Hindawi
Journal of Applied Mathematics
Volume 2022, Article ID 6974166, 11 pages
https://doi.org/10.1155/2022/6974166

*Research Article*

# An Automated Profile-Likelihood-Based Algorithm for Fast Computation of the Maximum Likelihood Estimate in a Statistical Model for Crash Data

**Issa Cherif Geraldo**

*Département de Mathématiques, Faculté des Sciences, Université de Lomé, 1 B.P. 1515 Lomé 1, Togo*

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

Received 15 August 2022; Accepted 1 October 2022; Published 26 October 2022

Academic Editor: Wei-Chiang Hong

Numerical computation of maximum likelihood estimates (MLE) is one of the most common problems encountered in applied statistics. Even if there exist many algorithms considered as performing, they can suffer in some cases for one or many of the following criteria: global convergence (capacity of an algorithm to converge to the true unknown solution from all starting guesses), numerical stability (ascent property), implementation feasibility (for example, algorithms requiring matrix inversion cannot be implemented when the involved matrices are not invertible), low computation time, low computational complexity, and capacity to handle high dimensional problems. The reality is that, in practice, no algorithm is perfect, and for each problem, it is necessary to find the most performing of all existing algorithms or even develop new ones. In this paper, we consider the computing of the maximum likelihood estimate of the vector parameter of a statistical model of crash frequencies. We split the parameter vector, and we develop a new estimation algorithm using the profile likelihood principle. We provide an automatic starting guess for which convergence and numerical stability are guaranteed. We study the performance of our new algorithm on simulated data by comparing it to some of the most famous and modern optimization algorithms. The results suggest that our proposed algorithm outperforms these algorithms.

## 1. Introduction

Let $\ell(\theta)$ be a log-likelihood function where $\theta \in \mathbb{R}^d$ is a parameter vector whose structure will be specified later. In computing the maximum likelihood estimate (MLE) which is the point $\widehat{\theta}$ at which $\ell(\theta)$ attains its maximum, the very first algorithm one can try is the Newton-Raphson (NR) algorithm which is no longer to be presented. The success of this algorithm comes from an enviable and unequalled property: that of quadratic convergence when the starting guess is well chosen (i.e., close to the unknown solution). However, it also has drawbacks: it is not globally convergent (i.e., its success depends on the starting guess); it can be costly or impossible to implement in high-dimensional problems because of its need of inverting the Hessian matrix at each iteration. Many other algorithms can be used whenever NR cannot. We refer the reader to [1–4] for a comprehensive review of these algorithms. Of all these algorithms, we can mention quasi-Newton (which use approximations of the inverse of the Hessian matrix rather than inverting it), Fisher scoring (a purely statistical method which consists in replacing the Hessian matrix with its mathematical expectation in order to ensure the ascent property and therefore numerical stability), block optimization, and derivative-free optimization algorithms. We can also mention minorization-maximization (MM) algorithms [5, 6] which have made an extraordinary breakthrough in computational statistics and are increasingly used. The MM philosophy for maximizing a function $\ell(\theta)$ is to define, in the first M step, a minorizing function for the objective function, and to maximize, in the second M step, the minorizing function with respect to the parameter vector $\theta$.

Even if all these algorithms are considered as performing, they can suffer in some cases for one or many of the following criteria: global convergence (capacity of an algorithm to converge to the true unknown solution from all starting guesses), numerical stability (ascent property), implementation feasibility (for example, algorithms requiring matrix inversion cannot be implemented when the involved matrices are not invertible), low computation time, low computational complexity, and capacity to handle high dimensional problems. The reality is that, in practice, no algorithm is perfect, and for each problem, it is necessary to find the most performing of all existing algorithms or even develop new ones.

In this paper, we consider the computing of the maximum likelihood estimate (MLE) of the vector parameter of a statistical model of crash frequencies proposed by [7]. The vector parameter has the form $\theta = (\alpha, \beta^{\mathrm{T}})^{\mathrm{T}}$ where $\alpha > 0$ is the parameter of interest and $\beta$ is a vector of secondary parameters. Although secondary, the subvector $\beta$ can contain an important number of components (up to several hundreds) depending on the structure of the data. Thus, the classical algorithms mentioned above may need a great number of iterations (and thus have a slow convergence) or simply fail to converge. For the model considered in this paper, Newton-Raphson and Minorization-Maximization (MM) algorithms have been implemented by [7, 8], but the numerical study of these algorithms has been limited to simple cases. The NR algorithm is known to be fast in simple cases and inefficient (or even impossible to implement) for high-dimensional problems. Moreover, its convergence depends on the starting guess (initial solution $\theta^{(0)}$ from which the algorithm starts). The MM algorithm, on the other hand, often requires a large number of iterations before converging to the solution.

The main contribution of this paper is to build an estimation algorithm which converges faster than the other algorithms and whose convergence is guaranteed and is not affected by the dimension (the number of components of $\theta$). For this purpose, we exploit the splitting of the parameter vector into two blocks, and we apply the profile likelihood principle (see for example [9], p. 231) to reduce the search of $\theta$ to only the search of the parameter of interest $\alpha$. Then, we develop a new estimation algorithm for which we provide an automatic starting guess from which convergence and numerical stability (ascent property) are guaranteed. This automatization of the starting guess reveals to be a true advantage for our algorithm over the others because it allows to circumvent the difficulty of finding an adequate starting guess. We show using simulated data that our algorithm outperforms Minorization-Maximization (MM) and Newton-Raphson algorithms which are two of the most famous and modern optimization algorithms.

The remainder of this paper is organized as follows. In Section 2, we describe the data and the estimation problem. The statistical model and the constrained maximum likelihood estimation of the parameters are also presented. In Section 3, we present the profile likelihood principle and then use it to design our new profile-likelihood-based algorithm (PLBA) for computing the MLE of $\theta$. We also provide an automatic starting guess, and we prove that it guarantees convergence of the proposed algorithm. In Section 4, we prove that the proposed algorithm satisfies the ascent property. In Section 5, we report the results of the comparison of the PLBA with MM and NR algorithms. The paper finishes with some discussions and concluding remarks in Section 6.

## 2. Data, Statistical Model, and Problem Setup

The framework of this paper is the statistical analysis of crash data before and after the implementation of a road safety measure at $s$ $(s > 0)$ experimental sites (called treated sites) where crashes are classified by severity in $r$ $(r > 0)$ levels. This analysis is aimed at estimating the mean effect $\alpha > 0$ of the safety measure simultaneously on all the $s$ sites. This mean effect must be understood in the multiplicative sense and therefore must be compared to 1. A value $\alpha < 1$ indicates a positive effect of the measure (an average reduction of $100 \times (1 - \alpha)\%$ in the number of crashes) while a value $\alpha > 1$ indicates a negative effect of the measure (an average increase of $100 \times (\alpha - 1)\%$ in the number of crashes) and $\alpha = 1$ indicates that the measure had no significant influence on the number of crashes.

