STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **16**, August 2026, pp 1640–1663.
Published online in International Academic Press (www.IAPress.org)

Estimation of the Dirichlet–multinomial distribution parameter using Schur Complement based Newton–Raphson method for modeling a road safety measure

Aboubacari Abdou Amadou [1], Assi N'guessan [2], Ibrahim Sidi Zakari [1,*], Bisso Saley [1]

[1] *Department of Mathematics and Computer Science, Abdou Moumouni University, 10896, Niamey, Niger*
abdouaboubacar27@gmail.com, sidizakariibrahim@gmail.com, bsaley@yahoo.fr
[2] *Paul Painlevé Laboratory (UMR CNRS 8524), University of Lille, 59655 Villeneuve d'Ascq CEDEX, France*
assi.n-guessan@univ-lille.fr

Abstract This paper investigates parameter estimation for the one-observation Dirichlet-–Multinomial (DM) model using several optimization methods, with particular emphasis on the Newton-Raphson (NR) procedure based on the Schur complement (NR-Schur) and on different initialization strategies. The simulation results show that the NR-Schur and NR methods, which use numerical inversion via the solve function (NR-Solve), produce identical estimates, while the NR-Schur method is consistently faster due to more efficient inversion of the Hessian matrix. The study also highlights the importance of the choice of the initial distribution parameter values, with the uniform initialization generally providing the best performance in terms of bias, mean squared error, and convergence speed. Compared with the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm, and Nelder–Mead (NM) methods, NR-Schur achieves the best balance between statistical accuracy, numerical stability, and computational cost. The real-data analysis confirms the simulation findings and demonstrates the robustness of the proposed method. Overall, the results obtained confirm the suitability of the DM model for modeling road accident data. They also demonstrate that combining the NR-Schur algorithm with a uniform initialization strategy provides a reliable, robust, and computationally efficient estimation procedure for the one-observation DM model.

Keywords Dirichlet–multinomial distribution, Overdispersion parameter, Newton-Raphson method, Maximum likelihood estimation, Schur complement, Statistical modeling of road safety

DOI: 10.19139/soic-2310-5070-3457

1. Introduction

The Dirichlet distribution is widely used for modeling compositional data and proportion vectors in a variety of application domains. For example, [7] employed the Dirichlet distribution to model compositional data arising from serum protein measurements in white Pekin ducklings. Within a Bayesian framework, [17] adopted a Dirichlet prior distribution to model the frequencies of different congenital heart diseases. More recently, the Dirichlet–multinomial distribution has attracted considerable attention for the analysis of multivariate categorical data exhibiting greater variability than can be explained by the classical multinomial model [2, 3].

In the field of road safety, several multinomial models have been developed to evaluate the average effects of safety measures and to estimate the risks associated with different types of road accidents. For instance, [10] proposed a multinomial model to estimate the average effect of a road safety measure implemented across multiple experimental sites. Subsequently, [12, 13] introduced a conditional multinomial model designed for a single experimental site, in which the probability vector associated with the different accident categories is assumed to

*Correspondence to: Ibrahim Sidi Zakari (Email: ibrahim.sidi@uam.edu.ne, sidizakariibrahim@gmail.com). Department of Mathematics and Computer Science, Abdou Moumouni University, 10896, Niamey, Niger.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

be fixed. Under this framework, the likelihood function corresponds to the multinomial distribution, and parameter estimation is performed through an iterative procedure known as the cyclic algorithm. This approach is based on transforming the likelihood equations into a linear system that is solved using the Schur complement [21].

Despite their usefulness, multinomial models suffer from certain limitations when applied to real-world data. In particular, they assume that the observed variability is fully explained by the multinomial distribution, an assumption that is often violated in the presence of overdispersion. This issue was highlighted in the context of allelic count data [19]. To accommodate this additional variability, several extensions of the multinomial model have been proposed, among which the Dirichlet–multinomial distribution has emerged as one of the most important. By introducing randomness into the category probability vector, the Dirichlet–multinomial distribution naturally accounts for overdispersion. It has been successfully applied in a wide range of fields, including the analysis of pollen grain frequencies [2] and, more recently, Bayesian Dirichlet–multinomial regression for single-cell RNA sequencing data [22].

Parameter estimation for the Dirichlet–multinomial distribution is generally performed using maximum likelihood methods and relies on numerical optimization algorithms [18] such as Newton-Raphson, Fisher scoring [20], Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm, and the Nelder–Mead (NM) simplex method. Several of these approaches have been implemented in statistical software. For instance, the R package *DirichletMultinomial* implements Dirichlet–multinomial mixture models for microbial metagenomic data based on the methodology proposed by [4]. Another example is the R package *dirmult* developed by [19]; which employs the Newton-Raphson algorithm for maximum likelihood estimation of the Dirichlet–multinomial parameters. In this implementation, the initial value of the overdispersion parameter is obtained through a moment-based estimator derived from the sample variance–covariance matrix, and the resulting estimate is then used to initialize the Newton-Raphson procedure.

Despite their widespread use, these methods face several practical difficulties. First, their performance is often highly sensitive to the choice of starting values, which may affect both convergence and estimation accuracy. Second, Newton-Raphson based algorithms require the inversion of the Hessian matrix at each iteration, leading to substantial computational costs and potential numerical instability, particularly in high-dimensional settings. Furthermore, existing implementations are generally designed for data organized as an observations matrix, where rows represent observations and columns correspond to categories. Consequently, these approaches become less suitable when only a single observation of the count vector is available, since the information required for conventional initialization procedures and matrix-based estimation strategies may no longer be accessible. These limitations motivate the development of alternative estimation procedures that are computationally efficient, numerically stable, and applicable to the one-observation Dirichlet–multinomial setting.

In this paper, we propose a Dirichlet–multinomial model that generalizes the conditional multinomial model developed by [12] for the analysis of road accident data at a single experimental site. Unlike the conditional multinomial framework, our approach treats the probability vector as a random variable following a Dirichlet distribution. This formulation directly yields the marginal distribution of the data and allows potential overdispersion to be incorporated naturally into the model. We further develop a parameter estimation procedure based on the Newton-Raphson algorithm, in which the inverse of the Hessian matrix is derived explicitly using the Schur complement. Finally, we propose a method for constructing the initial parameter vector, inspired by the work of [7], which relies on estimating the overdispersion parameter and remains applicable even when only a single observation is available.

The main contributions of this paper are as follows:

- the generalization of the conditional multinomial model through the introduction of a Dirichlet–multinomial model specifically designed for the analysis of road accident data at a single experimental site;
- the development of a Newton-Raphson estimation procedure in which the inverse of the Hessian matrix is obtained explicitly using the Schur complement, thereby improving numerical stability;
- the proposition of an initialization strategy that is particularly suitable when only limited data are available.

The remainder of the paper is organized as follows. Section 2 introduces the Dirichlet–multinomial model for road accident data. Section 3 presents the estimation procedure and describes the use of the Schur complement for Hessian matrix inversion. Section 4 reports simulation results and compares the performance of the proposed method with several classical optimization algorithms. Section 5 illustrates the application of the proposed methodology to real-world data. Finally, Section 6 concludes the paper and outlines directions for future research.

2. Modeling road accident data using the Dirichlet–multinomial distribution

We consider a hazardous area where a road safety measure has been implemented. It is also assumed that information on the number of different types of accidents, both before and after the intervention, has been recorded. To model the road accident data and estimate the model parameter vector, we use the Dirichlet–multinomial distribution, which is an extension of the multinomial model proposed by [12]. On a treated site, these authors assumed that X, a random vector containing accident data, follows a multinomial distribution with parameter (n, P), where n is the total number of accidents during both periods, and P is a probability vector associated with X.

Following the notation used in [10, 11, 12, 13, 14], let R $(R > 1)$ be the number of accident types, and let x be a vector containing the number of accidents for each type before and after the implementation of the road safety measure. Let x_{1j} (resp. x_{2j}) denote the number of accidents of type j $(j = 1, \cdots, R)$ before (resp. after) the intervention, and let $n = \sum_{j=1}^{R}(x_{1j} + x_{2j})$ be the total number of accidents.

We now denote:

- $X = (X_{11}, X_{12}, \cdots, X_{1R}, X_{21}, X_{22}, \cdots, X_{2R})$ is the random vector modeling the number of accidents of each type before and after the intervention.
- $P = (P_{11}, P_{12}, \cdots, P_{1R}, P_{21}, P_{22}, \cdots, P_{2R})$ is a random vector containing the probabilities associated with X. For example, the component P_{1j} (resp. P_{2j}) represents the probability that an accident occurring in the hazardous zone is of type j $(j = 1, \cdots, R)$ before (resp. after) the intervention.
  We assume that the random probability vector P follows a Dirichlet distribution with parameter a (see [2]),

$$P \sim \mathrm{Dir}(a) \tag{1}$$

  where a is a vector in $\mathbb{R}^{2R}$ such that $a_j > 0$ for $j = 1, \cdots, 2R$.

- Using the assumptions of [10], the conditional distribution of the random vector X given P follows a multinomial distribution with parameters n and P:

$$X|P \sim M(n, P) \tag{2}$$

- Thus, using the densities of P and $X|P$, we obtain the marginal density of X:

$$f_X(x) = \int_{S_C(1)} f(x|p) \times f(p)\, dp = \binom{n}{x} \frac{\mathcal{B}(a + x)}{\mathcal{B}(a)} \tag{3}$$

where $S_C(1) = \left\{(p_1, p_2, \cdots, p_d) \in \mathbb{R}^d; p_i > 0; \sum_{i=1}^{d} p_i = 1\right\}$ is the closed simplex and $\mathcal{B}$ is the multivariate Beta function defined by:

$$\mathcal{B}(a) = \frac{\prod_{i=1}^{2R} \Gamma(a_i)}{\Gamma\left(\sum_{i=1}^{2R} a_i\right)} \tag{4}$$

and Γ is the gamma function defined as:

$$\Gamma(s) = \int_0^{+\infty} t^{s-1} e^{-t} dt \tag{5}$$

The function given in (3) is called the Dirichlet–multinomial distribution, and we say that the random vector X follows a Dirichlet–multinomial distribution with parameter (n, a):

$$X \sim \text{DMultinomial}_{2R}(n, a) \tag{6}$$

One of the objectives of this work is to use the Newton-Raphson method to estimate the parameter of the density function defined by equation (3).

