Hindawi
Journal of Applied Mathematics
Volume 2023, Article ID 3377201, 10 pages
https://doi.org/10.1155/2023/3377201

Research Article

A Fast and Efficient Estimation of the Parameters of a Model of Accident Frequencies via an MM Algorithm

Issa Cherif Geraldo,[1] **Edoh Katchekpele**,[2] **and Tchilabalo Abozou Kpanzou**[2]

[1]*Laboratoire d'Analyse, de Modélisations Mathématiques et Applications (LAMMA), Département de Mathématiques, Faculté des Sciences, Université de Lomé, 1 B.P. 1515 Lomé 1, Togo*
[2]*Laboratoire de Modélisation Mathématique et d'Analyse Statistique Décisionnelle (LaMMASD), Département de Mathématiques, Faculté des Sciences et Techniques, Université de Kara, Kara, Togo*

Correspondence should be addressed to Issa Cherif Geraldo; cherifgera@gmail.com

Received 8 November 2022; Revised 6 March 2023; Accepted 24 March 2023; Published 19 April 2023

Academic Editor: Qing-Wen Wang

In this paper, we consider a multivariate statistical model of accident frequencies having a variable number of parameters and whose parameters are dependent and subject to box constraints and linear equality constraints. We design a minorization-maximization (MM) algorithm and an accelerated MM algorithm to compute the maximum likelihood estimates of the parameters. We illustrate, through simulations, the performance of our proposed MM algorithm and its accelerated version by comparing them to Newton-Raphson (NR) and quasi-Newton algorithms. The results suggest that the MM algorithm and its accelerated version are better in terms of convergence proportion and, as the number of parameters increases, they are also better in terms of computation time.

1. Introduction

A large majority of the problems encountered in applied statistics (maximum likelihood estimation, least squares, data fitting, machine learning, data analysis, experimental design, clustering, and classification) involve the numerical optimization of a real-valued function $L(\boldsymbol{\theta})$ depending on a parameter vector $\boldsymbol{\theta} \in \mathbb{R}^d$, where d is a positive integer. Of all the numerical optimization algorithms developed over the years, the Newton-Raphson (NR) algorithm is the most famous and the most widely used. This algorithm starts from an initial guess $\boldsymbol{\theta}^{(0)}$ and iterates according to the scheme

$$\boldsymbol{\theta}^{(m+1)} = \boldsymbol{\theta}^{(m)} - \left[\nabla^2 L\left(\boldsymbol{\theta}^{(m)}\right)\right]^{-1} \nabla L\left(\boldsymbol{\theta}^{(m)}\right), \tag{1}$$

where $\nabla L(\boldsymbol{\theta}^{(m)})$ and $\nabla^2 L(\boldsymbol{\theta}^{(m)})$ are, respectively, the gradient vector and the Hessian matrix evaluated at $\boldsymbol{\theta}^{(m)}$. The popularity of this algorithm is explained by its fast convergence when the starting guess $\boldsymbol{\theta}^{(0)}$ is chosen close enough to the maximum that is unknown in practice [1]. Unfortunately, the source of NR algorithm's popularity is also that of his first flaw (its success strongly depends on the appropriate choice of $\boldsymbol{\theta}^{(0)}$ in a neighbourhood of the unknown solution). The second major drawback of the NR algorithm inherited from its mathematical formulation is that it requires the numerical inversion of the Hessian matrix $\nabla^2 L(\boldsymbol{\theta}^{(m)})$ (a square matrix of order d) at each iteration, which can be very tricky if at the iteration m, the matrix $\nabla^2 L(\boldsymbol{\theta}^{(m)})$ is singular or ill-conditioned. When the NR algorithm is unsuccessful, alternatives may be considered. Among them, we can mention quasi-Newton algorithms [2] (which compute approximations of $[\nabla^2 L(\boldsymbol{\theta}^{(m)})]^{-1}$ at each iteration), block-relaxation algorithms (which divide parameters into disjoint blocks and proceed to optimization by cycling through these blocks) [3], and derivative-free optimization (DFO) algorithms [4, 5].

In statistics, the last decades have seen the development and the fast breakthrough of the minorization-maximization (MM) principle for constructing maximization algorithms [6–8], and the expectation-maximization (EM) algorithm [9] which is considered as a special case of MM algorithm for statistical estimation with incomplete data [6]. The design of MM algorithms for maximizing $L(\boldsymbol{\theta})$ consists in constructing a special surrogate function whose maximization is simpler but equivalent to that of $L(\boldsymbol{\theta})$ and then maximizing that surrogate function. Since they do not need any matrix inversion, MM algorithms generally outperform the NR algorithm and are considered as relevant for high-dimensional problems and for some discrete multivariate distributions [10].

In this paper, we are interested in one of the very important issues in statistics applied to road safety which is the statistical evaluation of road safety measures. We consider a discrete multivariate statistical model coupling the distributions of accidents classified by level of severity on the sites where the measure has been implemented (called treated sites) and on the sites where the measure has not been implemented (called control sites). This model has a constrained parameter vector $\boldsymbol{\theta}$ of $1 + sr$ components where s is the number of treated sites and r is the number of accident severity levels. Our main purpose is to design an MM algorithm for the maximum likelihood estimation of the parameter vector $\boldsymbol{\theta}$. Since MM algorithms may be slow to converge, we also consider the acceleration of our proposed MM algorithm.

The rest of this paper is structured as follows. In Section 2, we describe the model and we present the maximum likelihood estimation problem. In Section 3, we devise our new MM algorithm for estimating the parameter vector $\boldsymbol{\theta}$; afterwards, we apply an acceleration scheme in order to get the accelerated version. In Section 4, we present the results of the comparison of our proposed MM algorithm and its accelerated version to NR and quasi-Newton algorithms.

2. Statistical Model and Parameter Estimation

Let us consider $s\,(s > 0)$ geographical sites (hereafter referred to as *treated sites*) where a road safety measure (maximum speed reduction, installation of roundabouts, and so on) has been implemented for a certain period of time. The accidents occurring on these sites are assumed to be classified by level of severity in r levels ($r > 0$). Suppose that, in order to avoid confusing the effect of the measure with those of other factors likely to influence the number of crashes, each treated site is paired with another geographical site (hereafter called *control site*) with the same characteristics (traffic flow, weather conditions, and so on) as the treated site but where the measure has not been implemented. For $k = 1, \cdots, s$, let

$$\mathbf{X}_k = (X_{11k}, X_{12k}, \cdots, X_{1rk}, X_{21k}, X_{22k}, \cdots, X_{2rk})^{\mathrm{T}} \tag{2}$$

be a vector composed of $2r$ random variables, where for all $j = 1, \cdots, r$, X_{1jk} (respectively, X_{2jk}) is a random variable representing the number of crashes of type j occurred on treated site k in the period before (respectively, after) the implementation of the measure. Also consider the non-random vector

$$\mathbf{z}_k = (z_{1k}, \cdots, z_{rk})^{\mathrm{T}}, \tag{3}$$

where for all $j = 1, \cdots, r$, z_{jk} is a non-random variable representing the ratio of the number of accidents of severity level j in the "after" period to the number of accidents of the same severity level in the "before" period on the control site.

