Comparison of Newton-Raphson Algorithm and MaxLik Function

Kim-Hung PHO[1,*], *Vu-Thanh NGUYEN*[2]

[1]Faculty of Mathematics and Statistics, Ton Duc Thang University, Ho Chi Minh City, Vietnam
[2]Faculty of Mathematics and Statistics, University of Natural Sciences, Ho Chi Minh City, Vietnam

*Corresponding Author: Kim-Hung PHO (Email: phokimhung@tdtu.edu.vn)
(Received: 04-December-2018; accepted: 22-January-2019; published: 25-January-2019)
DOI: http://dx.doi.org/10.25073/jaec.201824.213

Abstract. *Our main objective is in antagonizing the performance of two approaches: the Newton-Raphson (N-R) algorithm and maxLik function in the statistical software R to obtain optimization roots of estimating functions. We present the approach of algorithms, examples and discussing about two approaches in detail. Besides, we prove that the N-R algorithm can perform if our data set contain missing values, while maxLik function cannot execute in this situation. In addition, we also compare the results, as well as, the time to run code to output the result of two approaches through an example is introduced in [1].*

Keywords

Newton-Raphson algorithm, maxLik function, optimization, comparison.

1. Introduction

In statistical inference and applied mathematics, estimating functions play an extremely vital position in researches. If having the estimating function we can execute some of approaches to figure out this issue. Comprehensive theory and its applications can be obtained from numerous reference books on statistics. In [2], Godlambe presented about estimating functions in which a function includes the data set and parameters need to be estimated. An overview, the estimating function can be described by H with provided that $H(data, \psi) = 0$, where $\psi \in \Psi$ and Ψ is a parameter space. The issues are associated with finding an optimization root to estimating functions are exceedingly crucial in several areas, such as: statistical inferences, mathematics, technology and economics, etc. Therefore, it is extremely meaningful to study of these problems. There are numerous approaches to obtain optimization roots for instance: the secant method, gradient method, Newton-Raphson algorithm, etc where the repetitive Newton-Raphson algorithm is one of the most widespread executed. About this regard, there are several scholars had researched and performed it. For example: Riks [3] presented the application of Newton's method to the problem of elastic stability, Broyden [4] introduced about quasi-Newton methods and their application to function mini-misation, Polyak [5] researched about the N-R method and its application in optimization, Chalco et al. [6] presented about on the Newton method for solving fuzzy optimization problems. Bakari et al. [7] introduced the application of Newton-Raphson method to non-linear models, Wu et al. [8] researched about a regularized Newton method for computing ground states of Bose-Einstein condensates, Chin et al. [9] presented an efficient alternating newton method for learning factorization machines, Ferreira et al. [10] introduced about inexact Newton method for nonlinear functions with values in a cone, Mokhtari et al. [11] researched about IQN: An incremental quasi-Newton method with local superlinear convergence rate, etc.

Furthermore, there are several available functions in R to obtain optimization roots, for example: the maxLik function is first introduced by Henningsen et al. [12]. Nash [13] developed an optim function. This function is derived from the algorithm of Zhu et al. [14]. Hasselman [15] proposed about Nleqslv function, etc. Although the problem about finding the optimization solution of estimating functions has been extensively studied and widely applied in various fields, the issue about comparing the performance of these approaches has not been yet researched. To circumvent the difficulty, in this paper, our main objective is in comparing the performance of two approaches: Newton-Raphson algorithm and maxLik function. In this study, we are interested in the following issues: basically, the maxLik function is an available function in the statistical software R, so it is convenient to use, but can maxLik function be executed if our data set contain missing values? When using two approaches to obtain the optimization root for the same model, are the results of two approaches similar? Time to run code to output the results from two approaches, which the approach is faster?

The rest of the paper is structured as follows. We present the procedure and examples of two approaches in Section 2. In Section 3, we review of three widespread methods to handle with missing data to compare the performance of two methods if our data set contain missing values. In Section 4, comparing the results and the time to run code to output the result of two approaches through an simulation study is introduced in [1]. Eventually, some concluding remarks and work extension are stated in the last section.

2. The procedure and examples

We introduce the procedure and examples of two approaches: Newton-Raphson algorithm and Maxlik function to obtain the optimization root to estimating functions. We present the Maxlik function owing to the fact that it is an available widespread function to get the optimization root and it does not use the Hessian matrix in formula. In numerical analysis, the Newton-Raphson is an ubiquitous repetitive algorithm to get roots to a target function $g(u)$ (solutions of $g(u) = 0$). In Statistics and optimization, the N-R algorithm is one of the most widespread performed algorithms to find roots of the derivative of function $g(u)$ (solutions of $g'(u) = 0$). Our main objective is comparing the performance of two approaches to obtain the optimization root. Hence, we do not introduce about how to set up algorithms in detail. We only present formulas and some examples of two approaches.

2.1. Newton-Raphson algorithm

(a) Case 1: One-dimension

Let $g(u)$ be a target function need to be found its roots by performing the Newton-Raphson algorithm. The expression root of the N-R algorithm can be described as follows:

$$u_{n+1} = u_n - \frac{g(u_n)}{g'(u_n)} \tag{1}$$

To obtain optimization roots by executing the Newton-Raphson algorithm. Usually we reiterate the expression (1) until the difference of two adjacent roots is smaller than γ (where γ a very small value provided).
Example 1: Performing the Newton-Raphson algorithm to obtain the root of the following equation: $3u^3+7u-9 = 0$. Utilizing an initial value $u_0 = 1$. Executing three iterations.
Solution:
Let $g(u) = 3u^3+7u-9$. It can be seen that $g(0)g(1) = -9 < 0$. Therefore $g(u) = 0$ has root in the $(0, 1)$ interval, $g'(u) = 9u^2 + 7$.