3. Estimation of the Dirichlet–multinomial distribution parameter using the Newton-Raphson method

Given x, a vector containing the accident data before and after the intervention, and considering notation (3), the probability function associated with the random vector X is defined as:

$$\begin{aligned} L(a, x) &= \binom{n}{x} \frac{\mathcal{B}(a + x)}{\mathcal{B}(a)} \\ &= \binom{n}{x} \frac{\prod_{j=1}^{R} \Gamma(x_{1j} + a_j) \times \Gamma(x_{2j} + a_{R+j})}{\Gamma\left(\sum_{j=1}^{R} x_{1j} + x_{2j} + a_j + a_{R+j}\right)} \times \frac{\Gamma\left(\sum_{j=1}^{2R} a_j\right)}{\prod_{j=1}^{2R} \Gamma(a_j)} \\ &= \binom{n}{x} \frac{\prod_{j=1}^{R} \Gamma(x_{1j} + a_j) \times \Gamma(x_{2j} + a_{R+j})}{\Gamma(n + a_+)} \times \frac{\Gamma(a_+)}{\prod_{j=1}^{2R} \Gamma(a_j)} \\ &= \binom{n}{x} \frac{\Gamma(a_+)}{\Gamma(n + a_+)} \prod_{j=1}^{R} \frac{\Gamma(x_{1j} + a_j)}{\Gamma(a_j)} \times \frac{\Gamma(x_{2j} + a_{R+j})}{\Gamma(a_{R+j})} \end{aligned} \tag{7}$$

where a is a parameter vector in $\mathbb{R}^{2R}$ such that $a_j > 0$ for $j = 1, \cdots, 2R$; $a_+ = \sum_{j=1}^{2R} a_j$, $n = \sum_{j=1}^{R} (x_{1j} + x_{2j})$, and Γ is the gamma function defined by equation (5).

The problem consists in finding an estimator $\hat{a}$, if it exists, such that:

$$\hat{a} = \arg \max_{a \in \mathbb{R}^{2R}} L(a, x) \tag{8}$$

We equivalently use the logarithm of this function to determine the estimator of the parameter a. Thus, problem (8) is equivalent to:

$$\hat{a} = \arg \max_{a \in \mathbb{R}^{2R}} \ell(a, x) \tag{9}$$

where

$$\begin{aligned} \ell(a,x) &= \log\left[L(a,x)\right] \\ &= \text{const} + \log\Gamma(a_+) - \log\Gamma(n+a_+) + \sum_{j=1}^{R} \log\Gamma(x_{1j}+a_j) \\ &+ \sum_{j=1}^{R} \log\Gamma(x_{2j}+a_{R+j}) - \sum_{j=1}^{R}\Big[\log\Gamma(a_j) + \log\Gamma(a_{R+j})\Big] \end{aligned} \tag{10}$$

with $\text{const} = \log\left[\binom{n}{x}\right]$ and $a_+ = \sum_{j=1}^{2R} a_j$.

Let us denote: $y = (x_{11}, x_{12}, \cdots, x_{1R}, x_{21}, x_{22}, \cdots, x_{2R})$ such that:

$$y_1 = x_{11},\ y_2 = x_{12},\ \cdots,\ y_R = x_{1R},\ y_{R+1} = x_{21},\ y_{R+2} = x_{22},\ \cdots,\ y_{2R} = x_{2R}$$

Then the log-likelihood function ℓ becomes:

$$\begin{aligned} \ell(a,y) &= \text{const} + \log\Gamma(a_+) - \log\Gamma(n+a_+) \\ &+ \left[\sum_{j=1}^{2R} \log\Gamma(y_j+a_j) - \log\Gamma(a_j)\right] \end{aligned} \tag{11}$$

Remark 3.1. Since the components of the parameter vector a are strictly positive, i.e., $a_j > 0$ for $j = 1, \cdots, 2R$, and the gamma function Γ is strictly positive, the log-likelihood function $\ell(a, y) = \log[L(a, y)]$ is well defined.

To solve problem (9), we use the Newton-Raphson algorithm, which is based on the following iterative scheme:

$$a^{(k+1)} = a^{(k)} - \left[H\left(\ell(a^{(k)}, y)\right)\right]^{-1} \times \nabla_{\hat{a}} \ell(a^{(k)}, y) \tag{12}$$

where $a^{(k+1)}$ is the estimate of the parameter vector a at iteration $k+1$, $\nabla_{\hat{a}} \ell(a^{(k)}, y)$ is the gradient of ℓ, and $H(\ell(a^{(k)}, y))$ is the Hessian matrix of ℓ.

One of the main issues with the Newton-Raphson method lies in ensuring that the matrix $H(\ell(a^{(k)}, y))$ is invertible at each iteration. To address this, we will use the Schur complement to demonstrate that the Hessian matrix $H(\ell(a^{(k)}, y))$ is formally invertible.

To implement the iterative scheme defined in (12), we first compute the gradient $\nabla_{\hat{a}} \ell(a^{(k)}, y)$ and the Hessian matrix $H(\ell(a^{(k)}, y))$.

3.1. Gradient of the log-likelihood function ℓ

The function $\ell(a,y)$ given by (11) is continuous and infinitely differentiable since it is composed of two infinitely differentiable functions (the logarithm and the Gamma function Γ). Therefore, the gradient of ℓ is well-defined:

$$\nabla\ell(a,y) = \left(\frac{\partial\ell(a,y)}{\partial a_1}, \frac{\partial\ell(a,y)}{\partial a_2}, \cdots, \frac{\partial\ell(a,y)}{\partial a_{2R}}\right)^T. \tag{13}$$

Thus, for all $j = 1, \cdots, 2R$, we have

$$\frac{\partial\ell(a,y)}{\partial a_j} = \frac{\Gamma'(y_j+a_j)}{\Gamma(y_j+a_j)} - \frac{\Gamma'(a_j)}{\Gamma(a_j)} + \frac{\Gamma'(a_+)}{\Gamma(a_+)} - \frac{\Gamma'(n+a_+)}{\Gamma(n+a_+)}. \tag{14}$$

By setting $\phi(a) = \frac{\Gamma'(a)}{\Gamma(a)}$, for all $j = 1, \cdots, 2R$ we have:

$$\frac{\partial\ell(a,y)}{\partial a_j} = \phi(y_j+a_j) - \phi(a_j) + \phi(a_+) - \phi(n+a_+). \tag{15}$$

Thus, the gradient $\nabla\ell(a,y)$ is given by:

$$\nabla\ell(a,y) = \begin{pmatrix} \phi(y_1+a_1) - \phi(a_1) + \phi(a_+) - \phi(n+a_+) \\ \phi(y_2+a_2) - \phi(a_2) + \phi(a_+) - \phi(n+a_+) \\ \vdots \\ \phi(y_{2R}+a_{2R}) - \phi(a_{2R}) + \phi(a_+) - \phi(n+a_+) \end{pmatrix}.$$

3.2. Determination of the associated Hessian matrix

The function ℓ being infinitely differentiable, the Hessian matrix associated to ℓ is well-defined:

$$H(\ell(a,y)) = \begin{bmatrix} \frac{\partial^2\ell(a,y)}{\partial a_1^2} & \frac{\partial^2\ell(a,y)}{\partial a_1\partial a_2} & \cdots & \frac{\partial^2\ell(a,y)}{\partial a_1\partial a_{2R}} \\ \frac{\partial^2\ell(a,y)}{\partial a_2\partial a_1} & \frac{\partial^2\ell(a,y)}{\partial a_2^2} & \cdots & \frac{\partial^2\ell(a,y)}{\partial a_2\partial a_{2R}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2\ell(a,y)}{\partial a_{2R}\partial a_1} & \frac{\partial^2\ell(a,y)}{\partial a_{2R}\partial a_2} & \cdots & \frac{\partial^2\ell(a,y)}{\partial a_{2R}^2} \end{bmatrix} \tag{16}$$

Let us determine the second derivatives $\dfrac{\partial^2\ell(a)}{\partial a_j\partial a_k}$.

For $j = k$, we have:

$$\frac{\partial^2\ell(a)}{\partial a_j^2} = \Psi(y_j+a_j) - \Psi(a_j) + \Psi(a_+) - \Psi(n+a_+), \tag{17}$$

and for $j \neq k$:

$$\frac{\partial^2\ell(a)}{\partial a_j\partial a_k} = \Psi(a_+) - \Psi(n+a_+), \tag{18}$$

Where $\Psi(s) = (\log\Gamma)''(s)$.