Different models combining accident frequencies from the treated sites and the control sites have been proposed in order to estimate mainly the average effect α of the measure and, secondarily, the accident risks [11–13]. Let $\mathbb{S}_{r-1} = \{(p_1, \cdots, p_r)^{\mathrm{T}} \in [0,1]^r, \sum_{j=1}^r p_j = 1\}$, $\langle , \rangle$ be the classical inner product on $\mathbb{R}^r$, and n_k ($k = 1, \cdots, s$) be the total number of accidents observed at treated site k. In this paper, we consider the statistical model proposed in [13] under the following assumptions:

(A1) For all $k = 1, \cdots, s$, $\mathbf{X}_k$ follows the multinomial distribution $\mathscr{M}(n_k\,; \pi_k(\boldsymbol{\theta}))$, where

(A2) $\boldsymbol{\theta} = (\alpha, \beta^{\mathrm{T}})^{\mathrm{T}}$, $\alpha > 0$, $\beta = (\beta_1^{\mathrm{T}}, \cdots, \beta_s^{\mathrm{T}})^{\mathrm{T}} \in (\mathbb{S}_{r-1})^s$, and $\beta_k = (\beta_{1k}, \cdots, \beta_{rk})^{\mathrm{T}} \in \mathbb{S}_{r-1}$ for all $k = 1, \cdots, s$

(A3) $\pi_k(\boldsymbol{\theta}) = (\pi_{11k}(\boldsymbol{\theta}), \cdots, \pi_{1rk}(\boldsymbol{\theta}), \pi_{21k}(\boldsymbol{\theta}), \cdots, \pi_{2rk}(\boldsymbol{\theta}))^{\mathrm{T}}$,

$$\begin{aligned} \pi_{1jk}(\boldsymbol{\theta}) &= \frac{\beta_{jk}}{1 + \alpha\langle \mathbf{z}_k, \beta_k\rangle}, \\ \pi_{2jk}(\boldsymbol{\theta}) &= \frac{\alpha\beta_{jk}\langle \mathbf{z}_k, \beta_k\rangle}{1 + \alpha\langle \mathbf{z}_k, \beta_k\rangle}, \\ j &= 1, \cdots, r. \end{aligned} \tag{4}$$

Model (4) has a parameter vector $\boldsymbol{\theta} = (\alpha, \beta^{\mathrm{T}})^{\mathrm{T}} \in \mathbb{R}_+^* \times (\mathbb{S}_{r-1})^s$ where α is the mean effect of the measure and for all $j = 1, \cdots, r$, $k = 1, \cdots, s$, β_{jk} is the probability that an accident occurring on the treated site k is of severity level j. In this paper, we are interested in estimating the unknown parameter vector $\boldsymbol{\theta}$.

The likelihood of observed data $\mathbf{x}_1, \cdots, \mathbf{x}_s$, where for all $k = 1, \cdots, s$, $\mathbf{x}_k = (x_{11k}, \cdots, x_{1rk}, x_{21k}, \cdots, x_{2rk})^{\mathrm{T}}$, is given to one additive constant by

$$\begin{aligned} L(\boldsymbol{\theta}) = {} & \left(\sum_{k=1}^{s}\sum_{j=1}^{r} x_{\bullet jk} \log \beta_{jk}\right) + x_{2\bullet\bullet} \log \alpha \\ & - \sum_{k=1}^{s} n_k \log\left(1 + \alpha\langle \mathbf{z}_k, \beta_k\rangle\right) + \sum_{k=1}^{s} x_{2\bullet k} \log \langle \mathbf{z}_k, \beta_k\rangle, \end{aligned} \tag{5}$$

where for all $k = 1, \cdots, s$, $x_{\bullet jk} = x_{1jk} + x_{2jk}$ and $x_{2\bullet\bullet} = \sum_{k=1}^{s} \sum_{j=1}^{r} x_{2jk}$. The maximum likelihood estimate (MLE) $\widehat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}$ is the solution to the following constrained maximization problem:

$$\begin{aligned} &\text{maximize } L(\boldsymbol{\theta}) \\ &\text{subject to} \end{aligned} \tag{6a}$$

$$\begin{aligned} &\alpha > 0, \\ &\beta_{jk} > 0,\ j = 1, \cdots, r,\ k = 1, \cdots, s, \end{aligned} \tag{6b}$$

$$\sum_{j=1}^{r} \beta_{jk} = 1,\ k = 1, \cdots, s. \tag{6c}$$

In the case $s = 1$, there exists a closed-form expression of $\widehat{\boldsymbol{\theta}}$ [14]. However, in the general case $s \geq 1$, the optimization problem (6a), (6b), and (6c) looks impossible to solve in closed-form and therefore calls for numerical optimization. As stated earlier, different iterative algorithms may be used to solve (6a), (6b), and (6c), each with strengths and weaknesses. The minorization-maximization (MM) strategy for constructing optimization algorithms has been shown to yield algorithms (then called MM algorithms) that outperform the classical NR and quasi-Newton algorithms [6, 8, 10]. Moreover, it can handle constraints easily. In problems such as (6a), (6b), and (6c), it consists in defining a special surrogate function and afterwards maximizes this latter rather than the log-likelihood. MM algorithms are considered as relevant for high-dimensional problems and for discrete multivariate distributions [10]. In the next section, we explain the MM principle; afterwards, we devise an MM algorithm to solve the problem (6a), (6b), and (6c).

3. An MM Algorithm for Computing the MLE

3.1. Brief Reminder of the MM Principle for Constructing Maximization Algorithms. The MM principle for constructing an iterative maximization algorithm consists of two steps. Let $\boldsymbol{\theta}^{(m)}$ be the iterate after m iterations. For the maximization problem (6a), (6b), and (6c), the first M step consists in defining a minorizing function $g(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})$ and the second M step consists in maximizing the surrogate function $g(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})$ with respect to $\boldsymbol{\theta}$ rather than the log-likelihood $L(\boldsymbol{\theta})$, and the next iterate $\boldsymbol{\theta}^{(m+1)}$ is obtained as the value in which $g(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})$ attains its maximum. Before going further, let us remind the definition of a minorizing function [6].

For a given $\boldsymbol{\theta}^{(m)}$, a function $g(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})$ is said to minorize the function $L(\boldsymbol{\theta})$ at $\boldsymbol{\theta}^{(m)}$ provided

$$g\left(\boldsymbol{\theta}\middle|\boldsymbol{\theta}^{(m)}\right) \leq L(\boldsymbol{\theta}) \text{ for all } \boldsymbol{\theta}, \tag{7}$$

$$g\left(\boldsymbol{\theta}^{(m)}\middle|\boldsymbol{\theta}^{(m)}\right) = L\left(\boldsymbol{\theta}^{(m)}\right). \tag{8}$$

We have successively $L(\boldsymbol{\theta}^{(m+1)}) \geq g(\boldsymbol{\theta}^{(m+1)}|\boldsymbol{\theta}^{(m)})$ by Formula (7), $g(\boldsymbol{\theta}^{(m+1)}|\boldsymbol{\theta}^{(m)}) \geq g(\boldsymbol{\theta}^{(m)}|\boldsymbol{\theta}^{(m)})$ by Equation (21), and $g(\boldsymbol{\theta}^{(m)}|\boldsymbol{\theta}^{(m)}) = L(\boldsymbol{\theta}^{(m)})$ by Formula (8).

Thus, $L(\boldsymbol{\theta}^{(m+1)}) \geq L(\boldsymbol{\theta}^{(m)})$ and, therefore, a maximization MM algorithm is an ascent algorithm. This ascent property ensures its numerical stability [6].

3.2. Design and Maximization of the Minorizing Function. To develop our MM algorithm, we must define and maximize a minorizing function for $L(\boldsymbol{\theta})$. To this purpose, we use some mathematical inequalities related to convex functions presented in [6]. The following lemma gives the expression of the minorizing function.