Let

$$\mathbf{x} = \begin{bmatrix} x_{111} & x_{121} & \cdots & x_{1r1} & x_{211} & x_{221} & \cdots & x_{2r1} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ x_{11k} & x_{12k} & \cdots & x_{1rk} & x_{21k} & x_{22k} & \cdots & x_{2rk} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ x_{11s} & x_{12s} & \cdots & x_{1rs} & x_{21s} & x_{22s} & \cdots & x_{2rs} \end{bmatrix} \tag{1}$$

be a matrix of order $s \times (2r)$ where $x_{1jk}$ (respectively $x_{2jk}$) is the number of accidents of severity level $j$ $(j = 1, \cdots, r)$ which occurred at site $k$ in the period before (respectively, after) the implementation of the measure. Also let

$$\mathbf{z} = \begin{bmatrix} z_{11} & z_{21} & \cdots & z_{r1} \\ \vdots & \vdots & \cdots & \vdots \\ z_{1k} & z_{2k} & \cdots & z_{rk} \\ \vdots & \vdots & \cdots & \vdots \\ z_{1s} & z_{2s} & \cdots & z_{rs} \end{bmatrix} \tag{2}$$

be a matrix of order $s \times r$ where $z_{jk}$ $(j = 1, \cdots, r,\ k = 1, \cdots, s)$ is the ratio of the number of accidents of severity $j$ in the "after" period to the number of accidents of the same severity in the "before" period on a control site (a site where the measure has not been implemented) paired with treated site $k$.

Let $\mathbb{S}_{r-1} = \{(p_1, \cdots, p_r) \in [0, 1]^r, \sum_{j=1}^{r} p_j = 1\}$ and $n_k = \sum_{i=1}^{2} \sum_{j=1}^{r} x_{ijk}$ be the total number of accidents observed on the treated site $k$. N'Guessan et al. [7] proposed the following multinomial-based probability distribution of parameter vector $\theta$ for $\mathbf{x}$:

$$P(\mathbf{x}|\theta) = \prod_{k=1}^{s}\left(\frac{n_k!}{\prod_{i=1}^{2}\prod_{j=1}^{r}x_{ijk}!}\prod_{i=1}^{2}\prod_{j=1}^{r}\left(\pi_{ijk}(\theta|\mathbf{z})\right)^{x_{ijk}}\right), \quad (3)$$

where $\theta = (\alpha, \beta^{\mathrm{T}})^{\mathrm{T}}$ is the parameter vector, $\alpha > 0$ is the mean effect, $\beta = (\beta_1^{\mathrm{T}}, \cdots, \beta_s^{\mathrm{T}})^{\mathrm{T}} \in (\mathbb{S}_{r-1})^s$ is a vector of $sr$ secondary parameters such that for all $k = 1, \cdots, s$, $\beta_k = (\beta_{1k}, \cdots, \beta_{rk})^{\mathrm{T}} \in \mathbb{S}_{r-1}$, and

$$\pi_{ijk}(\theta|\mathbf{z}) = \begin{cases} \dfrac{\beta_{jk}}{1 + \alpha\langle \mathbf{z}_k, \beta_k\rangle}, & \text{if } i = 1, j = 1, \cdots, r, \\[2ex] \dfrac{\alpha z_{jk}\beta_{jk}}{1 + \alpha\langle \mathbf{z}_k, \beta_k\rangle}, & \text{if } i = 2, j = 1, \cdots, r, \end{cases} \quad (4)$$

where $\langle \mathbf{z}_k, \beta_k\rangle = \sum_{j'=1}^{r} z_{j'k}\beta_{j'k}$.

The parameter of the model is the vector $\theta = (\alpha, \beta^{\mathrm{T}})^{\mathrm{T}} \in \mathbb{R}_+^* \times (\mathbb{S}_{r-1})^s$, and the log-likelihood ($\log P(\mathbf{x}|\theta)$) of observed data $\mathbf{x}$ is given to one additive constant by

$$\ell(\theta) = \sum_{k=1}^{s}\sum_{j=1}^{r}\left\{x_{\bullet jk}\log\beta_{jk} + x_{2jk}\log\alpha - x_{\bullet jk}\log\left(1 + \alpha\langle \mathbf{z}_k, \beta_k\rangle\right)\right\}, \quad (5)$$

where $x_{\bullet jk} = x_{1jk} + x_{2jk}$ (see [7] for more details).

In this paper, we are interested in computing the maximum likelihood estimate (MLE) $\widehat{\theta}$ of the unknown vector parameter $\theta$ defined by

$$\widehat{\theta} = \operatorname{argmax}_{\theta \in \mathbb{R}_+^* \times (\mathbb{S}_{r-1})^s}\ell(\theta). \quad (6)$$

In the case $s = 1$, [10] built an algorithm to solve Equation (6). In the next section, we present a profile-likelihood-based algorithm (PLBA) for computing the MLE in the general case $s \geq 1$. Our proposed PLBA is automated since we provide an automatic starting guess which guarantees convergence.

## 3. The Automated Profile-Likelihood-Based Algorithm (PLBA) for Computing the MLE

*3.1. A Brief Reminder on Profile Likelihood.* Let us rewrite the log-likelihood as $\ell(\theta) = \ell(\alpha, \beta)$. If, for a given $\alpha$, the MLE $\widehat{\beta}$ of $\beta$ may be written as a function $\widehat{\beta}(\alpha)$ of $\alpha$, that is,

$$\widehat{\beta} = \widehat{\beta}(\alpha) = \operatorname{argmax}_{\beta \in (\mathbb{S}_{r-1})^s}\ell(\alpha, \beta), \quad (7)$$

then the profile likelihood function (see for example [9], p. 231) is

$$\ell_p(\alpha) = \ell\left(\alpha, \widehat{\beta}(\alpha)\right), \quad (8)$$

expressed as a function of $\alpha$ only. The maximization of $\ell_p(\alpha)$ is equivalent to that of $\ell(\alpha, \beta)$ [11].

*3.2. Computation of $\widehat{\beta}$ in Closed Form*

**Lemma 1.** *Given $\alpha > 0$, let $\widehat{\beta} = (\widehat{\beta}_1^T, \cdots, \widehat{\beta}_s^T)^T$ be the solution to Equation (7), where for all $k = 1, \cdots, s$, $\widehat{\beta}_k = (\widehat{\beta}_{1k}, \cdots, \widehat{\beta}_{rk})^T$. The components of $\widehat{\beta}$ are given by*

$$\widehat{\beta}_{jk} = \frac{x_{\bullet jk}/(1 + \alpha z_{jk})}{\sum_{m=1}^{r} x_{\bullet mk}/(1 + \alpha z_{mk})}, \quad j = 1, \cdots, r;\ k = 1, \cdots, s. \quad (9)$$

*Proof.* Given $\alpha > 0$, [7] proved that for all $k = 1, \cdots, s$, the components of $\widehat{\beta}_k$ satisfy the following system of $r$ equations:

$$x_{\bullet jk} - \frac{n_k\widehat{\beta}_{jk}(1 + \alpha z_{jk})}{1 + \alpha\left\langle \mathbf{z}_k, \widehat{\beta}_k\right\rangle} = 0, \quad j = 1, \cdots, r. \quad (10)$$