Proposition 3.2
Let's consider $a \in \mathbb{R}^{2R}$, $R > 0$, so that $a_+ = \sum_{j=1}^{2R}$. For all indices j and k, we have:

$$\frac{\partial \Gamma(a_+)}{\partial a_j} = \frac{\partial \Gamma(a_+)}{\partial a_k}. \tag{19}$$

Proof

Indeed, the Gamma function Γ is continuous and infinitely differentiable on $\mathbb{R}_+^*$, and its p-th derivative for any $s > 0$ is given by:

$$\Gamma^{(p)}(s) = \int_0^{+\infty} t^{s-1} e^{-t} (\log t)^p \, dt. \tag{20}$$

Since for all $j = 1, \ldots, 2R$, $a_j > 0$ and $a_+ = \sum_j^{2R} a_j > 0$, we have:

$$\begin{aligned}
\frac{\partial \Gamma(a_+)}{\partial a_j} &= \frac{\partial}{\partial a_j} \int_0^{+\infty} t^{a_+ - 1} e^{-t} dt \\
&= \int_0^{+\infty} \frac{\partial}{\partial a_j} \left(t^{a_+ - 1} e^{-t} \right) dt \\
&= \int_0^{+\infty} t^{a_+ - 1} e^{-t} \log(t) \, dt,
\end{aligned}$$

and similarly,

$$\frac{\partial \Gamma(a_+)}{\partial a_k} = \int_0^{+\infty} t^{a_+ - 1} e^{-t} \log(t) \, dt.$$

Hence, for all indices j and k:

$$\frac{\partial \Gamma(a_+)}{\partial a_j} = \frac{\partial \Gamma(a_+)}{\partial a_k}.$$

Proposition 3.3
Let's consider $a \in \mathbb{R}^{2R}$, $R > 0$, so that $a_+ = \sum_{j=1}^{2R}$. For all indices j and k, we have:

$$\frac{\partial^2 \Gamma(a_+)}{\partial a_j^2} = \frac{\partial^2 \Gamma(a_+)}{\partial a_j \partial a_k}. \tag{21}$$

Proof

Indeed, we have:

$$\frac{\partial^2 \Gamma(a_+)}{\partial a_j^2} = \int_0^{+\infty} t^{a_+ - 1} e^{-t} (\log t)^2 \, dt, \tag{22}$$

and similarly,

$$\frac{\partial^2 \Gamma(a_+)}{\partial a_j \partial a_k} = \int_0^{+\infty} t^{a_+ - 1} e^{-t} (\log t)^2 \, dt. \tag{23}$$

Thus, for all indices j and k:

$$\Gamma''_{jj}(a_+) = \Gamma''_{jk}(a_+).$$

Proposition 3.4
For any real $s > 0$, the second derivative of log-gamma function $(\log \Gamma)''$ is strictly positive and decreasing on $(0, +\infty)$.

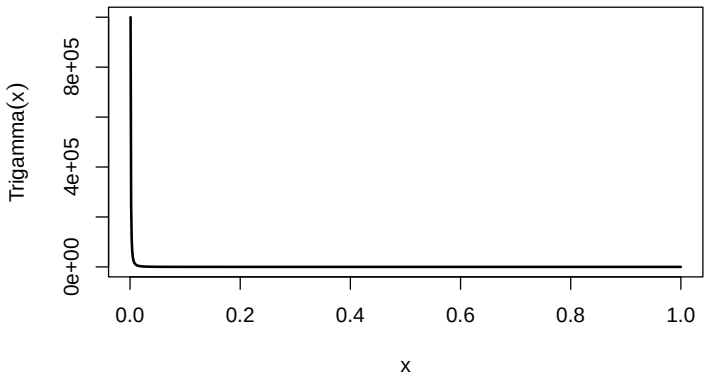

Figure 1. Graphical representation of the second derivative of log-gamma.

Proof

A representation of the second derivative of log-gamma function (also known as trigamma function) is given in figure (1). This confirms a well-known property of the trigamma function which is strictly positive and strictly decreasing on $(0, +\infty)$. In other words, for any $s > 0$,

$$(\log \Gamma)''(s) > 0,$$

and for any $0 < s < t$,

$$(\log \Gamma)''(s) > (\log \Gamma)''(t).$$

The following lemma provides a decomposition of the Hessian matrix associated to the function $\ell(a)$ (16), which is essential for the application of the Schur complement.

Lemma 3.5
The Hessian matrix associated to the function $\ell(a)$ (16) can be written as:

$$H(a, y) = \alpha(a) \times U + D_a,$$

where $\alpha(a) = \dfrac{\partial^2 \ell(a)}{\partial a_1 \partial a_2}$, U is a $2R \times 2R$ matrix with all entries equal to 1, and D_a is a diagonal $2R \times 2R$ matrix with entries $D_{jj} = \Psi(y_j + a_j) - \Psi(a_j)$.

Proof

Let the matrix $H(a, y)$ be defined as:

$$H(\ell(a, y)) = \begin{bmatrix} \dfrac{\partial^2 \ell(a,y)}{\partial a_1^2} & \dfrac{\partial^2 \ell(a,y)}{\partial a_1 \partial a_2} & \cdots & \dfrac{\partial^2 \ell(a,y)}{\partial a_1 \partial a_{2R}} \\ \dfrac{\partial^2 \ell(a,y)}{\partial a_2 \partial a_1} & \dfrac{\partial^2 \ell(a,y)}{\partial a_2^2} & \cdots & \dfrac{\partial^2 \ell(a,y)}{\partial a_2 \partial a_{2R}} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial^2 \ell(a,y)}{\partial a_{2R} \partial a_1} & \dfrac{\partial^2 \ell(a,y)}{\partial a_{2R} \partial a_2} & \cdots & \dfrac{\partial^2 \ell(a,y)}{\partial a_{2R}^2} \end{bmatrix}$$

Using propositions (3.2) and (3.3), we have:

$$\frac{\partial^2 \ell(a,y)}{\partial a_i \partial a_j} = \frac{\partial^2 \ell(a,y)}{\partial a_k \partial a_l} = \frac{\partial^2 \ell(a,y)}{\partial a_1 \partial a_2}$$

for all i, j, l, k with $i \neq j$ and $l \neq k$. Moreover:

$$\begin{aligned} \frac{\partial^2 \ell(a,y)}{\partial a_j^2} &= K_j(a) + \frac{\partial^2 \ell(a,y)}{\partial a_1 \partial a_2} \\ K_j(a) &= \Psi(y_j + a_j) - \Psi(a_j) \end{aligned}$$

We thus have,

$$H(\ell(a,y)) = \begin{bmatrix} K_1(a) + \frac{\partial^2 \ell(a,y)}{\partial a_1 \partial a_2} & \frac{\partial^2 \ell(a,y)}{\partial a_1 \partial a_2} & \cdots & \frac{\partial^2 \ell(a,y)}{\partial a_1 \partial a_2} \\ \frac{\partial^2 \ell(a,y)}{\partial a_1 \partial a_2} & K_2(a) + \frac{\partial^2 \ell(a,y)}{\partial a_1 \partial a_2} & \cdots & \frac{\partial^2 \ell(a,y)}{\partial a_1 \partial a_2} \\ \vdots & & \ddots & \vdots \\ \frac{\partial^2 \ell(a,y)}{\partial a_1 \partial a_2} & \frac{\partial^2 \ell(a,y)}{\partial a_1 \partial a_2} & \cdots & K_{2R}(a) + \frac{\partial^2 \ell(a,y)}{\partial a_1 \partial a_2} \end{bmatrix}$$

Hence,

$$H(\ell(a,y)) = \frac{\partial^2 \ell(a,y)}{\partial a_1 \partial a_2} \times \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix} + \begin{bmatrix} K_1(a) & 0 & \cdots & 0 \\ 0 & K_2(a) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & K_{2R}((a) \end{bmatrix}$$

We conclude that:

$$H(a,y) = \alpha(a) \times U + D_a$$

where U is the matrix with all entries equal to one(1) and D_a is a diagonal matrix.

A common problem encountered in the Newton-Raphson method is the invertibility of the Hessian matrix at each iteration. The following theorem guarantees that the matrix $H(a,y)$ is indeed invertible and provides its inverse by using the Schur complement approach.

Theorem 3.6
The matrix $H(a,y)$ is invertible, and its inverse is given by:

$$H^{-1}(a,y) = D_a^{-1} - \alpha(a) \times \left(1 + \alpha(a)\|V_u\|^2_{D_a^{-1}}\right)^{-1} \times D_a^{-1} V_u V_u^T D_a^{-1}$$

Proof

Let's define:

$$\Psi(a) = \frac{\partial^2 \log \Gamma(a)}{\partial a^2} \tag{24}$$

and

$$M = \begin{bmatrix} D_a & -\alpha(a)V_u \\ V_u^T & 1 \end{bmatrix} \tag{25}$$

where D_a is a $2R \times 2R$ diagonal matrix and $V_u = (1, 1, \cdots, 1)^T$ is a $2R \times 1$ vector such that:

$$\begin{aligned} D_{jj} &= K_j(a) \\ &= \Psi(y_j + a_j) - \Psi(a_j) \end{aligned}$$

$$U = V_u \times V_u^T$$

From proposition (3.4) and by assuming that $y_j > 0$, the values $K_j(a) = \Psi(y_j + a_j) - \Psi(a_j) \neq 0$ for all $j = 1, \ldots, 2R$. Hence, the diagonal matrix D_a is invertible.

Now, consider the Schur complement of 1 in the matrix M:

$$(M/1) = D_a + \alpha(a)V_u V_u^T \tag{26}$$

From lemma (3.5), we have $H(a, y) = (M/1)$. Moreover, according to [21], since D_a is invertible, the matrix $(M/1)$, thus $H(a, y)$, is invertible and its inverse is given by:

$$\begin{aligned} (M/1)^{-1} &= D_a^{-1} + D_a^{-1} \times (-\alpha(a))V_u \times (M/D_a)^{-1} \times V_u^T \times D_a^{-1} \\ &= D_a^{-1} - \alpha(a)\left(1 + \alpha(a)V_u^T D_a^{-1} V_u\right)^{-1} \times D_a^{-1} V_u V_u^T D_a^{-1} \\ &= D_a^{-1} - \alpha(a)\left(1 + \alpha(a)\|V_u\|^2_{D_a^{-1}}\right)^{-1} \times D_a^{-1} V_u V_u^T D_a^{-1} \end{aligned}$$

Therefore, we conclude:

$$[H(a, y)]^{-1} = (M/1)^{-1} = D_a^{-1} - \alpha(a)\left(1 + \alpha(a)\|V_u\|^2_{D_a^{-1}}\right)^{-1} \times D_a^{-1} V_u V_u^T D_a^{-1} \tag{27}$$

Remark 3.7. From lemma (3.5), we can also use the Sherman-Morrison formula as an alternative method for computing the inverse of the Hessian matrix given in (27).

The following corollary establishes an explicit expression for the inverse of the Hessian matrix.

