



An Introductory Study on “How the Genetic Algorithm Works in the Parameter Estimation of Binary Logit Model?”

Emre Demir^a, Özge Akkuş^{b*}

^a Hitit University, Department of Technical Programs, Osmancık Ömer Derindere Voc. School, Çorum, Turkey

^b Muğla Sıtkı Koçman University, Department of Statistics, Faculty of Science, Muğla, 48000, Turkey

^aEmail: emredemir@hitit.edu.tr

^bEmail: ozge.akkus@mu.edu.tr

Abstract

In this study, we mainly dealt with the introduction and comparison of two optimization techniques over the estimated parameters and model results of the Binary Logistic Regression (LR) model. These are the traditional Newton-Raphson (NR) algorithm which requires the differentiable objective function and appropriate starting values related to the parameters and Genetic Algorithm (GA) approach which does not need any strict assumptions as the algorithm NR. The results suggest that NR algorithm and GA give very similar results when the assumptions of NR are satisfied. This indicates that GA could successfully be used instead of the NR method considering its more flexible assumptions. Moreover, if the objective function does not satisfy the differentiability condition, NR could not be used in the optimization process and fails to find the optimum values. Therefore, this study actually reveals the success of GA in the parameter estimation of LR model. All the model outputs are compared in terms of the estimated parameter values, ease of application and convergence rates. Matlab commands are also given with their explanations for GA for researchers studying in this area.

Keywords: Binary dependent variable; Genetic Algorithm; maximum likelihood estimator; Newton-Raphson algorithm.

* Corresponding author.

E-mail address: ozge.akkus@mu.edu.tr.

1. Introduction

Parameters in nonlinear regression models with categorical dependent variable could be estimated by the method of Maximum Likelihood Estimation (MLE). In this method, it is aimed to find the optimum parameter values that maximize the likelihood of observing of an individual. To do this, first order derivatives are taken on the model parameters. This obviously requires the differentiability of the objective function (likelihood function). Because the model parameters are not in a linear form after the derivation step, we could not obtain optimum parameter values, analytically. Therefore, lots of iterative methods such as Newton-Raphson, Gauss-Newton, Direct Search, Hooke-Jeeves, Nelder-Mead, Gradient, Steepest Descent, *etc.*, have been developed.

These algorithms are generally suitable for a certain problem. A great majority of them have some restrictive assumptions. That is, they require restrictive assumptions such as studying with continuous and differentiable objective function and determining the most appropriate starting points for the parameters that could yield the rapid convergence in iterative process. If a good starting point could not be determined for each parameter, the solution may get stuck on local optimums and only regional optimums could be found [1]. In addition to this, in the event that the rule of differentiability of the objective function could not be satisfied, this assumption validation, not surprisingly, will prevent applicability of the algorithms based on iteration and obtaining the best estimation of the model parameters. As is well known, the classical Newton-Raphson (NR) algorithm is the most popular one in this field.

Unlike the traditional optimization approaches based on iteration, Genetic Algorithm (GA) approach has some important properties such as not requiring a differentiable objective function and other supporting information. GA is also able to find global optimum points without being stuck with local optimums, successfully.

When studies on the parameter estimation of the nonlinear regression models are examined in literature, we could find studies investigating the success of GA when the dependent variable is continuous. However, in practice, especially in survey studies, one can often encounter situations where the dependent variable is discrete with two or more levels. In categorical dependent variable models, we work with a structure which is not linear with respect to its parameters, and thus the objective function can have many local maxima, there has been a need for developing an alternative method that can solve this problem. Thus, due to the restrictions of the iterative methods, we thought that GA could be a good alternative to analytical and other iterative methods in order to form the best parameter set that would maximize the log-likelihood function. We mainly focused on the comparison of two optimization techniques over the model results of the binary Logistic Regression (LR) [2].

In the next part of the study, we gave some information about the theory of the LR model, NR and GA approaches. Matlab commands of GA are introduced with their explanations, in detail. Subsequently, we held discussion on whether GA could be used instead of the traditional NR algorithm when all the requirements are satisfied for the method of NR. The last part is on the comparison and recommendation of the results obtained from both methods.

2. Methodology

The realization of this work supposes the availability of a great number of repetitions of samples responding to the same known theoretical model. In practice, as the theoretical model is unknown, we use the Monte-Carlo method based on the generation of the data by computer according to a fixed theoretical model.

2.1. Binary logistic regression models

Binary Logistic Regression (LR) model is used when the dependent variable has two cases (e.g. healthy-not healthy, medication is effective-ineffective). The aim of the LR model is to set up an acceptable model which can define the relation between the dependent and independent variables using the minimum number of variables in most compatible way. Additionally, LR is a method used in data classification based on independent variables' effects and in data assignment to specific classes in accordance with the rules of the probability [3, 4, 5].

The LR analysis is basically a regression analysis, but has three important differences from the linear regression model. That is,

- Dependent variable Y is numerical in the linear regression while it is categorical in the LR analysis.
- We obtain the expected values of the dependent variable in the linear regression while we obtain the probability of realization of an event related to the explanatory variables in LR analysis.
- While we study with the normally distributed dependent variable in order to control for the significance of the estimated parameters in linear regression, Bernoulli distribution is assumed for the dependent variable in LR analysis.