Thus, for all $k = 1, \cdots, s$, we apply Theorem 3.5 of [12] to Equation (10) and get

$$\widehat{\beta}_k = \frac{1}{n_k}(\mathbf{M}_{\alpha,k}/\mathbf{\Delta}_{\alpha,k})^{-1}\left(\frac{x_{\bullet 1k}}{1 + \alpha z_{1k}}, \cdots, \frac{x_{\bullet rk}}{1 + \alpha z_{rk}}\right)^{\mathrm{T}}, \quad (11)$$

where

$$\mathbf{M}_{\alpha,k} = \begin{bmatrix} \mathbf{\Delta}_{\alpha,k} & \alpha\mathbf{X}_{\bullet k} \\ \mathbf{z}_k^{\mathrm{T}} & 1 \end{bmatrix} \quad (12)$$

is a block-defined square matrix of order $r + 1$, $\mathbf{\Delta}_{\alpha,k} = \operatorname{diag}(1 + \alpha z_{1k}, \cdots, 1 + \alpha z_{rk})$ is a diagonal matrix of order $r$, $\mathbf{X}_{\bullet k} = (x_{\bullet 1k}, \cdots, x_{\bullet rk})^{\mathrm{T}}$, and

$$(\mathbf{M}_{\alpha,k}/\mathbf{\Delta}_{\alpha,k}) = \frac{1}{n_k}\sum_{m=1}^{r}\frac{x_{\bullet mk}}{1 + \alpha z_{mk}} \quad (13)$$

is the Schur complement of $\mathbf{\Delta}_{\alpha,k}$ in $\mathbf{M}_{\alpha,k}$ (see for example [13] (p. 34) for a reminder on the use of Schur complement for inverting a block matrix). The proof is thus completed. □

*3.3. Profile Likelihood*

**Theorem 2.** *The profile likelihood function is defined, up to an additive constant independent of $\alpha$, by*

$$\ell_p(\alpha) = x_{2\bullet\bullet}\log\alpha - \sum_{k=1}^{s}\sum_{j=1}^{r} x_{\bullet jk}\log\left(1 + \alpha z_{jk}\right), \quad (14)$$

*where $x_{2\bullet\bullet} = \sum_{k=1}^{s}\sum_{m=1}^{r} x_{2mk}$.*

*Proof.* Expression (5) is equivalent to

$$\ell(\theta) = \sum_{k=1}^{s}\sum_{j=1}^{r} x_{\bullet jk}\log\beta_{jk} + x_{2\bullet\bullet}\log\alpha - \sum_{k=1}^{s} n_k\log\left(1 + \alpha\langle \mathbf{z}_k, \beta_k\rangle\right). \quad (15)$$

For all $k = 1, \cdots, s$, Equation (9) yields

$$\begin{aligned} 1+\alpha\left\langle \mathbf{z}_k, \widehat{\beta}_k \right\rangle &= 1+\alpha\sum_{j=1}^{r} z_{jk}\widehat{\beta}_{jk} \\ &= 1+\frac{\alpha\left(\sum_{j=1}^{r}\left(z_{jk}x_{\bullet jk}\right)/\left(1+\alpha z_{jk}\right)\right)}{\left(\sum_{m=1}^{r}(x_{\bullet mk})/(1+\alpha z_{mk})\right)} \\ &= \frac{\left(\sum_{j=1}^{r}\left((x_{\bullet jk})/(1+\alpha z_{jk})\right)+\sum_{j=1}^{r}\left(\alpha z_{jk}x_{\bullet jk}\right)/\left(1+\alpha z_{jk}\right)\right)}{\left(\sum_{m=1}^{r}(x_{\bullet mk})/(1+\alpha z_{mk})\right)} \\ &= \frac{\left(\sum_{j=1}^{r}\left(x_{\bullet jk}\left(1+\alpha z_{jk}\right)\right)/\left(1+\alpha z_{jk}\right)\right)}{\left(\sum_{m=1}^{r}(x_{\bullet mk})/(1+\alpha z_{mk})\right)} \\ &= \frac{n_k}{\sum_{m=1}^{r}(x_{\bullet mk})/(1+\alpha z_{mk})}, \end{aligned} \tag{16}$$

and the relationships (9) and (16) enable us to write

$$\begin{aligned} \ell_p(\alpha) = {} & \sum_{k=1}^{s}\sum_{j=1}^{r} x_{\bullet jk}\log\left(\frac{x_{\bullet jk}}{1+\alpha z_{jk}}\right)-\sum_{k=1}^{s}\sum_{j=1}^{r} x_{\bullet jk}\log\left(\sum_{m=1}^{r}\frac{x_{\bullet mk}}{1+\alpha z_{mk}}\right) \\ & + x_{2\bullet\bullet}\log\alpha-\sum_{k=1}^{s} n_k\log\left(\frac{n_k}{\sum_{m=1}^{r}(x_{\bullet mk})/(1+\alpha z_{mk})}\right). \end{aligned} \tag{17}$$

After some manipulations on the first, second and fourth terms, we get:

$$\begin{aligned} \ell_p(\alpha) = {} & \sum_{k=1}^{s}\sum_{j=1}^{r} x_{\bullet jk}\log x_{\bullet jk}-\sum_{k=1}^{s}\sum_{j=1}^{r} x_{\bullet jk}\log\left(1+\alpha z_{jk}\right) \\ & -\sum_{k=1}^{s} n_k\log\left(\sum_{m=1}^{r}\frac{x_{\bullet mk}}{1+\alpha z_{mk}}\right)+x_{2\bullet\bullet}\log\alpha \\ & -\sum_{k=1}^{s} n_k\log n_k+\sum_{k=1}^{s} n_k\log\left(\sum_{m=1}^{r}\frac{x_{\bullet mk}}{1+\alpha z_{mk}}\right). \end{aligned} \tag{18}$$

Removing the third and sixth terms and the constants (first and fifth terms), we get (14). □

*3.4. Design of the PLBA*

**Lemma 3.** *Let F be the mapping defined on $\mathbb{R}_+$ by*

$$F(u) = -x_{1\bullet\bullet}+\sum_{k=1}^{s}\sum_{j=1}^{r}\frac{x_{\bullet jk}}{1+uz_{jk}}, \tag{19}$$

*where $x_{1\bullet\bullet}=\sum_{k=1}^{s}\sum_{j=1}^{r}x_{1jk}$. The MLE $\widehat{\alpha}$ of $\alpha$ is the unique root of F.*

*Proof.*

(i) On the one hand, function $F$ is a one-to-one decreasing function (it is continuous and its derivative $F'(u)$ is strictly negative for all $u \geq 0$) and $F(\mathbb{R}_+) = ]F_\infty, F(0)]$, where

$$\begin{aligned} F_\infty &= \lim_{u\longrightarrow+\infty} F(u) = -x_{1\bullet\bullet}, \\ F(0) &= -x_{1\bullet\bullet}+\sum_{k=1}^{s}\sum_{j=1}^{r} x_{\bullet jk} \\ &= -x_{1\bullet\bullet}+\sum_{k=1}^{s}\sum_{j=1}^{r}\left(x_{1jk}+x_{2jk}\right) \\ &= -x_{1\bullet\bullet}+x_{1\bullet\bullet}+x_{2\bullet\bullet} = x_{2\bullet\bullet}. \end{aligned} \tag{20}$$

Since $-x_{1\bullet\bullet} < 0 < x_{2\bullet\bullet}$, Equation $F(\alpha)=0$ has a unique solution.