Corollary 3.8
The matrix $[H(a, y)]^{-1}$ can be written in the form:

$$[H(\ell(a, y))]^{-1} = \begin{bmatrix} \frac{K_1(a) - C(a)}{K_1^2(a)} & \frac{-C(a)}{K_1(a)K_2(a)} & \cdots & \frac{-C(a)}{K_1(a)K_d(a)} \\ \frac{-C(a)}{K_1(a)K_2(a)} & \frac{K_2(a) - C(a)}{K_2^2(a)} & \cdots & \frac{-C(a)}{K_2(a)K_d(a)} \\ \vdots & & \ddots & \vdots \\ \frac{-C(a)}{K_1(a)K_d(a)} & \frac{-C(a)}{K_2(a)K_d(a)} & \cdots & \frac{K_{2R}((a) - C(a)}{K_d^2(a)} \end{bmatrix}$$

where

$$C(a) = \frac{\alpha(a)}{1 + \alpha(a)\sum_{j=1}^{2R} \frac{1}{K_j(a)}}.$$

Proof
Indeed, from (27), $[H(a,y)]^{-1}$ is given by:

$$[H(a,y)]^{-1} = D_a^{-1} - \alpha(a)\left(1+\alpha(a)\|V_u\|^2_{D_a^{-1}}\right)^{-1} \times D_a^{-1}V_uV_u^TD_a^{-1}.$$

Moreover,

$$D_a^{-1} = \begin{bmatrix} \frac{1}{K_1(a)} & 0 & \cdots & 0 \\ 0 & \frac{1}{K_2(a)} & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{K_{2R}((a)} \end{bmatrix}$$

and

$$\|V_u\|^2_{D_a^{-1}} = V_u^TD_a^{-1}V_u = \sum_{j=1}^{2R}\frac{1}{K_j(a)}.$$

Also,

$$D_a^{-1}V_uV_u^TD_a^{-1} = \begin{bmatrix} \frac{1}{K_1^2(a)} & \frac{1}{K_1(a)K_2(a)} & \cdots & \frac{1}{K_1(a)K_{2R}(a)} \\ \frac{1}{K_1(a)K_2(a)} & \frac{1}{K_2^2(a)} & \cdots & \frac{1}{K_2(a)K_{2R}(a)} \\ \vdots & & \ddots & \vdots \\ \frac{1}{K_1(a)K_{2R}(a)} & \frac{1}{K_2(a)K_{2R}(a)} & \cdots & \frac{1}{K_{2R}^2(a)} \end{bmatrix}.$$

Replacing these terms in (27) and setting

$$C(a) = \frac{\alpha(a)}{1+\alpha(a)\sum_{j=1}^{2R}\frac{1}{K_j(a)}},$$

We obtain:

$$[H(\ell(a,y))]^{-1} = \begin{bmatrix} \frac{K_1(a)-C(a)}{K_1^2(a)} & \frac{-C(a)}{K_1(a)K_2(a)} & \cdots & \frac{-C(a)}{K_1(a)K_{2R}((a)} \\ \frac{-C(a)}{K_1(a)K_2(a)} & \frac{K_2(a)-C(a)}{K_2^2(a)} & \cdots & \frac{-C(a)}{K_2(a)K_{2R}((a)} \\ \vdots & & \ddots & \vdots \\ \frac{-C(a)}{K_1(a)K_{2R}((a)} & \frac{-C(a)}{K_2(a)K_{2R}((a)} & \cdots & \frac{K_{2R}((a)-C(a)}{K_{2R}^2(a)} \end{bmatrix}.$$

We have thus explicitly determined, through these results, the inverse of the Hessian matrix associated with the Newton-Raphson iterative scheme. Hence, with a suitable choice of the initial parameter vector, the convergence of the method is guaranteed.

3.3. Presentation of the Newton-Raphson method

The Newton-Raphson method is a numerical procedure to find a critical point of a function. By applying this method to the log-likelihood function, one hopes to find its maximum, thus the maximum likelihood estimator. We aim to determine the parameter vector a of a real-valued function f using the Newton-Raphson method.

Let $d > 1$ a natural number and f be a function defined by:

$$f : D \subset \mathbb{R}^d \longrightarrow \mathbb{R}.$$

We assume that f is twice differentiable with respect to the parameter vector a.
This method is based on the Taylor expansion, more precisely on the linear approximation of the gradient of f:

$$\nabla f(a) = \nabla f_x(\xi) + H(\xi)(a - \xi) + r(a, \xi), \tag{28}$$

where $\xi \in D$, $H(\xi)$ is the Hessian matrix of f at point ξ, and $r(a, \xi)$ is a remainder term.
The Newton-Raphson method consists in neglecting the remainder term $r(a, \xi)$ and then seeking the solution to the equation:

$$\nabla f(a) = 0, \tag{29}$$

which is equivalent to:

$$\nabla f(\xi) + H(\xi)(a - \xi) = 0. \tag{30}$$

Hence,

$$a = \xi - [H(\xi)]^{-1} \times \nabla f(\xi). \tag{31}$$

The iterative scheme is given by the following relation:

$$a^{(0)} = \xi, \tag{32}$$

$$a^{(k+1)} = a^{(k)} - [H(a^{(k)})]^{-1} \times \nabla f(a^{(k)}). \tag{33}$$

For the algorithm to be well-defined, the matrix $[H(a^{(k)})]^{-1}$ must exist at each iteration.
A stopping criterion is fixed. For any $\epsilon > 0$, we check if:

$$\|a^{(k+1)} - a^{(k)}\|_2 < \epsilon. \tag{34}$$

In general, it is difficult to guarantee the convergence of the sequence $(a^{(k)})_k$. However, if it converges, it does so quite quickly.
However, several problems can occur when using this method.
The first one is the evaluation and inversion of the Hessian matrix which may be difficult and very costly numerically if the spatial dimension d of the parameter a is high or if the expression of $f(a)$ is complex. The second one may occur when $a^{(k)}$ is far from the true solution. It should be noted that the Newton-Raphson method is not an ascent method, meaning we do not necessarily have $f(a^{(k+1)}) > f(a^{(k)})$.

Algorithm 3.9
The Newton-Raphson algorithm can be described as follows:

1- Initial data: a^0, $\nabla_{a^0}\ell$, $[H(a^0)]^{-1}$, $iter = 0$, $stop = 0$, and $\epsilon > 0$.
2- For k while $stop = 0$:
   a- Update
   $$a^{(k+1)} = a^{(k)} - [H(a^{(k)})]^{-1} \times \nabla_a f(a^{(k)}).$$
   b- If $\|a^{(k+1)} - a^{(k)}\|_2 > \epsilon$, then
      i. $a^{(k)} = a^{(k+1)}$,
      ii. $k = k + 1$,

iii. $iter = iter + 1$,
iv. return to step a-.

c- Otherwise,
i. $\hat{a} = a^{(k+1)}$,
ii. $iter = iter + 1$,
iii. $stop = 1$.

3- End.

Among the disadvantages of the Newton-Raphson method are notably the requirement of the Hessian matrix invertibility and the choice of the initial parameter. In this work, we have formally demonstrated, using the Schur complement, that the Hessian matrix is invertible. It remains now to determine an appropriate initial parameter to guarantee the convergence of the method.

3.4. Determination of the initial parameter for the Newton-Raphson method

Let $x = (x_{11}, x_{12}, \cdots, x_{1R}, x_{21}, x_{22}, \cdots, x_{2R})^T$ be a realization of the random vector X, where X models the road accident data before and after the experimental site improvement. We have:

$$X \sim \text{DMultinomial}_{2R}(n, a) \tag{35}$$

where $n = \sum_{j=1}^{R}(x_{1j} + x_{2j})$ and a is the parameter of the Dirichlet distribution. The expectation of X is defined by:

$$E[X] = n \times \frac{a}{a_+} \tag{36}$$

where $a_+ = \sum_{j=1}^{2R} a_j$.

Since X is a random vector, we write:

$$E[X] = (E[X_{11}], E[X_{12}], \cdots, E[X_{1R}], E[X_{21}], E[X_{22}], \cdots, E[X_{2R}])^T. \tag{37}$$

From this, for $j = 1, \ldots, R$, we have:

$$\begin{aligned} E[X_{1j}] &= n \times \frac{a_j}{a_+}, \\ E[X_{2j}] &= n \times \frac{a_{R+j}}{a_+}. \end{aligned} \tag{38}$$

Since there is only one observed realization for each component of the vector X and using the method of moments (which assumes that empirical moment equals to theoretical one), we have:

$$E[X_{ij}] = \bar{x}_{ij} = x_{ij}$$

for $i = 1, 2$ and $j = 1, \ldots, R$.
From relations (38) and (39), we deduce that for $j = 1, \ldots, R$:

$$\begin{aligned} a_j^0 &= a_+ \times \frac{x_{1j}}{n}, \\ a_{R+j}^0 &= a_+ \times \frac{x_{2j}}{n}. \end{aligned} \tag{39}$$

From these formulas, the components a_j^0 depend on a_+. Following [8], the parameter a_+ is related to an overdispersion parameter ρ defined by:

$$\rho^2 = \frac{1}{1+a_+}, \tag{40}$$

where $0 < \rho < 1$.
This parameter allows additional variability to be incorporated into the basic model, as in the cases of beta-binomial and Dirichlet-multinomial distributions. In contrast, binomial and multinomial models do not take this overdispersion into account, which often leads to higher variability observed in the data than predicted by the model.
From this formula (40), we deduce:

$$\hat{a}_+ = \frac{1-\rho^2}{\rho^2}. \tag{41}$$

Thus, determining the initial parameter a^0 is reduced to choosing the parameter ρ between 0 and 1 and using formula (39) with the recorded accident data on site.

Remark 3.10. The determination of the initial parameter vector a^0 depends both on the theoretical mean and the empirical mean of X, as well as on the model's overdispersion parameter. Instead of manually assigning an initial value to each component of a^0, we propose to select a value of ρ between 0 and 1 and then use formula (39) to compute a^0.

4. Numerical Study

In this section, we evaluate the performance of the proposed estimation method through a simulation study. The objective is to assess its accuracy, convergence behavior, and computational efficiency under various configurations.

To highlight the contribution of using the Schur complement for computing the inverse of the Hessian matrix, we compare this approach with the standard Newton–Raphson method based on direct numerical inversion of the Hessian (NR-Solve).
In addition, our procedure is compared with several benchmark optimization algorithms, namely BFGS [1, 5, 6, 16] and Nelder–Mead (NM) [9], implemented via the `optim` function in R [15].