As mentioned above, in LR model, the categorical dependent variable (Y) has two cases: the occurrence of an event (Y=1) and non-occurrence of an event (Y=0). The probability of an event, $\pi(x_i)$, depending on the values of explanatory variables for observation 'I' could be estimated by the following equation.

$$\pi(x_i) = \frac{1}{1+e^{-(\beta_0+\beta_1 X_{1i}+\dots+\beta_p X_{pi})}} = \frac{e^{\beta_0+\beta_1 X_{1i}+\dots+\beta_p X_{pi}}}{1+e^{\beta_0+\beta_1 X_{1i}+\dots+\beta_p X_{pi}}} \quad (1)$$

In Eq.(1), β_0 is a constant term; $\beta_1, \beta_2, \dots, \beta_p$ represent regression coefficients and X_1, X_2, \dots, X_p are independent variables. The other expression of the model in Eq.(1) which is called "logit" is as following [3, 4, 5].

$$\ln \frac{\pi(x_i)}{1-\pi(x_i)} = \beta_0 + \beta_1 X_{1i} + \dots + \beta_p X_{pi} = X_i' \beta \quad (2)$$

We focus on how to estimate the model parameters in Eq.(2) in easier and efficient way. In the following sub-sections we could find some basic fundamentals on the traditional NR and alternative GA methods in the parameter estimation process.

2.2. The classical NR approach to the parameter estimation in LR model

In categorical dependent variable models, parameter estimations could be obtained by the method of Maximum Likelihood Estimation (MLE). In LR model, for the (x_i, y_i) pair, when $y_i = 1$, its contribution to the likelihood function is $\pi(x_i)$, and when $y_i = 0$, its contribution to the likelihood function is $1 - \pi(x_i)$. Accordingly, the (x_i, y_i) pair's contribution to the likelihood function is as in Eq.(3) [3, 4, 5].

$$\pi(x_i)^{y_i} [1 - \pi(x_i)]^{1-y_i} \tag{3}$$

Under the assumption that all the observations are independent, the Likelihood Function $l(\beta)$ obtained by multiplying the terms in Eq.(3) is as in Eq.(4).

$$l(\beta) = \prod_{i=1}^n \pi(x_i)^{y_i} [1 - \pi(x_i)]^{1-y_i} \tag{4}$$

The estimation of the β parameters which maximize $l(\beta)$ forms the basis of MLE. For convenience, when we take the logarithm of this statement, we obtain log-likelihood function, $L(\beta)$, given below.

$$L(\beta) = \ln[l(\beta)] = \sum_{i=1}^n \{y_i \ln [\pi(x_i)] + (1 - y_i) \ln [1 - \pi(x_i)]\} \tag{5}$$

When $L(\beta)$ is taken as a nonlinear objective function and β s are taken as decision variables, the problem can be thought an unrestricted nonlinear optimization problem. In order to find the estimation of β parameters that maximize the function $L(\beta)$, we should take the first order derivative with respect to β and equalize to zero, thus get the equation in Eq.(6).

$$\frac{\partial L(\beta)}{\partial \beta} = \sum_{i=1}^n [y_i - \pi(x_i)] x_{ij} = 0 \quad i = 1, \dots, n; \quad j = 1, \dots, p \tag{6}$$

Particularly, for LR model, let's substitute $\pi(x_i)$ with the statement given by Eq.(1) and get the open form of the equation in Eq.(5) and (6), respectively.

$$L(\beta) = \sum_{i=1}^n \left[y_i \ln \left(\frac{e^{(x_i^T \beta)}}{1 + e^{(x_i^T \beta)}} \right) + (1 - y_i) \ln \left(1 - \frac{e^{(x_i^T \beta)}}{1 + e^{(x_i^T \beta)}} \right) \right] \tag{7}$$

$$\frac{\partial L(\beta)}{\partial \beta} = \sum_{i=1}^n \left[y_i - \frac{e^{(x_i^T \beta)}}{1 + e^{(x_i^T \beta)}} \right] x_{ij} \tag{8}$$

The simultaneous solution of these p equations will give the likelihood estimates of the parameters. Because the likelihood equations given in Eq.(6) are not linear in β' , we need iterative methods such as NR in this step of the

estimation procedure. In the NR iteration process, we determine starting values in order to acquire an estimate of

$\hat{\beta}$ which maximizes the likelihood function. However, various alternatives have been suggested for these starting values. Newton and Raphson suggested that the least squares estimates should be used as base. We go on with the solution until the change in the likelihood function can be ignored. There are computer algorithms designed to make all these calculations in order to form the best parameter set that will maximize the likelihood function. Sometimes, the NR method can be stuck with many local maximum or minimum values and the iteration can be completed without reaching the global maximum or minimum points.

The iterative NR algorithm which finds $\hat{\beta}$ value in (m+1) iteration is written as:

$$\hat{\beta}^{(m+1)} = \hat{\beta}^{(m)} + (X^T V_M X)^{-1} X^T (Y - P_m) \tag{9}$$

Here, $P_m^T = (\pi_{1,m}, \pi_{2,m}, \dots, \pi_{n,m})$ is the likelihood vector and $V_m = \text{diag}[\pi_{i,m} (1 - \pi_{i,m})]$ is the diagonal matrix acquired from the previous iteration. The iteration goes on with the Iteratively Reweighted Least Square Method until $\hat{\beta}^{(m)}$ is close enough to $\hat{\beta}^{(m+1)}$. As convergence measure, for the difference between $\hat{\beta}^{(m+1)}$ and $\hat{\beta}^{(m)}$ values such as 1×10^{-8} ; 1×10^{-7} ; 1×10^{-6} (1e-6) are used [1, 3]. The correct selection of the starting values has a great effect on iteration number. We can find the optimum solution with the minimum number of iterations using a good starting value [6].