Lemma 1. *Let $\boldsymbol{\theta}^{(m)}$ be a value of the vector parameter $\boldsymbol{\theta}$ and g the real-valued function of $\boldsymbol{\theta}$ defined by*

$$\begin{aligned} g\left(\boldsymbol{\theta}\middle|\boldsymbol{\theta}^{(m)}\right) = {} & C\left(\boldsymbol{\theta}^{(m)}\right) + \left(\sum_{k=1}^{s}\sum_{j=1}^{r} x_{\bullet jk} \log \beta_{jk}\right) \\ & + x_{2\bullet\bullet} \log \alpha - \sum_{k=1}^{s} \frac{n_k(1+\alpha\langle \mathbf{z}_k, \beta_k\rangle)}{1+\alpha^{(m)}\left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle} \\ & + \sum_{k=1}^{s}\sum_{j=1}^{r} x_{2\bullet k} \frac{z_{jk}\beta_{jk}^{(m)}}{\left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle} \log \beta_{jk}, \end{aligned} \tag{9}$$

where

$$\begin{aligned} C\left(\boldsymbol{\theta}^{(m)}\right) = {} & -\sum_{k=1}^{s} n_k \log\left(1+\alpha^{(m)}\left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle\right) + \sum_{k=1}^{s} n_k \\ & + \sum_{k=1}^{s}\sum_{j=1}^{r} x_{2\bullet k} \frac{z_{jk}\beta_{jk}^{(m)}}{\left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle} \log\left(\frac{\left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle}{\beta_{jk}^{(m)}}\right) \end{aligned} \tag{10}$$

is an additive constant independent of $\boldsymbol{\theta}$. Then, $g(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})$ minorizes $L(\boldsymbol{\theta})$ at $\boldsymbol{\theta}^{(m)}$.

Proof. By convexity of $-\log$, we know that, for all a, $b > 0$, $-\log(a/b) \geq -(a/b) + 1$. So, for all $k = 1, \cdots, s$,

$$-\log\left(\frac{1+\alpha\langle \mathbf{z}_k, \beta_k\rangle}{1+\alpha^{(m)}\left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle}\right) \geq -\frac{1+\alpha\langle \mathbf{z}_k, \beta_k\rangle}{1+\alpha^{(m)}\left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle} + 1. \tag{11}$$

Hence,

$$\begin{aligned} & -\log(1+\alpha\langle \mathbf{z}_k, \beta_k\rangle) \\ & \quad \geq -\log\left(1+\alpha^{(m)}\left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle\right) - \frac{1+\alpha\langle \mathbf{z}_k, \beta_k\rangle}{1+\alpha^{(m)}\left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle} + 1. \end{aligned} \tag{12}$$

We can then write

$$\begin{aligned}
&-n_k \log\left(1+\alpha\langle \mathbf{z}_k, \beta_k\rangle\right) \\
&\quad \geq -n_k \log\left(1+\alpha^{(m)}\left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle\right) \\
&\qquad - \frac{n_k(1+\alpha\langle \mathbf{z}_k, \beta_k\rangle)}{1+\alpha^{(m)}\left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle} + n_k,
\end{aligned} \tag{13}$$

and, after summing on the k indexes, we get

$$\begin{aligned}
&-\sum_{k=1}^{s} n_k \log\left(1+\alpha\langle \mathbf{z}_k, \beta_k\rangle\right) \\
&\quad \geq -\sum_{k=1}^{s} n_k \log\left(1+\alpha^{(m)}\left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle\right) \\
&\qquad - \sum_{k=1}^{s} \frac{n_k(1+\alpha\langle \mathbf{z}_k, \beta_k\rangle)}{1+\alpha^{(m)}\left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle} + \sum_{k=1}^{s} n_k.
\end{aligned} \tag{14}$$

By convexity of $-\log$ and by Equation (10) of [6], we also have

$$\begin{aligned}
\log\langle \mathbf{z}_k, \beta_k\rangle &= \log\left(\sum_{j=1}^{r} z_{jk}\beta_{jk}\right) \\
&\geq \sum_{j=1}^{r} \frac{z_{jk}\beta_{jk}^{(m)}}{\left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle} \log\left(\frac{\left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle \beta_{jk}}{\beta_{jk}^{(m)}}\right).
\end{aligned} \tag{15}$$

Hence,

$$\begin{aligned}
&\sum_{k=1}^{s} x_{2\bullet k} \log\langle \mathbf{z}_k, \beta_k\rangle \\
&\quad \geq \sum_{k=1}^{s}\sum_{j=1}^{r} x_{2\bullet k} \frac{z_{jk}\beta_{jk}^{(m)}}{\left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle} \log\left(\frac{\left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle \beta_{jk}}{\beta_{jk}^{(m)}}\right),
\end{aligned} \tag{16}$$

which is equivalent to

$$\begin{aligned}
&\sum_{k=1}^{s} x_{2\bullet k} \log\langle \mathbf{z}_k, \beta_k\rangle \\
&\quad \geq \sum_{k=1}^{s}\sum_{j=1}^{r} x_{2\bullet k} \frac{z_{jk}\beta_{jk}^{(m)}}{\left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle} \log \beta_{jk} \\
&\qquad + \sum_{k=1}^{s}\sum_{j=1}^{r} x_{2\bullet k} \frac{z_{jk}\beta_{jk}^{(m)}}{\left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle} \log\left(\frac{\left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle}{\beta_{jk}^{(m)}}\right).
\end{aligned} \tag{17}$$

From (14), (17), and the definition of $L(\boldsymbol{\theta})$ (see (5)), we

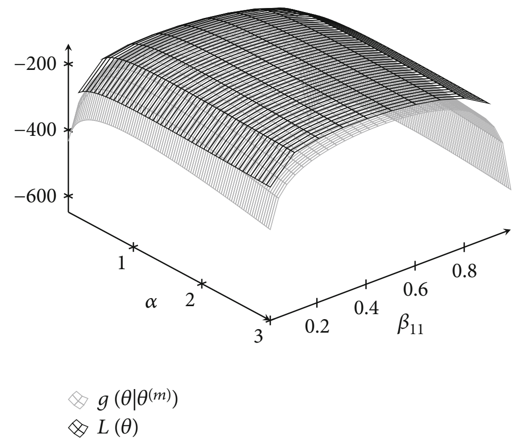

Figure 1: Illustration of the log-likelihood for $s = 1$ and $r = 2$ and its minorization at point $(\alpha^{(m)}, \beta_{11}^{(m)}) = (1, 0.5)$.

can write

$$\begin{aligned}
L(\boldsymbol{\theta}) \geq &\left(\sum_{k=1}^{s}\sum_{j=1}^{r} x_{\bullet jk} \log \beta_{jk}\right) + x_{2\bullet\bullet} \log \alpha \\
&- \sum_{k=1}^{s} n_k \log\left(1+\alpha^{(m)}\left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle\right) \\
&- \sum_{k=1}^{s} \frac{n_k(1+\alpha\langle \mathbf{z}_k, \beta_k\rangle)}{1+\alpha^{(m)}\left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle} + \sum_{k=1}^{s} n_k \\
&+ \sum_{k=1}^{s}\sum_{j=1}^{r} x_{2\bullet k} \frac{z_{jk}\beta_{jk}^{(m)}}{\left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle} \log \beta_{jk} \\
&+ \sum_{k=1}^{s}\sum_{j=1}^{r} x_{2\bullet k} \frac{z_{jk}\beta_{jk}^{(m)}}{\left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle} \log\left(\frac{\left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle}{\beta_{jk}^{(m)}}\right),
\end{aligned} \tag{18}$$

which means that $L(\boldsymbol{\theta}) \geq g(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})$ where $g(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})$ is defined by (9). Moreover, we have $g(\boldsymbol{\theta}^{(m)}|\boldsymbol{\theta}^{(m)}) = L(\boldsymbol{\theta}^{(m)})$. We can then conclude that $g(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})$ minorizes $L(\boldsymbol{\theta})$ at $\boldsymbol{\theta}^{(m)}$. □

Figure 1 gives an example of representation of $L(\boldsymbol{\theta})$ and $g(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})$ where $s = 1$, $r = 2$, and $\boldsymbol{\theta} = (\alpha, \beta_{11}, \beta_{21})^{\mathrm{T}}$. Since $\beta_{21} = 1 - \beta_{11}$, $L(\boldsymbol{\theta})$ and $g(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})$ are considered as functions of two variables $\alpha > 0$ and β_{11} such that $0 < \beta_{11} < 1$.

