# Maximum Likelihood Estimators on MCMC Sampling Algorithms for Decision Making

Christos Karras[1(✉)], Aristeidis Karras[1], Markos Avlonitis[3], Ioanna Giannoukou[2], and Spyros Sioutas[1]

[1] Computer Engineering and Informatics Department, University of Patras, 26504 Patras, Greece
{c.karras,akarras,sioutas}@ceid.upatras.gr
[2] Department of Management Science and Technology, University of Patras, 26334 Patras, Greece
igian@upatras.gr
[3] Department of Informatics, Ionian University, 49100 Corfu, Greece
avlon@ionio.gr

**Abstract.** Monte Carlo simulations using Markov chains as the Gibbs sampler and Metropolis algorithm are widely used techniques for modelling stochastic problems for decision making. Like all other Monte Carlo approaches, MCMC exploits the law of large numbers via repeated random sampling. Samples are formed by running a Markov Chain that is constructed in such a way that its stationary distribution closely matches the input function, which is represented by a proposal distribution. In this paper, the fundamentals of MCMC methods are discussed, including the algorithm selection process, optimizations, as well as some efficient approaches for utilizing generalized linear mixed models. Another aim of this paper is to highlight the usage of the EM method to get accurate maximum likelihood estimates in the context of generalized linear mixed models.

**Keywords:** MCMC methods · Gibbs sampler · Maximum likelihood · Estimators · Generalized linear mixed models · Decision making

## 1 Introduction

Generalized linear mixed models (GLMMs) are variations of generalized linear models (GLMs) that include unobservable factors as extra components of variability. As a consequence, they have a variety of applications and practical relevance [3,4,12,16]. Typically, unobserved effects are handled by including random effects in the predictor of the generalised linear model. The marginal likelihood function of the GLM is then derived by integrating the likelihood of another GLM with regard to the mixing distribution, which is the anticipated


Published by Springer Nature Switzerland AG 2022
I. Maglogiannis et al. (Eds.): AIAI 2022 Workshops, IFIP AICT 652, pp. 345–356, 2022.
https://doi.org/10.1007/978-3-031-08341-9_28

distribution of the random effects. While GLMMs are a robust family of statistical models, their practical use has been constrained by the complexity of the likelihood function. As a result, numerous techniques based on analytical approximations to the probability have been developed. In the context of this paper, we provide one distinct implementation of the Monte Carlo EM algorithm in which the intractable integral at the E-step (our S-step) is evaluated using simulation approaches. The first technique employs simulated random samples from the precise conditional distribution of the random effects vector $u$ given the data $y$, which was derived by rejection sampling using the marginal distribution of $u$ as the candidate distribution.

## 2 Related Work

Sampling is a crucial process across every discipline. With random sampling as indicated in [14], elements are taken in a probabilistic way for further processing. The underlying distribution of data is although hard to estimate requiring more accurate models. The capacity of conventional generalised linear models to accommodate non-Gaussian distributions and non-linear link processes is combined with the ability of classic (Gaussian) mixed models to express complicated dependent structures using random components. As a result, GLMMs appear to be well suited for many applications [1,5,6,16]. To be effective, GLMMs need more inference tools than ordinary statistical models.

For example, when doing probability inference, conditional probability values must be considered. Other integration simplifications used in conventional Gaussian mixed models (e.g., defining a Gaussian distribution on Gaussian random components that results in a Gaussian marginal distribution) are not applicable to GLMMs. The literature discusses several interpretative techniques, for example, [3,17] and [15] for a full study comparing different methods.

Many complicated stochastic systems may be simulated using Markov chain Monte Carlo [8,11,18]. Integrals may be calculated via simulation for various statistical inferences while there is a lot of study on Bayesian inference [2,7,8]. To describe stochastic processes, Markov chain Monte Carlo is a general-purpose technique that has been proved to be successful for sampling across difficult geometric objects [13] while it is also employed for probability inference. Several Monte Carlo approximation techniques have been created for complicated stochastic processes such as Markov random fields (Gibbs distributions) utilised in spatial statistics. One method is to use Monte Carlo simulations [9,10,21]. Another is to use stochastic models [19] and third, the likelihood situation [20]. Only the first allows for quick parametric bootstrapping and simulation experiments using a single Monte Carlo sample.