 © 2018 Journal of Advanced Engineering and Computation (JAEC)

We construct an expression $\{u_n\}$:

$$u_{n+1} = u_n - \frac{g(u_n)}{g'(u_n)} = u_n - \frac{3u_n^3 + 7u_n - 9}{9u_n^2 + 7}$$

With $u_0 = 1$, we have

$$\begin{aligned} u_1 &= 0.937500, \quad f(u_1) = 0.0344238 \\ u_2 &= 0.935191, \quad f(u_2) = 0.0000449 \\ u_3 &= 0.935188, \quad f(u_3) = 0 \end{aligned}$$

Remark: if the target function is $g'(u)$ then the expression (1) can be rewritten:

$$u_{n+1} = u_n - \frac{g'(u_n)}{g''(u_n)} \tag{2}$$

(b) Case 2: Multi-dimension

The expression (2) can be extrapolated to the N-R algorithm in numerous dimensions by substituting the derivative of the target function by a gradient, $\nabla g(u)$, and substituting the reciprocal of the second derivative by the inverse of the Hessian matrix $Hg(u)$. The expression root of the N-R algorithm in multi-dimension can be then illustrated by:

$$u^{(n+1)} = u^{(n)} - \left[Hg(u^{(n)})\right]^{-1} \nabla g\left(u^{(n)}\right)$$

where

$$\nabla g(u) = \left[\frac{\partial g(u)}{\partial u_1}; \frac{\partial g(u)}{\partial u_2}; \ldots; \frac{\partial g(u)}{\partial u_n}\right]^T$$

and

$$Hg(u) = \begin{bmatrix} \frac{\partial^2 g(u)}{\partial u_1^2} & \frac{\partial^2 g(u)}{\partial u_1 \partial u_2} & \cdots & \frac{\partial^2 g(u)}{\partial u_1 \partial u_n} \\ \frac{\partial^2 g(u)}{\partial u_2 \partial u_1} & \frac{\partial^2 g(u)}{\partial u_2^2} & \cdots & \frac{\partial^2 g(u)}{\partial u_2 \partial u_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 g(u)}{\partial u_n \partial u_1} & \frac{\partial^2 g(u)}{\partial u_n \partial u_2} & \cdots & \frac{\partial^2 g(u)}{\partial u_n^2} \end{bmatrix}$$

Likewise in one dimension, to get the optimization root by the Newton-Raphson algorithm in multi-dimension. We need to reiterate the above expression until a sufficiently accurate value is reached.

Example 2: Performing the N-R algorithm to minimize the following function:

$$g(u_1, u_2, u_3, u_4) = (u_1 + 10u_2)^2 + \\ + 5(u_3 - u_4)^2 + (u_2 - 2u_3)^4 + 10(u_1 - u_4)^4$$

Choosing as an initial value

$$u^{(0)} = [1; -0.5; 0.5; 0.5]^T$$

Executing four iterations.
We have,

$$g(u^{(0)}) = 21.6875$$

and,

$$\nabla g\,(u) = \left[\frac{\partial g}{\partial u_1}; \quad \frac{\partial g}{\partial u_2}; \quad \frac{\partial g}{\partial u_3}; \quad \frac{\partial g}{\partial u_4}\right]^T = \begin{bmatrix} 2(u_1 + 10u_2) + 40(u_1 - u_4)^3 \\ 20(u_1 + 10u_2) + 4(u_2 - 2u_3)^3 \\ 10(u_3 - u_4) - 8(u_2 - 2u_3)^3 \\ -10(u_3 - u_4) - 40(u_1 - u_4)^3 \end{bmatrix}$$

Let

$$Hg(u) = [Hg_1(u); Hg_2(u); Hg_3(u); Hg_4(u)]$$

where $Hg_1(u); \ldots; Hg_4(u)$ be the first, second, third and fourth column of $Hg(u)$. then

$$Hg_1(u) = \begin{bmatrix} 2 + 120(u_1 - u_4)^2 \\ 20 \\ 0 \\ -120(u_1 - u_4)^2 \end{bmatrix}$$

$$Hg_2(u) = \begin{bmatrix} 20 \\ 200 + 12(u_2 - 2u_3)^2 \\ -24(u_2 - 2u_3)^2 \\ 0 \end{bmatrix}$$

$$Hg_3(u) = \begin{bmatrix} 0 \\ -24(u_2 - 2u_3)^2 \\ 10 + 48(u_2 - 2u_3)^2 \\ -10 \end{bmatrix}$$

$$Hg_4(u) = \begin{bmatrix} -120(u_1 - u_4)^2 \\ 0 \\ -10 \\ 10 + 120(u_1 - u_4)^2 \end{bmatrix}$$

Iteration 1

$$\nabla g(u^{(0)}) = \left[-3; -93.5; 27; -5\right]^T,$$

$$Hg(u^{(0)}) = \begin{bmatrix} 32 & 20 & 0 & -30 \\ 20 & 227 & -54 & 0 \\ 0 & -54 & 118 & -10 \\ -30 & 0 & -10 & 40 \end{bmatrix},$$

$$\left[Hg(u^{(0)})\right]^{-1} = \begin{bmatrix} 130 & -10 & 3 & 98 \\ -10 & 5 & 2 & -7 \\ 3 & 2 & 9 & 5 \\ 98 & -7 & 5 & 100 \end{bmatrix} * 10^{-3}.$$