Because of the problems encountered in NR and restrictions of the iterative methods, GA approach to the parameter estimation could be seen as a good alternative. The heuristic methods give solutions that are very close to the exact solution and they do it in a very fast manner. The GA is one of the fairly common methods, and is used frequently especially in optimization of the non-linear functions. Details related to the method of GA are presented in the following sub-sections.

2.3. Genetic algorithm approach

The main principles of the Genetic Algorithms (GA) are based on Charles Darwin's idea of 'Survival of the Fittest' [7]. John Holland from Michigan University started his works on genetic algorithms in the early 1970s based on Charles Darwin's idea. These works pioneered fields such as evolution strategies, evolutionary programming, artificial intelligence, classification systems and genetic programming [8]. GA was also used as an artificial intelligence optimization technique with the contributions of Goldberg [9] and was used in solving the real life problems. Goldberg's book on GA's use in optimization problems, (1989) "Genetic Algorithms in Search, Optimization and Machine Learning", extended the genetic algorithms' application fields [9, 10].

GA starts with a set of solutions taken from a population which is constituted by chromosomes. These solutions are then used to create a new population. This process is done based on the assumption that the new population will be better than the old population. Solutions selected in order to create new solutions (generations) are determined based on the competence or appropriateness value. In GA's evolution process, the chromosome populations are run by GA's operators through the potential solution space. The operators used in GA process

are Copying, Crossover and Mutation operators. With the help of these operators, the GA aims to find the best solution by scanning the solution space.

Chromosomes are evaluated as good or bad according to their calculated appropriateness values for the population. By increasing the survival likelihoods of good individuals and dying likelihood of bad individuals, the GA aims to increase the population of good individuals while wiping out bad individuals. During the process of evolution, the chromosomes create a different population with the help of the GA's evolution operators and an iteration step is completed. When a certain number of iterations are completed or when the stopping condition is met, the solution coded by the chromosome with the best appropriateness value in the population is determined as the solution of the problem.

According to Goldberg [9], the GAs' differences from the traditional search methods can be explained as following [1, 7, 11].

- GA does not work with the parameters themselves, but with coding of the parameter set.
- GA does not use a single point as the search tool, but a point set. These decrease the risk of being stuck with the local best.
- GA does not need additional information such as differentiability condition during the search. It is sufficient for it to know the objective function.
- GA works with *stochastic* transition rules, not with *deterministic* transition rules [8, 12].

In the following sections, the introductory information and some new expressions related to the GA will be presented step by step [13, 14].

Coding a Representation Mechanism in GA: In GA, the commonly used mechanism for complex structures' representation is the binary bit string that uses (0,1) alphabet. Direct binary coding is the most commonly used coding technique. However, representations such as Uniform coding and Gray coding are possible in the binary system. In order to use the GA to solve a problem, first of all we must be able to state the problem's solution in chromosomes. The chromosomes represent the individuals forming a population and they are made up of gene strings. The chromosome representation consists of the coding of values of the parameters. The chromosomes are converted to binary to be appropriate for crossover and mutation [15, 16, 17, 18]. For example, the number 32 may be written as 100000 (1×2^5) in the binary system. While 100000 represents a chromosome, each 1 and 0 represents a gene that forms the chromosome. For example, the binary number 1101010 is equivalent in decimal system to 106.

$$1 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 106$$

Starting Population in GA: GA starts with a starting population consisting of individuals coded according to the solution cycle. Each chromosome in the starting population represents a possible solution of the problem.

Based on the starting population, GA makes it possible to create new populations that come close to the optimum solution in each iteration step. The number of starting populations of the best solution is different for each problem type. If the number of population is not sufficient, the search space cannot be searched effectively and this causes a lot of values that can be used in the algorithm to be skipped without being processed. A large number of populations, on the other hand, increase the workload and causes the algorithm to slow down. In the literature, it is accepted that the starting population should be determined with the randomness principle. In some studies, it is seen that the first population was produced based on previously acquired information.

In the traditional optimization methods, the search for solution starts from a certain point and it goes on from different points according to the results. However, depending on the starting point selection, problems such as not being able to find a solution or a prolonged process of solution may occur. But in GA, the search is carried out on a potential solution set and the solutions are evaluated until the best solution is found [15, 16, 17, 18].

Genetic Algorithm Operators: Forming of new populations and getting closer to the solution space are possible with the help of “Selection”, “Crossover” and “Mutation” operators, which are three main parameters of GA [15, 16, 17, 18].

Selection operator

A suitability value is assigned to each individual in the population. This value indicates that an individual is in the solution set. While some of the strings in the population are transmitted to the next generation, some of the strings disappear. Determination of the strings to be transmitted to the next generation is done by the selection mechanism. Selection process is the process of selecting a chromosome in the old population to be transmitted to the new generation, which will be formed with stochastic methods, according to the suitability value. This process makes it possible to create individual with higher suitability values in oncoming generations. Therefore, this process is referred as the “survival of the fittest test”. Selection methods are usually formed according to the survival of the fittest principle. The aim is to increase the number of the individuals with high suitability values in the next generation. The most commonly used selection operators are roulette wheel selection, tournament selection and truncation selection.