# 3 Methodology

## 3.1 Problem Definition

In this paper a clustering issue is examined. Assume there are $n$ items, each of which has a binary answer of type:

$$Y_{ij} = 0, 1, \text{for } i = n, \ldots, 1, \text{for } j = n, \ldots, T \tag{1}$$

where $n$ signifies all observed variables and $T$ denotes the observation time. Typically, the time of observation varies across components; as a consequence, time points may also vary. For the purposes of this paper, we will assume that all elements are exactly equivalent in length and time points. Additionally, we suppose that these topics fall into two separate clusters. The dependent expectation of a variable for each cluster, responds as follows:

$$\begin{aligned} P_{1,ij} &= \mathbb{E}(Y_{ij}|U_i, X_{1,ij}, Z_{1,i}) = f^{-1}(\beta_1 \cdot X_{1,ij} + Z_{1,i}) \\ P_{2,ij} &= \mathbb{E}(Y_{ij}|U_i, X_{2,ij}, Z_{2,i}) = f^{-1}(\beta_2 \cdot X_{2,ij} + Z_{2,i}) \end{aligned} \tag{2}$$

where cluster membership is denoted by $U$, and fixed and random effects, are denoted by $X_{c,ij}$ and $Z_c, i$, $(c = 1, 2)$ respectively. The function of connection is specified as:

$$f^{-1}(x) = \frac{\exp(x)}{1 + \exp(x)} \tag{3}$$

Due to the fact that $U$ is often unknown in a typical clustering scenario, it is treated as an effect of randomness. In (2), $u = 1$ for $P_{1,ij}$ while $u = 2$ for $P_{2,ij}$. For randomness, it is assumed:

$$Z_{c,i} \sim N(0, \sigma_c^2), \mathbb{P}(U) = 1 \tag{4}$$

Thus, $\Omega = \{\beta_1, \beta_2, \sigma_1, \sigma_2, \pi_1\}$ is the parameter to be assessed. By interpreting random effects as data missing, the function of likelihood for the whole set of data may be represented as in (5).

$$L(\Omega|Y_{ij}, U_i, Z_{U_i,i}) = \prod_{i=1}^{n} \prod_{c=1}^{2} \{\pi_c f_c(Z_{c,i})[\prod_{j=1}^{T} f_c(Y_{ij}|Z_{c,i})]\}^{w_{ic}} \tag{5}$$

where the normal distribution $f_c(Y_{ij}|Z_{c,i}) = \mathbb{P}^{Y_{ij}}(1 - \mathbb{P}_{ij})^{1-Y_{ij}}$ and $f_c(Z_{c,i})$ signifies the density of it. The dummy variable $w_{ic}$ is associated with $U_i$ , hence

$$w_{ic} = \begin{cases} 1, & \text{element belongs to cluster } c \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

### 3.2 Generalized Linear Mixed Models (GLMMs)

Given the simulation parameters: $n$, $T$, $\beta_1, \beta_2$, $\pi_1$, $\sigma_1$, $\sigma_2$, we can obtain, Observed variables as $Y$

$$\mathbf{Y} = \begin{bmatrix} Y_{11} & Y_{12} & \cdots Y_{1T} \\ Y_{21} & Y_{22} & \cdots Y_{2T} \\ \vdots & \vdots & \ddots \quad \vdots \\ Y_{n1} & Y_{n2} & \cdots Y_{nT}) \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \\ \vdots \\ \mathbf{Y}_n \end{bmatrix}$$

Additional unobserved or unobservable variables as $U$, $Z$

$$\mathbf{U} = \begin{bmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \\ \vdots \\ \mathbf{U}_n \end{bmatrix}, \mathbf{Z} = \begin{bmatrix} \mathbf{Z}_{U_1,1} \\ \mathbf{Z}_{U_2,2} \\ \vdots \\ \mathbf{Z}_{U_n,n} \end{bmatrix}$$

Explanatory variables (fixed effect) as $X$

$$X = \begin{vmatrix} X_{U_1,11} & X_{U_1,12} & \cdots & X_{U_1,1T} \\ X_{U_2,21} & X_{U_2,22} & \cdots & X_{U_2,2T} \\ \vdots & \vdots & \ddots & \vdots \\ X_{U_n,n1} & X_{U_n,n2} & \cdots & X_{U_n,nT} \end{vmatrix}$$

**Computing Log-Likelihood.** Given the necessary parameters for each element of $\Omega$, the enhanced logged likelihood might be expressed as follows.

$$\begin{aligned} L(\Omega|\mathbf{Y}_{ij}, \mathbf{U}_i, \mathbf{Z}_{1,i}, \mathbf{Z}_{2,i}) &= \prod_{i=1}^{n} \prod_{c=1}^{2} \left\{ \pi_c f_c(Z_{c,i}) [\prod_{j=1}^{T} f_c(Y_{ij}|Z_{c,i})] \right\}^{\omega_{ic}} \\ &= exp\Bigg\{ \sum_{i=1}^{n} \sum_{c=1}^{2} \omega_{ic} \Big[ ln\pi_c - ln(\sqrt{2\pi}\sigma_c) - \frac{Z_{c,i}^2}{2\sigma_c^2} \\ &+ \sum_{j=1}^{T} [Y_{ij} ln P_{ij}^{(c)} + (1 - Y_{ij}) ln(1 - P_{ij}^{(c)})] \Big] \Bigg\} \end{aligned} \tag{7}$$

