Variable selection in Poisson regression model based on chaotic meta-heuristic search algorithm

heyaa Nadhim Ahmed Alangood [1*], Zakariya Yahya Algamal [2], and Mundher Abdullah Khaleel[3]
[1] Department of Mathmatics, College of Computer science & Mathematics, University of Tikrit, Iraq
[2] Department of Statistics and Informatics, College of Computer science & Mathematics, University of Mosul, Mosul, Iraq
[3] Department of Mathematics , University of Tikrit, Tikrit, Iraq

Abstract. By determining the most significant variables that are connected to the response variable, Increasing prediction accuracy and processing speed can be achieved through the process of variable selection. Regression modeling has drawn a lot of interest from several scientific domains. One of the most effective nature-inspired algorithms that has been suggested recently and can be used effectively for variable selection is the Firefly algorithm. The chaotic firefly algorithm is presented in this work to carry out the Poisson regression model's variable selection. A simulation study is carried out to assess how well the suggested strategy performs in terms of variable selection criteria and prediction accuracy. Its effectiveness is also contrasted with alternative approaches. The outcomes demonstrated the effectiveness of our suggested strategies, which beat other widely used approaches.

1. Introduction

Poisson regression analysis is frequently used to examine a variety of real data issues, including mortality studies, which look at the number of deaths, and health insurance, which aims to explain the number of claims made by the individual [1-3].Recent technological advancements have made it possible to measure a huge number of variables in many real-world applications. Huge numbers have a detrimental effect on regression modeling because they cause the model to overfit. Therefore, a key component of developing predictive regression models is identifying a small sample of significant factors from a vast set of variables for effective prediction [4]. When the number of variables increases, traditional variable selection approaches like stepwise selection, forward selection, and backward elimination computationally become exhaustive searches and become slow to process .Naturally inspired algorithms have gained popularity recently and have shown to be effective variable selection techniques. The crow search algorithm, the firefly method, the genetic algorithm, and the particle swarm optimization algorithm are a few examples of these [5]. This is because the primary goal of variable selection is to reduce the total number of variables chosen while preserving the highest level of prediction accuracy; as a result, it may be thought of as an optimization issue [6]. The organically inspired approaches for variable selection in regression models have been used by numerous researchers. Broadhurst, Goodacre [7] used the genetic process in chemometrics to choose variables for regression models that were linear and partial least squares. Drezner, Marcoulides [8] suggested selecting the model for the linear regression model using the tabu search technique

* Corresponding author: Dheyaaan39@gmail.com

Conversely, a hybrid approach combining simulated annealing and evolutionary algorithms was presented as a linear regression model subset selection technique by Örkcü [9]. Brusco [10] compared the simulated annealing algorithms used in discriminant analysis and principal component analysis for variable selection. Additionally, Dünder and Gümüştekin [11] employed the differential evolution approach to choose variables in a linear regression model. Extended linear models [12, 13], Poisson regression models [14, 15], and gamma regression models [16] also use natural inspired variable selection procedures similar to the logistic regression model.The suggested algorithm will effectively assist in determining which Poisson regression model variables have the highest prediction accuracy. The superiority of the suggested approach is demonstrated in several simulation scenarios.

2. Poisson regression model

Count data are frequently used in social, economic, and epidemiological research. The values in this type of data are positive integers. A well-known distribution that fits this kind of data is the Poisson distribution. The link between the counts as the response variable and possibly explanatory variables is modeled using the Poisson regression model [17, 18].

Let y_i be "the response variable and follows a Poisson distribution with mean θ_i, then the probability density function is defined as

$$f(y_i) = \frac{e^{-\theta_i}\theta_i^{y_i}}{y_i!}, \quad y_i = 0,1,\ldots; \; i = 1,2,\ldots,n. \tag{1}$$

In a Poisson regression model, $ln(\theta_i) = \boldsymbol{x}_i^T\boldsymbol{\beta}$ is expressed as a linear combination of explanatory variables $\boldsymbol{x}_i = (x_{i1},\ldots,x_{ip})^T$. The $ln(\theta_i)$ is called as canonical link function which making the relationship between explanatory variables and response variable linear. Using the maximum likelihood method is the most popular approach for estimating the coefficients of the Poisson regression model. Assuming that each observation is independent, the log-likelihood function can be expressed as follows: $\ell(\boldsymbol{\beta}) = \sum_{i=1}^{n}\{y_i\boldsymbol{x}_i^T\boldsymbol{\beta} - exp(\boldsymbol{x}_i^T\boldsymbol{\beta}) - ln\, y_i\,!\}$. (2)