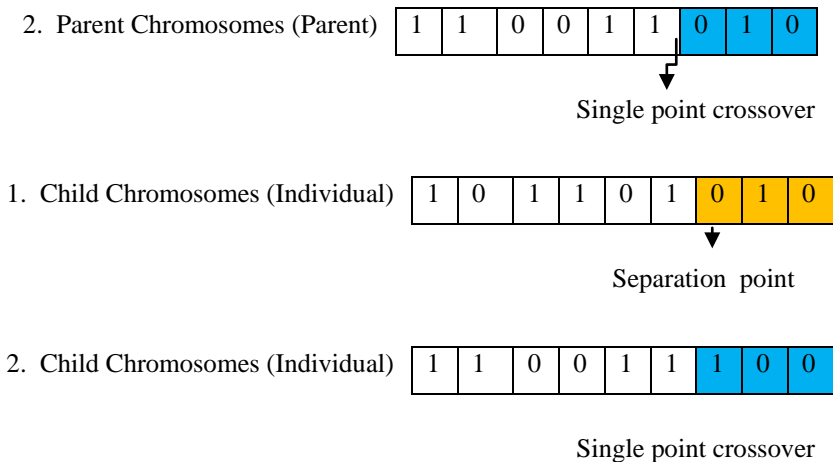
Crossover Operator

Crossover is the most important operator in GA and is responsible for production of new solutions. It is mostly done by swapping two randomly selected individual’s chromosomes. In this process, by swapping different parts of strings that forms individuals’ chromosomes, new individual are produced. This way, the newly created individual replaces the individual with the low suitability value, and the weak individual in the population are eliminated. An example of a crossover for unique point is given below.

1. Parent Chromosomes (Parent)

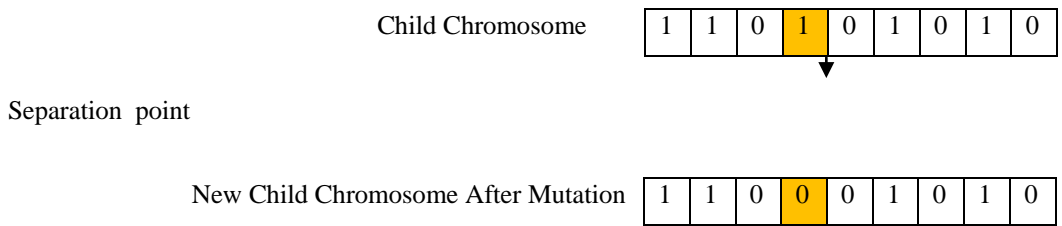
1	0	1	1	0	1	1	0	0
---	---	---	---	---	---	---	---	---

Separation point



Mutation Operator

Mutation is to randomly change one or more “genes” in a string that forms an individual’s chromosome according to their mutation probability. Mutation is the operator that prevents newly formed solutions from copying each other, thus speeding up the process. In this method, one can face with the problem of elimination of good individuals. To prevent this, a structure is formed that ensures the best individuals are to be transmitted to the next generation. An example of a mutation process is explained below.



A simple genetic algorithm process is explained over a numerical example in Appendix. The following section consists of the comparison of the LR model results obtained by NA and GA algorithms over a real data set on “Alopecia Diagnosis”.

3. Application

The application part of the study is on the comparison of two optimization method finding the optimal parameter estimations of LR model over a real data set. For this purpose first, the likelihood equations are obtained using the method of MLE. Then it has been seen that the analytical solutions of the parameter estimations could not be obtained by the simultaneous solution of the equations. Thus, both NR and GA methods are applied to get the LR model results. The results are compared in terms of the success of the methods in finding the optimum points, simplicity and applicability.

3.1. The Data Set

The data set used in the study has been taken from the Dermatology Unit of Çorum Training and Research Hospital of Hitit University. From patients who applied to Dermatology Unit with hair loss problem and

diagnosed with Alopecia in 2013, 52 were assigned to experimental group and 43 patients who were not diagnosed with alopecia were assigned to control group.

The binary dependent variable is the “Alopecia Diagnosis”. The diagnosed patients are coded as “1” and the non-diagnosed patients are coded as “0”. Variables affecting the Alopecia diagnosis and the necessary abbreviations are given in Table 1.

Table 1. Variables used in the analysis

Variable Definition		Variable Type
Y	Alopecia Disease (dependent variable)	Qualitative 1 = Diseased 0 = Nondiseased
X ₁	Age	Quantitative
X ₂	Gender	Qualitative 1 = Male 0 = Female
X ₃	ALT (Alanine Aminotransferase)	Quantitative
X ₄	AST (Aspartate Aminotransferase)	Quantitative
X ₅	Glucose	Quantitative
X ₆	HDL (High Density Lipoprotein)	Quantitative
X ₇	LDL (Low Density Lipoprotein)	Quantitative
X ₈ , X ₉ , X ₁₀	Cholesterol, Free t3 and Free t4 thyroids	Quantitative
X ₁₁	TSH (Thyroid Stimulating Hormone)	Quantitative
X ₁₂	Triglycerit (Tg)	Quantitative

In Table 1, ALT and AST are enzymes existent in liver cell. Cholesterol is a greasy element existent in circulation system. HDL which is a special type of molecular proteins generated in liver transports good cholesterol by blood. LDL which is a special type of molecular proteins generated in liver transports bad cholesterol by blood. Free t3 and Free t4 are thyroid gland hormone. TSH is a thyroid stimulating hormone and Triglycerit is a principle component of grease. Since the explanatory variable “Gender” has a categorical structure, “Male” is coded as “1” and “Female” is “0”.