The logged likelihood could be expressed as in (8).

$$l = \sum_{i=1}^{n} \sum_{c=1}^{2} \omega_{ic} \Big[ ln\pi_c - ln(\sqrt{2\pi}\sigma_c) - \frac{Z_{c,i}^2}{2\sigma_c^2} + \sum_{j=1}^{T} [Y_{ij} ln P_{ij}^{(c)} + (1 - Y_{ij}) ln(1 - P_{ij}^{(c)})] \Big] \tag{8}$$

The result of (8) can be expressed in a simpler form as in (9)

$$\begin{aligned} l(\Omega|\mathbf{Y},\mathbf{U},\mathbf{Z}) &= \sum_{i=1}^{n} \ln f_{(U_i,Z_{U_i,i})}(U_i, Z_{U_i,i}|\pi_c,\sigma_1,\sigma_2) \\ &+ \sum_{i=1}^{n}\sum_{j=1}^{T} \ln f_{Y_{ij}|(U_i,Z_{U_i,i})}(Y_{ij}|(U_i,Z_{U_i,i}),\beta_1,\beta_2) \\ &=\triangleq \ln f_{(\mathbf{U},\mathbf{Z})}(\mathbf{U},\mathbf{Z}|\pi_c,\sigma_1,\sigma_2) + \ln f_{\mathbf{Y}|(\mathbf{U},\mathbf{Z})}(\mathbf{Y}|\mathbf{U},\mathbf{Z},\beta_1,\beta_2) \end{aligned} \tag{9}$$

### 3.3 Monte Carlo Simulation Maximization

**Simulation Maximization Algorithm.** To perform maximization on the Monte Carlo method, the augmented logged likelihood must be approximated first. By taking expectation of $\mathbf{U}$ and $\mathbf{Z}$ given $\mathbf{Y}$ under the current estimate of the parameters $\Omega^{(m)}$, the expected augmented logged likelihood could be defined as:

$$\begin{aligned} &Q(\Omega|\Omega^m) = \\ &E(l|Y_{ij},\Omega^{(m)}) = \frac{1}{N}\sum_{k=1}^{N}\sum_{i=1}^{n}\sum_{c=1}^{2}\omega_{ic} \\ &\Big[ln\pi_c - ln(\sqrt{2\pi}\sigma_c) - \frac{Z_{i,k}^2}{2\sigma_c^2} + \sum_{j=1}^{T}[Y_{ij}lnP_{ij}^{(c)} + (1-Y_{ij}) \\ &ln(1-P_{ij}^{(c)})]\Big]. \end{aligned} \tag{10}$$

Notice that in the expected log-likelihood, $\Omega^{(m)}$ could be decomposed into separate component as in (11).

$$\begin{aligned} Q(\Omega,\Omega^{(m)}) &= \mathbb{E}_{(\mathbf{U},\mathbf{Z})|(\mathbf{Y},\Omega^{(m)})} \ln f_{(\mathbf{U},\mathbf{Z})}(\mathbf{U},\mathbf{Z}|\pi_c,\sigma_1,\sigma_2) \\ &+ \mathbb{E}_{(\mathbf{U},\mathbf{Z})|(\mathbf{Y},\Omega^{(m)})} \ln f_{\mathbf{Y}|(\mathbf{U},\mathbf{Z})}(\mathbf{Y}|\mathbf{U},\mathbf{Z},\beta_1,\beta_2) \\ &=\triangleq P(\Omega,\Omega^{(m)}) + R(\Omega,\Omega^{(m)}) \end{aligned} \tag{11}$$

### 3.4 Monte Carlo Integration

In order to compute the integral above, we use Monte Carlo Integrating to approximate it. Suppose that $\{(\mathbf{U}_{(k)},\mathbf{Z}_{(k)}, k = 1,2,\cdots,K)\} \overset{i.i.d}{\sim} f_{(\mathbf{U},\mathbf{Z}|\mathbf{Y})}(\mathbf{U},\mathbf{Z}|\mathbf{Y}),\Omega)$ and we sample $m$ times to approximate. Based on Mean Value Method we get:

$$\begin{aligned} Q\left(\Omega,\Omega^{(m)}\right) \approx \frac{1}{m}\sum_{k=1}^{m}\sum_{i=1,c=U_{(k),i}} \Bigg[&\ln\pi_c - \frac{1}{2}\ln\left(2\pi\sigma_c^2\right) - \frac{Z_{c,i}^2}{2\sigma_c^2} \\ &+ \sum_{j=1}^{T}\left[Y_{ij}\left(\beta_c X_{c,ij} + Z_{c,i}\right) - \ln\left(1+\exp\left(\beta_c X_{c,ij} + Z_{c,i}\right)\right)\right] \Bigg] \end{aligned} \tag{12}$$