Therefore

$$u^{(1)} = u^{(0)} - \left[Hg(u^{(0)})\right]^{-1} \nabla g(u^{(0)})$$
$$= [0.7937; -0.0794; 0.4603; 0.4603]^T,$$
$$g\left(u^{(1)}\right) = 1.1234.$$

Iteration 2,

$$u^{(2)} = u^{(1)} - \left[Hg(u^{(1)})\right]^{-1} \nabla g(u^{(1)})$$
$$= [0.5291; -0.0529; 0.3069; 0.3069]^T,$$
$$g\left(u^{(2)}\right) = 0.2219.$$

Iteration 3,

$$u^{(3)} = u^{(2)} - \left[Hg(u^{(2)})\right]^{-1} \nabla g(u^{(2)})$$
$$= [0.3527; -0.0353; 0.2046; 0.2046]^T,$$
$$g\left(u^{(3)}\right) = 0.0438.$$

Iteration 4,

$$u^{(4)} = u^{(3)} - \left[Hg(u^{(3)})\right]^{-1} \nabla g(u^{(3)})$$
$$= [0.2352; -0.0235; 0.1364; 0.1364]^T,$$
$$g\left(u^{(4)}\right) = 0.0086.$$

2.2. MaxLik function

The maxLik function is an ubiquitous available function to obtain the optimization root. This function is first introduced by Henningsen et al. [12]. Similarly, the other R packages, the maxLik package needs to be installed and loaded before using. The command to install and load maxLik function are as follows:

```
>install.packages("maxLik")
>library(maxLik)
```

The simplest formula of the maxLik function is: maxLik*(logLik, start)*, where *logLik* is the log-likelihood function of a target function, *start* is a starting value of parameters need to be estimated, which can get the real value or vector.

Its whole formula as follows: maxLik(*logLik*, *grad* = NULL, *hess* = NULL, *start*, *method*, *constraints*=NULL, ...). In which *grad* is a gradient of an objective function. If NULL, numerical gradient will be performed, *hess* is a hessian matrix of an objective function. If NULL, numerical Hessian will be executed, *method*: we can select "NM" (Nelder-Mead), "CG" (Conjugate Gradients), "BFGS" (Broyden-Fletcher-Goldfarb-Shanno), etc, *constraints*: if we can select NULL for unconstrained maximization. It can be seen that: To apply a maxLik function, we need to have an objective function (a log-likelihood function) and an initial value.

Next, we investigate an example in article of Henningsen et al. [12].
Example 3: Assuming that x is generated from a standard normal distribution and the sample size is $(n = 500)$. We need to estimate parameters of the standard normal distribution is derived from this data set.

The log-likelihood function can be written as follows:

$$\ln\left(L(x;\mu,\sigma)\right) = -\frac{N\ln(2\pi)}{2} + \\ - N\ln(\sigma) - \sum_{i=1}^{n} \frac{(x_i-\mu)^2}{2\sigma^2}$$

Choosing a starting value is (1,2) vector and execute the statistical software R to write code:

```
>x = rnorm(500, mean = 0, sd = 1)
>logLikFun = function(parameter){
    mu = parameter[1]
    sigma = parameter[2]
    sum(dnorm(x, mean = mu,
                   sd = sigma,
                log = TRUE))}
>mle = maxLik(logLik = logLikFun,
      start = c(mu = 1, sigma = 2))
```

To see the full results of mle, executing the following command: *summary(mle)*
For simplicity's sake, let *coef* be parameters need to be estimated and *stdEr* be standard errors, we can use *coef(mle)* and *stdEr(mle)* to see the result of the estimated parameter and its standard error.

```
> summary(mle)
> coef(mle)
     mu            sigma
     0.03542      1.02518
> stdEr(mle)
```


© 2018 Journal of Advanced Engineering and Computation (JAEC)

```
mu          sigma
0.04585    0.03242
```

3. Comparison of the performance of two approaches when data set having missing values

For comparison of the performance of two approaches: Newton-Raphson algorithm and maxLik function, it is necessary to review of three widespread approaches to handle the issues having missing data, they are: complete-case (CC), inverse probability weighting (IPW) and joint conditional likelihood (JCL) estimator.

3.1. Complete cases (CC) estimator

Assuming that our data set having missing values. The complete case (CC) estimator is only performed on the data set no missing values, while the data set having missing values will be removed. As a result, the sample size in our data set will be reduced significantly. This issue will seriously affect the results in researches. Let η_i be a missing-ness status of X_i i.e. $\eta_i = 1$ when X_i is observed and $\eta_i = 0$ otherwise. Let T be a surrogate variable of X and T is independent of Y given (X, Z). The validation data set $(\eta_i = 1)$ includes (Y_i, X_i, V_i) and non-validation data set $(\eta_i = 0)$ includes (Y_i, V_i), with $V_i = (Z_i^T, T_i^T)^T$. The general estimating function by CC estimators of the regression model parameters when covariates are missing at random (MAR), denoted by $U_{CC,n}(\alpha)$, can be represented as follows:

$$U_{CC,n}(\alpha) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \eta_i S_i(\alpha) \qquad (3)$$

where α are interested parameters and $S_i(\alpha)$ is the first derivative of the log-likelihood function of $P(Y_i = 1|X_i, V_i)$ with respect to α.
By figuring out $U_{CC,n}(\alpha) = 0$ can be acquired $\hat{\alpha}_{CC}$ which is an estimator of α.