(ii) On the other hand, the MLE $\widehat{\alpha}$, if it exists, is solution to the optimization problem

$$\widehat{\alpha} = \operatorname{argmax}_{\alpha>0}\ell_p(\alpha). \tag{21}$$

The profile log-likelihood $\ell_p(\alpha)$ being differentiable for every $\alpha > 0$, the MLE $\widehat{\alpha}$ is then solution to Equation $\ell_p'(\alpha) = 0$, where

$$\begin{aligned} \ell_p'(\alpha) &= \frac{x_{2\bullet\bullet}}{\alpha}-\sum_{k=1}^{s}\sum_{j=1}^{r}\frac{x_{\bullet jk}z_{jk}}{1+\alpha z_{jk}} \\ &= \frac{1}{\alpha}\left(x_{2\bullet\bullet}-\sum_{k=1}^{s}\sum_{j=1}^{r}\left(x_{\bullet jk}-\frac{x_{\bullet jk}}{1+\alpha z_{jk}}\right)\right) \\ &= \frac{1}{\alpha}\left(-x_{1\bullet\bullet}+\sum_{k=1}^{s}\sum_{j=1}^{r}\frac{x_{\bullet jk}}{1+\alpha z_{jk}}\right) \\ &= \frac{F(\alpha)}{\alpha} \end{aligned} \tag{22}$$

and $\sum_{k=1}^{s}\sum_{j=1}^{r}x_{\bullet jk} = x_{1\bullet\bullet}+x_{2\bullet\bullet}$. Thus, Equation $\ell_p'(\widehat{\alpha})=0$ is equivalent to $F(\widehat{\alpha})=0$, and $\widehat{\alpha}$ is the unique root of $F$. □

Equation $F(u)=0$ seems fairly complicated to solve in closed form for any $s>1$ and must therefore be solved numerically. Obviously, there are many root-finding algorithms (see for example [14] (Chapter 3) or [15] (Chapter 3)). Here, we propose a numerical approximation of $\widehat{\alpha}$ using the following one-dimensional Newton-Raphson (NR) root-finding algorithm:

$$\alpha^{(m+1)} = \alpha^{(m)}-\frac{F\left(\alpha^{(m)}\right)}{F'\left(\alpha^{(m)}\right)}, m = 0, 1, 2, \cdots, \tag{23}$$

where the starting guess $\alpha^{(0)}$ should be chosen in $\mathbb{R}_+$ by the user. Our choice of NR algorithm is motivated by the fact that it converges quadratically to the solution if $\alpha^{(0)}$ is chosen near the unknown solution. To overcome the difficulty of the choice of $\alpha^{(0)}$, we prove that, if we set $\alpha^{(0)}=0$ as an automatic starting guess, then, the convergence of NR iterations (23) is always guaranteed.

**Theorem 4.** *If $\alpha^{(0)} = 0$, then, the sequence $(\alpha^{(m)})$ defined by NR iterations (23) converges to the MLE $\widehat{\alpha}$.*

To prove Theorem 4, we need the following Lemma 5.

**Lemma 5.** *Let $\varphi$ be the function linked to NR iterations (23) such that $\alpha^{(m+1)} = \varphi(\alpha^{(m)})$, i.e., the function defined for all $u \in \mathbb{R}_+$ by*

$$\varphi(u) = u - \frac{F(u)}{F'(u)}. \tag{24}$$

*(i) Function $\varphi$ is increasing on $[0, \widehat{\alpha}]$ and decreasing on $[\widehat{\alpha}, +\infty[$.*

*(ii) The MLE $\widehat{\alpha}$ is the unique fixed point of $\varphi$.*

*(iii) For all $u \geq 0$,*

$$\begin{cases} \varphi(u) \geq u, & \text{if } u \leq \widehat{\alpha}, \\ \varphi(u) \leq u, & \text{if } u \geq \widehat{\alpha}. \end{cases} \tag{25}$$

*(iv) For all $u \geq 0$, $\varphi(u) \leq \widehat{\alpha}$.*

*Proof of Lemma 5.*

(i) For all $u \geq 0$, we have

$$F'(u) = -\sum_{k=1}^{s}\sum_{j=1}^{r} \frac{x_{\bullet jk} z_{jk}}{\left(1 + u z_{jk}\right)^2} < 0. \tag{26}$$

Therefore, $\varphi$ is differentiable and

$$\varphi'(u) = 1 - \left( \frac{\left(F'(u)\right)^2 - F(u)F''(u)}{\left(F'(u)\right)^2} \right) = \frac{F(u)F''(u)}{\left(F'(u)\right)^2}, \tag{27}$$

where for all $u \in [0,+\infty[$,

$$F''(u) = \sum_{k=1}^{s}\sum_{j=1}^{r} \frac{2x_{\bullet jk}\left(z_{jk}\right)^2}{\left(1 + u z_{jk}\right)^3} > 0. \tag{28}$$

So the sign of $\varphi'(u)$ is the same as the one of $F(u)$. As $F$ is a decreasing function and $F(\widehat{\alpha}) = 0$, we have

$$\begin{cases} F(u) \geq 0, & \text{if } u \leq \widehat{\alpha}, \\ F(u) \leq 0, & \text{if } u \geq \widehat{\alpha}. \end{cases} \tag{29}$$

It means that $\varphi$ is increasing on $[0, \widehat{\alpha}]$ and decreasing on $[\widehat{\alpha}, +\infty[$.

(ii) Equation $\varphi(u) = u$ is equivalent to $F(u) = 0$, where $F$ is defined by Formula (19). By Lemma 3, this equation has $\widehat{\alpha}$ as unique solution; hence, $\widehat{\alpha}$ is the unique fixed point of $\varphi$.

(iii) For all $u \geq 0$, $\varphi(u) - u = -F(u)/F'(u)$. The relation (25) is a simple consequence of (26) and (29).

(iv) Let $u \geq 0$. If $u \leq \widehat{\alpha}$, then $\varphi(u) \leq \varphi(\widehat{\alpha}) = \widehat{\alpha}$ because $\varphi$ is increasing on $[0, \widehat{\alpha}]$ (item (i)). If $u \geq \widehat{\alpha}$, then $\varphi(u) \leq \varphi(\widehat{\alpha}) = \widehat{\alpha}$ because $\varphi$ is decreasing on $[\widehat{\alpha}, +\infty[$. Thus, for all $u \geq 0$, $\varphi(u) \leq \widehat{\alpha}$.