**Maximum Likelihood Estimators.** To calculate the maximum likelihood estimators the partial derivatives must be approximated whose parameters are given by (13), (14), (15).

$$\frac{\partial Q(\Omega, \Omega^{(m)})}{\partial \pi_1} = \frac{1}{m} \sum_{k=1}^{m} \sum_{i=1}^{n} \mathbb{I}_{\{U_{(k),i}, i=1\}} \frac{1}{\pi_1} - \frac{1}{m} \sum_{k=1}^{m} \sum_{i=1}^{n} \mathbb{I}_{\{U_{(k),i}, i=2\}} \frac{1}{1-\pi_1} \tag{13}$$

$$\frac{\partial Q(\Omega, \Omega^{(m)})}{\partial \sigma_c^2} = \frac{1}{m} \sum_{k=1}^{m} \sum_{i=1}^{n} \mathbb{I}_{\{U_{(k),i=c}\}} (-\frac{1}{2\sigma_c^2} + \frac{Z_{(k),c,i}^2}{2\sigma_c^4}) \tag{14}$$

$$\frac{\partial Q(\Omega, \Omega^{(m)})}{\partial \beta_c} = \frac{1}{m} \sum_{k=1}^{m} \sum_{i=1}^{n} \mathbb{I}_{\{U_{(k),i=c}\}} \sum_{j=1}^{T} \left[ Y_i j X_{c,ij} - \frac{X_{c,ij} \exp(\beta_c X_{c,ij} + Z_{(k),c,i})}{1 + \exp(\beta_c X_{c,ij} + Z_{(k),c,i})} \right] \tag{15}$$

By setting the above partial derivatives to 0, we get the maximum likelihood estimators as in (16).

$$\hat{\pi}_1 = \frac{1}{mn} \sum_{k=1}^{m} \sum_{i=1}^{n} \mathbb{I}_{\{U_{(k)}, i=1\}} \quad \hat{\sigma}_c = \sqrt{\frac{\sum_{k=1}^{m} \sum_{i=1}^{n} \mathbb{I}_{\{U_{(k)}, i=c\}} Z_{(k),c,i}^2}{\sum_{k=1}^{m} \sum_{i=1}^{n} \mathbb{I}_{\{U_{(k),i=c}\}}}} \tag{16}$$

To compute the MLE of $\beta_c$, we use direct numerical maximization proposed by Newton-Raphson Method. The second order partial derivative of $\beta_c$ is denoted as in (17).

$$\frac{\partial^2 Q(\Omega, \Omega^{(m)})}{\partial \beta_c^2} = -\frac{1}{m} \sum_{k=1}^{m} \sum_{i=1}^{n} \mathbb{I}_{\{U_{(k),i=c}\}} \sum_{j=1}^{T} \frac{X_{c,ij}^2 \exp(\beta_c X_{c,ij} + Z_{(k),c,i})}{(1 + \exp(\beta_c X_{c,ij} + Z_{(k),c,i}))^2} \tag{17}$$

### 3.5 Markov Chain Sampler

Since it difficult to sample directly from a multivariate distribution of the type $f_{(\mathbf{U},\mathbf{Z}|\mathbf{Y})}(\mathbf{U}, \mathbf{Z}|\mathbf{Y}), \Omega)$, we can use Gibbs Sampling, a Markov chain Monte Carlo (MCMC) algorithm to obtain a sequence of observations which are approximated from the multivariate distribution. First, we need to calculate the conditional distributions (18) and (19).

$$\frac{f_{(U_i, Z_{(U_i,i)}|\mathbf{Y}_i)}(U_i, Z_{U_i,i}|\mathbf{Y}_i, \Omega)}{f_{Z_{(U_i,i)}|\mathbf{Y}_i}(Z_{U_i,i}|\mathbf{Y}_i, \Omega)} = f_{U_i|(Z_{U_i},i,\mathbf{Y}_i)}(U_i|Z_{U_i,i}, \mathbf{Y}_i) \tag{18}$$

$$\frac{f_{(U_i, Z_{(U_i,i)}|\mathbf{Y}_i)}(U_i, Z_{U_i,i}|\mathbf{Y}_i, \Omega)}{f_{U_i|\mathbf{Y}_i}(U_i|\mathbf{Y}_i, \Omega)} = f_{Z_{U_i},i|(U_i,\mathbf{Y}_i)}(Z_{U_i,i}|(U_i, \mathbf{Y}_i)) \tag{19}$$

