Thailand Statistician
January 2025; 23(1): 29-40
http://statassoc.or.th
Contributed paper

Bayesian Analysis of Generalized Linear Models for Count Data using Stan

Firdoos Yousuf [a]* and Athar Ali Khan [b]
[a] Department of Statistics, Government Degree College,
Bijbehara Anantnag, Jammu and Kashmir, India.
[b] Department of Statistics and Operations Research,
Aligarh Muslim University, Aligarh, Uttar Pradesh, India.
*Corresponding author; e-mail: firdoos1990@gmail.com

Received: 13 April 2021
Revised: 21 July 2022
Accepted: 27 July 2022

Abstract

In this paper, Bayesian approach is used to model count data for generalized linear models (GLMs) using Stan language. Commonly used GLMs include logistic regression for binary data and Poisson regression or Negative binomial regression for count data. The loo package is used for model selection. The Bayesian model comparison criteria LOOIC and WAIC are applied to evaluate the models. Furthermore, parallel simulations tools are also implemented with an extensive use of R.

Keywords: Bayesian inference, poisson regression, negative binomial regression , posterior, Stan, LOOIC, WAIC.

1. Introduction

An important milestone in the development of count data models for regression was the emergence of the "generalized linear models", of which the Poisson regression is a special case, first introduced by Nelder and Wedderburn (1972) and detailed in McCullagh and Nelder (1989). Building on these contributions, the work on longitudinal or panel count data models by Hausman et al. (1984) have also been very influential in stimulating applied work in the econometric literature. The existing econometrics literature on count data models has largely ignored the Bayesian paradigm of inference. Similarly, in Zellner (1971) important book on Bayesian inference in econometrics, the Poisson regression model is not addressed. The feasible reasons for this overlook are computational complexities that made the Bayesian analysis of count data models unattractive in the past. However, increased computer capability now allows for fast evaluation of posterior distributions using simulation methods. The fundamental approaches to Bayesian inference by simulation are briefly discussed in this paper. The development of the generalized linear models (GLM) provided a solution to the latter approach.

Generalized linear models, originally introduced by Nelder and Wedderburn (1972), provide a uniform family of models that are widely used in practical regression analysis for which either the assumption of a linear relationship between covariate and response variable or the assumption of normal variation is not appropriate (Gelman et al., 2014). GLMs can be regarded as the natural extension of normal linear regression models and are based on the exponential family of distributions, including the normal, exponential, gamma, binomial, Poisson, negative binomial and many others. Due to the generality and wide range of applications, they can be considered as one of the most prominent and important components of the modern statistical theory. They have provided not only a family of models that are widely used in practice but also a unified, general way of thinking concerning the formulation of statistical models (Ntzoufras, 2009).

In this paper, we discuss Poisson regression as a generalized linear model to analyze the count data under Bayesian framework, a comparison is made with Negative binomial regression model has done using Stan and R languages for fitting the models. Selecting the best-fitted model from among the several competitive models is always crucial in Bayesian inference and well as in Classical Statistics (Frequentist). Based on information criteria Leave-One-Out Cross-Validation (LOO-CV, or LOO) and the Widely Applicable or Watanable Akaike Information Criterion WAIC (Vehtari et al., 2017; McElreath, 2020; Gelman et al., 2014) are usually used to compare the Bayesian fitted models and to select the better one than the others. The packages rstan and loo are used for this purpose.

2. Structure of the GLM Model

A GLM is specified in the following three components (Gelman et al., 2014):

1. Random component: The random component specifying the distribution of the response variable y with mean $E(y|X) = \mu$. In general, the distribution of y given x can also depend on a dispersion parameter ϕ.

2. Linear predictor: The linear predictor, $\eta = X\beta$, is a systematic or deterministic component specifying the linear model defined by the explanatory variables.

3. Link function: The link function $g(\cdot)$ is the mathematical expression that relates the linear predictor to the expectation of the response variable: $\mu = g^{-1}(\eta) = g^{-1}(X\beta)$.

Thus, the expectation of the distribution of y, given X is determined by $X\beta : E(y|X) = g^{-1}(X\beta)$. The variable X is the $n \times p$ matrix of explanatory variables and $\eta = X\beta$ is the vector of n linear predictor values.

3. Poisson and Negative Binomial Regression

The Poisson regression is used when the outcome variable represents count data. In this type of data, the observations take only the non-negative integer values $\{0, 1, 2, 3, \ldots\}$, which arise from counting rather than ranking, or grouping. The distribution of counts is discrete, and typically skewed (Wang et al., 2018).