All simulations are conducted using R (version 4.3.1) on a machine equipped with an Intel(R) Core(TM) i7-8650U processor (1.90 GHz), 20 GB of RAM, and a 64-bit operating system (x64 architecture).

4.1. Data generation principle

Considering R and n as the number of accident types and the total number of accidents observed at the experimental site, respectively, we first generate a parameter ρ from a uniform distribution $\mathcal{U}[0.1,\ 0.2]$. This interval was selected to avoid extreme situations in which the estimation procedures tend to become unstable or degenerate, particularly when ρ takes values close to 0 or 1. Based on this value, we compute the quantity a_+ using formula (41). Finally, the observations x are generated from a Dirichlet–multinomial distribution with parameters (n, a_{true}), where $a_{\text{true}} = a_+ \times p$, and p is drawn from a Dirichlet distribution.
After generating the data, we consider two additional initialization strategies for the parameter vector, in addition to our proposed approach, as defined below:

- Proposed initialization: $a_{\text{rho}} \leftarrow a_+ \times \dfrac{x}{n}$
- Moment-based initialization with $\epsilon = 10^{-2}$: $a_{\text{mom}} \leftarrow a_+ \times \dfrac{x+\epsilon}{n + 2 \times R \times \epsilon}$
- Uniform initialization: $a_{\text{unif}} \leftarrow \text{rep}(1,\ 2 \times R)$

It should be noted that the initializations a_{rho} and a_{mom} are constructed from the observed count vector x, whose values vary from one replication to another. Consequently, they may transmit this variability to the resulting

parameter estimates. In contrast, the uniform initialization remains unchanged across replications and is therefore not directly affected by fluctuations in the simulated data.

4.2. Metrics for estimations assessment

We compare the proposed Newton-Raphson-based method (3.9), which explicitly uses the inverse of the Hessian matrix, with a quasi-Newton (BFGS) and the Nelder–Mead (NM) algorithms (implemented in the `optim` function in the R software), which do not explicitly involve second-order derivatives. To enforce the positivity constraint on the parameters a, we adopt the logarithmic transformation $b_j = \log(a_j)$, for $j = 1, \ldots, 2R$.
To assess the performance of the algorithms on the data, we study their accuracy using the Bias, MSE, and the convergence rate, the number of iterations required to converge to a solution, and the convergence time using the CPU time computed in seconds.

4.3. Results and discussion

The following tables report the results of estimating the parameter a from simulated data generated on an experimental site under various scenarios. For $R \in \{3, 5\}$, total accident counts of $n \in \{50, 100, 1000\}$ were considered, while for $R \in \{10, 20\}$, only $n = 1000$ were examined. This choice is motivated by the fact that, for large values of R combined with small sample sizes, the simulated observations often contain zero-count categories. Such configurations may result in a singular Hessian matrix, preventing its inversion and consequently hindering the implementation of Hessian-based estimation procedures (see details in the proof of the theorem (3.6)).
To evaluate the robustness and stability of the proposed approach, several simulation settings were investigated. The results reported in the following tables are averages over 100 independent simulation runs. For each run, the maximum number of iterations was set to 600. Convergence was declared when the Euclidean norm between two successive parameter estimates satisfied ($\|a^{(k+1)} - a^{(k)}\|_2 < 10^{-4}$), where $\|\cdot\|_2$ denotes the Euclidean (L_2) norm on $\mathbb{R}^{2R}$.
The performance of the proposed procedure was subsequently compared with that of the BFGS and Nelder–Mead (NM) algorithms.
It should be noted that the estimation problem examined in this work is particularly complex, as only a single Dirichlet-multinomial observation is available. In such contexts, the likelihood surface can become relatively flat, which may lead to incorrect parameter estimates despite the numerical convergence of the optimization algorithms.

4.3.1. Numerical study of the Newton-Raphson algorithm Table 1 compares the two Newton-Raphson procedures, NR-Schur and NR-Solve, under different initialization strategies (Mom, Rho, and Unif) for various values of n and R in the Dirichlet–multinomial model.
The results indicate that both procedures yield identical statistical performance in terms of bias, mean squared error (MSE), number of iterations, gradient norm, and convergence rate. This outcome is expected since both approaches are based on the same Newton-Raphson scheme and converge to the same numerical solution. Their main difference lies in the computational cost associated with the matrix inversion technique employed at each iteration.
Across all configurations, NR-Schur consistently outperforms NR-Solve in terms of computational efficiency, and this advantage becomes more pronounced as n and R increase. For instance, for $n = 1000$ and $R = 10$ with Mom initialization, NR-Schur requires only 0.02864 seconds, compared with 0.05404 seconds for NR-Solve, representing nearly a 50% reduction in CPU time. These findings highlight the computational benefits of exploiting the Schur complement for Hessian inversion rather than relying on direct matrix inversion.

Table 1. Comparison of Newton-Raphson procedures for the Dirichlet–multinomial model across $n \in \{50, 100, 1000\}$

n	R	Method	Initialization	Rate (%)	Bias	MSE	Iterations	Time (s)	Gradient
50	3	NR-Schur	Mom	100	1.70×10^{2}	4.38×10^{4}	63.15	0.00186	9.82×10^{-4}
		NR-Schur	Rho	100	1.62×10^{2}	4.02×10^{4}	52.19	0.00160	9.81×10^{-4}
		NR-Schur	Unif	100	8.91×10^{1}	1.48×10^{4}	37.96	0.00139	9.67×10^{-4}
		NR-Solve	Mom	100	1.70×10^{2}	4.38×10^{4}	63.15	0.00385	9.82×10^{-4}
		NR-Solve	Rho	100	1.62×10^{2}	4.02×10^{4}	52.19	0.00311	9.81×10^{-4}
		NR-Solve	Unif	100	8.91×10^{1}	1.48×10^{4}	37.96	0.00290	9.67×10^{-4}
	5	NR-Schur	Mom	100	1.49×10^{2}	6.23×10^{4}	94.16	0.00301	9.88×10^{-4}
		NR-Schur	Rho	100	1.47×10^{2}	6.05×10^{4}	82.39	0.00254	9.87×10^{-4}
		NR-Schur	Unif	100	8.26×10^{1}	2.04×10^{4}	56.47	0.00197	9.77×10^{-4}
		NR-Solve	Mom	100	1.49×10^{2}	6.23×10^{4}	94.16	0.00585	9.88×10^{-4}
		NR-Solve	Rho	100	1.47×10^{2}	6.05×10^{4}	82.39	0.00504	9.87×10^{-4}
		NR-Solve	Unif	100	8.26×10^{1}	2.04×10^{4}	56.47	0.00364	9.77×10^{-4}
100	3	NR-Schur	Mom	100	2.71×10^{2}	1.15×10^{5}	92.52	0.00274	9.88×10^{-4}
		NR-Schur	Rho	100	2.61×10^{2}	1.07×10^{5}	80.69	0.00241	9.88×10^{-4}
		NR-Schur	Unif	100	1.69×10^{2}	5.43×10^{4}	61.64	0.00218	9.78×10^{-4}
		NR-Solve	Mom	100	2.71×10^{2}	1.15×10^{5}	92.52	0.00564	9.88×10^{-4}
		NR-Solve	Rho	100	2.61×10^{2}	1.07×10^{5}	80.69	0.00491	9.88×10^{-4}
		NR-Solve	Unif	100	1.69×10^{2}	5.43×10^{4}	61.64	0.00407	9.78×10^{-4}
	5	NR-Schur	Mom	100	2.81×10^{2}	2.06×10^{5}	157.49	0.00482	9.93×10^{-4}
		NR-Schur	Rho	100	2.80×10^{2}	2.06×10^{5}	148.34	0.00453	9.93×10^{-4}
		NR-Schur	Unif	100	1.12×10^{2}	3.67×10^{4}	70.11	0.00244	9.83×10^{-4}
		NR-Solve	Mom	100	2.81×10^{2}	2.06×10^{5}	157.49	0.00980	9.93×10^{-4}
		NR-Solve	Rho	100	2.80×10^{2}	2.06×10^{5}	148.34	0.00887	9.93×10^{-4}
		NR-Solve	Unif	100	1.12×10^{2}	3.67×10^{4}	70.11	0.00456	9.83×10^{-4}
1000	3	NR-Schur	Mom	99	8.53×10^{2}	1.12×10^{6}	259.57	0.01136	9.96×10^{-4}
		NR-Schur	Rho	99	8.08×10^{2}	1.02×10^{6}	238.35	0.01200	9.95×10^{-4}
		NR-Schur	Unif	100	4.28×10^{2}	3.40×10^{5}	135.30	0.00636	9.92×10^{-4}
		NR-Solve	Mom	99	8.53×10^{2}	1.12×10^{6}	259.57	0.02721	9.96×10^{-4}
		NR-Solve	Rho	99	8.08×10^{2}	1.02×10^{6}	238.35	0.02664	9.95×10^{-4}
		NR-Solve	Unif	100	4.29×10^{2}	3.40×10^{5}	135.30	0.01188	9.92×10^{-4}
	5	NR-Schur	Mom	99	5.31×10^{2}	4.4×10^{5}	219.74	0.01810	9.96×10^{-4}
		NR-Schur	Rho	99	5.17×10^{2}	4.32×10^{5}	206.15	0.01590	9.95×10^{-4}
		NR-Schur	Unif	99	3.17×10^{2}	1.51×10^{5}	133.13	0.01070	9.93×10^{-4}
		NR-Solve	Mom	99	5.31×10^{2}	4.54×10^{5}	219.74	0.03540	9.96×10^{-4}
		NR-Solve	Rho	99	5.17×10^{2}	4.32×10^{5}	206.15	0.03280	9.95×10^{-4}
		NR-Solve	Unif	99	3.17×10^{2}	1.51×10^{5}	133.13	0.02150	9.93×10^{-4}
	10	NR-Schur	Mom	32	7.66×10^{2}	1.12×10^{6}	479.44	0.04540	9.98×10^{-4}
		NR-Schur	Rho	28	7.79×10^{2}	1.15×10^{6}	478.00	0.02797	9.98×10^{-3}
		NR-Schur	Unif	100	2.82×10^{2}	1.57×10^{5}	184.44	0.01010	9.95×10^{-4}
		NR-Solve	Mom	32	7.66×10^{2}	1.12×10^{6}	479.44	0.08340	9.98×10^{-4}
		NR-Solve	Rho	28	7.79×10^{2}	1.15×10^{6}	478.00	0.08350	9.98×10^{-4}
		NR-Solve	Unif	100	2.82×10^{2}	1.57×10^{5}	184.44	0.01844	9.95×10^{-4}
	20	NR-Schur	Mom	0	5.90×10^{2}	8.57×10^{5}	600	0.04570	2.07×10^{-3}
		NR-Schur	Rho	0	5.97×10^{2}	8.75×10^{5}	600	0.04740	2.41×10^{-3}
		NR-Schur	Unif	100	2.30×10^{2}	1.31×10^{5}	240.02	0.01940	9.96×10^{-4}
		NR-Solve	Mom	0	5.90×10^{2}	8.57×10^{5}	600	0.10880	2.07×10^{-3}
		NR-Solve	Rho	0	5.97×10^{2}	8.75×10^{5}	600	0.10330	2.41×10^{-3}
		NR-Solve	Unif	100	2.30×10^{2}	1.31×10^{5}	240.02	0.04110	9.96×10^{-4}