Then, suppose that $(U_{(k),i}, Z_{(k),U_{(k),i},i})$ is the $i$-th component of the $k$-th sample, we want to draw the $i$-th component of the $(k+1)$-th sample. We draw

$$U_{(k+1),i} \sim f_{U_i|Z_{U_i},i,\mathbf{Y}_i}(u|Z_{U_i,i}, \mathbf{Y}_i, \Omega) \tag{20}$$

**Algorithm 1.** MCMC incorporated Metropolis-Hastings

**for** $i = 1, \cdots, n$ **do**
  Initialize$(U_{(0),i}, Z_{(0),1,i}, Z_{(0),2,i})$
  **for** c=1:2 **do**
    $k \leftarrow 0$
    **for** k **do**=1:$K_2$
      Draw $z^* \sim f_c(z|\Omega)$
      Accept $z^*$ as $Z_{(k+1),c,i}$ with probability $A_{k,\mathbf{Y}_i}(z, z^*)$; otherwise, retain the original $Z_{(k),c,i}$
    **end for**
    Burn-in procedure and let the last K + 1 samples be the final samples $\{Z_{(k),c,i}, k = 0, 1, \cdots, K\}$
  **end for**
  $k \leftarrow 0$
  **for** k=1:$K$ **do**
    Draw $U_{(k+1),i} \sim f_{U_i|Z_{U_i},i,\mathbf{Y}_i}(u|Z_{U_i}, i, \mathbf{Y}_i, \Omega)$
  **end for**
  Let the last $m$ samples be the final samples $\{U_{(k),i}, k = 0, 1, \cdots, K\}$
**end for**
Burn-in procedure and return the $m$ samples $\{(U_{(i),i}, Z_{(i),1,i}, Z_{(i),2,i}), k = 0, 1, \cdots, m\}$

$$Z_{(k+1),U_{(k+1),i},i} \tag{21}$$

where (21) can be approximated as:

$$f_{Z_{U_i,i}|U_i,\mathbf{Y}_i}(z|U_i, \mathbf{Y}_i, \Omega). \tag{22}$$

Let (23) be a candidate distribution for (22).

$$h_{Z_{U_{(k),i},i}}(z) \tag{23}$$

**Metropolis-Hastings Algorithm.** To sample (21) from (22), we use (23). Since the candidate distribution should be similar to (22), we can choose $h_{Z_{U_{(k),i},i}}(z) = f_{U_i}(z|\Omega)$ and the acceptance function is

$$A_{k,\mathbf{Y}_i}(z, z^*) = \min\left[1, \frac{f_{Z_{U_i,i}|U_i,\mathbf{Y}_i}(z^*|U_i, \mathbf{Y}_i, \Omega) f_{U_i}(z|\Omega)}{f_{Z_{U_i,i}|U_i,\mathbf{Y}_i}(z|U_i, \mathbf{Y}_i, \Omega) f_{U_i}(z^*|\Omega)}\right] \tag{24}$$

where $\frac{f_{Z_{U_i,i}|U_i,\mathbf{Y}_i}(z^*|U_i,\mathbf{Y}_i,\Omega) f_{U_i}(z|\Omega)}{f_{Z_{U_i,i}|U_i,\mathbf{Y}_i}(z|U_i,\mathbf{Y}_i,\Omega) f_{U_i}(z^*|\Omega)}$ can be expressed as,

$$\frac{f_{Z_{U_i,i}|U_i,\mathbf{Y}_i}(z^*|U_i, \mathbf{Y}_i, \Omega) f_{U_i}(z|\Omega)}{f_{Z_{U_i,i}|U_i,\mathbf{Y}_i}(z|U_i, \mathbf{Y}_i, \Omega) f_{U_i}(z^*|\Omega)} = \exp\left[\sum_{j=1}^{T} Y_{ij}(z^* - z)\right] \prod_{j=1}^{T} \frac{1 + \exp(\beta_i X_{ij} + z)}{1 + \exp(\beta_i X_{ij} + z^*)} \tag{25}$$

We begin our Gibbs sampler incorporated a Metropolis-Hastings step as in Algorithm 1.