3.2. NR results of the LR model

LR model parameters are estimated by the method of MLE in the optimal iteration number using the programme MATLAB. Primarily, we got the NR results given in Table 2.

The maximum value of the log-likelihood function is 52.8833. The optimal values of the statistically significant parameters (i.e. the roots of the log-likelihood function) are presented bold.

Table 2. The NR results in LR model

	$\hat{\beta}$	Standard Error	Wald Test	Degrees of Freedom	Sig.	Exp ($\hat{\beta}$)
Constant	2.985	3.165	0.889	1	0.346	19.782
Age	-0.025	0.024	1.029	1	0.310	0.976
Gender	1.618	0.619	6.840	1	0.009*	5.044
ALT	-0.051	0.038	1.761	1	0.185	0.951
AST	0.135	0.059	5.237	1	0.022*	1.145
Glucose	-0.004	0.010	0.162	1	0.687	0.996
HDL	-0.012	0.045	0.070	1	0.791	0.988
Cholesterol	0.011	0.038	0.083	1	0.774	1.011
LDL	-0.032	0.037	0.750	1	0.387	0.968
Free t3 thyroid	-1.611	0.777	4.306	1	0.038*	0.200
Free t4 thyroid	2.357	1.056	4.977	1	0.026*	10.557
TSH	-0.039	0.240	0.026	1	0.871	0.962
Trigliserit	0.006	0.008	0.423	1	0.515	1.006
Maximum value of the log-likelihood function: 52.8833						

*Coefficient is statistically significant at a significance level of 5%.

*Base category is Female for the variable Gender

3.2. GA Results of the LR Model

GA results are also obtained by MATLAB programming. There are two ways to optimize the objective function (here is the likelihood function) with GA. The first way is to use “the Optimization Toolbox” which does not require coding. The second way is to write the necessary commands in MATLAB’s “command window”. Parameter vector β which maximizes the log-likelihood function, optimization results and commands that we used in the parameter estimation are given below. By substituting patients’ independent variable values into Eq.(5), we get the following log-likelihood function.

$$f(\beta_0, \beta_1, \beta_2, \dots, \beta_{12}) = 52*x(1)+1481*x(2)+37*x(3)+1167*x(4)+1189*x(5)+4899*x(6)+2255.4*x(7)+8430*x(8)+4826.07*x(9)+178.45*x(10)+55.28*x(11)+103.67*x(12)+6695*x(13)-\log(1+\exp(x(1)+18*x(2)+x(3)+20*x(4)+17*x(5)+100*x(6)+39.1*x(7)+199*x(8)+76.7*x(9)+4.57*x(10)+1.31*$$

$$x(11)+2.5*x(12)+416*x(13))\log(1+\exp(x(1)+23*x(2)+14*x(4)+30*x(5)+92*x(6)+39.2*x(7)+143*x(8)+93.19$$

$$*x(9)+2.97*x(10)+0.79*x(11)+0.94*x(12)+53*x(13)))$$

...

$$\log(1+\exp(x(1)+25*x(2)+10.6*x(4)+16.5*x(5)+88*x(6)+41.3*x(7)+140*x(8)+89.09*x(9)+3.19*x(10)+0.62*x($$

$$11)+2.2*x(12)+136.7*x(13)))-\log(1+\exp(x(1)+25*x(2)+13*x(4)+17*x(5)+81*x(6)+69.7*x(7)$$

$$+202*x(8)+118.7*x(9)+3.01*x(10)+0.86*x(11)+0.19*x(12)+68*x(13))) \quad (10)$$

Because the program is not based on maximization but minimization, function $f(\beta_0, \beta_1, \beta_2, \dots, \beta_{12})$ must be multiplied by (-1) in order to find the parameter estimates which minimize the function.

The GA optimization command in MATLAB is generally expressed as the following [19].

$$[x,fval] = \text{ga}(@\text{likelihood},13,\text{options}) \quad (11)$$

Here, the right side of the equation is constituted by the input statements while the left side shows the output expressions. The inputs give the definition of the function ('ga' stands for the genetic algorithm) by entering variable count ('13') and definition of the operators ('likelihood') to be used in solution. Additionally, the starting population, the selection technique, mutation and crossover rates, and all the stopping criteria of the iterations are set with options code in this section.

The output functions indicate the results to be displayed after the solution. 'x' displays the independent variables that optimize the function and 'fval' displays the optimum value of the function. The necessary GA commands to maximize the log-likelihood function and their descriptions can be found in Appendix in detail.

In GA parameter estimation, in order to find the optimum values of 52.833533819034635, we set:

Population size	: '50';
Selection function	: 'Roulette';
Reproduction crossover function	: '0.95';
Mutation function	: 'Gaussian';
Mutation scale	: '0.15';
Mutation shrink	: '0.88';
Crossover function	: 'Intermediate';

Crossover ratio : '1.0';
Stopping value was set to be generations : '50000';
Stall generations : '50000';
Function tolerance : '1e-6';
Nonlinear constraint tolerance : '1e-6'.

Other operators and rates are taken as default. In the following section, some basic explanations related to the GA coding in MATLAB are presented.

