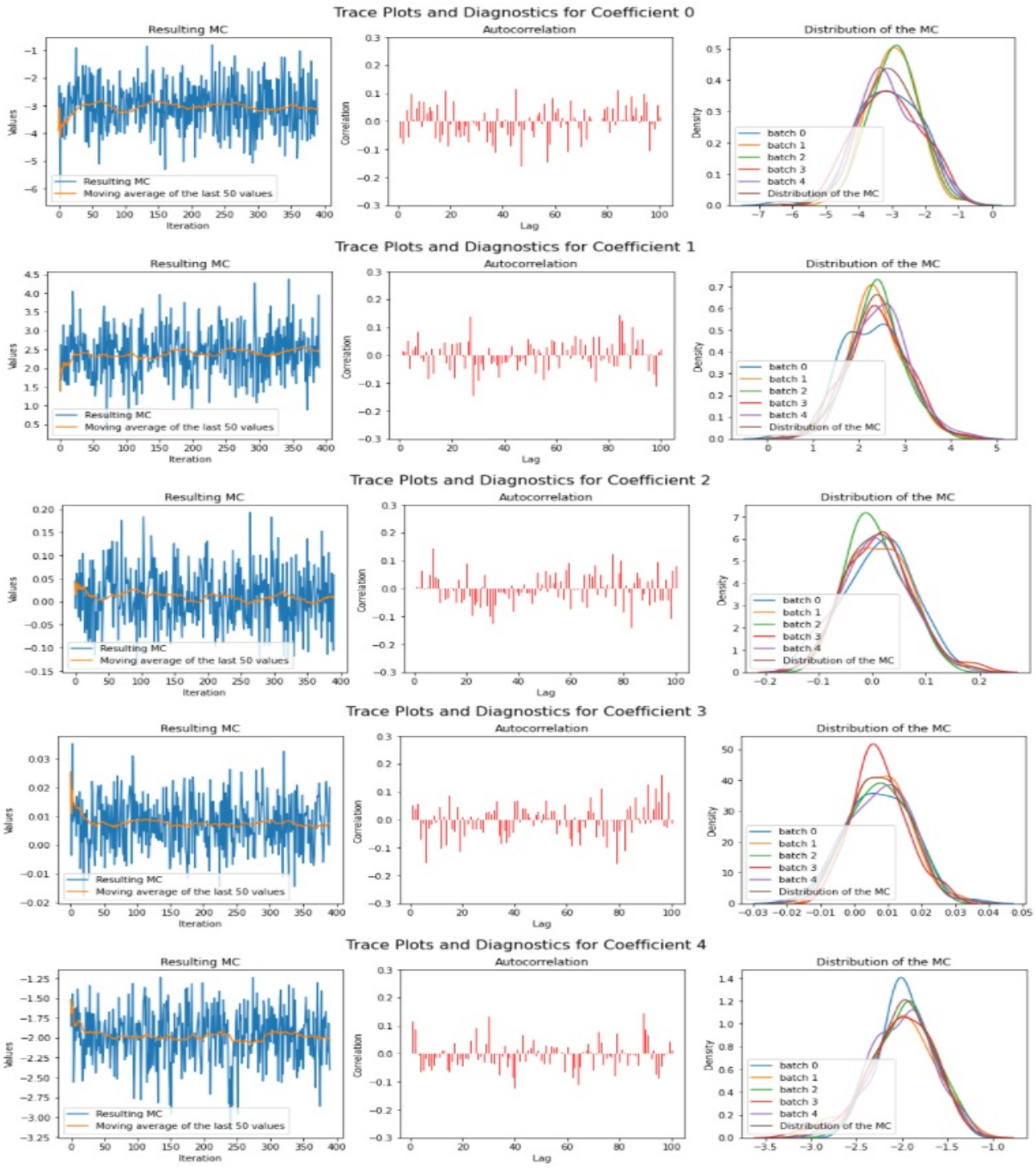

Figure 27: Traceplots, autocorrelations and densities for the coefficients β_0, β_1 and β_2 in the **Auxiliary Gibbs model** on **real data** with **randomly initialized coefficients** and as **prior a Normal centered in such values with the identity matrix as variance-covariance matrix**.