□

We may now prove Theorem 4.

*Proof of Theorem 4.* Assume that $\alpha^{(0)} = 0$. Then, $\alpha^{(0)} \leq \widehat{\alpha}$ and by item (iii) and (iv) of Lemma 5, $\alpha^{(0)} \leq \varphi(\alpha^{(0)}) = \alpha^{(1)} \leq \widehat{\alpha}$. It can be easily proved by induction that

$$\alpha^{(0)} \leq \alpha^{(1)} \leq \cdots \leq \alpha^{(m)} \leq \alpha^{(m+1)} \leq \widehat{\alpha}. \tag{30}$$

Thus, the sequence $(\alpha^{(m)})$ is increasing and bounded; hence, it converges to the unique fixed point of $\varphi$ which is $\widehat{\alpha}$. □

*Remark 6.* Actually, from the proof of Theorem 4, it is clear that any starting value $\alpha^{(0)} \in [0, \widehat{\alpha}]$ will guarantee convergence of the sequence $(\alpha^{(m)})$ generated by NR iterations (23). However, since $\widehat{\alpha}$ is unknown, it is difficult to find a value other than $\alpha^{(0)} = 0$.

We therefore propose an algorithm (see Algorithm 1) starting from $\alpha^{(0)} = 0$. The MLE $\widehat{\alpha}$ is computed using NR iterations (23); afterwards, $\widehat{\beta}$ is computed from $\widehat{\alpha}$.

## 4. Ascent Property

The ascent property of Algorithm 1 (the fact that the profile log-likelihood is increased monotonically by the algorithm) is given by Theorem 7.

**Theorem 7.** *The sequence $(\alpha^{(m)})$ generated by Algorithm 1 increases monotonically the profile log-likelihood $\ell_p(\alpha)$, that is*

$$\ell_p\left(\alpha^{(m+1)}\right) \geq \ell_p\left(\alpha^{(m)}\right), m = 0, 1, \cdots \tag{31}$$

*Proof.* From (22) and (29), we deduce that, for all $\alpha > 0$, $\ell_p'(\alpha) \geq 0$ if $\alpha \leq \widehat{\alpha}$ and $\ell_p'(\alpha) \leq 0$ if $\alpha \geq \widehat{\alpha}$. Hence, the

```
Input: $\mathbf{x}$, $\mathbf{z}$ and $\varepsilon > 0$
Output: MLE $\widehat{\theta}$
1 Initialize $m = 0$ and $\alpha^{(0)} = 0$ ;
2 repeat
3   $\alpha^{(m+1)} = \alpha^{(m)} - F(\alpha^{(m)})/F'(\alpha^{(m)})$;
4   Set $m \longleftarrow m + 1$ ;
5 until $|F(\alpha^{(m)})| < \varepsilon$;
6 Set $\widehat{\alpha} = \alpha^{(m)}$ ;
7 For every $k = 1, \cdots, s$, compute $\widehat{\beta}_k$ as
      $\widehat{\beta}_k = (1/\sum_{j=1}^{r} x_{\bullet jk}/(1 + \widehat{\alpha} z_{jk}))(x_{\bullet 1k}/(1 + \widehat{\alpha} z_{1k}), \cdots, x_{\bullet rk}/(1 + \widehat{\alpha} z_{rk}))^{\mathrm{T}}$ ;
8 Set $\widehat{\theta} = (\widehat{\alpha}, \widehat{\beta}_1^{\mathrm{T}}, \cdots, \widehat{\beta}_s^{\mathrm{T}})^{\mathrm{T}}$.
```

Algorithm 1: PLBA for computing $\widehat{\theta}$.

function $\ell_p$ is increasing on the interval $]0, \widehat{\alpha}]$ and decreasing on $[\widehat{\alpha}, +\infty[$. From (30), we know that the sequence $(\alpha^{(m)})$ is increasing and still belongs to the interval $]0, \widehat{\alpha}]$. Then, for all iteration $m$, we have $\alpha^{(m)} \leq \alpha^{(m+1)}$ and $\ell_p(\alpha^{(m)}) \leq \ell_p(\alpha^{(m+1)})$ because $\ell_p$ is increasing on $]0, \widehat{\alpha}]$. The proof of Theorem 7 is then completed. □

## 5. Simulation Study

We compare the performance of our PLBA with that of Newton-Raphson (NR) and MM algorithms in R software [16]. We implemented the NR algorithm using the R package pracma [17] and the MM algorithm using Theorem 3.3 of [8]. The choice of these two comparison algorithms is motivated by the following factors: (a) MM and NR algorithms are two of the most used algorithms in statistics for parameter estimation; (b) other algorithms such as quasi-Newton and derivative-free algorithms showed very low convergence proportions and their results are not reported here; (c) the results obtained for the particular case $s = 1$ in [10] suggest that MM and NR algorithms are much more efficient than quasi-Newton and derivative-free algorithms.

*5.1. Data Generation Principle.* For given values of $s$, $r$, $n_1$, ..., $n_s$, the components of matrix $\mathbf{z}$ are randomly generated from a uniform distribution on [0.5,2.5]. Given the true parameter vector denoted $\theta^0 = (\alpha^0, (\beta_1^0)^{\mathrm{T}}, \cdots, (\beta_s^0)^{\mathrm{T}})^{\mathrm{T}}$, the true class probabilities $\pi_{ijk}(\theta^0|\mathbf{z})$ (see Equation (4)) are computed afterwards data matrix $\mathbf{x}$ (Equation (1)) is generated using the random generation function linked to the probability distribution function (3).

The true parameter vector $\theta^0 = (\alpha^0, (\beta_1^0)^{\mathrm{T}}, \cdots, (\beta_s^0)^{\mathrm{T}})^{\mathrm{T}}$ is presented under the following six scenarios:

Scenario 1: $s = 2$, $r = 3$,

$$\alpha^0 = 0.85, \beta_1^0 = (0.52,0.31,0.17)^{\mathrm{T}}, \beta_2^0 = (0.25,0.45,0.30)^{\mathrm{T}}. \quad (32)$$

Table 1: Number of parameters $(1 + sr)$ for the different scenarios.

| Scenario | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Number of parameters | 7 | 25 | 43 | 61 | 81 | 101 |

Scenario 2: $s = 8$, $r = 3$,

$$\begin{aligned} \alpha^0 = 0.85, \quad & \beta_k^0 = (0.50,0.15,0.35)^{\mathrm{T}}, k \in \{1, 3, 5\}, \\ & \beta_k^0 = (0.43,0.32,0.25)^{\mathrm{T}}, k \in \{2, 4, 7\}, \\ & \beta_k^0 = (0.35,0.35,0.30)^{\mathrm{T}}, k \in \{6, 8\}. \end{aligned} \quad (33)$$

Scenario 3: $s = 14$, $r = 3$,

$$\begin{aligned} \alpha^0 = 1.02, \quad & \beta_k^0 = (0.50,0.15,0.35)^{\mathrm{T}}, k \in \{1, 3, 5, 9, 11\}, \\ & \beta_k^0 = (0.43,0.32,0.25)^{\mathrm{T}}, k \in \{2, 4, 7, 12, 13\}, \\ & \beta_k^0 = (0.35,0.35,0.30)^{\mathrm{T}}, k \in \{6, 8, 10, 14\}. \end{aligned} \quad (34)$$