**Monte Carlo Simulation Maximization.** Unfortunately, we do know the $f_{(\mathbf{U},\mathbf{Z})|\mathbf{Y}}(\mathbf{U},\mathbf{Z}|\mathbf{Y},\Omega)$, so we use

$$f_{(\mathbf{U},\mathbf{Z})|\mathbf{Y}}(\mathbf{U},\mathbf{Z}|\mathbf{Y},\Omega^{(m)}) \tag{26}$$

in the $(m+1)$-$th$ step from 2 to approximate the distribution so as to generate

$$\{(U_{(k)},Z_{(k)}),k=1,2,\cdots,m\} \overset{i.i.d}{\sim} f_{(\mathbf{U},\mathbf{Z})|\mathbf{Y}}(\mathbf{U},\mathbf{Z}|\mathbf{Y},\Omega^{(m)}) \tag{27}$$

The Monte Carlo Simulation-Maximization Algorithm we use in every experiment is given in Algorithm 2. Moreover, a flowchart of the proposed method is shown in Fig. 1.

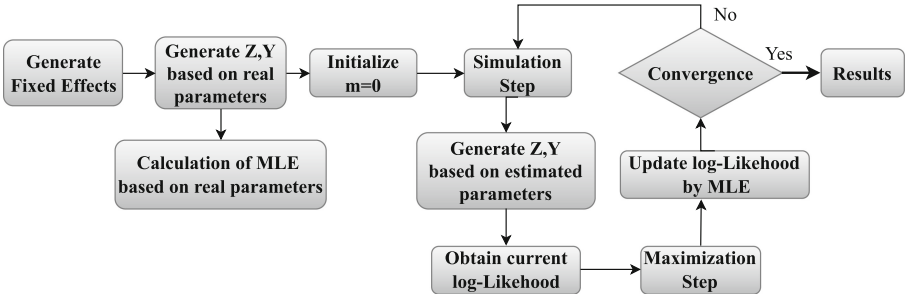


**Fig. 1.** Flow chart of the proposed method.

**Algorithm 2.** Monte Carlo Simulation Maximization (Proposed)

1: Start with the initial value for estimator $\Omega^{(0)}$. Set $m$=0.
2: Simulation-step:
3: a. Generate $m$ samples $\{(U_{(i),i},Z_{(i),1,i},Z_{(i),2,i}),k=0,1,\cdots,m\}$ from $f_{Z_{U_i,i}|U_i,\mathbf{Y}_i}(z|U_i,\mathbf{Y}_i,\Omega)$ throughAlgorithm 1
4: b. Calculate the partial derivatives of $Q(\Omega,\Omega^{(m)})$, the Monte Carlo estimator for every parameters.
5: Maximization-step
6: $\Omega^{(m+1)} \leftarrow \arg\max_{\Omega} Q(\Omega,\Omega^{(m)})$
7: $m \leftarrow m+1$
8: Repeat step 2-6 until convergence and then output the maximum likelihood estimators $\Omega^{(m)}$.

## 4 Experimental Results

For $Y_{ij} = 0, 1$, for $i = n, \ldots, 1$, for $j = n$, $N = 100$, we start our experiments by setting the initial values of the parameter $\Omega = \{\beta_1, \beta_2, \sigma_1, \sigma_2, \pi_1\}$. For the first experiment we set $\Omega$ as $\beta_1 = 1.3$, $\beta_2 = 1.3$, $\sigma_1 = 2.0$, $\sigma_2 = 10$, $\pi_1 = 0.6$. These values are set after several experiments as they make the model operate smoothly. We perform variable step-size; we start the proposed Monte Carlo estimation with a modest sample size and gradually raise our sampling intervals as the Simulation Maximization (SM) iterates. We perform Gibbs sampling in each SM iteration. In each experiment, we repeat the SM iteration process for 50 times. We conduct 1000 tests with various random seeds and mark down the results of the first 100 experiments as well as the Mean Squared Error of the 1000 experiments. The results of the simulation (step 2 of Algorithm 2) are shown in Table 1. The aim here is to meet convergence for all values as close as possible in a relative short period of time. Next, we use the results derived from the previous process to perform the maximization step of 2. The results are shown in Table 1. Our convergences are pretty good, as all parameters are converged in less than 50 steps, which costs about 1 min.

We monitor the convergence of the algorithm by plotting $\Omega^*$ vs. iteration number $i$ and the plot reveals random fluctuation about the line $\Omega = \Omega^*$. So, we may continue with a large value of $m$ to decrease the system variability. In Figs. 2, 3, 4 the convergence of $\beta$, $\sigma$ and $\pi$ is shown. The blue line represents the actual value while the orange represents the converged value.