It has been seen that, $E[U_{CC,n}(\alpha)] \neq 0$. so it is called a biased estimating function. Wang et al. [16] and Lukusa et al. [17] also stated that the complete-cases estimator is not a trustworthy approach.

3.2. Inverse probability weighting (IPW) estimator

The inverse probability weighting (IPW) estimator is an improvement to complete case (CC) estimator. Zhao and Lipsitz [18] proposed an IPW estimator. Basically how this approach works as the complete case estimator, i.e. it only considers and works on data set with no missing values. Nevertheless this approach is based on wighting observations. The authors have shown that this is a reliable method. Below MAR mechanism. Assuming that $\pi(Y_i, V_i) = P(\eta_i = 1|Y_i, X_i, V_i)$ is a selection probability of covariates X_i. The general its formula can be illustrated as follows:

$$U_{W,n}(\alpha, \pi) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \frac{\eta_i}{\pi(Y_i, V_i)} S_i(\alpha) \qquad (4)$$

where α are interested parameters and $S_i(\alpha)$ is the first derivative of the log-likelihood function of $P(Y_i = 1|X_i, V_i)$ with respect to α.
Let $\hat{\alpha}_W$ be an estimator of α that can be acquired by figuring out $U_{W,n}(\alpha, \pi) = 0$.
In general, we have $E[U_{W,n}(\alpha, \pi)] = 0$. Therefore, $U_{W,n}(\alpha, \pi)$ is an unbiased estimating function.
In practice, $\pi(Y_i, V_i)$ is usually unknown and it is usually estimated by non-parametrically method [16]. If $\pi(Y_i, V_i)$ is a correctly estimated, $\hat{\alpha}_W$ will usually be a consistent estimator of α.

Let $v_1, v_2, \ldots, v_m$ be distinct values of the V_is. The non-parametric estimator of $\pi(y, v)$ is provided as follows:

$$\hat{\pi}(y, v) = \frac{\sum_{k=1}^{n} \eta_k I(Y_k = y, V_k = v)}{\sum_{i=1}^{n} I(Y_i = y, V_i = v)}, \qquad (5)$$

where I(A) is an indicator function of A, y are natural numbers and $v \in \{v_1, v_2, \ldots, v_m\}$.
The function (4) can be expressed as follows:

$$U_{Ws,n}(\alpha, \hat{\pi}) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \frac{\eta_i}{\hat{\pi}(y_i, v_i)} S_i(\alpha) \qquad (6)$$

3.3. Joint conditional likelihood (JCL) estimator

The joint conditional likelihood (JCL) estimator is first introduced by Wang et al. [19]. This approach is based on both the validation and non-validation data set. The general its formula can be described as follows:

$$U_{J,n}(\alpha, \pi) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} [\eta_i S_{1i}(\alpha) + (1 - \eta_i) S_{2i}(\alpha)] \tag{7}$$

Where α are interested parameters and $S_{1i}(\alpha)$ and $S_{2i}(\alpha)$ is the first derivative of the log-likelihood function of $P(Y_i = 1|X_i, V_i, \eta_i = 1)$ and $P(Y_i = 1|V_i, \eta_i = 0)$ with respect to α, respectively. The authors have shown that $E[U_{J,n}(\alpha, \pi)] = 0$ as a result, $U_{J,n}(\alpha, \pi)$ is an unbiased estimating function.

Missing data is an ubiquitous issue is usually encountered in, e.g., health, education and transportation, etc. This issue arises by numerous reasons, such as: respondents do not response to a certain item in the survey questions, nonacceptance to response, incomprehensible response, etc. [20]. The issues are associated with missing data can also be classified by 2 different types: missing outcome and missing covariates. The problems about estimate parameters in regression models with missing data have been extensively studied and widely applied in various fields by several scholars. for instance: Wang et al. [19] performed an JCL estimator to estimate parameters in logistic regression with missing covariates. This method aslo extended by Hsieh et al. [21] and Lee et al. [22] in their studies. All above authors performed a Newton-Raphson algorithm to estimate parameters in regression models with missing data. Lukusa et al. [17] introduced a semiparametric inverse probability weighting (SIPW) estimator and the authors also executed a Newton-Raphson algorithm to estimate parameters of a zero-inflated Poisson (ZIP) regression model with missing covariates. Diallo et al. [1] presented an IPW estimator and performed a maxLik function to estimate parameters in the zero-inflated Binomial (ZIB) model with missing covariates.

Derived from the general estimating function by CC and IPW estimator. It is possible to observe that two approaches only deliberate on validation data set ($\eta_i = 1$) but they do not deliberate on non-validation data set ($\eta_i = 0$). Consequently, two approaches can be executed to obtain the optimization root. Notwithstanding, both CC and IPW estimator are not a trustworthy approach.

The joint conditional likelihood (JCL) estimator is applied in regression models with missing data to estimate parameters which deliberates both validation and non-validation data set. The three articles: Wang et al. (2002), Hsieh et al. (2009) and Lee et al. (2012) have shown that the JCL estimator executes better than other approaches (CC and IPW). Owing to the fact that this approach deliberates both the validation data set ($\eta_i = 1$) and non-validation data set ($\eta_i = 0$), i.e. our data set contain missing values. Therefore, the maxLik function and some available functions in the statistical software can not perform in this situation.

It has been seen that, if the data set no missing values then the N-R algorithm and maxLik function can be executed to find the optimization root. Notwithstanding, in numerous applications, the data set often contain missing values. Therefore, maxLik function and some available functions in the statistical software are no longer acceptable in practice, the N-R algorithm still can execute in this situation. Although the algorithm of this approach is more unintelligible than some available functions in the statistical software, the N-R algorithm is regarded as a vigorous apparatus to find the optimization solution and parameters need to be estimated in researches. This method can be performed if the data set contain missing values that maxLik function and some available functions in the statistical software are unworkable. We compare the results of two approaches in the next section.