NR-Schur: Newton-Raphson method using Schur complement
NR-Solve: Newton-Raphson method using direct matrix inversion
Mom, **Rho**, **Unif**: initialization strategies
MSE: mean squared error; **Time**: CPU time; **Gradient**: final gradient norm

The choice of initialization has a substantial impact on estimation accuracy. In nearly all scenarios, the uniform initialization (Unif) yields the smallest bias and MSE values, followed by the Rho initialization, whereas the Mom initialization generally provides the least accurate estimates. For example, when $n = 50$ and $R = 3$, the bias decreases from 1.70×10^2 and 1.62×10^2 under the Mom and Rho initializations, respectively, to 8.91×10^1 under the Unif initialization. Similarly, the MSE decreases from 4.38×10^4 and 4.02×10^4 to 1.48×10^4. A comparable pattern is observed for $n = 100$, suggesting that uniform initialization substantially improves estimation accuracy. The convergence diagnostics further confirm the numerical robustness of both procedures. In almost all configurations, the final gradient norm remains close to 10^{-3}, indicating satisfactory convergence to a stationary point of the log-likelihood function, even in relatively high-dimensional settings such as $n = 1000$ and $R = 10$. The inclusion of the convergence rate, computed over 100 simulation runs, provides additional evidence regarding the robustness of the estimation procedures. The results show that the Unif initialization achieves the highest convergence rates, reaching near-perfect convergence in most configurations and attaining a 100% convergence rate in some cases when $R = 20$. By contrast, the Mom and Rho initializations exhibit lower convergence rates in the most challenging settings, particularly as the dimension of the problem increases. These findings emphasize the critical role of starting values, not only in improving estimation accuracy but also in ensuring reliable convergence of the Newton-Raphson algorithm.

Finally, the results reveal that the computational burden increases with both n and R, as reflected by the number of iterations and CPU time. Nevertheless, NR-Schur consistently maintains a substantial computational advantage, demonstrating superior scalability for higher-dimensional problems. Overall, the combination of NR-Schur and uniform initialization emerges as an efficient, robust, and reliable estimation strategy for the one-observation Dirichlet–multinomial model.

Table 2. Comparison of optimization methods under rho initialization

n	R	Method	Rate (%)	Bias	MSE	Iterations	Time (s)	Gradient
50	3	BFGS	100	3.44×10^3	1.86×10^7	24.70	0.00029	4.16×10^{-4}
		NM	100	1.14×10^4	3.14×10^9	220.14	0.00145	1.86×10^{-3}
		NR-Schur	100	1.62×10^2	4.02×10^4	52.19	0.00160	9.81×10^{-4}
	5	BFGS	100	3.78×10^3	4.50×10^7	26.20	0.00031	5.96×10^{-4}
		NM	60	4.10×10^2	1.87×10^7	346.6	0.00520	1.010×10^{-1}
		NR-Schur	100	1.47×10^2	6.05×10^4	82.39	0.00254	9.87×10^{-4}
100	3	BFGS	100	3.71×10^3	2.16×10^7	28.21	0.00031	2.56×10^{-3}
		NM	100	1.17×10^4	3.08×10^9	207.30	0.00144	4.55×10^{-3}
		NR-Schur	100	2.61×10^2	1.07×10^5	80.69	0.00241	9.88×10^{-4}
	5	BFGS	100	3.19×10^3	3.39×10^7	30.13	0.00034	1.14×10^{-3}
		NM	73	3.93×10^2	2.81×10^7	346.12	0.00530	2.87×10^{-1}
		NR-Schur	100	2.80×10^2	2.06×10^5	148.34	0.00453	9.93×10^{-4}
1000	3	BFGS	100	1.73×10^3	9.20×10^6	34.02	0.00047	2.66×10^{-2}
		NM	100	9.67	9.06×10^3	20.58	0.00026	2.08×10^{-1}
		NR-Schur	99	8.08×10^2	1.02×10^6	238.35	0.01200	9.95×10^{-4}
	5	BFGS	100	7.12×10^2	1.21×10^6	36.88	0.00040	2.07×10^{-2}
		NM	100	1.29×10^{-1}	7.46	22.54	0.00027	3.75×10^{-1}
		NR-Schur	99	5.17×10^2	4.31×10^5	206.15	0.01580	9.95×10^{-4}
	10	BFGS	100	3.30×10^2	2.66×10^5	43.21	0.00069	6.80×10^{-2}
		NM	100	2.37×10^{-3}	3.61	31.96	0.00045	7.34
		NR-Schur	28	7.79×10^2	1.14×10^6	478.84	0.04670	9.97×10^{-4}
	20	BFGS	100	3.70×10^2	4.40×10^5	51.32	0.00090	4.16×10^{-1}
		NM	100	-2.42	1.19×10^1	503.76	0.00645	5.81
		NR-Schur	0	5.97×10^2	8.75×10^5	600	0.04740	2.41×10^{-3}

NR-Schur: Newton-Raphson method using Schur complement
BFGS: Broyden–Fletcher–Goldfarb–Shanno algorithm
NM: Nelder–Mead simplex method
Gradient: : final gradient norm
MSE: mean squared error; **Time**: CPU time

4.3.2. Comparison with other methods Tables 2, 3, and 4 present a comparison of the performance of the NR-Schur, BFGS, and Nelder–Mead (NM) methods under different initialization strategies (Rho, Mom, and Unif) for several values of n and R in the Dirichlet–multinomial model.

Overall, the results indicate that the NR-Schur method provides the best compromise between statistical accuracy, numerical stability, and computational cost. For small and moderate sample sizes $n = 50$ and $n = 100$, NR-Schur generally yields substantially lower bias and MSE values than BFGS and NM, while requiring a moderate number of iterations. In contrast, the BFGS and NM methods often exhibit very large estimation errors, particularly in low-dimensional settings. For example, for $n = 50$ and $R = 3$ under Rho initialization, the bias obtained with NR-Schur is 1.62×10^2, compared with 3.44×10^3 for BFGS and 1.14×10^4 for NM.

The convergence rates, computed over 100 simulation runs, provide additional insight into the robustness of the competing optimization methods. For small and moderate dimensions $n = 50$ and $n = 100$, NR-Schur and BFGS achieve convergence rates close to or equal to 100% across most configurations, whereas the NM method occasionally exhibits lower convergence rates, particularly under the Rho initialization. These results indicate that Newton-Raphson and quasi-Newton approaches are generally more reliable in terms of successful convergence.

The results also show that the initialization strategy strongly influences the quality of the estimates. Overall, uniform initialization (Unif) improves the performance of the different methods and often leads to the lowest bias and MSE values. Rho initialization also provides satisfactory results, whereas moment initialization (Mom) is generally less effective. This trend is particularly evident for NR-Schur, whose performance becomes more accurate and stable under uniform initialization.

Table 3. Comparison of optimization methods under moment initialization

n	R	Method	Rate (%)	Bias	MSE	Iterations	Time (s)	Gradient
50	3	BFGS	100	2.18×10^3	8.84×10^6	22.15	0.00044	8.09×10^{-4}
		NM	99	2.44×10^4	1.48×10^{10}	335.92	0.00287	5.89×10^{-2}
		NR-Schur	100	1.70×10^2	4.38×10^4	63.15	0.00298	9.82×10^{-4}
	5	BFGS	100	3.15×10^3	3.38×10^7	25.02	0.00035	3.65×10^{-4}
		NM	17	5.31×10^3	2.51×10^4	514.41	0.00610	1.361
		NR-Schur	100	1.49×10^2	6.23×10^4	94.16	0.00400	9.88×10^{-4}
100	3	BFGS	100	1.83×10^3	7.42×10^6	25.14	0.00029	1.50×10^{-3}
		NM	98	1.57×10^4	7.87×10^9	304.02	0.00300	1.15×10^{-1}
		NR-Schur	100	2.71×10^2	1.15×10^5	92.52	0.00297	9.88×10^{-4}
	5	BFGS	100	2.53×10^3	2.56×10^7	32.87	0.00037	2.43×10^{-3}
		NM	24	3.96×10^1	1.30×10^4	466.91	0.00601	2.34×10^1
		NR-Schur	100	2.81×10^2	2.06×10^5	157.49	0.00562	9.93×10^{-4}
1000	3	BFGS	100	1.72×10^3	8.86×10^6	38.00	0.00046	2.92×10^{-2}
		NM	100	2.84×10^2	1.43×10^6	160.46	0.00300	1.96
		NR-Schur	99	8.53×10^2	1.12×10^6	259.57	0.01300	9.96×10^{-4}
	5	BFGS	100	9.65×10^2	1.55×10^7	41.11	0.00041	2.22×10^{-2}
		NM	97	0.293	1.927×10^3	157.78	0.00964	2.95×10^1
		NR-Schur	99	5.31×10^2	4.54×10^5	219.73	0.01805	9.95×10^{-4}
	10	BFGS	100	3.51×10^2	2.84×10^5	47.67	0.00071	1.05×10^{-1}
		NM	1	-3.03	1.92×10^1	513	0.03400	4.77×10^2
		NR-Schur	32	7.66×10^2	1.115×10^6	479.43	0.04530	9.97×10^{-4}
	20	BFGS	100	3.70×10^2	4.40×10^5	51.32	0.00090	4.16×10^{-1}
		NM	0	-1.67	6.17	600	0.13610	750.70
		NR-Schur	0	5.90×10^2	8.57×10^5	600	0.04570	2.07×10^{-3}