To finalize the design of our MM algorithm, we need to deal with the constraints (6b) and (6c). In [6], the authors recommend to extend function $g(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})$ under a new form that takes into account the inequality constraints; afterwards, equality constraints will be enforced during the optimization of the extended form of $g(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})$. Their method is based on the logarithmic barrier method for handling inequality constraints. For q inequality constraints of the form $v_i(\boldsymbol{\theta}) \geq 0$, $i = 1, \cdots, q$, the extension of $g(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})$ presented in Equation (23) of [6] is an additive value composed of linear combinations of $\log v_i(\boldsymbol{\theta})$. In this paper, the form of inequality constraint (6b) implies that the extension of $g(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(m)})$ will be composed of linear combinations of $\log \alpha$ and $\log \beta_{jk}$. So the log-likelihood $L(\boldsymbol{\theta})$ defined in Equation (5) already contains the logarithmic barrier (it diverges to negative infinity if any of the parameters α or β_{jk} tends to zero). Thus, as the authors of [6] themselves recognize, if the initial point $\theta^{(0)}$ satisfies inequality constraints (6b), then the presence of the terms $\log \alpha$ and $\log \beta_{jk}$ in the expression of $L(\boldsymbol{\theta})$ prevents $\alpha^{(m+1)} \leq 0$ and $\beta_{jk}^{(m+1)} \leq 0$ from occurring.

Now, knowing $\boldsymbol{\theta}^{(m)}$, we just have to maximize $g(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(m)})$ under the equality constraints (6c) to obtain the next iterate $\boldsymbol{\theta}^{(m+1)}$.

Theorem 2. *Let $\boldsymbol{\theta}^{(m)}$ be the estimate of the parameter vector $\boldsymbol{\theta}$ after m steps of our proposed MM algorithm. Then, the components of the next iterate $\boldsymbol{\theta}^{(m+1)}$ are given by*

$$\alpha^{(m+1)} = \frac{x_{2\bullet\bullet}}{\left(\sum_{k=1}^{s}\left(\left(n_k\left\langle \mathbf{z}_k, \beta_k^{(m+1)}\right\rangle\right) / \left(1+\alpha^{(m)}\left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle\right)\right)\right)}, \tag{19}$$

and for all $k = 1, \cdots, s$, $j = 1, \cdots, r$,

$$\beta_{jk}^{(m+1)} = \frac{x_{\bullet jk} + \left(\left(x_{2\bullet k} z_{jk} \beta_{jk}^{(m)}\right) / \left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle\right)}{n_k + x_{2\bullet k} + \left(\left(n_k \alpha^{(m+1)}\left(z_{jk} - \left\langle \mathbf{z}_k, \beta_k^{(m+1)}\right\rangle\right)\right) / \left(1+\alpha^{(m)}\left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle\right)\right)}. \tag{20}$$

Proof. If $\boldsymbol{\theta}^{(m)}$ is the estimate of $\boldsymbol{\theta}$ after m steps, then the next iterate denoted by $\boldsymbol{\theta}^{(m+1)}$ is obtained as

$$\boldsymbol{\theta}^{(m+1)} = \operatorname{argmax}_{\boldsymbol{\theta}} g\left(\boldsymbol{\theta} \middle| \boldsymbol{\theta}^{(m)}\right) \tag{21}$$

under equality constraints (6c). Solving problem (21) is equivalent to looking for the stationary point of the Lagrangian

$$\tilde{L}(\boldsymbol{\theta}, \lambda) = g\left(\boldsymbol{\theta} \middle| \boldsymbol{\theta}^{(m)}\right) + \sum_{k=1}^{s} \lambda_k \left(1 - \sum_{j=1}^{r} \beta_{jk}\right), \tag{22}$$

where $\boldsymbol{\lambda} = (\lambda_1, \cdots, \lambda_s)$ is a vector of s Lagrange's multipliers and g is defined by Equation (9). Setting $\partial \tilde{L}(\boldsymbol{\theta}, \lambda)/\partial \alpha$ and $\partial \tilde{L}(\boldsymbol{\theta}, \lambda)/\partial \beta_{jk}$ to zero, we get

$$\frac{x_{2\bullet\bullet}}{\alpha} - \sum_{k=1}^{s} \frac{n_k \langle z_k, \beta_k \rangle}{1+\alpha^{(m)}\left\langle z_k, \beta_k^{(m)}\right\rangle} = 0, \tag{23a}$$

$$\frac{x_{\bullet jk}}{\beta_{jk}} - \frac{n_k \alpha z_{jk}}{1+\alpha^{(m)}\left\langle z_k, \beta_k^{(m)}\right\rangle} + \frac{x_{2\bullet k}}{\beta_{jk}} \frac{z_{jk}\beta_{jk}^{(m)}}{\left\langle z_k, \beta_k^{(m)}\right\rangle} - \lambda_k = 0, \quad k = 1, \cdots, s, \ j = 1, \cdots, r. \tag{23b}$$

Multiplying (23a) by α and (23b) by β_{jk}, one gets

$$x_{2\bullet\bullet} - \alpha \sum_{k=1}^{s} \frac{n_k \langle z_k, \beta_k \rangle}{1+\alpha^{(m)}\left\langle z_k, \beta_k^{(m)}\right\rangle} = 0, \tag{24a}$$

$$x_{\bullet jk} - \frac{n_k \alpha z_{jk} \beta_{jk}}{1+\alpha^{(m)}\left\langle z_k, \beta_k^{(m)}\right\rangle} + \frac{x_{2\bullet k} z_{jk} \beta_{jk}^{(m)}}{\left\langle z_k, \beta_k^{(m)}\right\rangle} - \lambda_k \beta_{jk} = 0, \quad k = 1, \cdots, s, \ j = 1, \cdots, r. \tag{24b}$$

For all $k = 1, \cdots, s$, summing on the index j in Equation (24b) and noting that $\sum_{j=1}^{r} \beta_{jk} = 1$ and $\sum_{j=1}^{r} x_{\bullet jk} = n_k$ lead to

$$n_k - \frac{n_k \alpha \langle \mathbf{z}_k, \beta_k \rangle}{1+\alpha^{(m)}\left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle} + \frac{x_{2\bullet k}\left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle}{\left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle} - \lambda_k = 0. \tag{25}$$

Hence,

$$\lambda_k = n_k + x_{2\bullet k} - \frac{n_k \alpha \langle \mathbf{z}_k, \beta_k \rangle}{1+\alpha^{(m)}\left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle}. \tag{26}$$

Combination of (24b) and (26) leads to

$$\left(n_k + x_{2\bullet k} + \frac{n_k \alpha \left(z_{jk} - \langle \mathbf{z}_k, \beta_k \rangle\right)}{1+\alpha^{(m)}\left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle}\right) \beta_{jk} = x_{\bullet jk} + \frac{x_{2\bullet k} z_{jk} \beta_{jk}^{(m)}}{\left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle}. \tag{27}$$