4. Comparing about the results of two methods

In this section, we compare about the results of two approaches: the Newton-Raphson algorithm

 © 2018 Journal of Advanced Engineering and Computation (JAEC)

and maxLik function, when both approaches are employed to estimate parameters to the same regression model (the zero-inflated binomial (ZIB) regression model). This model is introduced in article of Diallo et al. [1].

Example 4: Generating data set from the ZIB regression model:

$$\begin{aligned}\mathrm{logit}(\pi_{\mathrm{i}}) = {} & \alpha_1 X_{i1} + \alpha_2 X_{i2} + \alpha_3 X_{i3} \\ & + \alpha_4 X_{i4} + \alpha_5 X_{i5} + \alpha_6 X_{i6} + \alpha_7 X_{i7}\end{aligned}$$

and

$$\begin{aligned}\mathrm{logit}(\mathrm{p_i}) = {} & \beta_1 \mathrm{Z_{i1}} + \beta_2 Z_{i2} + \beta_3 Z_{i3} \\ & + \beta_4 Z_{i4} + \beta_5 Z_{i5},\end{aligned}$$

where $X_{i1} = 1$, $X_{i2} \sim \mathcal{N}(0,1)$, $X_{i3} \sim \mathcal{U}(2,5)$, $X_{i4} \sim \mathcal{N}(1,1.5)$, $X_{i5} \sim \mathcal{E}(1)$, $X_{i6} \sim \mathcal{B}(1,0.3)$ and $X_{i7} \sim \mathcal{N}(-1,1)$ are independently. Assuming that $Z_{i1} = 1$, $Z_{i4} \sim \mathcal{N}(-1,1)$ and $Z_{i5} \sim \mathcal{B}(1,0.5)$ are independently. In this study, we assume that $Z_{i2} = X_{i2}$ and $Z_{i3} = X_{i6}$ and choosing initial values as follows:

$$\alpha = (-0.3, 1.2, 0.5, -0.75, -1, 0.8, 0)^T$$

and

$$\beta = (-0.55, -0.7, -1, 0.45, 0)^T$$

Investigating numerous sample sizes ($n = 150, 300, 500$) and $h_i \in \{4,5,6\}$. The numbers h_i are allowed to change across subjects.
Let

$$\begin{aligned}(k_4, k_5, k_6) = {} & (card\{i : h_i = 4\}, \\ & card\{i : h_i = 5\}, card\{i : h_i = 6\})\end{aligned}$$

With $n = 150$, using $(k_4, k_5, k_6) = (60, 50, 40)$. When $n = 300$, performing $(k_4, k_5, k_6) = (120, 100, 80)$ and with $n = 500$, choosing $(k_4, k_5, k_6) = (200, 170, 130)$.

Utilizing above values, the average proportion of zero-inflation in our data set is 25%. The number of repetitions in simulation is chosen $N = 5000$ times and figure out the maximum likelihood estimation (MLE) $\hat{\gamma}_n = \left(\hat{\alpha}_n^T, \hat{\beta}_n^T\right)^T$.

In this study, we execute two approaches: the Newton-Raphson method and maxLik function to estimate parameters. These results are provided in Tab. 1 and Tab. 2, respectively (in Appendix). It can seen be that, the biases of estimators are very small, the values of SD and ASE are very close and the values of CP are very close to 0.95. These prove that our estimated results are very trustworthy. In addition, it has been seen that the bias, SE, SD, and l(CI) of all estimators decrease as the sample size increases. Furthermore, it can be seen that the normal Q - Q plots are provided in Figs. 1-4 (in Appendix) that the Gaussian approximation of the distribution of the MLE in the zero-inflated Binomial (ZIB) regression model is reasonably satisfied.

About the results, the authors in article of Diallo et al. (2017) have executed a maxLik function to study simulation. The results in this paper is performed by utilizing two approaches: maxLik function and N-R method. It can be observed from the above results of two approaches most are the same. We employed the HP desktop computer is configured with Intel Core i5, 8GB of RAM, 1TB of hard drive to check the time to run code to output the result of two approaches. To obtain the above results, it takes 60 minutes for maxLik function while the Newton-Raphson method is only 30 minutes. Thus the Newton-Raphson algorithm provides the results is faster than the maxLik function.

5. Concluding Remarks and Inference

It can be observed that, in general, maxLik function and some available functions in the statistical software only can be performed to get the optimization root in case of the data set with no missing values. Furthermore, its structure is easier than the N-R algorithm. Notwithstanding, the N-R algorithm is a robust apparatus to get the optimization root and to estimate parameters in regression models. It can be executed if our data set contain missing values that some available functions in the statistical software are unworkable. These functions only can perform if our data set with no missing values (the validation data set ($\eta = 1$), they can not execute in case of the non-validation data set ($\eta = 0$)). In the meantime, the Newton-Raphson algorithm can employ in all situations.

For the results, the authors in article of Diallo et al. (2017) have performed a maxLik function to study simulation. The results in this article is employed by utilizing two approaches: maxLik function and N-R algorithm. It can be observed from the results of two approaches in Section 4 most are the same. For the time to run code to output these results of two approaches. It takes 60 minutes for maxLik function while the Newton-Raphson algorithm is only 30 minutes. Thus the Newton-Raphson algorithm provides the results is faster than the maxLik function.