The ML estimator is then obtained by computing the first derivative of the Eq. (3) and setting it equal to zero, as

$$\frac{\partial\ell(\boldsymbol{\beta})}{\partial\boldsymbol{\beta}} = \sum_{i=1}^{n}[y_i - exp(\boldsymbol{x}_i^T\boldsymbol{\beta})]\,\boldsymbol{x}_i = 0. \tag{3}$$

Because Eq. (4) is nonlinear in $\boldsymbol{\beta}$, the iteratively weighted least squares (IWLS) algorithm can be used to obtain the ML estimators of the Poisson regression parameters (PR) as

$$\widehat{\boldsymbol{\beta}}_{PRM} = (\boldsymbol{X}^T\widehat{\boldsymbol{W}}\boldsymbol{X})^{-1}\boldsymbol{X}^T\widehat{\boldsymbol{W}}\widehat{\boldsymbol{v}}, \tag{4}$$

where $\widehat{\boldsymbol{W}} = \mathrm{diag}(\hat{\theta}_i)$ and $\widehat{\boldsymbol{v}}$ is a vector where i^{th} element equals to $\hat{v}_i = ln(\hat{\theta}_i) + ((y_i - \hat{\theta}_i)/\hat{\theta}_i)$". The ML estimator is asymptotically normally distributed with a covariance matrix that corresponds to the inverse of the Hessian matrix

$$cov(\widehat{\boldsymbol{\beta}}_{PRM}) = \left[-E\left(\frac{\partial^2\ell(\boldsymbol{\beta})}{\partial\beta_i\,\partial\beta_k}\right)\right]^{-1} = (\boldsymbol{X}^T\widehat{\boldsymbol{W}}\boldsymbol{X})^{-1}. \tag{5}$$

3. Firefly algorithm

Many algorithms inspired by nature have been put out as effective methods of solving continuous optimization problems in recent years. An optimization task is to minimize the number of variables while maximizing forecast accuracy [6]. One of the most effective nature-inspired algorithms that has been proposed recently is the Firefly Optimization Algorithm (FA), which is firstly introduced by Yang [19]. When compared to other methods, the use of FA is a simple algorithm for solving optimization problems. FA draws inspiration from fireflies' social behavior as their lights flash. FA makes it possible for a swarm of low-light fireflies to migrate in the direction of nearby, brighter fireflies with better search skills to solve optimization difficulties. Three rules are held in FA [20]. “One firefly will be drawn to other fireflies regardless of their sex, according to the first rule, which states that all fireflies are unisex. The second rule states that a firefly's appeal is directly correlated with its brightness. This means that if two fireflies are flashing, the less brilliant one will gravitate toward the brighter one. It will move at random if there isn't a firefly that is brighter than it. The third rule states that there is a connection between a firefly's brightness and the fitness function's analytical form. The brightness of each firefly in a maximizing issue is proportional to the cost function's value.” Let d represents “the dimension of the object function that will optimized, n_f represents the number of fireflies, δ refers the light absorption coefficient, I_i is the light intensity, and r is the distance between any two firefly locations i (s_i) and j (s_j). This Cartesian distance can be defined as

$$r(s_i, s_j) = \sqrt{\sum_{c=1}^{d} \left(s_{i,c} - s_{j,c}\right)^2}. \quad (6)$$

Because I_i decreases when the distance from the source increases, the variations of I_i should be monotonically decreasing function. As a result, in most applications, the I_i can be approximated as

$$I(r) = I_0 e^{-\delta r^2}, \quad (7)$$

where I_0 is the original light intensity. Because the attractiveness of a firefly is proportional to the I_i, the attractiveness φ of a firefly is defined as

$$\varphi(r) = \varphi_0 e^{-\delta r^2}, \quad (8)$$

where φ_0 represents the attractiveness at $r = 0$. The movement of any firefly to the best position will be attracted to another firefly, which is more attractive firefly, by

$$s_i^{(t+1)} = s_i^{(t)} + \varphi_0 e^{-\delta r_{i,j}^2}\left(s_j^{(t)} - s_i^{(t)}\right) + \alpha\,(k_1 - 0.5), \quad (9)$$

where α and k_1, respectively, is the randomization parameter and a generated random number from uniform distribution with [0, 1]”.