Thus, the solution $\boldsymbol{\theta}$ to (21) satisfies

$$\alpha = \frac{x_{2\bullet\bullet}}{\left(\sum_{k=1}^{s}\left(\left(n_k \langle \mathbf{z}_k, \beta_k \rangle\right) / \left(1+\alpha^{(m)}\left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle\right)\right)\right)}, \tag{28}$$

and for all $k = 1, \cdots, s$, $j = 1, \cdots, r$,

$$\beta_{jk} = \frac{x_{\bullet jk} + \left(\left(x_{2\bullet k} z_{jk} \beta_{jk}^{(m)}\right) / \left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle\right)}{n_k + x_{2\bullet k} + \left(\left(n_k \alpha \left(z_{jk} - \langle \mathbf{z}_k, \beta_k \rangle\right)\right) / \left(1+\alpha^{(m)}\left\langle \mathbf{z}_k, \beta_k^{(m)}\right\rangle\right)\right)}. \tag{29}$$

The updates $\alpha^{(m+1)}$ and $\beta_{jk}^{(m+1)}$ are obtained by replacing α by $\alpha^{(m+1)}$, β_k by $\beta_k^{(m+1)}$, and β_{jk} by $\beta_{jk}^{(m+1)}$ in Equations (28) and (29). □

Remark 3. The updates in Formulas (19) and (20) are ideal but impossible to apply in practice since (a) the computation of $\alpha^{(m+1)}$ depends on the unknown $\beta_{jk}^{(m+1)}$ and vice versa and (b) in Equation (20), the computation of each $\beta_{jk}^{(m+1)}$ depends on $\beta_k^{(m+1)}$ and vice versa. To circumvent these difficulties, we can replace $\beta_k^{(m+1)}$ by $\beta_k^{(m)}$ on the right side of Equations (19) and (20).

Input: $\varepsilon > 0$, $\mathbf{x}_1, \dots \mathbf{x}_s$ and $\mathbf{z}_1, \dots, \mathbf{z}_s$
Output: MLE $\widehat{\boldsymbol{\theta}}$
Initialize $m = 0$ and $\boldsymbol{\theta}^{(0)} = \left(\alpha^{(0)}, \left(\beta_1^{(0)}\right)^{\mathrm{T}}, \cdots, \left(\beta_s^{(0)}\right)^{\mathrm{T}}\right)^{\mathrm{T}}$ where $\alpha^{(0)} > 0$ and for all $k = 1, \cdots, s$, $\beta_k^{(0)} = \left(\beta_{1k}^{(0)}, \cdots, \beta_{rk}^{(0)}\right)^{\mathrm{T}} \in \mathbb{S}_{r-1}$;
repeat
 $\alpha^{(m+1)} = x_{2\bullet\bullet} / \left(\sum_{k=1}^{s}\left(\left(n_k\langle \mathbf{z}_k, \beta_k^{(m)}\rangle\right)/\left(1 + \alpha^{(m)}\langle \mathbf{z}_k, \beta_k^{(m)}\rangle\right)\right)\right)$;
 For all $k = 1, \cdots, s$, update $\beta_k^{(m+1)} = \left(\beta_{1k}^{(m+1)}, \cdots, \beta_{rk}^{(m+1)}\right)^{\mathrm{T}}$, where for all $j = 1, \cdots, r$,
 $\beta_{jk}^{(m+1)} = \left(x_{\bullet jk} + \left(x_{2\bullet k} z_{jk}\beta_{jk}^{(m)}/\langle \mathbf{z}_k, \beta_k^{(m)}\rangle\right)\right)/\left(n_k + x_{2\bullet k} + \left(n_k\alpha^{(m+1)}\left(z_{jk} - \langle \mathbf{z}_k, \beta_k^{(m)}\rangle\right)/\left(1 + \alpha^{(m)}\langle \mathbf{z}_k, \beta_k^{(m)}\rangle\right)\right)\right)$;
 Set $\boldsymbol{\theta}^{(m+1)} = \left(\alpha^{(m+1)}, \left(\beta_1^{(m+1)}\right)^{\mathrm{T}}, \cdots, \left(\beta_s^{(m+1)}\right)^{\mathrm{T}}\right)^{\mathrm{T}}$ and $m \longleftarrow m + 1$
until $\left|L\left(\boldsymbol{\theta}^{(m)}\right) - L\left(\boldsymbol{\theta}^{(m-1)}\right)\right| < \varepsilon$
Set $\widehat{\boldsymbol{\theta}} \longleftarrow \boldsymbol{\theta}^{(m)}$.

Algorithm 1: MM algorithm for computing the MLE $\widehat{\boldsymbol{\theta}}$.

3.3. The Proposed MM Algorithm. Our proposed MM algorithm is Algorithm 1. It starts from $\boldsymbol{\theta}^{(0)} = \left(\alpha^{(0)}, \left(\beta^{(0)}\right)^{\mathrm{T}}\right)^{\mathrm{T}}$, where $\alpha^{(0)} > 0$ and $\beta^{(0)} = \left(\left(\beta_1^{(0)}\right)^{\mathrm{T}}, \cdots, \left(\beta_s^{(0)}\right)^{\mathrm{T}}\right)^{\mathrm{T}}$ are randomly set such that for all $k = 1, \cdots, s$, $\beta_k^{(0)} = \left(\beta_{1k}^{(0)}, \cdots, \beta_{rk}^{(0)}\right)^{\mathrm{T}} \in \mathbb{S}_{r-1}$. At the $(m+1)$-iteration, the update $\alpha^{(m+1)}$ is computed from $\alpha^{(m)}$ and $\beta^{(m)}$ using Formula (28) and Remark 3; afterwards, $\beta^{(m+1)}$ is updated from $\alpha^{(m+1)}$, $\alpha^{(m)}$, and $\beta^{(m)}$ using Formula (29) and Remark 3. This process is repeated until a convergence criterion is satisfied. Since the $\beta_{jk}^{(m+1)}$ are computed from $\alpha^{(m+1)}$, our MM algorithm (Algorithm 1) is thus a cyclic MM algorithm [6].

3.4. Acceleration of the MM Algorithm. In many statistical estimation problems, MM algorithms may need a great number of iterations and converge very slowly. In consequence, different acceleration schemes have been developed [15–17].

To the best of our knowledge, the acceleration schemes developed in [16] are among the most popular, and they will be also considered in this paper. The authors of [16] presented a new class of iterative methods, called SQUAREM (squared iterative methods for EM acceleration), for accelerating the expectation-minimization (EM) algorithm and state that SQUAREM may be also used to accelerate MM algorithms.