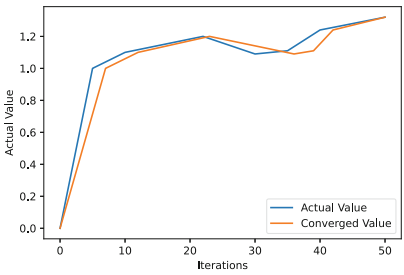


**Fig. 2.** Convergence of $\beta$. (Color figure online)

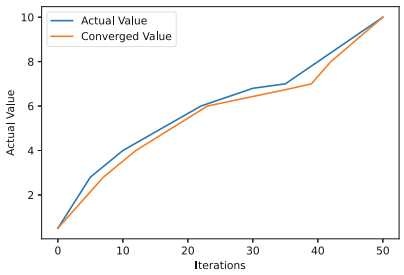


**Fig. 3.** Convergence of $\sigma$. (Color figure online)

As depicted in Figs. 2, 3 the converged value is relatively close to the true value at each given point across all 50 iterations while at some points they are even identical. Similarly the convergence of $\sigma$ is relatively close to the true value across all iterations. Likewise, the convergence of $\pi$ is close to the actual value as with the two previous results.

**Table 1.** True values and initial values vs converged values.

| Variables | True value | Initial value | Converged value |
|---|---|---|---|
| $\beta_1$ | 1.3 | 0 | 1.2953680 |
| $\beta_2$ | 1.3 | 0 | 1.3076125 |
| $\sigma_1$ | 2 | 1 | 1.987342 |
| $\sigma_2$ | 10 | 5 | 9.132040 |
| $\pi_1$ | 0.6 | 0.8 | 0.480500 |

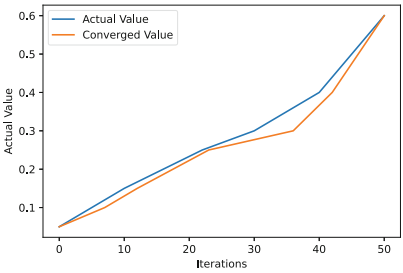


**Fig. 4.** Convergence of $\pi$. (Color figure online)

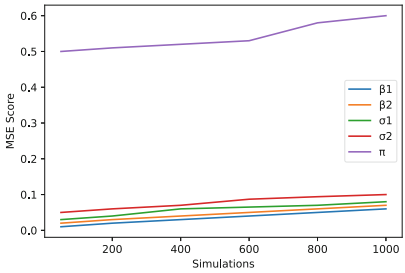


**Fig. 5.** MSE of parameters.

The simulations of the proposed algorithm are summarized in Table 2. The results show satisfactory performance across all five parameters of $\Omega$.

**Table 2.** Simulations for $\beta_1 = 1.3$, $\beta_2 = 1.3$, $\sigma_1 = 2.0$, $\sigma_2 = 10$, $\pi_1 = 0.6$, $N = 100$.

| N | $\beta_1$ | $\beta_2$ | $\sigma_1$ | $\sigma_2$ | $\pi_1$ |
|---|---|---|---|---|---|
| 100 | 1.2993 | 1.3045 | 1.9676 | 9.5722 | 0.4730 |
| 200 | 1.2983 | 1.3039 | 1.9675 | 9.5673 | 0.4731 |
| 300 | 1.2982 | 1.3059 | 1.9691 | 9.6156 | 0.4747 |
| 400 | 1.2982 | 1.3071 | 1.9681 | 9.6091 | 0.4744 |
| 500 | 1.2985 | 1.3059 | 1.9692 | 9.6096 | 0.4748 |
| 600 | 1.2984 | 1.3057 | 1.9683 | 9.6032 | 0.4746 |
| 700 | 1.2987 | 1.3054 | 1.9695 | 9.6325 | 0.4752 |
| 800 | 1.2985 | 1.3051 | 1.9683 | 9.6270 | 0.4749 |
| 900 | 1.2982 | 1.3047 | 1.9669 | 9.6238 | 0.4746 |
| 1000 | 1.2982 | 1.3051 | 1.9668 | 9.6233 | 0.4744 |

### 4.1 Evaluation

To evaluate the proposed method we use the Mean Squared Error Metric. The MSE is calculated as in (28).

$$MSE_\theta = \frac{1}{N}\sum_{n=1}^{N}(\theta^{(n)} - \hat{\theta}^{(n)})^2 \tag{28}$$

where $\theta \in \Omega$, $\theta^{(n)}$ is the true MLE of $\theta$ in the $n$-th experiments and $\hat{\theta}^{(n)}$ is the estimator of $\theta^{(n)}$. The MSE score of $\beta_1$, $\beta_2$, $\sigma_1$, $\sigma_2$ and $\pi$ is shown in Fig. 5. Generally MSE should be within the value range of 0–2 whereabouts zero value indicates that the model is perfect and the value of two indicates that the performance is marginally acceptable.