About the extension of the Newton-Raphson algorithm, we can execute this algorithm to estimate parameters to regression models with missing covariates data, for example: the zero-inflated power series (ZIPS) regression models, zero-inflated generalized Poisson (ZIGP) regression models, zero-inflated negative binomial (ZINB) regression models, or multivariate zero-inflated regression models, etc. All the structures can look at in Lukusa et al. [23] and the authors also stated that all of these models have not yet been researched with missing covariates data. These are interesting research directions in the hereafter.

References

[1] Diallo, A. O., Diop, A., & Dupuy, J.-F. (2017). Asymptotic properties of the maximum-likelihood estimator in zero-inflated binomial regression. Communications in Statistics-Theory and Methods, 46(20), 9930–9948.

[2] Godambe, V. (1991). Estimating Functions. Oxford Statistical Science Series. Oxford University Press.

[3] Riks, E. (1972). The application of Newton's method to the problem of elastic stability. Journal of Applied Mechanics, 39(4), 1060–1065.

[4] Broyden, C. G. (1967). Quasi-Newton methods and their application to function minimisation. Mathematics of Computation, 21(99), 368–381.

[5] Polyak, B. T. (2007). Newton's method and its use in optimization. European Journal of Operational Research, 181(3), 1086–1096.

[6] Chalco-Cano, Y., Silva, G. N., & Rufián-Lizana, A. (2015). On the Newton method for solving fuzzy optimization problems. Fuzzy Sets and Systems, 272, 60–69.

[7] Bakari H.R, A. T., & A.M, Y. (2016). Application of Newton-Raphson method to non-linear models. International Journal of Mathematics and Statistics Studies, 4(11), 21–31.

[8] Wu, X., Wen, Z., & Bao, W. (2017). A regularized Newton method for computing ground states of Bose–Einstein condensates. Journal of Scientific Computing, 73(1), 303–329.

[9] Chin, W.-S., Yuan, B.-W., Yang, M.-Y., & Lin, C.-J. (2018). An efficient alternating newton method for learning factorization machines. ACM Transactions on Intelligent Systems and Technology (TIST), 9(6), 72.

[10] Ferreira, O., & Silva, G. (2018). Inexact Newton method for non-linear functions with values in a cone. Applicable Analysis, 1–17.

[11] Mokhtari, A., Eisen, M., & Ribeiro, A. (2018). IQN: An incremental quasi-Newton method with local superlinear convergence rate. SIAM Journal on Optimization, 28(2), 1670–1698.

[12] Henningsen, A., & Toomet, O. (2011). maxLik: A package for maximum likelihood estimation in R. Computational Statistics, 26(3), 443–458.

[13] Nash, J. C. (1990). Compact numerical methods for computers: linear algebra and function minimisation. CRC press.

[14] Zhu, C., Byrd, R. H., Lu, P., & Nocedal, J. (1997). Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. ACM Transactions on Mathematical Software (TOMS), 23(4), 550–560.

 © 2018 Journal of Advanced Engineering and Computation (JAEC)

[15] Hasselman, B. (2009). nleqslv: Solve systems of non linear equations. R package version, 1.

[16] Wang, C., Wang, S., Zhao, L.-P., & Ou, S.-T. (1997). Weighted semiparametric estimation in regression analysis with missing covariate data. Journal of the American Statistical Association, 92(438), 512–525.

[17] Lukusa, T. M., Lee, S.-M., & Li, C.-S. (2016). Semiparametric estimation of a zero-inflated Poisson regression model with missing covariates. Metrika, 79(4), 457–483.

[18] Zhao, L., & Lipsitz, S. (1992). Designs and analysis of two-stage studies. Statistics in medicine, 11(6), 769–782.

[19] Wang, C., Chen, J., Lee, S., & Ou, S. (2002). Joint conditional likelihood estimator in logistic regression with missing covariate data. Statistica Sinica, 555–574.

[20] Schafer, J. L., & Graham, J. W. (2002). Missing data: our view of the state of the art. Psychological methods, 7(2), 147.

[21] Hsieh, S.-H., Lee, S.-M., & Shen, P. (2009). Semiparametric analysis of randomized response data with missing covariates in logistic regression. Computational Statistics & Data Analysis, 53(7), 2673–2692.

[22] Lee, S.-M., Li, C.-S., Hsieh, S.-H., & Huang, L.-H. (2012). Semiparametric estimation of logistic regression model with missing covariates and outcome. Metrika, 75(5), 621–653.

[23] T. Martin Lukusa, S.-M. L., & Li, C.-S. (2017). Review of Zero-Inflated Models with Missing Data. Current Research in Biostatistics, 7, 1–12.

About Authors

Kim-Hung PHO is a Ph.D. Student in Applied Statistics in Feng Chia University, Taiwan. In 2014, he became a Lecturer in Faculty of Mathematics-Statistics in Ton-Duc-Thang university, Ho Chi Minh City, Vietnam. His current research interests include Regression models with missing data, Randomized Response Technique, Copula, Mathematics education models and Mathematics in Finance.

Vu-Thanh NGUYEN is a Master Student in Applied Statistics in Faculty of Mathematics and Statistics, University of Natural Sciences, Ho Chi Minh City, Vietnam. His current research interests include Copula and Mathematics in Finance.

"This is an Open Access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited (CC BY 4.0)."