Let F be the MM map of Algorithm 1, i.e., the function from $\mathbb{R}_+^* \times (\mathbb{S}_{r-1})^s$ to itself defined by

$$F(\boldsymbol{\theta}) = \left(a, \mathbf{b}_1^{\mathrm{T}}, \cdots, \mathbf{b}_s^{\mathrm{T}}\right)^{\mathrm{T}}, \tag{30}$$

where

$$a = \frac{x_{2\bullet\bullet}}{\sum_{k=1}^{s}\left(\left(n_k\langle \mathbf{z}_k, \beta_k\rangle\right)/\left(1 + \alpha\langle \mathbf{z}_k, \beta_k\rangle\right)\right)}, \tag{31}$$

and for all $k = 1, \cdots, s$, $\mathbf{b}_k = (b_{1k}, \cdots, b_{rk})^{\mathrm{T}}$, where for all $j = 1, \cdots, r$,

$$b_{jk} = \frac{x_{\bullet jk} + \left(\left(x_{2\bullet k} z_{jk}\beta_{jk}\right)/\langle \mathbf{z}_k, \beta_k\rangle\right)}{n_k + x_{2\bullet k} + \left(\left(n_k a\left(z_{jk} - \langle \mathbf{z}_k, \beta_k\rangle\right)\right)/\left(1 + \alpha\langle \mathbf{z}_k, \beta_k\rangle\right)\right)}. \tag{32}$$

Knowing iterate $\boldsymbol{\theta}^{(m)}$, the SQUAREM consists in computing the next iterate $\boldsymbol{\theta}^{(m+1)}$ as

$$\boldsymbol{\theta}^{(m+1)} = \boldsymbol{\theta}^{(m)} - 2\gamma\mathbf{u} + \gamma^2\mathbf{v}, \tag{33}$$

where $\mathbf{u} = F\left(\boldsymbol{\theta}^{(m)}\right) - \boldsymbol{\theta}^{(m)}$, $\mathbf{v} = F \circ F\left(\boldsymbol{\theta}^{(m)}\right) - 2F\left(\boldsymbol{\theta}^{(m)}\right) - \boldsymbol{\theta}^{(m)}$, and γ is a scalar steplength. Varadhan and Roland [16] described three choices for the steplength, but, in many numerical experiments (see, for example, [16, 17]), the steplength

$$\gamma = -\frac{\|\mathbf{u}\|}{\|\mathbf{v}\|} \tag{34}$$

gives a faster convergence. As in [16, 17], the accelerated MM algorithm with steplength (34) (the third choice of steplength proposed by [16]) will be denoted SqS3. The SqS3 algorithm is given hereafter (see Algorithm 2).

Note that lines 7 to 9 of Algorithm 2 allow to correct the new iterate $\boldsymbol{\theta}^{(m+1)}$ by performing a simple step of the non-accelerated MM if $\boldsymbol{\theta}^{(m+1)}$ does not belong to the constrained parameter space $\mathbb{R}_+^* \times (\mathbb{S}_{r-1})^s$.

4. Simulation Study

We compare, in R software [18], our MM algorithm and its accelerated version SqS3 to the NR algorithm (package nleqslv [19]) and quasi-Newton BFGS algorithm (package alabama [20]). The design of the simulation study is inspired from [21, 22].

4.1. Data Generation. We have generated the data under assumptions (A1), (A2), and (A3), where the true parameter vector denoted by $\boldsymbol{\theta}^0 = \left(\alpha^0, \left(\beta_1^0\right)^{\mathrm{T}}, \cdots, \left(\beta_s^0\right)^{\mathrm{T}}\right)^{\mathrm{T}}$ has taken five values defined as follows:

Input: F, $\varepsilon > 0$, $\mathbf{x}_1, \ldots \mathbf{x}_s$ and $\mathbf{z}_1, \ldots, \mathbf{z}_s$
Output: MLE $\widehat{\theta}$
Initialize $m = 0$ and $\theta^{(0)} = (\alpha^{(0)}, (\beta_1^{(0)})^{\mathrm{T}}, \cdots, (\beta_s^{(0)})^{\mathrm{T}})^{\mathrm{T}}$ where $\alpha^{(0)} > 0$ and for all $k = 1, \cdots, s$, $\beta_k^{(0)} = (\beta_{1k}^{(0)}, \cdots, \beta_{rk}^{(0)})^{\mathrm{T}} \in \mathbb{S}_{r-1}$;
repeat
  $\mathbf{u} = F(\boldsymbol{\theta}^{(m)}) - \boldsymbol{\theta}^{(m)}$;
  $\mathbf{v} = F \circ F(\boldsymbol{\theta}^{(m)}) - 2F(\boldsymbol{\theta}^{(m)}) - \boldsymbol{\theta}^{(m)}$;
  $\gamma = -\|\mathbf{u}\|/\|\mathbf{v}\|$;
  $\boldsymbol{\theta}^{(m+1)} = \boldsymbol{\theta}^{(m)} - 2\gamma\mathbf{u} + \gamma^2\mathbf{v}$;
  if $\boldsymbol{\theta}^{(m+1)} \notin \mathbb{R}_+^* \times (\mathbb{S}_{r-1})^s$ **then**
    $\boldsymbol{\theta}^{(m+1)} = F(\boldsymbol{\theta}^{(m)})$;
  end
  $m \longleftarrow m + 1$;
until $|L(\boldsymbol{\theta}^{(m)}) - L(\boldsymbol{\theta}^{(m-1)})| < \varepsilon$
Set $\widehat{\boldsymbol{\theta}} \longleftarrow \boldsymbol{\theta}^{(m)}$.

Algorithm 2: Accelerated MM algorithm (SqS3) for computing the MLE $\widehat{\boldsymbol{\theta}}$.

Scenario 1. $s = 2$, $r = 2$,

$$\begin{aligned} \alpha^0 &= 0.8, \\ \beta_1^0 &= (0.65, 0.35)^{\mathrm{T}}, \\ \beta_2^0 &= (0.25, 0.75)^{\mathrm{T}}. \end{aligned} \tag{35}$$

Scenario 2. $s = 5$, $r = 3$,

$$\begin{aligned} \alpha^0 &= 1, \\ \beta_1^0 &= (0.80, 0.15, 0.05)^{\mathrm{T}}, \\ \beta_2^0 &= (0.10, 0.30, 0.60)^{\mathrm{T}}, \\ \beta_3^0 &= (0.35, 0.30, 0.35)^{\mathrm{T}}, \\ \beta_4^0 &= (0.70, 0.20, 0.10)^{\mathrm{T}}, \\ \beta_5^0 &= (0.30, 0.40, 0.30)^{\mathrm{T}}. \end{aligned} \tag{36}$$

Scenario 3. $s = 10$, $r = 3$,

$$\begin{aligned} \alpha^0 &= 1, \\ \beta_k^0 &= (0.40, 0.15, 0.45)^{\mathrm{T}}, k \in \{1, 3, 5, 9\}, \\ \beta_k^0 &= (0.55, 0.25, 0.20)^{\mathrm{T}}, k \in \{2, 4, 7\}, \\ \beta_k^0 &= (0.20, 0.30, 0.50)^{\mathrm{T}}, k \in \{6, 8, 10\}. \end{aligned} \tag{37}$$

Scenario 4. $s = 10$, $r = 5$,

$$\begin{aligned} \alpha^0 &= 1, \\ \beta_k^0 &= (0.40, 0.10, 0.05, 0.25, 0.20)^{\mathrm{T}}, k \in \{1, 3, 5, 9\}, \\ \beta_k^0 &= (0.30, 0.15, 0.10, 0.25, 0.20)^{\mathrm{T}}, k \in \{2, 4, 7\}, \\ \beta_k^0 &= \left(\underbrace{0.20, \cdots, 0.20}_{5}\right)^{\mathrm{T}}, k \in \{6, 8, 10\}. \end{aligned} \tag{38}$$

Scenario 5. $s = 20$, $r = 5$,

$$\begin{aligned} \alpha^0 &= 1.2, \\ \beta_k^0 &= (0.65, 0.15, 0.05, 0.05, 0.10)^{\mathrm{T}}, \; k \in \{1, 3, 5, 9, 11, 13, 15, 19\}, \\ \beta_k^0 &= (0.30, 0.20, 0.25, 0.10, 0.15)^{\mathrm{T}}, \; k \in \{2, 4, 7, 12, 14, 17\}, \\ \beta_k^0 &= \left(\underbrace{0.20, \cdots, 0.20}_{5}\right)^{\mathrm{T}}, \; k \in \{6, 8, 10, 16, 18, 20\}. \end{aligned} \tag{39}$$

For all $k = 1, \cdots, s$, we have given n_k two values: $n = 50$ and $n = 5000$. The starting guess $\boldsymbol{\theta}^{(0)} = (\alpha^{(0)}, (\beta^{(0)})^{\mathrm{T}})^{\mathrm{T}}$ is randomly generated as follows: $\alpha^{(0)}$ is a random observation of the uniform distribution $\mathscr{U}[0.1\,;5]$ and for all $k = 1, \cdots, s$, $\beta_k^{(0)} = (\beta_{1k}^{(0)}, \cdots, \beta_{rk}^{(0)})^{\mathrm{T}}$ is randomly generated using the formula

$$\beta_k^{(0)} = \frac{1}{\sum_{j=1}^{r} u_j}(u_1, \cdots, u_r)^{\mathrm{T}}, \tag{40}$$

where for all $j = 1, \cdots, r$, u_j is a random observation from the uniform distribution on $[0.05, 0.95]$.