Scenario 4: $s = 15$, $r = 4$,

$$\begin{aligned} \alpha^0 = 1.02, \quad & \beta_k^0 = (0.40,0.10,0.30,0.20)^{\mathrm{T}}, k \in \{1, 3, 5, 9, 13, 14\}, \\ & \beta_k^0 = (0.45,0.10,0.25,0.20)^{\mathrm{T}}, k \in \{2, 4, 7, 12\}, \\ & \beta_k^0 = \left(\underbrace{0.25, \cdots, 0.25}_{4}\right)^{\mathrm{T}}, k \in \{6, 8, 10, 11, 15\}. \end{aligned} \quad (35)$$

Scenario 5: $s = 20$, $r = 4$,

$$\begin{aligned} \alpha^0 = 1.25, \quad & \beta_k^0 = (0.62,0.16,0.07,0.15)^{\mathrm{T}}, k \in \{1, 3, 5, 9, 11, 13, 15, 19\}, \\ & \beta_k^0 = (0.50,0.15,0.10,0.25)^{\mathrm{T}}, k \in \{2, 4, 7, 12, 14, 17\}, \end{aligned}$$

Table 2: Results for Scenario 1 (values in brackets are standard deviations).

| | | PLBA | MM | NR |
|---|---|---|---|---|
| | $\alpha$ | 0.864 (0.180) | 0.864 (0.180) | 0.878 (0.181) |
| | $\beta_{11}$ | 0.515 (0.071) | 0.515 (0.071) | 0.513 (0.071) |
| | $\beta_{21}$ | 0.310 (0.066) | 0.310 (0.066) | 0.309 (0.067) |
| | $\beta_{31}$ | 0.175 (0.054) | 0.175 (0.054) | 0.179 (0.054) |
| | $\beta_{12}$ | 0.248 (0.062) | 0.248 (0.062) | 0.249 (0.063) |
| | $\beta_{22}$ | 0.453 (0.068) | 0.453 (0.068) | 0.448 (0.068) |
| $n = 50$ | $\beta_{32}$ | 0.299 (0.064) | 0.299 (0.064) | 0.302 (0.064) |
| | Convergence proportion (%) | 100 | 100 | 54.3 |
| | Iterations | 5.7 (0.5) | 27.8 (5.6) | 7 (1.2) |
| | CPU time (secs) | 0.0005 | 0.0043 | 0.0029 |
| | Time ratio | 1 | 8 | 6 |
| | Log-likelihood | −191.95 | −191.95 | −191.95 |
| | MSE | $8.2e-03$ | $8.2e-03$ | $8.4e-03$ |
| | $\alpha$ | 0.851 (0.017) | 0.851 (0.017) | 0.852 (0.017) |
| | $\beta_{11}$ | 0.520 (0.007) | 0.520 (0.007) | 0.520 (0.007) |
| | $\beta_{21}$ | 0.310 (0.007) | 0.310 (0.007) | 0.310 (0.007) |
| | $\beta_{31}$ | 0.170 (0.005) | 0.170 (0.005) | 0.170 (0.005) |
| | $\beta_{12}$ | 0.250 (0.006) | 0.250 (0.006) | 0.250 (0.006) |
| | $\beta_{22}$ | 0.450 (0.007) | 0.450 (0.007) | 0.450 (0.007) |
| $n = 5000$ | $\beta_{32}$ | 0.300 (0.007) | 0.300 (0.007) | 0.300 (0.007) |
| | Convergence proportion (%) | 100 | 100 | 55.9 |
| | Iterations | 6 (0.1) | 36.3 (4.5) | 7 (1.3) |
| | CPU time (secs) | 0.0004 | 0.0056 | 0.0030 |
| | Time ratio | 1 | 14 | 7 |
| | Log-likelihood | −19380 | −19380 | −19380 |
| | MSE | $7.8e-05$ | $7.8e-05$ | $7.6e-05$ |

$$\beta_k^0 = \left(\underbrace{0.25, \cdots, 0.25}_{4}\right)^{\mathrm{T}}, k \in \{6, 8, 10, 16, 18, 20\}. \quad (36)$$

Scenario 6: $s = 20$, $r = 5$,

$$\alpha^0 = 1.25, \quad \beta_k^0 = (0.63, 0.16, 0.06, 0.05, 0.10)^{\mathrm{T}}, k \in \{1, 3, 5, 9, 11, 13, 15, 19\},$$
$$\beta_k^0 = (0.32, 0.18, 0.25, 0.10, 0.15)^{\mathrm{T}}, k \in \{2, 4, 7, 12, 14, 17\},$$
$$\beta_k^0 = \left(\underbrace{0.20, \cdots, 0.20}_{5}\right)^{\mathrm{T}}, k \in \{6, 8, 10, 16, 18, 20\}. \quad (37)$$

For these different scenarios, the number of parameters $(1 + sr)$ is given by Table 1.

For the $n_k$'s $(k = 1, \cdots, s)$, we have chosen two common values: a low value $(n = 50)$ and a great value $(n = 5000)$. Except for the proposed algorithm whose starting guess is automated, the other algorithms were given randomly generated starting guesses.

*5.2. Results.* Tables 2–7 present the average results obtained for the different scenarios over 1000 replications (i.e., 1000 repetitions of the data generation and computation of $\widehat{\theta}$). In these tables, CPU times are given in seconds and CPU time ratios are calculated as the ratio of the mean CPU time of a given algorithm to the CPU time of the PLBA. Thus, the CPU time ratio of the PLBA is always equal to 1. The Mean Square Error (MSE) is defined as

$$\mathrm{MSE}\left(\widehat{\theta}, \theta^0\right) = \frac{1}{1 + sr}\left(\left(\widehat{\alpha} - \alpha^0\right)^2 + \sum_{k=1}^{s}\sum_{j=1}^{r}\left(\widehat{\beta}_{jk} - \beta_{jk}^0\right)^2\right). \quad (38)$$

Due to the increasing number of parameters, the MLE $\widehat{\theta}$ has only been included for Scenario 1 (Table 2).