Tab. 1: The results by performing Newton-Raphson algorithm

The results of estimator of α

n		$\hat{\alpha}_{1,n}$	$\hat{\alpha}_{2,n}$	$\hat{\alpha}_{3,n}$	$\hat{\alpha}_{4,n}$	$\hat{\alpha}_{5,n}$	$\hat{\alpha}_{6,n}$	$\hat{\alpha}_{7,n}$
150								
	bias	-0.0118	0.0287	0.0125	-0.0235	-0.0323	0.0281	0.0010
	SD	0.5676	0.1674	0.1425	0.1019	0.1635	0.2743	0.1263
	ASE	0.5552	0.1642	0.1443	0.0996	0.1598	0.2715	0.1242
	CP	0.9458	0.9419	0.9427	0.9449	0.9466	0.9485	0.9467
	l(CI)	2.1652	0.6369	0.5689	0.3873	0.6228	1.0575	0.4849
300								
	bias	-0.0112	0.0145	0.0065	-0.0109	-0.0145	0.0088	-0.0017
	SD	0.3881	0.1136	0.1005	0.0705	0.1116	0.1871	0.0856
	ASE	0.3799	0.1127	0.0993	0.0689	0.1098	0.1857	0.0859
	CP	0.9473	0.9497	0.9489	0.9508	0.9427	0.9458	0.9429
	l(CI)	1.4856	0.4417	0.3916	0.2668	0.4328	0.7181	0.3307
500								
	bias	0.0013	0.0105	0.0031	-0.0061	-0.0097	0.0085	-0.0017
	SD	0.2925	0.0854	0.0776	0.0519	0.0823	0.1425	0.0668
	ASE	0.2922	0.0851	0.0769	0.0525	0.0837	0.1419	0.0652
	CP	0.9511	0.9496	0.9481	0.9499	0.9495	0.9438	0.9520
	l(CI)	1.1398	0.3356	0.2995	0.2045	0.3197	0.5493	0.2541

The results of estimator of β

n		$\hat{\beta}_{1,n}$	$\hat{\beta}_{2,n}$	$\hat{\beta}_{3,n}$	$\hat{\beta}_{4,n}$	$\hat{\beta}_{5,n}$
150						
	bias	-0.0759	-0.0507	-0.0735	0.0513	0.0079
	SD	0.5202	0.3576	0.7386	0.3156	0.5809
	ASE	0.5035	0.3483	0.7723	0.3078	0.5778
	CP	0.9603	0.9529	0.9643	0.9573	0.9581
	l(CI)	1.9368	1.3243	2.8927	1.1751	2.2243
300						
	bias	-0.0352	-0.0205	-0.0731	0.0243	0.0121
	SD	0.3401	0.2281	0.5012	0.2067	0.3807
	ASE	0.3353	0.2249	0.4952	0.2016	0.3799
	CP	0.9508	0.9512	0.9591	0.9508	0.9573
	l(CI)	1.2811	0.8792	1.8905	0.7823	1.4827
500						
	bias	-0.0170	-0.0096	-0.0397	0.0151	0.0015
	SD	0.2497	0.1723	0.3681	0.1552	0.2914
	ASE	0.2491	0.1701	0.3646	0.1525	0.2862
	CP	0.9501	0.9458	0.9509	0.9515	0.9512
	l(CI)	0.9767	0.6659	1.4272	0.5893	1.1219

SD: empirical standard deviation, ASE: asymptotic standard error.
CP: Empirical confidence intervals at 95% level and l(CI): the average of the length of the confidence intervals. All of results are executed the number of repetitions N = 5000 times.

 © 2018 Journal of Advanced Engineering and Computation (JAEC)

Tab. 2: The results by executing a maxLik function

The results of estimator of α

n		$\hat{\alpha}_{1,n}$	$\hat{\alpha}_{2,n}$	$\hat{\alpha}_{3,n}$	$\hat{\alpha}_{4,n}$	$\hat{\alpha}_{5,n}$	$\hat{\alpha}_{6,n}$	$\hat{\alpha}_{7,n}$
150								
	bias	-0.0119	0.0287	0.0126	-0.0238	-0.0315	0.0273	0.0089
	SD	0.5677	0.1712	0.1397	0.1021	0.1619	0.2723	0.1280
	ASE	0.5649	0.1707	0.1418	0.0999	0.1597	0.2708	0.1269
	CP	0.9498	0.9439	0.9485	0.9488	0.9489	0.9479	0.9463
	l(CI)	2.1743	0.6357	0.6013	0.3728	0.6419	1.0685	0.4786
300								
	bias	-0.0058	0.0145	0.0068	-0.0109	-0.0132	0.0087	0.0013
	SD	0.3811	0.1137	0.0995	0.0711	0.1099	0.1869	0.0859
	ASE	0.3813	0.1132	0.0998	0.0705	0.1096	0.1861	0.0852
	CP	0.9479	0.9472	0.9428	0.9529	0.9497	0.9459	0.9508
	l(CI)	1.4952	0.4412	0.3971	0.2638	0.4427	0.7178	0.3369
500								
	bias	-0.0061	0.0104	0.0045	-0.0067	-0.0079	0.0070	-0.0011
	SD	0.2921	0.0889	0.0782	0.0529	0.0843	0.1412	0.0651
	SE	0.2928	0.0875	0.0773	0.0525	0.0847	0.1411	0.0659
	CP	0.9501	0.9517	0.9486	0.9469	0.9497	0.9471	0.9439
	l(CI)	1.1369	0.3358	0.2985	0.2051	0.3127	0.5447	0.2552