4.2. Results. For the different scenarios and values of n, the average values obtained over 1000 replications are given by Tables 1–5. In these tables, convergence proportion refers to the percentage of convergence over 1000 replications, CPU (central processing unit) times are given in seconds, and time ratios are obtained by dividing the CPU time of each algorithm by that of the MM algorithm (which explains why the time ratio of MM is always equal to 1). The mean square error (MSE) linked to an estimate $\widehat{\boldsymbol{\theta}}$ is

Table 1: Results for Scenario 1 ($s = 2$ and $r = 2$). Values in brackets are standard deviations.

		MM	SqS3	NR	BFGS
$n = 50$	α	0.826 (0.176)	0.826 (0.176)	0.826 (0.176)	0.824 (0.174)
	β_{11}	0.649 (0.068)	0.649 (0.068)	0.649 (0.068)	0.648 (0.068)
	β_{21}	0.351 (0.068)	0.351 (0.068)	0.351 (0.068)	0.352 (0.068)
	β_{12}	0.247 (0.062)	0.247 (0.062)	0.247 (0.062)	0.246 (0.061)
	β_{22}	0.753 (0.062)	0.753 (0.062)	0.753 (0.062)	0.754 (0.061)
	Convergence proportion (%)	100	100	99.5	76.1
	Iterations	24 (4.1)	7 (1)	5.5 (1.5)	13 (0.1)
	CPU time (secs)	0.004	0.002	0.002	0.087
	Time ratio	1.00	0.51	0.62	23.63
	Log-likelihood	-126.66	-126.66	-126.67	-126.71
	MSE	$9.7e-03$	$9.7e-03$	$9.7e-03$	$9.5e-03$
$n = 5000$	α	0.800 (0.016)	0.800 (0.016)	0.800 (0.016)	0.800 (0.030)
	β_{11}	0.650 (0.007)	0.650 (0.007)	0.650 (0.007)	0.650 (0.010)
	β_{21}	0.350 (0.007)	0.350 (0.007)	0.350 (0.007)	0.350 (0.010)
	β_{12}	0.250 (0.006)	0.250 (0.006)	0.250 (0.006)	0.250 (0.009)
	β_{22}	0.750 (0.006)	0.750 (0.006)	0.750 (0.006)	0.750 (0.009)
	Convergence proportion (%)	100	100	99.7	77.3
	Iterations	31.2 (4.3)	8.2 (1)	5.7 (1.4)	16 (0.5)
	CPU time (secs)	0.006	0.003	0.003	0.165
	Time ratio	1.00	0.45	0.50	26.25
	Log-likelihood	-12832.34	-12832.34	-12832.39	-12838.15
	MSE	$8.6e-05$	$8.6e-05$	$8.6e-05$	$2.4e-04$

Table 2: Results for Scenario 2 ($s = 5$ and $r = 3$). Values in brackets are standard deviations.

		MM	SqS3	NR	BFGS
$n = 50$	Convergence proportion (%)	100	100	99.6	50.4
	Iterations	29.3 (4.2)	7.8 (0.8)	8.1 (3)	14 (0.1)
	CPU time (secs)	0.007	0.003	0.015	0.283
	Time ratio	1.00	0.47	2.14	39.93
	Log-likelihood	-391.24	-391.24	-391.21	-391.6
	MSE	$4.3e-03$	$4.3e-03$	$4.3e-03$	$4.2e-03$
$n = 5000$	Convergence proportion (%)	100	100	99.7	53.3
	Iterations	38.2 (4.1)	9.1 (0.8)	8.4 (2.9)	18.3 (3.4)
	CPU time (secs)	0.009	0.004	0.016	0.533
	Time ratio	1.00	0.43	1.80	59.55
	Log-likelihood	-39181.57	-39181.57	-39181.5	-39942.08
	MSE	$4.3e-05$	$4.3e-05$	$4.3e-05$	$1.1e-02$

$$\text{MSE}\left(\widehat{\boldsymbol{\theta}} \middle| \boldsymbol{\theta}^0\right) = \frac{1}{1+sr}\left(\left(\widehat{\alpha} - \alpha^0\right)^2 + \sum_{k=1}^{s}\sum_{j=1}^{r}\left(\widehat{\beta}_{jk} - \beta_{jk}^0\right)^2\right). \tag{41}$$

To avoid overloading the tables, the estimate $\widehat{\boldsymbol{\theta}}$ has been included for Scenario 1 only (see Table 1).

In Tables 1–5, we can notice that the estimated values, the standard deviation, the log-likelihoods, and the MSE are globally the same for all algorithms except BFGS when $n = 5000$. Regarding the convergence proportions, our proposed MM algorithm and its accelerated version SqS3 always have a 100% convergence proportion while the convergence proportion of NR is close but strictly lower than 100%. The convergence proportion of BFGS varies between 4.1% and

Table 3: Results for Scenario 3 ($s = 10$ and $r = 3$). Values in brackets are standard deviations.

		MM	SqS3	NR	BFGS
$n = 50$	Convergence proportion (%)	100	100	99.7	28
	Iterations	30.2 (3.9)	7.9 (0.7)	8.6 (2.5)	14 (0.2)
	CPU time (secs)	0.009	0.004	0.043	0.426
	Time ratio	1.00	0.50	4.90	48.60
	Log-likelihood	-831.26	-831.26	-831.27	-831.07
	MSE	$4.1e-03$	$4.1e-03$	$4.1e-03$	$4.1e-03$
$n = 5000$	Convergence proportion (%)	100	100	99.6	30.9
	Iterations	39.5 (3.6)	9.2 (0.8)	8.8 (2.3)	16 (6.4)
	CPU time (secs)	0.012	0.005	0.045	0.669
	Time ratio	1.00	0.41	3.70	55.22
	Log-likelihood	-84027.58	-84027.58	-84027.15	-88978.38
	MSE	$4.2e-05$	$4.2e-05$	$4.2e-05$	$2.4e-02$

Table 4: Results for Scenario 4 ($s = 10$ and $r = 5$). Values in brackets are standard deviations.