NR-Schur: Newton-Raphson method using Schur complement
BFGS: Broyden–Fletcher–Goldfarb–Shanno algorithm
NM: Nelder–Mead simplex method
Gradient: : final gradient norm
MSE: mean squared error; **Time**: CPU time

When the sample size becomes large $n = 1000$, the NM method becomes particularly competitive in terms of bias and MSE, especially for $R = 5$ and $R = 10$, where it produces estimates very close to the true parameter values. However, these good statistical performances are accompanied by very large gradient norms, sometimes

exceeding 1, indicating unstable or incomplete convergence toward a stationary point of the log-likelihood function. The effect of increasing the number of categories becomes more evident for $R = 20$. Under the Rho initialization, the convergence rate of NR-Schur decreases from 99% for $R = 3$ and $R = 5$ to 28% for $R = 10$, and eventually reaches 0% for $R = 20$. In contrast, both BFGS and NM maintain convergence rates of 100% across these configurations. These results suggest that the Newton-Raphson procedure becomes increasingly sensitive to the choice of starting values as the dimensionality of the problem increases.
By contrast, NR-Schur consistently maintains a gradient norm close to 10^{-3} across all configurations in which convergence is achieved, regardless of the initialization strategy. This confirms the numerical robustness and stability of the method, even in high-dimensional problems. Although its convergence rate may deteriorate under unfavorable initializations in very challenging settings, the use of appropriate starting values, particularly the uniform initialization, substantially improves its reliability.

Table 4. Comparison of optimization methods under uniform initialization

n	R	Method	Rate (%)	Bias	MSE	Iterations	Time (s)	Gradient
50	3	BFGS	100	1.66×10^{3}	5.33×10^{6}	22.04	0.00041	5.26×10^{-4}
		NM	84	9.42×10^{3}	3.63×10^{9}	396.73	0.00367	3.48×10^{-1}
		NR-Schur	100	8.91×10^{1}	1.48×10^{4}	37.96	0.00195	9.67×10^{-4}
	5	BFGS	100	3.83×10^{3}	4.53×10^{7}	26.27	0.00042	5.64×10^{-4}
		NM	43	-6.99	1.58×10^{2}	442.25	0.00651	2.85
		NR-Schur	100	8.26×10^{1}	2.04×10^{4}	56.47	0.00257	9.77×10^{-4}
100	3	BFGS	100	1.51×10^{3}	4.56×10^{6}	26.43	0.00034	2.72×10^{-3}
		NM	90	1.47×10^{4}	8.26×10^{9}	336.53	0.00332	9.03×10^{-1}
		NR-Schur	100	1.69×10^{2}	5.43×10^{4}	61.64	0.00223	9.78×10^{-4}
	5	BFGS	100	2.99×10^{3}	3.20×10^{7}	33.18	0.00042	1.69×10^{-3}
		NM	57	-4.93	7.16×10^{1}	436.36	0.00650	4.37
		NR-Schur	100	1.12×10^{2}	3.67×10^{4}	70.11	0.00280	9.83×10^{-4}
1000	3	BFGS	100	4.86×10^{3}	1.72×10^{9}	36.11	0.00056	7.68×10^{-2}
		EM	0	1.12×10^{2}	1.73×10^{4}	600.00	0.00443	4.62×10^{-3}
		NR-Schur	100	4.29×10^{2}	3.40×10^{5}	135.30	0.00636	9.92×10^{-4}
	5	BFGS	100	1.20×10^{3}	4.49×10^{6}	48.64	0.00054	9.64×10^{-2}
		NM	100	-7.12	7.34×10^{1}	62.88	0.00056	2.51
		NR-Schur	99	3.17×10^{2}	1.50×10^{5}	133.13	0.01070	9.92×10^{-4}
	10	BFGS	100	3.70×10^{2}	4.40×10^{5}	51.32	0.00090	4.16×10^{-1}
		NM	65	-2.50	1.26×10^{1}	50451.4	0.04500	5.36
		NR-Schur	100	2.82×10^{2}	1.57×10^{5}	184.44	0.01010	9.95×10^{-4}
	20	BFGS	100	3.70×10^{2}	4.40×10^{5}	51.32	0.00090	4.16×10^{-1}
		NM	1	-0.75	4.10	65	0.25530	6.88
		NR-Schur	100	2.82×10^{2}	1.57×10^{5}	184.44	0.01010	9.95×10^{-4}

NR-Schur: Newton-Raphson method using Schur complement
BFGS: Broyden–Fletcher–Goldfarb–Shanno algorithm
NM: Nelder–Mead simplex method
Gradient: : final gradient norm
MSE: mean squared error; **Time**: CPU time

Overall, the results demonstrate that NR-Schur combined with uniform initialization offers the best balance between estimation accuracy, numerical stability, and computational efficiency. This combination consistently achieves low bias and MSE values, satisfactory convergence diagnostics, and competitive computational times, making it a particularly attractive estimation strategy for the Dirichlet–multinomial model.

4.3.3. Model assumptions and limitation. The proposed estimation procedure is derived under the assumption that the random probability vector P follows a Dirichlet distribution. Consequently, the resulting Dirichlet–Multinomial model inherits the flexibility of the Dirichlet mixing distribution for modeling overdispersed multinomial counts.
However, if the true mixing distribution differs substantially from a Dirichlet distribution, the model may be misspecified. In such situations, parameter estimates may remain numerically stable, but their statistical interpretation and goodness-of-fit may be affected. Therefore, model adequacy should be assessed before drawing substantive conclusions.

In practice, the suitability of the Dirichlet–Multinomial assumption may be evaluated through goodness-of-fit criteria such as the log-likelihood, AIC, and BIC, or by comparing the Dirichlet–Multinomial model with alternative count models. Graphical comparisons between observed and fitted category proportions may also provide useful diagnostic information.
The study of the robustness of the proposed estimator under alternative mixing distributions constitutes an interesting direction for future research.

5. Application to road accident data

5.1. Data description and parameter estimation

We consider road accident data collected on National Road 17 (RN17), located in the municipalities of Vimy and Avion, in the Pas-de-Calais department, France. In this study, three types of accidents ($R = 3$) are distinguished: fatal (F), serious (S), and minor (M) accidents [11].
The data are recorded over two periods of four years, before and after the road improvement. Let x_{11}, x_{12} and x_{13} (resp. x_{21}, x_{22} and x_{23}) denote the numbers of fatal, serious, and minor accidents observed before (resp. after) the road improvement. We define the random vector by:

$$X = (x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23})^\top,$$

which is assumed to follow a Dirichlet–multinomial distribution, denoted by $DMultinomial(n, a)$, where

$$n = \sum_{j=1}^{3} x_{1j} + x_{2j}$$

represents the total number of observed accidents and $a = (a_1, a_2, a_3, a_4, a_5, a_6)$ is the parameter to be estimated.
The observed data are summarized in the following table.

Table 5. Distribution of observations before and after the intervention

Before (4 years)			After (4 years)			Total
F	S	M	F	S	M	
x_{11}	x_{12}	x_{13}	x_{21}	x_{22}	x_{23}	
4	4	16	1	1	7	33

The results reported in Table 6 were obtained from the real dataset using the NR-Schur, BFGS, and NM methods under different initialization strategies. Overall, the parameter estimates vary considerably across optimization methods; however, the convergence diagnostics indicate that NR-Schur provides the most stable solutions. In particular, NR-Schur consistently maintains a gradient norm close to 10^{-3} for all initialization strategies. In contrast, NM exhibits relatively large gradient norms in some cases, suggesting less reliable convergence despite occasionally producing substantially different estimates. Although BFGS converges rapidly and requires fewer iterations, it produces parameter estimates that differ markedly from those obtained with NR-Schur.

Table 6. Comparison of optimization methods under different initialization strategies

Initialization	Initial vector	NR-Schur	SE	BFGS	SE	NM	SE
Uniform	1	49.05996	84.74786	1034.777	10.20880	2.025492	1.607877
	1	49.05996	84.74786	1034.777	10.20880	1.482312	1.900926
	1	192.5898	320.6182	4142.092	10.20324	6.769819	1.749407
	1	12.67142	23.97638	257.5862	10.23106	0.795192	1.434861
	1	12.67142	23.97638	257.5862	10.23106	0.922921	1.385687
	1	85.18378	144.5004	1809.817	10.20566	1.662155	2.872615
	Iterations	31		17		285	
	Time (s)	6.499×10^{-2}		5.691×10^{-4}		5.910×10^{-3}	
	Gradient	9.6×10^{-4}		3.38×10^{-5}		7.6×10^{-1}	
Rho	7.106	110.5225	271.6366	4185.746	20.48503	1062.935	10.17134
	7.106	110.5225	271.6366	4185.746	20.48503	1046.538	10.17197
	28.424	442.0406	1075.964	16747.76	20.48227	4244.403	10.16583
	1.776	27.58868	70.15772	1047.212	20.49599	242.2335	10.19825
	1.776	27.58868	70.15772	1047.212	20.49599	257.1566	10.19530
	12.435	193.4074	472.7566	7327.988	20.48345	1814.845	10.16914
	Iterations	54		16		319	
	Time (s)	5.02×10^{-3}		5.249×10^{-4}		5.537×10^{-3}	
	Gradient	9.6×10^{-4}		5.09×10^{-7}		3.4×10^{-4}	
Moment	6.026	112.3981	278.4219	4414.428	21.03254	57.53037	2.357280
	6.026	112.3981	278.4219	4414.428	21.03254	45.68587	2.397655
	24.060	449.5923	1103.252	17658.00	21.02987	191.2961	2.362550
	1.517	28.03470	71.76695	1103.677	21.04324	11.56247	2.490456
	1.517	28.03470	71.76695	1103.677	21.04324	8.502026	2.568280
	10.534	196.7170	484.7394	7724.908	21.03102	86.94490	2.366675
	Iterations	62		25		600	
	Time (s)	4.41×10^{-3}		1.11×10^{-3}		1.44×10^{-2}	
	Gradient	9.7×10^{-4}		5.09×10^{-7}		4.1×10^{-2}	

NR–Schur: Newton-Raphson method using Schur complement
BFGS: Broyden–Fletcher–Goldfarb–Shanno algorithm
NM: Nelder–Mead algorithm
Gradient: : final gradient norm
SE: Standard Error

These findings are consistent with the conclusions drawn from the simulation study. In particular, the uniform initialization remains the most effective strategy for NR-Schur, as it reduces the number of iterations required for convergence compared with the Rho and Moment initializations. Furthermore, the numerical robustness of NR-Schur observed in the simulations is also evident in the real-data analysis, where the method yields coherent estimates and stable convergence regardless of the initialization strategy.
Overall, the empirical results corroborate the simulation findings and demonstrate that the NR-Schur method, combined with uniform initialization, provides the most robust and well-balanced estimation approach in this study.