The Poisson generalized linear model, often known as a Poisson regression model, assumes that y is Poisson having only one parameter that specifies its shape, resulting in a likelihood definition will be of the form

$$y \sim \text{Poisson}\,(\lambda)$$

where λ is the expected value of the outcome y. The distribution has the canonical parameter $\theta = \log\lambda$, the dispersion parameter $\phi = 1$, and its variance function equals λ. To build a GLM with the likelihood, we also require a link function. The canonical link function is the logarithm link, leads to the Poisson regression model,

$$\begin{cases} y_i \sim \text{Poisson}\,(\lambda_i)\,, \\ \log\,(\lambda_i) = \beta_0 + \sum_{j=1}^{J} \beta_j x_{ij} = \eta_i. \end{cases}$$

The distribution for data $y = (y_1, \ldots, y_n)$ is thus

$$p\,(y|\beta) = \prod_{i=1}^{n} \frac{1}{y_i!} e^{-exp(\eta_i)}\,(exp\,(\eta_i))^{y_i}$$

where $\eta_i = X_i\beta$ is the linear predictor for the i^{th} case. When considering the Bayesian posterior distribution, we condition on y, and so the factors of $1/y_i!$ can be absorbed into an arbitrary constant.

In applied work, Poisson regression is restrictive in the analysis of count data. It is acknowledged that counts often display substantial extra-Poisson variation, or overdispersion. Overdispersion refers to the situation when the variance of an observed dependent variable exceeds the nominal variance, given the respective assumed distribution. The assumption in Poisson model that the conditional mean and variance of Y given X are equal may be too strong and thus fail to account for the overdispersion. Inappropriate imposition of this restriction may result in unreasonably small estimated standard errors of the parameter estimates. In this scenario, the negative binomial regression is perhaps the most convenient way to relax the Poisson restriction and deal with the overdispersion (Wang et al., 2018).

The log link is commonly used in negative binomial regression. Let us suppose we have a vector of k explanatory variables, $(x_{i1}, x_{i2}, \ldots, x_{ik})$, which is related to the response variable y_i. The model is written as:

$$\begin{cases} y_i \sim \text{NB}\left(\alpha, \mu_i\right), \\ \log\left(\mu_i\right) = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_i k. \end{cases}$$

4. Description of Data

The data set “DoctorVisits” is previously analyzed by Cameron et al. (1988) in the brightness of an economic model of the joint determination of health service utilization and health insurance choice with classical parametric and nonparametric methods. It is the cross-section data based on the number of doctor visits in the past two weeks for a single-adult of sample size 5190 from the Australian Health Survey 1977-1978. Here, we shall discuss the Bayesian approach of the same data set. It is in the form of a data frame containing 5190 observations on 12 variables. The following R commands show how to load this data frame:

```
library(AER)
```

To load the data DoctorVisits, use the function data

```
data(“DoctorVisits”,package=“AER”)
```

The data frame DoctorVisits has 5190 rows and 12 columns, but we can display only the few rows using the function head

```
head(DoctorVisits)
```

	visits	gender	age	income	illness	reduced	health	private	freepoor	freerepat	nchronic	lchronic
1	1	female	0.19	0.55	1	4	1	yes	no	no	no	no
2	1	female	0.19	0.45	1	2	1	yes	no	no	no	no
3	1	male	0.19	0.90	3	0	0	no	no	no	no	no
4	1	male	0.19	0.15	2	5	1	no	no	no	no	no
5	1	male	0.19	0.45	2	5	1	no	no	no	yes	no
6	1	female	0.19	0.35	5	1	9	no	no	no	yes	no

5. Stan Modeling for Poisson Regression Model

In this section, Bayesian analysis of Poisson regression model with Stan (Carpenter et al., 2017). Stan uses the “no-U-turn sampler”, which is a type of Hamiltonian Monte Carlo (HMC) simulation (Hoffman and Gelman, 2014; Betancourt, 2013). HMC can be more effective than the various samplers in JAGS and BUGS, especially for large data and complex models. Moreover, Stan operates with compiled C++ and allows greater programming flexibility, which again is especially useful for unusual or complex models. For large data sets or complex models, Stan can provide solutions when JAGS (or BUGS) takes too long or fails. In Stan, a model is defined in a text file, the data is prepared within R, and then Stan is called from R. The stan code blocks for implementing the Bayesian Poisson regression model are given below using the package rstan (Stan Development Team, 2020).

Before fitting the Poisson regression model with Stan, we require a model matrix X (also known as design matrix) which can be extracted by the function model.matrix() from a glm() fitted object, we use the following command:

```
M0<-glm(visits˜.+I(ageˆ2),data=DoctorVisits,family=poisson
(link="log"))
```

Now the Stan code for the Poisson model are as follows:

```
library("rstan")
// data block
stan_code<-"
data{
int<lower=0> N;//the number of observations
int<lower=0> M;//the number of covariates
matrix[N,M] X;//matrix of covariates with N rows and
M columns
vector[N] y;//observed times
}
// parameters block
parameters{
vector[M] beta;//coefficients in the linear predictor
}
// transformed parameters block
transformed parameters{
vector[N] linpred;
vector[N] mu;
linpred=X*beta;
for(i in 1:N){
mu[i]=exp(linpred[i]);
}
}
// model block
model{
beta ˜ student_t(2,0,5);//prior for regression coefficients
y ˜ poisson(mu);//likelihood
}
// generated quantities block
generated quantities{
real log_lik;
log_lik=poisson_lpmf(y|mu);
}"
```

The whole code blocks are saved as stan_code that is used afterwards.