3.3. MATLAB commands and definitions for GA

Some important GA commands in Matlab are given with their explanations as follows.

[x,fval,exitflag,output,population,scores]=GA(FITNESSFCN,NVARS,A,b,Aeq,beq,lb,ub,NONLCON,options);

The left side of the expression above consists of the output functions. The descriptions are given below.

([x, fval, exitflag, output, population, scores]);

X :Displays the variable values that minimize the function.

Fval :Displays the optimum value of the objective function.

Exitflag :Displays the integer value that shows the stopping reason of the algorithm.

When there are no linear constraints, this integer value is -5,-4,-2,-1, 0, 1, or 5. For example, the integer value '5' indicates that the iteration has been stopped because the time limit has been exceeded. Below is a brief description of all the integer values.

- (1) The optimum result and variable values have been found.
- (5) Set function accuracy constraint violation has been exceeded.
- (0) The number of generations has been exceeded.
- (-1) The optimization has been stopped by the output or drawing function.
- (-2) The appropriate point could not be found.
- (-4) Stop time limit has been exceeded.

(-5) Time limit has been exceeded.

Output : Gives an output containing information on the performance of the algorithms for each generation.

Rngstate : State of the random number generator

Generations : Total generation count

Funccount : Total function count

Maxconstraint : Maximum constraint violation

Message : Termination message

Population : Gives the output of populations created for the solution.

Scores : Prints the optimum value(s).

Descriptions of the input variables on the right side of the equation are as the following.

GA(FITNESSFCN,NVARS,A,b,Aeq,beq,lb,ub,NONLCON,options);

@fitnessfcn : Calls the objective function saved as m file.

Nvars : Indicates the number of variables in the objective function.

A : Unequal constraints matrix

B : Unequal constraints vector

Aeq : Equality constraints matrix

Beq : Equality constraints matrix

Lb : Lower limit

Ub : Upper limit

Nonlcon : Nonlinear constraint function

Options : Determines the structure of the options created by 'gaoptimset'.

Detailed usage of the code is given below.

options = gaoptimset('param1',value1,'param2',value2,...).

'gaoptimset' starts a solution search for parameter values put in quotes according to the values after coma. It is one of the most important commands in GA. The operations such as selection, crossover and mutation which we use to find the optimum solution are performed with 'options' code.

The function 'likelihood' is actually the log-likelihood function which we try to find the values that maximize this function and we take it as the objective function here.

function [a] = likelihood(x)

```
a=-(52*x(1)+1481*x(2)+37*x(3)+1167*x(4)+1189*x(5)+4899*x(6)+2255.4*x(7)
+8430*x(8)+4826.07*x(9)+178.45*x(10)+55.28*x(11)+103.67*x(12)+6695*x(13)-
log(1+exp(x(1)+18*x(2)+x(3)+20*x(4)+17*x(5)+100*x(6)+39.1*x(7)+199*x(8)+76.7*x(9)+4.57*x(10)+1.31 *
x(11)+2.5*x(12)+416*x(13)))+log(1+exp(x(1)+23*x(2)+14*x(4)+30*x(5)+92*x(6)+39.2*x(7)+143*x(8)+93.1
9*x(9)+2.97*x(10)+0.79*x(11)+0.94*x(12)+53*x(13)))+.....log(1+exp(x(1)+25*x(2)+10.6*x(4)+16.5*x(5)+
88*x(6)+41.3*x(7)+140*x(8)+89.09*x(9)+3.19*x(10)+0.62*x(11)+2.52*x(12)+136.7*x(13)))-
log(1+exp(x(1)+25*x(2)+13*x(4)+17*x(5)+81*x(6)+69.7*x(7)+202*x(8)+118.7*x(9)+3.01*x(10)+0.86*x(11)
+0.19*x(12)+68*x(13))));
```

end

After defining the objective function, we write the commands in the "command window" in order to find the parameter estimates in LR and the optimum result. We can examine these commands in two sections.

Section 1

```
options=gaoptimset('StallGenLimit',50000,'PopulationSize',50,'SelectionFcn',{@selectionroulette},'CrossoverFr
action',0.95,'MutationFcn',{@mutationgaussian,0.15,0.88},'CrossoverFcn',{@crossoverintermediate,1.0},'PlotF
cns',@gplotbestf,'Generations',50000,'TolFun',1e-6,'TolCon',1e6,'Display','iter', 'Display','final',
'Display','diagnose');
```

In this section,

- (i) Entering 50 between the comas after 'PopulationSize' indicates that the population size is 50. In Matlab, the default value for the population size is 20.
- (ii) With 'SelectionFcn', we set 'roulette' as the selection operator.
- (iii) With 'CrossoverFcn', the crossover rate was set to be 0.95.
- (iv) With 'MutationFcn', 'gaussian' was set as the mutation operator. Mutation rate was set to be 0.15 and reduction rate was set to be 0.88.
- (v) With 'CrossoverFcn' the crossover function was set to be 'intermediate' and the crossover rate was set to

be 1.0.

(vi) Written after 'PlotFcns', @gaplotbestf gives a plot of the result. You can find the plot below.