## 5 Conclusions and Future Work

In the context of this paper, the basic functions of MCMC methods were shown as well as the inner workings of these methods along with Gibbs sampling and a proposed method for Monte Carlo Simulation Maximization. The results show that the proposed method performed smoothly using the estimators created by our system. In respect to time, the system was capable to sample elements in a quite speedy way (approximately 1 min) for up to 1000 experiments. The MSE for 1000 experiments was also in relevantly low levels (approximately 0.5 for $\pi$) while the parameters $\beta_1$, $\beta_2$, $\sigma_1$, $\sigma_2$ and $\pi_1$ were simulated efficiently and effectively.

Future directions of this work include the integration of the proposed algorithm with a Bayesian Neural Network to better highlight the findings in a more accurate yet speedy way and to approximate the underlying evolving distributions in a more steady way. Another future scope is to increase the sampling rates and to reduce the MSE as low as possible compared to the existing method. One last but significant improvement could be the reduction of the difference among the real and estimated coefficients.

**Acknowledgements.** This paper is funded in the framework of THLEMAXOS project which is funded by the Ionian Region Islands with MIS code 5007986 in the context of Operational Program Ionian Islands 2014-2020.

## References

1. Agresti, A.: Categorical Data Analysis. Wiley, Hoboken (2003)
2. Besag, J., York, J., Mollié, A.: Bayesian image restoration, with two applications in spatial statistics. Ann. Inst. Stat. Math. **43**(1), 1–20 (1991)
3. Breslow, N.E., Clayton, D.G.: Approximate inference in generalized linear mixed models. J. Am. Stat. Assoc. **88**(421), 9–25 (1993)
4. Clayton, D.G.: Generalized linear mixed models. Markov Chain Monte Carlo Pract. **1**, 275–302 (1996)
5. Demidenko, E.: Mixed Models: Theory and Applications (Wiley Series in Probability and Statistics). Wiley-Interscience, New York (2004)
6. Fahrmeir, L., Tutz, G., Hennevogl, W., Salem, E.: Multivariate Statistical Modelling Based on Generalized Linear Models, vol. 425. Springer, New York (1994). https://doi.org/10.1007/978-1-4899-0010-4
7. Gelfand, A.E., Smith, A.F.: Sampling-based approaches to calculating marginal densities. J. Am. Stat. Assoc. **85**(410), 398–409 (1990)
8. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Trans. Pattern Anal. Mach. Intell. PAMI **6**(6), 721–741 (1984). https://doi.org/10.1109/TPAMI.1984.4767596
9. Geyer, C.J.: Likelihood inference in exponential families and directions of recession. Electron. J. Stat. **3**, 259–289 (2009). https://doi.org/10.1214/08-EJS349
10. Geyer, C.J., Thompson, E.A.: Constrained Monte Carlo maximum likelihood for dependent data. J. Roy. Stat. Soc. Ser. B (Methodol.) **54**(3), 657–683 (1992)
11. Hastings, W.K.: Monte Carlo sampling methods using Markov chains and their applications. Biometrika **57**, 97–109 (1970)
12. Hedeker, D.: Generalized linear mixed models. Encyclopedia of statistics in behavioral science (2005)
13. Karras, C., Karras, A.: DBSOP: an efficient heuristic for speedy MCMC sampling on polytopes. arXiv preprint arXiv:2203.10916 (2022)
14. Karras, C., Karras, A., Sioutas, S.: Pattern recognition and event detection on IoT data-streams. arXiv preprint arXiv:2203.01114 (2022)
15. McCulloch, C.E.: Maximum likelihood algorithms for generalized linear mixed models. J. Am. Stat. Assoc. **92**(437), 162–170 (1997)
16. Mcculloch, C.E., Neuhaus, J.M.: Generalized Linear Mixed Models. Wiley (2014). https://doi.org/10.1002/9781118445112.stat07540
17. McCulloch, C.E., Searle, S.R.: Generalized, Linear, and Mixed Models. Wiley, Hoboken (2004)
18. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equation of state calculations by fast computing machines. J. Chem. Phys. **21**(6), 1087–1092 (1953)
19. Moyeed, R., Baddeley, A.J.: Stochastic approximation of the MLE for a spatial point pattern. Scand. J. Stat., 39–50 (1991)
20. Ogata, Y., Tanemura, M.: Likelihood estimation of soft-core interaction potentials for Gibbsian point patterns. Ann. Inst. Stat. Math. **41**(3), 583–600 (1989)
21. Penttinen, A.: Modelling interactions in spatial point patterns: parameter estimation by the maximum-likelihood method. Comp. Sci. Econ. Statist. **7**, 1–107 (1984)