		MM	SqS3	NR	BFGS
$n = 50$	Convergence proportion (%)	100	100	98.9	21.1
	Iterations	32.8 (3.9)	8.3 (0.8)	11.4 (4)	13.8 (0.5)
	CPU time (secs)	0.010	0.004	0.089	0.454
	Time ratio	1.00	0.46	9.33	47.45
	Log-likelihood	-1046.83	-1046.83	-1046.88	-1047.3
	MSE	$2.9e-03$	$2.9e-03$	$2.9e-03$	$3e-03$
$n = 5000$	Convergence proportion (%)	100	100	99.4	24.8
	Iterations	43.9 (4.1)	9.9 (0.9)	11.5 (3.9)	15.8 (5.5)
	CPU time (secs)	0.014	0.006	0.090	0.576
	Time ratio	1.00	0.41	6.32	40.50
	Log-likelihood	-104802.55	-104802.55	-104800.93	-108716.18
	MSE	$3e-05$	$3e-05$	$3e-05$	$1.1e-02$

Table 5: Results for Scenario 5 ($s = 20$ and $r = 5$). Values in brackets are standard deviations.

		MM	SqS3	NR	BFGS
$n = 50$	Convergence proportion (%)	100	100	97.9	4.1
	Iterations	34.9 (3.5)	8.4 (0.7)	14.3 (4.9)	14 (0)
	CPU time (secs)	0.017	0.007	0.338	1.162
	Time ratio	1.00	0.42	19.54	67.25
	Log-likelihood	-2045.46	-2045.46	-2045.65	-2046.96
	MSE	$2.7e-03$	$2.7e-03$	$2.7e-03$	$2.8e-03$
$n = 5000$	Convergence proportion (%)	100	100	96.8	16.2
	Iterations	46.3 (3.6)	10.1 (0.8)	14.9 (5.2)	6.6 (7.2)
	CPU time (secs)	0.018	0.006	0.347	0.582
	Time ratio	1.00	0.35	19.25	32.31
	Log-likelihood	-203353.48	-203353.48	-203355.72	-245597.55
	MSE	$2.8e-05$	$2.8e-05$	$2.8e-05$	$4.5e-02$

77.3% and decreases when the number of parameters increases.

For all the algorithms, we notice a decrease of the MSE when n increases from 50 to 5000 which suggests a good fitting between the true and estimated values when n increases. When $n = 5000$, the MSE of BFGS is greater than those of the other algorithms which suggests a convergence of BFGS to bad values.

As far as the CPU times are concerned, we notice that, except for Scenario 1, the computation time of NR algorithm is greater than that of the MM algorithm (more than 2 times for $n = 50$ and more than 1.8 times for $n = 5000$ in Table 2, more than 3 times in Table 3, more than 6 times in Table 4, and more than 19 times in Table 5). The proposed MM algorithm is 23 to 67 times faster than BFGS.

The performance of the MM is even better with acceleration. Indeed, the CPU time ratio of SqS3 is between 0.35 and 0.51. The percentage of computation time reduction yielded by the acceleration varies between 49% and 65% (since $(1 - 0.51) \times 100\% = 49\%$ and $(1 - 0.35) \times 100\% = 65\%$).

5. Conclusion

In this paper, we built a minorization-maximization (MM) algorithm to compute, under box constraints and linear equality constraints, the maximum likelihood estimates of the parameters of a multivariate statistical model used in the analysis of accident frequencies. This statistical model is composed of several multinomial distributions whose parameters are dependent. Since MM algorithms are generally considered as slow to converge, we have also proposed an accelerated version of our MM algorithm using a square iterative acceleration scheme developed in [16]. Using simulated data, we have proven that our proposed MM algorithm and its accelerated version are better than Newton-Raphson (NR) and quasi-newton BFGS algorithms in terms of convergence proportion and computation time.

Data Availability

This research uses simulated data and the data generation process is described in the paper.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] I. Griva, S. G. Nash, and A. Sofer, *Linear and Nonlinear Optimization*, Society for Industrial and Applied Mathematics, Philadelphia, 2nd edition, 2009.

[2] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, New York, second edition, 2006.

[3] J. de Leeuw, "Block-relaxation algorithms in statistics," in *Information Systems and Data Analysis*, Studies in Classification, Data Analysis, and Knowledge Organization, H.-H. Bock, W. Lenski, and M. M. Richter, Eds., pp. 308–324, Springer, Berlin, Heidelberg, 1994.

[4] J. Larson, M. Menickelly, and S. M. Wild, "Derivative-free optimization methods," *Acta Numerica*, vol. 28, pp. 287–404, 2019.

[5] L. M. Rios and N. V. Sahinidis, "Derivative-free optimization: a review of algorithms and comparison of software implementations," *Journal of Global Optimization*, vol. 56, no. 3, pp. 1247–1293, 2013.

[6] D. R. Hunter and K. Lange, "A tutorial on MM algorithms," *The American Statistician*, vol. 58, no. 1, pp. 30–37, 2004.

[7] K. Lange, *MM Optimization Algorithms*, Society for Industrial and Applied Mathematics, Philadelphia, 2016.

[8] K. Lange, D. R. Hunter, and I. Yang, "Optimization transfer using surrogate objective functions," *Journal of Computational and Graphical Statistics*, vol. 9, no. 1, pp. 1–20, 2000.

[9] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM Algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–22, 1977.

[10] H. Zhou and K. Lange, "MM algorithms for some discrete multivariate distributions," *Journal of Computational and Graphical Statistics*, vol. 19, no. 3, pp. 645–665, 2010.

[11] A. N'Guessan and F. Bellavance, "A confidence interval estimation problem using Schur complement approach," *Comptes Rendus Mathématiques de l'Académie des Sciences du Canada*, vol. 27, no. 3, pp. 84–91, 2005.

[12] A. N'Guessan, A. Essai, and C. Langrand, "Estimation multidimensionnelle des contrôles et de l'effet moyen d'une mesure de sécurité routière," *Revue de Statistique Appliquée*, vol. 49, no. 2, pp. 85–102, 2001.

[13] A. N'Guessan, A. Essai, and M. N'Zi, "An estimation method of the average effect and the different accident risks when modelling a road safety measure: a simulation study," *Computational Statistics & Data Analysis*, vol. 51, no. 2, pp. 1260–1277, 2006.

[14] I. C. Geraldo, "On the maximum likelihood estimator for a discrete multivariate crash frequencies model," *Afrika Statistika*, vol. 15, no. 2, pp. 2335–2348, 2020.

[15] N. C. Henderson and R. Varadhan, "Damped Anderson acceleration with restarts and monotonicity control for accelerating EM and EM-like algorithms," *Journal of Computational and Graphical Statistics*, vol. 28, no. 4, pp. 834–846, 2019.

[16] R. Varadhan and C. Roland, "Simple and globally convergent methods for accelerating the convergence of any EM algorithm," *Scandinavian Journal of Statistics*, vol. 35, no. 2, pp. 335–353, 2008.

[17] H. Zhou, D. Alexander, and K. Lange, "A quasi-Newton acceleration for high-dimensional optimization algorithms," *Statistics and Computing*, vol. 21, no. 2, pp. 261–273, 2011.

[18] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2022, https://www.R-project.org/.

[19] B. Hasselman, "nleqslv: solve systems of nonlinear equations," 2018, https://CRAN.R-project.org/package=nleqslv. R package version 3.3.2.

[20] R. Varadhan, "alabama: constrained nonlinear optimization," 2015, https://CRAN.R-project.org/package=alabama. R package version 2015.

[21] I. C. Geraldo, "An automated profile-likelihood-based algorithm for fast computation of the maximum likelihood estimate in a statistical model for crash data," *Journal of Applied Mathematics*, vol. 2022, Article ID 6974166, 11 pages, 2022.

[22] I. C. Geraldo, T. A. Kpanzou, and E. Katchekpele, "A new hybrid algorithm for maximum likelihood estimation in a model of accident frequencies," *Malaya Journal of Matematik*, vol. 11, no. 1, pp. 11–24, 2023.