The results of estimator of β

n		$\hat{\beta}_{1,n}$	$\hat{\beta}_{2,n}$	$\hat{\beta}_{3,n}$	$\hat{\beta}_{4,n}$	$\hat{\beta}_{5,n}$
150						
	bias	-0.0755	-0.0518	-0.0725	0.0514	0.0079
	SD	0.5209	0.3590	0.7339	0.3143	0.5871
	ASE	0.5072	0.3498	0.7298	0.3098	0.5799
	CP	0.9502	0.9515	0.9603	0.9571	0.9518
	l(CI)	1.9423	1.3258	2.9026	1.1695	2.2172
300						
	bias	-0.0308	-0.0212	-0.0475	0.0209	-0.0047
	SD	0.3398	0.2316	0.5019	0.2045	0.3818
	ASE	0.3395	0.2328	0.5712	0.2052	0.3783
	CP	0.9471	0.9525	0.9593	0.9505	0.9489
	l(CI)	1.2712	0.8895	1.8013	0.7815	1.4908
500						
	bias	-0.0179	-0.0145	-0.0457	0.0149	0.0013
	SD	0.2492	0.1748	0.3302	0.1528	0.2939
	ASE	0.2497	0.1745	0.3309	0.1521	0.2932
	CP	0.9541	0.9479	0.9532	0.9519	0.9512
	l(CI)	1.0021	0.6655	1.5272	0.5992	1.1912

SD: empirical standard deviation, ASE: asymptotic standard error.
CP: Empirical confidence intervals at 95% level and l(CI): the average of the length of the confidence intervals. All of results are executed the number of repetitions N = 5000 times.

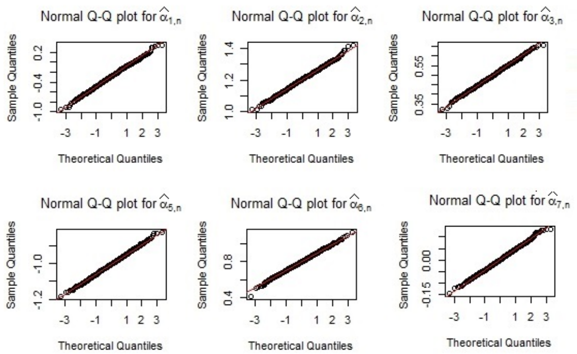

Fig. 1: Normal Q - Q plots for $\widehat{\alpha}_{1,n}, \ldots, \widehat{\alpha}_{7,n}$ with results are obtained from the maxLik function (n = 500)

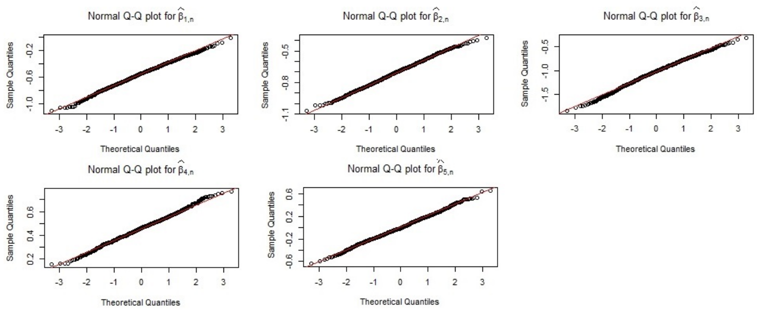

Fig. 2: Normal Q - Q plots for $\widehat{\beta}_{1,n}, \ldots, \widehat{\beta}_{5,n}$ with results are achieved by the maxLik function (n = 500)

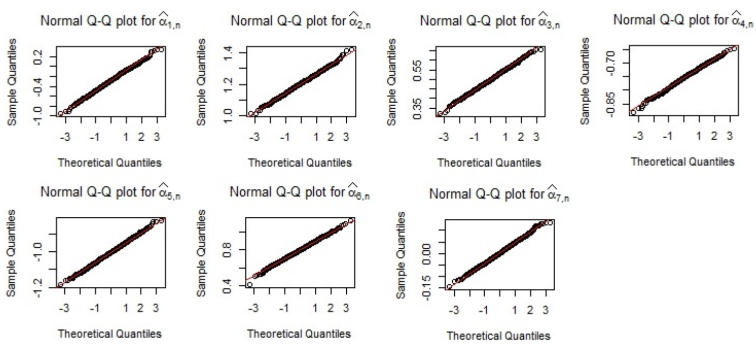

Fig. 3: Normal Q - Q plots for $\widehat{\alpha}_{1,n}, \ldots, \widehat{\alpha}_{7,n}$ with results are attained by the N-R algorithm (n = 500)

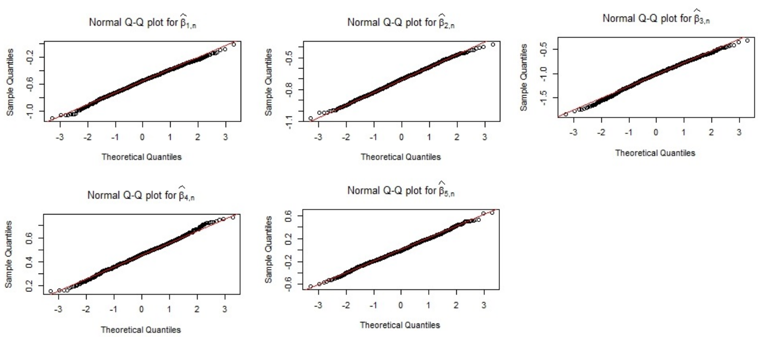

Fig. 4: Normal Q - Q plots for $\widehat{\beta}_{1,n}, \ldots, \widehat{\beta}_{5,n}$ with results are gained from the N-R algorithm (n = 500)

© 2018 Journal of Advanced Engineering and Computation (JAEC)