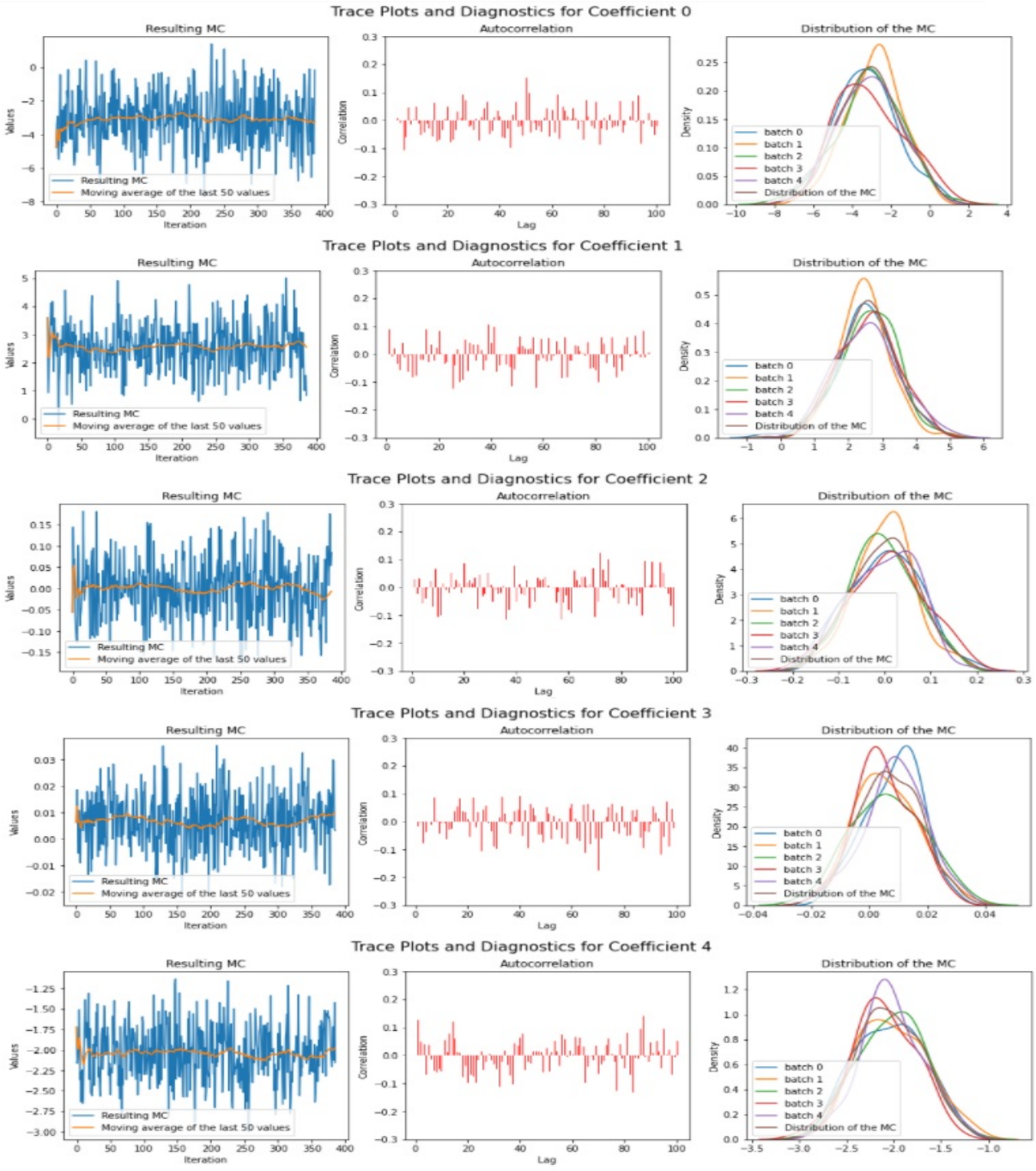

Figure 28: Traceplots, autocorrelations and densities for the coefficients β_0, β_1 and β_2 in the **Auxiliary Gibbs mode** on **real data** with **randomly initialized coefficients** and **noninformative prior**.