From these tables, it can be seen that PLBA and MM algorithms have always converged while the convergence proportion of NR algorithm decreases (from 55.9% to 19.6%) with the number of parameters (see Figure 1). As

Table 3: Results for Scenario 2 (values in brackets are standard deviations).

| | | PLBA | MM | NR |
|---|---|---|---|---|
| $n = 50$ | Convergence proportion (%) | 100 | 100 | 35.5 |
| | Iterations | 6 (0.1) | 29.9 (3) | 7.7 (1.3) |
| | CPU time (secs) | 0.0005 | 0.0087 | 0.0068 |
| | Time ratio | 1 | 18 | 14 |
| | Log-likelihood | −773.57 | −773.57 | −773.57 |
| | MSE | $4.4e-03$ | $4.4e-03$ | $4.4e-03$ |
| $n = 5000$ | Convergence proportion (%) | 100 | 100 | 38.7 |
| | Iterations | 6 (0) | 38.6 (2.6) | 7.7 (1.4) |
| | CPU time (secs) | 0.0005 | 0.0108 | 0.0069 |
| | Time ratio | 1 | 21 | 14 |
| | Log-likelihood | −78015.32 | −78015.32 | −78015.32 |
| | MSE | $4.5e-05$ | $4.5e-05$ | $4.5e-05$ |

Table 4: Results for Scenario 3 (values in brackets are standard deviations).

| | | PLBA | MM | NR |
|---|---|---|---|---|
| $n = 50$ | Convergence proportion (%) | 100 | 100 | 35.5 |
| | Iterations | 6 (0) | 35.2 (3) | 8.2 (1.6) |
| | CPU time (secs) | 0.0005 | 0.0134 | 0.0113 |
| | Time ratio | 1 | 25 | 21 |
| | Log-likelihood | −1356.98 | −1356.98 | −1356.98 |
| | MSE | $4.4e-03$ | $4.4e-03$ | $4.5e-03$ |
| $n = 5000$ | Convergence proportion (%) | 100 | 100 | 37.1 |
| | Iterations | 6 (0) | 45.4 (2.5) | 8.2 (1.6) |
| | CPU time (secs) | 0.0005 | 0.0160 | 0.0106 |
| | Time ratio | 1 | 35 | 23 |
| | Log-likelihood | −137298.2 | −137298.2 | −137298.2 |
| | MSE | $4.5e-05$ | $4.5e-05$ | $4.6e-05$ |

Table 5: Results for Scenario 4 (values in brackets are standard deviations).

| | | PLBA | MM | NR |
|---|---|---|---|---|
| $n = 50$ | Convergence proportion (%) | 100 | 100 | 29.6 |
| | Iterations | 6 (0) | 34.9 (2.9) | 8.3 (1.3) |
| | CPU time (secs) | 0.0006 | 0.0135 | 0.0149 |
| | Time ratio | 1 | 24 | 26 |
| | Log-likelihood | −1647.49 | −1647.49 | −1647.49 |
| | MSE | $3.6e-03$ | $3.6e-03$ | $3.6e-03$ |
| $n = 5000$ | Convergence proportion (%) | 100 | 100 | 29.4 |
| | Iterations | 6 (0) | 45.4 (2.6) | 8.3 (1.4) |
| | CPU time (secs) | 0.0005 | 0.0166 | 0.0142 |
| | Time ratio | 1 | 31 | 27 |
| | Log-likelihood | −166241.8 | −166241.8 | −166241.8 |
| | MSE | $3.8e-05$ | $3.8e-05$ | $3.7e-05$ |

Table 6: Results for Scenario 5 (Values in brackets are standard deviations).

| | | PLBA | MM | NR |
|---|---|---|---|---|
| $n = 50$ | Convergence proportion (%) | 100 | 100 | 25.9 |
| | Iterations | 6 (0) | 42.2 (3.2) | 8.7 (1.8) |
| | CPU time (secs) | 0.0005 | 0.0181 | 0.0213 |
| | Time ratio | 1 | 36 | 42 |
| | Log-likelihood | −2102.95 | −2102.95 | −2102.95 |
| | MSE | $3.2e-03$ | $3.2e-03$ | $3.3e-03$ |
| $n = 5000$ | Convergence proportion (%) | 100 | 100 | 26.3 |
| | Iterations | 6.6 (0.5) | 56.2 (2.9) | 8.5 (1.6) |
| | CPU time (secs) | 0.0006 | 0.0234 | 0.0196 |
| | Time ratio | 1 | 37 | 31 |
| | Log-likelihood | −210936.75 | −210936.75 | −210936.75 |
| | MSE | $3.5e-05$ | $3.5e-05$ | $3.5e-05$ |

Table 7: Results for Scenario 6 (values in brackets are standard deviations).

| | | PLBA | MM | NR |
|---|---|---|---|---|
| $n = 50$ | Convergence proportion (%) | 100 | 100 | 19.6 |
| | Iterations | 6 (0) | 41.1 (3.1) | 8.8 (1.4) |
| | CPU time (secs) | 0.0004 | 0.0177 | 0.0307 |
| | Time ratio | 1 | 43 | 74 |
| | Log-likelihood | −2308.19 | −2308.19 | −2308.19 |
| | MSE | $2.8e-03$ | $2.8e-03$ | $2.8e-03$ |
| $n = 5000$ | Convergence proportion (%) | 100 | 100 | 21.9 |
| | Iterations | 6.6 (0.5) | 56.4 (3.3) | 8.7 (1.6) |
| | CPU time (secs) | 0.0005 | 0.0227 | 0.0278 |
| | Time ratio | 1 | 50 | 61 |
| | Log-likelihood | −230371.92 | −230371.92 | −230371.92 |
| | MSE | $3e-05$ | $3e-05$ | $3e-05$ |

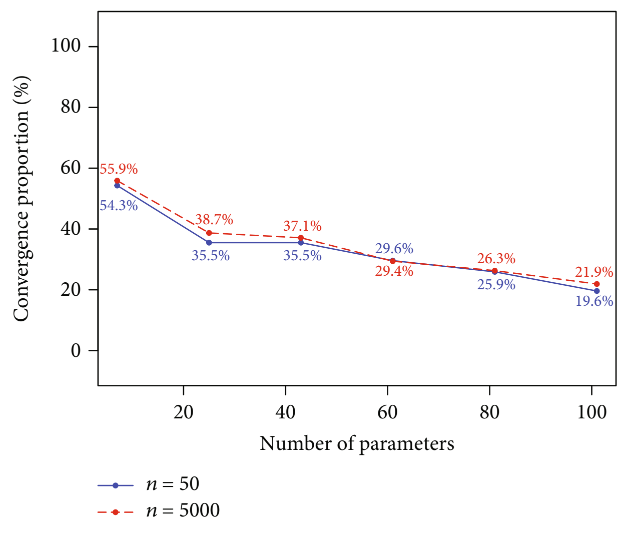

Figure 1: Convergence proportions for NR.

far as the MSE is concerned, the trend for all the algorithms is that the MSE decreases when the sample size $n$ increases.

By taking a look at the CPU times ratios, we notice that the CPU time ratios of the MM and NR algorithms are all well above 1. This means that the PLBA is significantly faster than these two algorithms. As shown on Figures 2 and 3, the CPU time ratios of MM and NR algorithms increase with the number of parameters. The PLBA is on average 8 to 50 times faster than MM and 6 to 74 times quicker than NR.