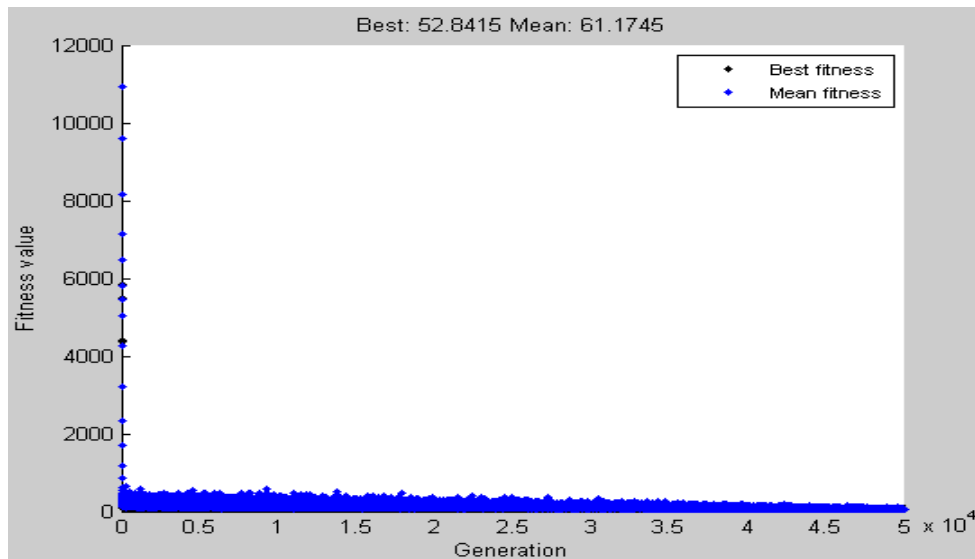


Fig.1. Fitness Value (50000 iteration)

In Fig. 1, we see the graphical representation of the results after 50000 iterations. Accordingly, while the objective function values are around 100 in the first iteration, they got closer to 50-55 range after the genetic operator processes, which is the optimization range of the function.

(vii) With 'StallGenLimit' and 'Generations' codes, we indicate that if there is no improvement for consecutive generations and if the iteration count exceeds 50000, the iterations will stop.

(viii) With 'TolFun' and 'TolCon' codes, the difference between to iteration values were set to be 1×10^{-6} as the convergence rate.

(ix) 'iter', 'final', and 'diagnose' codes written after 'Display' allow us to display all the iterations, the result value, the starting population of all stages and the objective function values of the optimization.

Section 2

`[x,fval,exitFlag,output,population,scores]=ga(@olabilirlik,13,[],[],[],[],[],[],[],options)`

In this section, the commands written on the left allow us to display the value which minimizes the function, the optimum value of the function and the populations. Other operators and rates, which are not presented in the coding, are processed as default. In this study, we tried more than one starting populations, crossover and mutation rates and different selection methods while using GA. The characteristics of the parameters we used to find the optimum results are given in detail in Table 3.

Table 3. Parameters used in coding

GA Parameter	Value / Method
PopulationType - PopulationSize	'doubleVector' -50
Selection Function	Roulette
Reproduction Crossover fraction	0.95
Mutation function	Gaussian
Mutation ratio and Mutation shrink	0.15 and 0.88
CrossoverFcn and Crossover ratio	Crossoverintermediate and 1.0
Migration Direction	'forward'
Others (InitialPopulation, InitialScores)	Use default
Stopping Criteria	
Generations and StallGenLimit	50000 and 5000
TolFun and TolCon	1e-6 and 1e-6
Output	
PlotFcns	@gaplotbestf
Display	'iter','final','diagnose'
Others	Use default

3.4. Comparison of the NR and GA Results

The results in Table 4 somewhat clarify the question that we discussed. The estimated parameter values and the maximum value of the log-likelihood function obtained by GA and NR are presented below [1, 11, 14, 16]. As can be seen in Table 4, both of the maximum values of the log-likelihood function are very close to each other. The difference between the optimum values is approximately '0.0498'. When the β parameters are examined, we see that the values are also close to each other.

Table 4. Comparison of the NR and GA Results

Estimated Parameters	Logistic Regression Analysis	
	NR Algorithm	Genetic Algorithm
$\hat{\beta}_0$ (constant)	2.985	1.984
$\hat{\beta}_1$	-0.025	-0.022
$\hat{\beta}_2$	1.618	1.548
$\hat{\beta}_3$	-0.051	-0.049
$\hat{\beta}_4$	0.135	0.135
$\hat{\beta}_5$	-0.004	-0.004
$\hat{\beta}_6$	-0.012	-0.01
$\hat{\beta}_7$	0.011	0.012

$\hat{\beta}_8$	-0.032	-0.032
$\hat{\beta}_9$	-1.611	-1.424
$\hat{\beta}_{10}$	2.357	2.283
$\hat{\beta}_{11}$	-0.039	-0.024
$\hat{\beta}_{12}$	0.006	0.005
Maximum value of the log-likelihood function	52.8833	52.833533819034635

These results show that we can obtain very close and sometimes even better results in LR model using GA with different operators and rates when the necessary conditions for applying the NR method are satisfied. Additionally, these findings represent the success of GA due to the fact that we could not use the NR algorithm in any event when the assumptions related to the method are not met. However, we could use the method of GA successfully in more flexible circumstances.

This study of course could be seen as an inception and introduction in this area. To be able to generalize the results, a simulation study should be done under different scenarios such as number of observations, different starting values for NR algorithm, different iteration numbers, different selection, mutation and crossover functions *etc.* for GA.