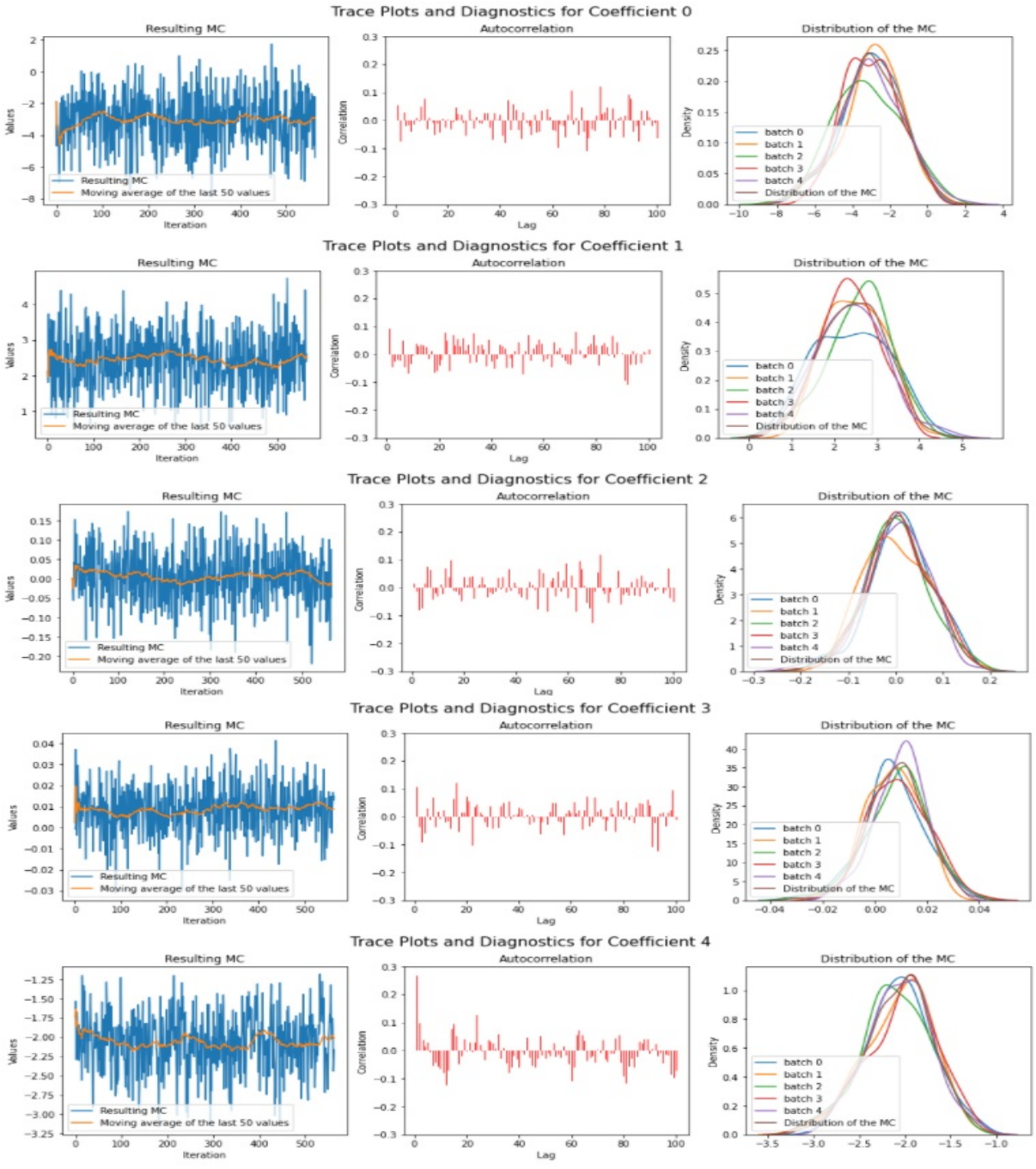

Figure 29: Traceplots, autocorrelations and densities for the coefficients β_0, β_1 and β_2 in the **Auxiliary Gibbs model** on **real data** with **coefficient initialized at the maximum likelihood estimates** and **noninformative prior**.