*5.3. Analysis of the Results.* It appears that our proposed PLBA outperforms the Minorization-Maximization (MM) and the full Newton-Raphson (NR) algorithms. It always converges; it requires a low number of iterations and little computation time. The number of iterations varies slightly and seems to stabilize around six iterations whatever the starting guess and the number of parameters to be estimated. The MM algorithm also has a convergence rate of 100%, but its number of iterations is quite high, and this may indicate some sensitivity to the starting guess. The NR algorithm has

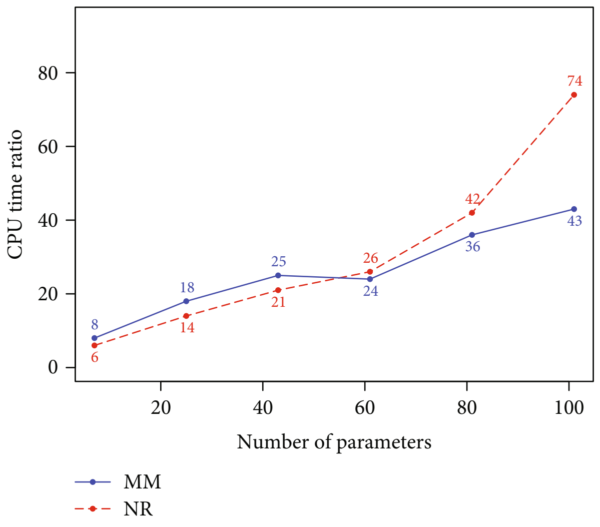

Figure 2: CPU time ratios for NR and MM algorithms ($n = 50$).

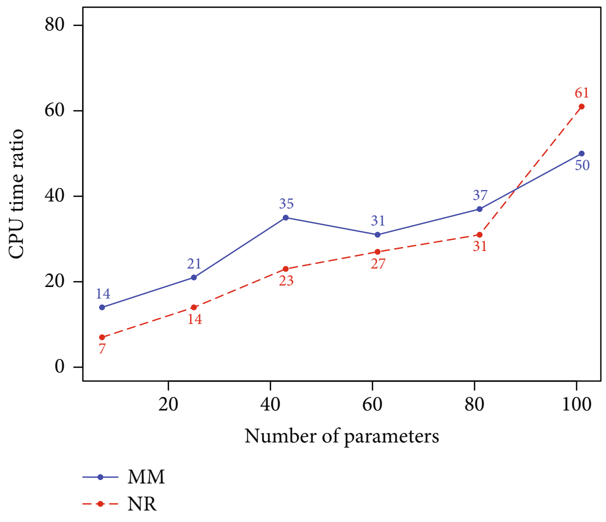

Figure 3: CPU time ratios for NR and MM algorithms ($n = 5000$).

a convergence proportion at most slightly higher than 50% and this proportion decreases when the number of parameters increases. This is not surprising because at each iteration of the NR algorithm, a square matrix of order $1 + sr$ is numerically inverted. This numerical inversion can be complicated or impossible when the matrix to be inverted is ill-conditioned or singular. The fact that NR does not converge for all the replications is also not surprising since it is well known that NR may converge to bad values or may not converge at all if the starting guess is far from the true parameter vector.

The good results obtained by our proposed algorithm PLBA can be explained by several factors. First, the principle of profile likelihood enables to reduce the search for $1 + sr$ solutions to that of the single solution $\widehat{\alpha}$ from which the remaining $sr$ estimates $\widehat{\beta}_{jk}$ $(j = 1, \cdots, r,\ k = 1, \cdots, s)$ can be obtained by a simple formula. This also enables to reduce the computation time required by our proposed PLBA to converge. Secondly, the fact that the estimation of $\alpha$ relies on a one-dimensional NR enables our algorithm to enjoy the quadratic convergence of the NR algorithm. Thirdly, the definition of the automated starting guess ensures the convergence of our proposed algorithm PLBA, and the ascension property ensures its numerical stability.

## 6. Conclusion

In this article, we have built a profile-likelihood-based algorithm (PLBA) to compute, under constraints, the maximum likelihood estimate (MLE) of the parameter vector of a statistical model used in the analysis of a road safety measure applied to $s$ sites presenting in total $r$ accident severity levels. The parameter vector of the model is of the form $\theta = (\alpha, \beta^{\mathrm{T}})^{\mathrm{T}}$, where $\alpha$ is the parameter of interest and $\beta$ is a vector of $sr$ secondary parameters. Using the likelihood equations, we obtained the closed-form expression of the components of $\beta$ as a function of the main parameter $\alpha$ and then used the principle of profile likelihood to express the log-likelihood only as a function of $\alpha$. We then built an algorithm mixing a one-dimensional Newton method to compute the estimate of the parameter of interest $\alpha$ and the computation of the secondary parameters from their closed-form expressions. The starting guess of our proposed algorithm is automated in such a way as to guarantee its convergence towards the MLE $\widehat{\theta}$. The numerical studies suggest that our PLBA outperforms the Minorization-Maximization (MM) and the full Newton-Raphson (NR) algorithms in terms of computation time. Our PLBA converges to estimates close to the true values of the parameter vector even for small sample sizes. They also suggest that the problem addressed in this paper is difficult to tackle for the full NR algorithm (the latter having a convergence proportion at most slightly higher than 50%).

## Data Availability

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

## Conflicts of Interest

The author declares that he/she has 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] K. Lange, *Optimization*, Springer Science & Business Media, New York, 2nd edition, 2013.

[3] K. Lange, E. C. Chi, and H. Zhou, “A brief survey of modern optimization for statisticians,” *International Statistical Review*, vol. 82, no. 1, pp. 46–70, 2014.

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

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

[6] 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.

[7] 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.

[8] A. Mkhadri, A. N'Guessan, and B. Hafidi, "An MM algorithm for constrained estimation in a road safety measure modeling," *Communications in Statistics – Simulation and Computation*, vol. 39, no. 5, pp. 1057–1071, 2010.

[9] J. F. Monahan, *Numerical Methods of Statistics*, Cambridge University Press, New York, 2nd edition, 2011.

[10] A. N'Guessan and I. C. Geraldo, "A cyclic algorithm for maximum likelihood estimation using Schur complement," *Numerical Linear Algebra with Applications*, vol. 22, no. 6, pp. 1161–1179, 2015.

[11] S. A. Murphy and A. W. Van der Vaart, "On profile likelihood," *Journal of the American Statistical Association*, vol. 95, no. 450, pp. 449–465, 2000.

[12] A. N'Guessan, "Analytical existence of solutions to a system of nonlinear equations with application," *Journal of Computational and Applied Mathematics*, vol. 234, no. 1, pp. 297–304, 2010.

[13] A. N'Guessan and M. Truffier, "Impact d'un aménagement de sécurité routière sur la gravité des accidents de la route," *Journal de la Société Française de Statistique*, vol. 49, pp. 23–41, 2008.

[14] J. F. Epperson, *An Introduction to Numerical Methods and Analysis*, John Wiley & Sons, Hoboken (USA), 3rd edition, 2021.

[15] R. K. Gupta, *Numerical Methods: Fundamentals and Applications*, Cambridge University Press, New Delhi, 2019.

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

[17] H. W. Borchers, *pracma: Practical Numerical Math Functions*, 2022, https://CRAN.R-project.org/package=pracma. R package version 2.3.8.