FA was first suggested as a solution for continuous optimization issues. Nevertheless, the optimization issue in variable selection is discrete. Zhang, Gao [21] proposes a binary firefly algorithm (BFA) to deal with the problem of variable selection where the position is binary. The answer to the variable selection issue is written as a binary vector, where the value 1 denotes a variable that should be selected and 0 otherwise. This is because the goal of the problem is to select a specific variable or not. In BFA, the term $\varphi_0 e^{-\delta r_{i,j}^2}\left(s_j^{(t)} - s_i^{(t)}\right) + \alpha\,(k_1 - 0.5)$ will transfer to probability vector by using the sigmoid (sigm) function as

$$\text{sigm} = \frac{1}{1+exp\left[\varphi_0 e^{-\delta r_{i,j}^2}\left(s_j^{(t)} - s_i^{(t)}\right)+\alpha\,(k_1 - 0.5)\right]}. \quad (10)$$

As a result, the firefly's location in Eq. (10) will be changed to the following:

$$s_i^{(t+1)} = \begin{cases} 1 & \text{ifsigm} \geq k_2 \\ 0 & \text{otherwise,} \end{cases} \quad (11)$$

where k_2 denotes a random number produced using [0, 1] in a uniform distribution.

x_1	x_2	x_3	....	x_{p-1}	x_p
1	0	0	....	1	0

Fig1: The representation of the firefly position

4. The suggested chaotic algorithm for firefly

Chaos theory makes use of chaotic maps to explain random behavior in nonlinear systems. In a restricted spectrum of nonlinear, dynamic, and nonlinear systems, chaotic maps can be seen and can move as particles with no clear regularity along their route [22]. The use of the chaos technique helps to enhance the quality of the global optimum search and prevent becoming stuck in local optima. As a result, chaos has been used in many optimization scenarios. Chaos can be utilized to optimize the feature selection problem since it is an optimization problem with a search range of [0, 1] [22].Chaotic maps are expected to improve the performance of the binary firefly technique by preventing it from being stuck at local optima and by speeding up the convergence for variable selection in the gamma regression model. Ten chaotic maps are used in this paper. The description of these maps is explained in Table 1

.

Table 1: An explanation of the 10 utilized maps

Name	Definition	Range
Chebyshev	„$x_{k+1} = cos(k\, cos^{-1}(x_k))$	(-1,1)
Circle	$x_{k+1} = mod(x_k + 0.2 - \frac{0.5}{2\pi} sin(2\pi x_k),1)$	(0,1)
Guass/mouse	$x_{k+1} = \begin{cases} 1 & x_k = 0 \\ \frac{1}{mod(x_k,1)}, & otherwise \end{cases}$	(0,1)
Iterative	$x_{k+1} = sin\left(\frac{(0.7)\pi}{x_k}\right)$	(-1,1)
Logistic	$x_{k+1} = 4x_k(1 - x_k)$	(0,1)
Piecewise	$x_{k+1} = \begin{cases} \frac{x_k}{0.4} & 0 \le x_k < 0.4 \\ \frac{x_k - 0.4}{0.1} & 0.4 \le x_k < 0.5 \\ \frac{0.6 - x_k}{0.1} & 0.5 \le x_k < 0.6 \\ \frac{1 - x_k}{0.4} & 0.6 \le x_k < 1 \end{cases}$	(0,1)
Sine	$x_{k+1} = sin(\pi x_k)$	(0,1)
Singer	$x_{k+1} = 1.07(7.86x_k - 23.31(x_k)^2 + 28.75(x_k)^3 - 13.302875(x_k)^4$	(0,1)
Sinusoidal	$x_{k+1} = 2.3x_k\, sin(\pi x_k)$	(0,1)
Tent	$x_{k+1} = \begin{cases} \frac{x_k}{0.7} & x_k < 0.7 \\ \frac{10}{3}(1 - x_k) & x_k \ge 0.7 \end{cases}$"	(0,1)

Consequently, our proposed algorithm setting is as follows:

Step 1: The number of fireflies is $n_f = 30$, $\varphi_0 = 1$, $\delta = 0.3$, $\alpha = 0.2$, and the maximum number of iterations is t=1000_{max}.

Step 2: For the original binary firefly algorithm, each firefly's position is randomly generated from a uniform distribution with 0 and 1. The maps listed in Table 1 are utilized for the suggested chaotic maps. Figure 1 shows the representation of a firefly's positions.

Step 3: The definition of the fitness function is

$$\text{fitness} = min\left[\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2\right]. \quad (12)$$

Step 4: Using Eq. (10), the firefly positions are updated.