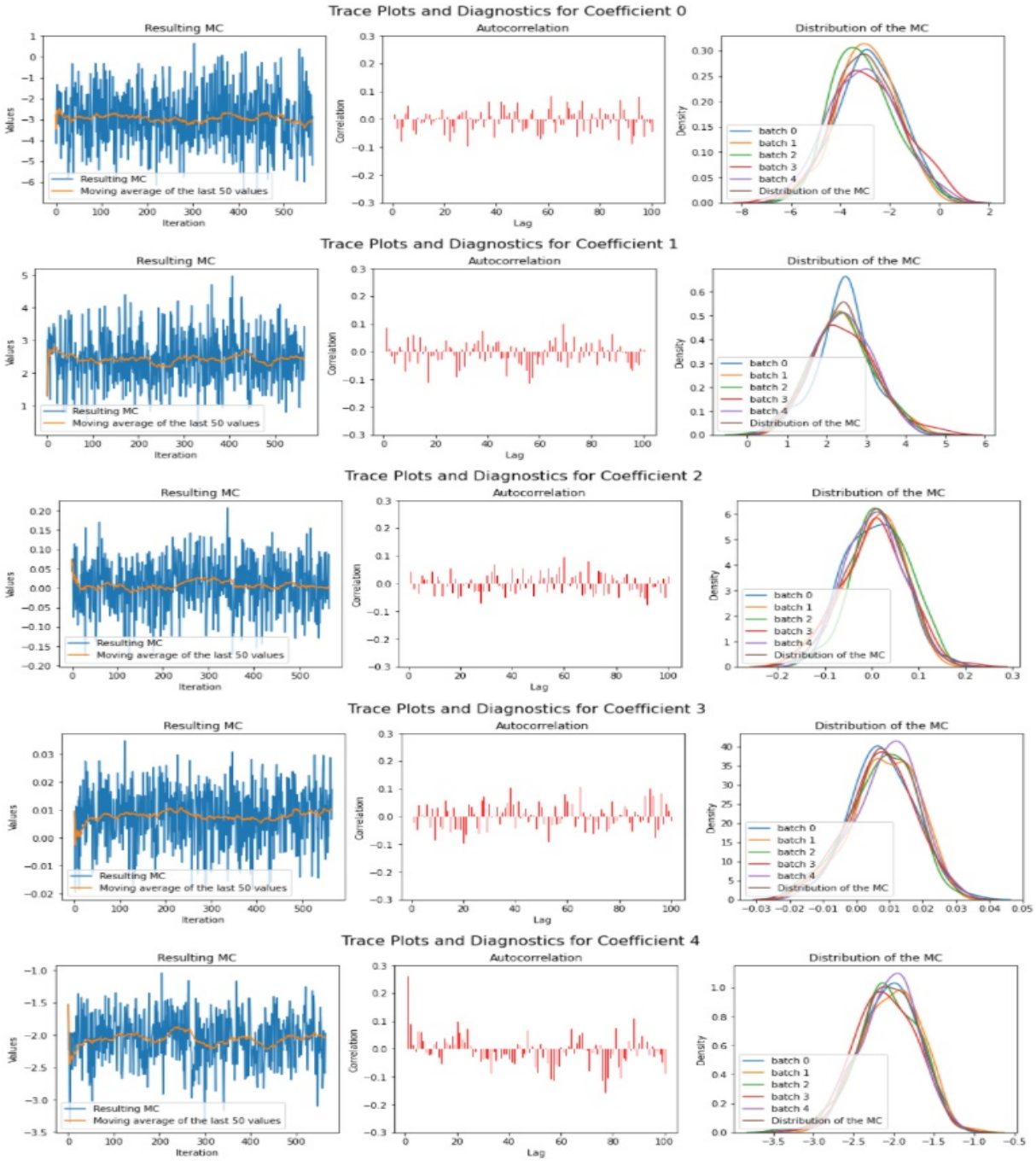

Figure 30: Traceplots, autocorrelations and densities for the coefficients β_0, β_1 and β_2 in the **Auxiliary Gibbs mode** on **real data** with **coefficient initialized at the maximum likelihood estimates** and as **prior a Normal centered in such values with the identity matrix scaled by a factor of 5 as variance-covariance matrix**.

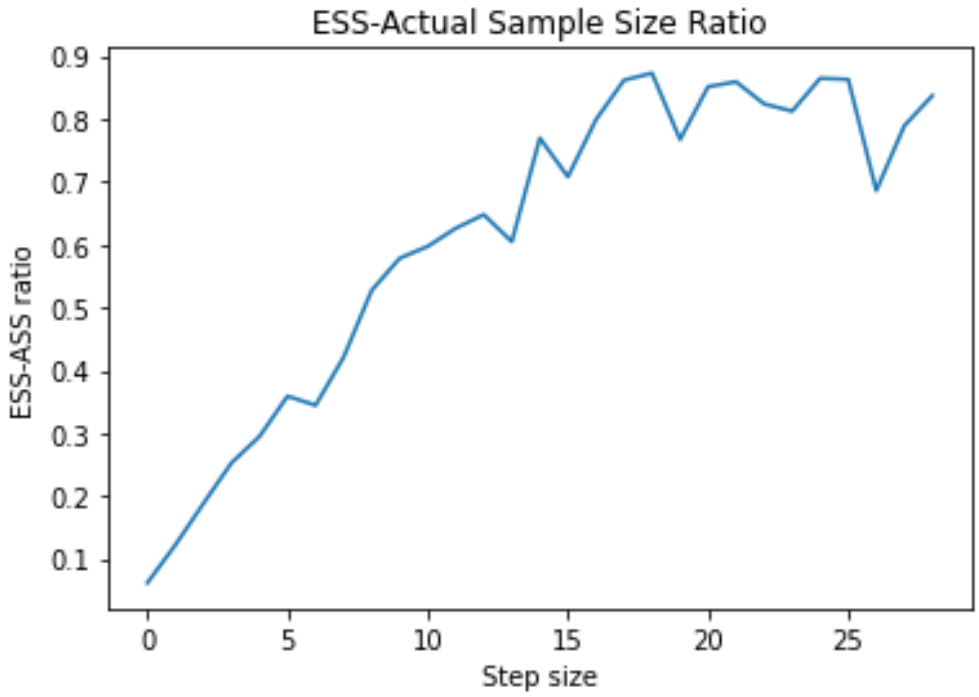

Figure 31: Example of a multivariate ESS graph, used to find the optimal cycle size for each model. In the figure, the ESS-Actual Sample Size Ratio of Model 1 of Metropolis algorithm, performing on simulated data.