5.2. Comparison of the fit of the Dirichlet–multinomial model with that of the multinomial model

In this section, we consider the estimator $\hat{a}$ obtained in (6) using the NR-Schur method with uniform initialization, which was identified as the most effective initialization strategy in the simulation study.

Although the dataset consists of a single observation of the count vector, the adequacy of the Dirichlet–multinomial model relative to the multinomial model is assessed through the maximized log-likelihood, the overdispersion parameter ρ, and the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC). These criteria are defined as

$$AIC = -2\ell(\hat{a}) + 2k, \tag{42}$$

$$BIC = -2\ell(\hat{a}) + k\log(n), \tag{43}$$

where $\ell(\hat{a})$ denotes the log-likelihood evaluated at the maximum likelihood estimate, n is the total number of observed counts, and k is the number of estimated parameters.

The overdispersion parameter is estimated as $\hat{\rho}^2 = 0.002486$ using Equation (40). The corresponding values of the log-likelihood, AIC, and BIC for the multinomial and Dirichlet–multinomial models are reported in Table 7.

Table 7. Comparison of goodness-of-fit criteria between the multinomial and Dirichlet–multinomial models

Model	LogLik	AIC	BIC
multinomial	-6.810	25.620	34.599
Dirichlet–multinomial	-7.008	26.016	34.995

LogLik: Maximized log-likelihood.

AIC: Akaike Information Criterion.

BIC: Bayesian Information Criterion.

The log-likelihood of the Dirichlet–multinomial distribution is computed using Equation (10), whereas the log-likelihood of the multinomial distribution is given by

$$\ell(\mathbf{p}) = \log(n!) - \sum_{i=1}^{2R} \log(x_i!) + \sum_{i=1}^{2R} x_i \log p_i, \tag{44}$$

where

$$p_i = \frac{x_i}{n}, \qquad i = 1, \ldots, 2R.$$

Table 7 shows that the multinomial and Dirichlet–multinomial models provide very similar fits to the observed data. The slightly lower AIC and BIC values obtained for the multinomial model suggest a marginally better fit; however, the differences between the two models are negligible.

This finding is consistent with the estimated overdispersion parameter ($\hat{\rho}^2 = 0.002486$), whose value is very close to zero, indicating the absence of substantial extra-multinomial variation in the data. Under such conditions, the Dirichlet–multinomial model naturally approaches the multinomial model, which explains the similarity of the goodness-of-fit measures.

These results highlight an important property of the Dirichlet–multinomial model: it is capable of accommodating overdispersion when present, while reducing to the multinomial model when overdispersion is negligible. Consequently, the Dirichlet–multinomial formulation offers a flexible and robust framework for modeling count data without sacrificing interpretability in situations where extra-multinomial variation is weak or absent.

6. Conclusion

In this paper, we investigated parameter estimation for the one-observation Dirichlet–multinomial model using several optimization methods, with particular emphasis on a Newton-Raphson procedure based on the Schur complement, referred to as NR-Schur, and on alternative parameter initialization strategies. The proposed approach was designed to reduce the computational cost associated with Hessian matrix inversion while preserving estimation accuracy and numerical stability.

The simulation results showed that NR-Schur and NR-Solve produce identical estimates and convergence diagnostics, since they rely on the same Newton-Raphson scheme. However, NR-Schur consistently achieved lower computational times by using the Schur complement to evaluate the inverse Hessian matrix. The study also highlighted the importance of initialization. Among the strategies considered, uniform initialization generally yielded the lowest bias and MSE and the highest convergence rate, whereas moment-based initialization yielded less favorable results.

The comparison with BFGS and Nelder–Mead demonstrated that NR-Schur offers an effective balance between statistical accuracy, numerical stability, and computational efficiency. Although the competing methods were sometimes faster, their performance was more sensitive to the model configuration and initialization strategy. In contrast, NR-Schur consistently maintained small gradient norms and reliable convergence across all scenarios investigated.

The real-data application confirmed the conclusions drawn from the simulation study. NR-Schur produced stable estimates under different initialization strategies and satisfactory convergence diagnostics. Furthermore, the comparison between the multinomial and Dirichlet–multinomial models showed very similar goodness-of-fit measures, which was consistent with the estimated overdispersion parameter being close to zero. This result illustrates the Dirichlet–multinomial model's ability to reduce to the multinomial model when overdispersion is negligible.

Overall, the findings indicate that NR-Schur, particularly when combined with uniform initialization, provides a robust and computationally efficient framework for parameter estimation in the one-observation Dirichlet–multinomial model. Future work may extend this methodology to more general Dirichlet–multinomial settings involving multiple observations.

Acknowledgement

This work is part of the doctoral research conducted jointly at Abdou Moumouni University and the University of Lille. We express our deep gratitude to the Erasmus+ International Credit Mobility Program (KA 107) for awarding a mobility grant that enabled five months (March to July, 2022) research stay at the mathematics laboratory of the University of Lille, contributing significantly to the advancement of this study. We sincerely thank the Editor and the anonymous reviewers for their valuable comments and constructive suggestions, which have greatly contributed to improving the quality of this manuscript.

REFERENCES

1. C. G. Broyden, *The convergence of a class of double-rank minimization algorithms*, Journal of the Institute of Mathematics and Its Applications, 6, pp. 76–90, 1970.
2. H. K. T. Ng, G.-L. Tian, and M.-L. Tang, *Dirichlet and Related Distributions: Theory, Methods and Applications*, John Wiley & Sons, Chichester, UK, 2011.
3. D. T. Fuller, S. Mondal, S. Sur, and N. Pal, *Dirichlet Distribution Parameter Estimation With Applications in Microbiome Analyses*, Statistics in Medicine, 45(3–5), e70454, 2026.
4. I. Holmes, K. Harris and C. Quince, *Dirichlet multinomial mixtures: Generative models for microbial metagenomics*, PLoS ONE, 7(2), e30126, 2012.
5. R. Fletcher, *A new approach to variable metric algorithms*, The Computer Journal, 13(3), pp. 317–322, 1970.
6. D. Goldfarb, *A family of variable metric updates derived by variational means*, Mathematics of Computation, 24(109), pp. 23–26, 1970.
7. J. E. Mosimann, *On the compound multinomial distribution, the multivariate beta-distribution, and correlations among proportions*, Biometrika, 49, pp. 65–82, 1962.
8. N. K. Neerchal and J. G. Morel, *An improved method for the computation of maximum likelihood estimates for multinomial overdispersion models*, Computational Statistics & Data Analysis, 49, pp. 33–43, 2005.
9. J. A. Nelder and R. Mead, *A simplex algorithm for function minimization*, The Computer Journal, 7(4), pp. 308–313, 1965.
10. 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, 49(2), pp. 85–102, 2001.
11. A. N'Guessan and Trufur, *Impact d'un aménagement de sécurité routière sur la gravité des accidents de la route*, Journal de la Société Française de Statistique, 149(3), pp. 23–41, 2008.
12. A. N'Guessan and I. C. Geraldo, *A cyclic algorithm for maximum likelihood estimation using Schur complement*, Numerical Linear Algebra with Applications, 22, pp. 1161–1179, 2015.
13. A. N'Guessan, I. C. Geraldo, and B. Hafidi, *An approximation method for a maximum likelihood equation system and application to the analysis of accident data*, Open Journal of Statistics, 7, pp. 132–152, 2017.
14. A. N'Guessan, *Analytical existence of solutions to a system of nonlinear equations with application*, Journal of Computational and Applied Mathematics, 234, pp. 297–304, 2010.
15. R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria. Available at https://www.r-project.org/.

16. D. F. Shanno, *Conditioning of quasi-Newton methods for function minimization*, Mathematics of Computation, 24(111), pp. 647–656, 1970.
17. D. J. Spiegelhalter, N. L. Harris, K. Bull, and R. C. G. Franklin, *Empirical evaluation of prior beliefs about frequencies: methodology and a case study in congenital heart disease*, Journal of the American Statistical Association, 89, pp. 435–443, 1994.
18. T. Rebafka, *Statistique appliquée, cours de Master 1, Mathématiques et Applications*, Université Pierre et Marie Curie (Paris 6), 2015.
19. T. Tvedebrink, *Overdispersion in allelic counts and theta-correction in forensic genetics*, Theoretical Population Biology, 78(3), pp. 200–210, 2010.
20. Z. F. Li, M. R. Osborne, and T. Prvan, *Numerical algorithms for constrained maximum likelihood estimation*, ANZIAM Journal, 45, pp. 91–114, 2003.
21. F. Zhang, *The Schur Complement and Its Applications*, Springer, New York, 2005.
22. Y. Guo, L. Yu, L. Guo, L. Xu, and Q. Li, *A regularized Bayesian Dirichlet-multinomial regression model for integrating single-cell-level omics and patient-level clinical study data*, Biometrics, vol. 81, no. 1, article ujaf005, 2025.