4. Discussion and Conclusion

When the dependent variable is binary, one of the most widely used methods in modelling the data is LR. Some interpretations which constitute the basis of the study can be made based on the statistically significant estimates of β parameters obtained as a result of LR model. Accordingly, these parameter estimations constitute an important part of the LR analysis. In the parameter estimation, the method of MLE is preferred due to its simplicity and flexibility in assumptions. In this method, after constructing the likelihood equations, LR model parameters could be obtained as a result of an iterative process. The traditional preference is to use NR algorithm in this step.

NR and other iterative methods are generally suitable for a certain problem. They require restrictive assumptions such as differentiable objective function and determination of a starting point for the solution. The search for solution starts from a certain point and it goes on from different points according to the results. For these methods to be successful, we may need to select a starting point by examining plots. Depending on the starting point selection, problems such as not being able to find a solution or a prolonged process of solution may occur.

However, GA has an edge on traditional algorithms because of its advantages such as not needing a derivative and other supporting information, and being able to find global optimum points without being stuck with local optimum points. In GA, the search is carried out on a potential solution set and the solutions are evaluated until the best solution is found.

In this study, the success of GA for a binary dependent variable model is shown by estimating β parameters that maximize the log-likelihood function and these estimates are compared with the estimates of traditional NR algorithm. When the results are examined we see that better optimized value for the log-likelihood function is obtained by GA. This reveals that we manage to go beyond our initial aim and show that the optimum value obtained using GA is better than NR. This also shows that GA is a admissible optimization method for the parameter estimation in models with categorical dependent variables and one of the best alternatives to the NR algorithm. Additionally, these findings represent the success of GA due to the fact that we could not use the NR algorithm in any event when the assumptions related to the method are not met.

In the next studies, GA's effectiveness in parameter estimation compared with NR method should be researched on a simulation study under different scenarios such as different sample sizes, different starting points, different starting populations or different convergence rates. This will allow us to find the best combination of the most suitable optimization method to use in studies with categorical dependent variable.

References

- [1] B. Altunkaynak, and A. Esin. "Doğrusal Olmayan Regresyonda Parametre Tahmini İçin Genetik Algoritma Yöntemi". *Gazi University Journal of Science*, 17(2), 43-51, 2004
- [2] D.C. Sekhar, and R. Ganguli. "Modified Newton, Rank-1 Broyden Update and Rank-2 Bfgs Update Methods in Helicopter Trim: A Comparative Study". *Aerospace Science and Technology*, 23(1), 187-200, 2012.
- [3] A. Agresti. *Categorical Data Analysis*. Canada: John Wiley & Sons, 2th edition, 2002.
- [4] D.W. Hosmer, and S. Lemeshow. *Applied Logistic Regression*. New York, USA: John Wiley & Sons Inc., 2th edition, 2000.
- [5] S. Menard. *Applied Logistic Regression Analysis*. USA: Sage Publications, 2th edition, 2002.
- [6] K. Matilainen, E. Mantysaari, M. Lidauer, I. Strandén, and R. Thompson. "Employing A Monte Carlo Algorithm in Newton-Type Methods For Restricted Maximum Likelihood Estimation of Genetic Parameters". *PLoS ONE*, 8(12), e80821, Doi:10.1371/journal.pone.0080821, 2013.
- [7] M. Mitchell. *An Introduction to Genetic Algorithms*. England: MIT Press, 5th edition, 1999.
- [8] J.H. Holland, *Adaptation in Natural and Artificial Systems*. USA: University of Michigan Press, 1975.
- [9] D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison. Massachusetts: Wesley Publishing, 1989.
- [10] D.E. Goldberg. *Frontiers of Evolutionary Computation*. New York: Kluwer Academic Publishers, 2004.

- [11] A. Pourrajabian, A. Ebrahimi, R.M. Mirzaei, and M. Shams. "Applying Genetic Algorithms or Solving Nonlinear Algebraic Equations". *Applied Mathematics and Computation*, 219, 11483-11494, 2013.
- [12] J. Alcock and K. Burrage. "A Genetic Estimation Algorithm For Parameters of Stochastic Ordinary Differential Equations". *Computational Statistics & Data Analysis*, 47, 255-275, 2004.
- [13] M. Thakur, S. Meghwani, and H. Jalota. "A Modified Real Coded Genetic Algorithm For Constrained Optimization", *Applied Mathematics and Computation*, 235, 292-317, 2014.
- [14] C. Reeves, and J.E. Rowe. *Genetic Algorithms Principles and Perspectives*. A Guide to GA Theory. USA: Kluwer Academic Press, 2002.
- [15] S. Chatterjee, M. Laudato, and L. Lynch. "Genetic Algorithms and Their Statistical Applications: An Introduction", *Computational Statistics & Data Analysis*, 22, 633-651, 1996.
- [16] C. Karr, and L.M. Freeman. *Industrial Applications of Genetic Algorithms*. USA: CRC Press, 1999
- [17] C.L. Karr, B. Weck, and L.M. Freeman, "Solutions to Systems of Nonlinear Equations via A Genetic Algorithm". *Engineering Applications of Artificial Intelligence*, 11, 369-375, 1998.
- [18] J. Pasia, A. Hermosilla, and H. Ombao. "A Useful Tool for Statistical Estimation: Genetic Algorithm", *Journal of Statistical Computation and Simulation*, 75(4), 237-251, 2005.
- [19] Z. Michalewicz, *Genetic Algorithms+ Data Structures = Evolution Programs*. USA: Springer, 3th edition, 1996.