Step 5: Until a t_{max} is reached , Steps 3 and 4 are repeated.

4. Simulation results

This section looks into how well the suggested chaotic FA performs using simulation studies. The response variable in each simulation was produced using a Poisson distribution with a conditional mean as (13)
where . Two cases were studied:

$$\theta_i = exp(\boldsymbol{x}_i^T \boldsymbol{\beta}), \qquad (13)$$

where $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)$. Two cases were studied:
Case 1: The true variables' parameters, $\boldsymbol{\beta}$, were $\beta = (\underbrace{0.3, \ldots, 0.3}_{9}, \underbrace{0, \ldots, 0}_{991})$.
Case 2: The true variables' parameters, $\boldsymbol{\beta}$, were $\beta = (\underbrace{1.5, \ldots, 1.3}_{9}, \underbrace{0, \ldots, 0}_{9991})$.

Every simulation scenario is 100 times repeated. We produce training and testing data for each replication and for each simulation scenario (n=50 and n=150). The results are reported in Tables 2-5.Tables 2–5 show that the binary firefly algorithm with various chaotic maps surpasses the BFA. Most chaotic maps choose fewer variables than the BFA in terms of the total number of variables selected. But the variables chosen by the logistic map were the same as those chosen by the BFA. In comparison to the other chaotic maps, the Tent map is the best one because it chose fewer variables. Furthermore, when compared to the other chaotic maps that were employed, it is evident that the Tent map yields the lowest MSE for both the train and test datasets. Table 2 illustrates how the MSEtrain of the Tent map differed from that of the Chebyshev, Circle, Guass, Iterative, Logistic, Piecewise, Sine, and Singer algorithms by approximately 27.69%, 25.47%, 29.81%, 32.74%, 32.19%, 20.68%, 7.88%, 3.92%, and 15.06%, respectively. Furthermore, the Tent map was around 29.83% lower than the BFA. It is also evident that the second-best approach is the Singer map.It is clear from Table 3's test data (MSEtest) that the Tent map produces better outcomes than the others in terms of MSEtest. Comparing the Tent map's MSEtest to those of Chebyshev, Circle, Guass, Iterative, Logistic, Piecewise, Sine, and Singer, the results were roughly 25.36%, 22.74%, 26.36%, 32.71%, 31.96%, 17.55%, 6.97%, 2.64%, and 13.66% lower, respectively.

Table 2: The performance of the used chaotic maps when case 1 and n=50

Method	# selected variables	MSEtrain	MSEtest
BFA	26	3.54	3.784
Chebyshev	24	3.439	3.68
Circle	23	3.327	3.568
Guass	25	3.478	3.739
Iterative	24	3.831	4.072
Logistic	26	3.767	4.008
Piecewise	20	3.067	3.308
Sine	15	2.581	2.827
Singer	14	2.466	2.707
Sinusoidal	18	2.863	3.104
Tent	11	2.388	2.646

Table 3: The performance of the used chaotic maps when case 1 and n=150

Method	# selected variables	MSEtrain	MSEtest
BFA	25	3.329	3.573
Chebyshev	22	3.228	3.469
Circle	22	3.116	3.357
Guass	23	3.267	3.528
Iterative	24	3.62	3.861
Logistic	24	3.556	3.797
Piecewise	19	2.856	3.097
Sine	14	2.37	2.616
Singer	13	2.255	2.496
Sinusoidal	17	2.652	2.893
Tent	10	2.177	2.435

Table 4: The performance of the used chaotic maps when case 2 and n=50

Method	# selected variables	MSEtrain	MSEtest
BFA	35	6.181	6.425
Chebyshev	33	6.08	6.321
Circle	32	5.968	6.209
Guass	33	6.119	6.38
Iterative	34	6.472	6.713
Logistic	35	6.408	6.649
Piecewise	29	5.708	5.949
Sine	24	5.222	5.468
Singer	23	5.107	5.348
Sinusoidal	26	5.504	5.745
Tent	20	5.029	5.287

Table 5: The performance of the used chaotic maps when case 2 and n=150

Method	# selected variables	MSEtrain	MSEtest
BFA	34	5.645	5.889
Chebyshev	31	5.544	5.785
Circle	30	5.432	5.673
Guass	32	5.583	5.844
Iterative	32	5.936	6.177
Logistic	33	5.872	6.113
Piecewise	27	5.172	5.413
Sine	22	4.686	4.932
Singer	21	4.571	4.812
Sinusoidal	24	4.968	5.209
Tent	17	4.493	4.751

5. Conclusion

The issue of choosing variables for Poisson regression models is examined in this study. As a variable selection technique, the simulation study's outcomes proved that the chaotic firefly algorithm was better in terms of MSE for both the train and test data as well as a few other factors.

REFERENCES

1 Algamal, Z.Y., Diagnostic in poisson regression models. Electronic Journal of Applied Statistical Analysis, 2012. 5(2): p. 178-186.
2 Cameron, A.C. and P.K. Trivedi, Regression analysis of count data. Vol. 53. 2013: Cambridge university press.
3 De Jong, P. and G.Z. Heller, Generalized linear models for insurance data. Vol. 10. 2008: Cambridge University Press Cambridge.
4 Algamal, Z.Y. and M.H. Lee, Penalized logistic regression with the adaptive LASSO for gene selection in high-dimensional cancer classification. Expert Systems with Applications, 2015. 42(23): p. 9326-9332.
5 Sayed, G.I., A.E. Hassanien, and A.T. Azar, Feature selection via a novel chaotic crow search algorithm. Neural Computing and Applications, 2017.
6 Sindhu, R., et al., Sine–cosine algorithm for feature selection with elitism strategy and new updating mechanism. Neural Computing and Applications, 2017. 28(10): p. 2947-2958.

7 Broadhurst, D., et al., Genetic algorithms as a method for variable selection in multiple linear regression and partial least squares regression, with applications to pyrolysis mass spectrometry. Analytica Chimica Acta, 1997. 348(1-3): p. 71-86.
8 Drezner, Z., G.A. Marcoulides, and S. Salhi, Tabu search model selection in multiple regression analysis. Communications in Statistics - Simulation and Computation, 1999. 28(2): p. 349-367.
9 Örkcü, H., Subset selection in multiple linear regression models: A hybrid of genetic and simulated annealing algorithms. Applied Mathematics and Computation, 2013. 219(23): p. 11018-11028.
10 Brusco, M.J., A comparison of simulated annealing algorithms for variable selection in principal component analysis and discriminant analysis. Computational Statistics & Data Analysis, 2014. 77: p. 38-53.
11 Dünder, E., et al., Variable selection in linear regression analysis with alternative Bayesian information criteria using differential evaluation algorithm. Communications in Statistics - Simulation and Computation, 2017. 47(2): p. 605-614.
12 Pacheco, J., S. Casado, and L. Núñez, A variable selection method based on Tabu search for logistic regression models. European Journal of Operational Research, 2009. 199(2): p. 506-511.
13 Unler, A. and A. Murat, A discrete particle swarm optimization method for feature selection in binary classification problems. European Journal of Operational Research, 2010. 206(3): p. 528-539.
14 Koç, H., et al., Particle swarm optimization-based variable selection in Poisson regression analysis via information complexity-type criteria. Communications in Statistics - Theory and Methods, 2017: p. 1-9.
15 Massaro, T.J. and H. Bozdogan, Variable subset selection via GA and information complexity in mixtures of Poisson and negative binomial regression models. arXiv preprint arXiv:1505.05229, 2015.
16 Dunder, E., S. Gumustekin, and M.A. Cengiz, Variable selection in gamma regression models via artificial bee colony algorithm. Journal of Applied Statistics, 2016. 45(1): p. 8-16.
17 Algamal, Z.Y. and M.H. Lee, Adjusted Adaptive LASSO in High-dimensional Poisson Regression Model. Modern Applied Science, 2015. 9(4): p. 170-177.
18 KaÇiranlar, S. and I. Dawoud, On the performance of the Poisson and the negative binomial ridge predictors. Communications in Statistics - Simulation and Computation, 2017: p. 0-0.
19 Yang, X.-S., Multiobjective firefly algorithm for continuous optimization. Engineering with Computers, 2013. 29(2): p. 175-184.
20 Yu, S., et al., Enhancing firefly algorithm using generalized opposition-based learning. Computing, 2015. 97(7): p. 741-754.
21 Zhang, J., et al., Identification of DNA-binding proteins using multi-features fusion and binary firefly optimization algorithm. BMC Bioinformatics, 2016. 17(1): p. 323.
22 Sayed, G.I., A. Darwish, and A.E. Hassanien, A New Chaotic Whale Optimization Algorithm for Features Selection. Journal of Classification, 2018. 35(2): p. 300-344.