Creation of data in R

Stan requires a data list that includes a matrix, vector and values. So, the data must be made within `R` for Stan. The response variate `visits` is entered with a vector y of length N. An object list is built and assigned it to dat.

```
y<-DoctorVisits$visits
X<-model.matrix(M0)
N<-length(y)
M<-ncol(X)
dat=list(X=X,N=N,M=M,y=y)
```

Model Fitting

To fit the Poisson regression model using the function stan, and an object Fitstanpois has been created. The function stan starts Stan and returns an object containing MCMCs for every model parameter, and its results are assigned with the object Fitstanpois. For model comparison criteria, we compute LOOIC and WAIC using the loo package.

```
Fitstanpois<-stan(model_code=stan_code,data=dat,
iter=5000,chains=3)
print(Fitstanpois,digit=3,pars=c('beta','sigma'),probs=
c(.025, .5, .975))
// compute looic and waic using loo package
library(loo)
log_lik_1 <- extract_log_lik(Fitstanpois,
merge_chains = FALSE)
r_eff1 <- relative_eff(exp(log_lik_1))
loo_1 <- loo(log_lik_1, r_eff = r_eff1, cores = 2)
print(loo_1)
waic1 <- waic(log_lik_1)
print(waic1)
```

Summarizing Output

The function print summarizes the MCMC output, and the results are reported in Table 1. It gives the mean and standard deviation (sd) of the marginal posterior distribution of each parameter, and some quantiles, effective sample size and Rhat. The Rhat described by Gelman et al. (2014) and implemented in Stan, called the "potential reduction factor". The convergence of an MCMC algorithm is assessed mainly by trace and autocorrelation plots. Figure 1 indicates the autocorrelation drops to near zero rapidly with the increase of lag indicating convergence of the MCMC sampling process to the joint posterior distribution, so the MCMC algorithm HMC-NUTS performs correctly to explore the target posterior distribution. Moreover,there is no indication of non-convergence and all Rhat are lower than 1.1 which can be seen in Figure 2. Also, from the posterior predictive density plot (Figure 3), indicated that the Poisson model is well matched to the data. The decision of significant regressor variables can be done by using a caterpillar plot (Figure 4) which are very popular in Bayesian inference for summarizing the quantiles of posterior samples.

Table 1 Summary of the simulated results of fitted Poisson regression model obtained by Hamiltonian Monte Carlo (HMC) algorithm using the function stan

Parameters	mean	sd	2.5%	50%	97.5%	n_eff	Rhat
beta[1]	-2.221	0.187	-2.591	-2.223	-1.862	2060	1.000
beta[2]	0.156	0.055	0.048	0.156	0.266	4434	1.000
beta[3]	1.036	0.971	-0.845	1.049	2.944	2008	1.000
beta[4]	-0.208	0.087	-0.379	-0.207	-0.038	4186	1.000
beta[5]	0.187	0.018	0.151	0.187	0.222	4250	1.001
beta[6]	0.127	0.005	0.117	0.127	0.137	4573	1.000
beta[7]	0.030	0.010	0.010	0.030	0.049	4042	1.000
beta[8]	0.124	0.070	-0.013	0.123	0.265	3479	1.001
beta[9]	-0.456	0.180	-0.816	-0.452	-0.111	4380	1.000
beta[10]	0.080	0.090	-0.097	0.078	0.258	3862	1.000
beta[11]	0.115	0.068	-0.022	0.116	0.247	3528	1.001
beta[12]	0.140	0.084	-0.026	0.140	0.304	3374	1.001
beta[13]	-0.830	1.043	-2.872	-0.839	1.170	2147	1.000

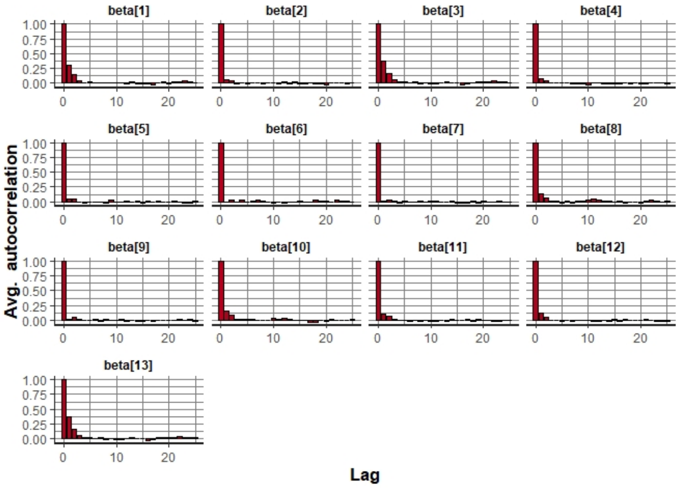

Figure 1 Autocorrelation plot of fitted Poisson regression model using Stan shows that autocorrelation drops to values close to zero at around lags of 2

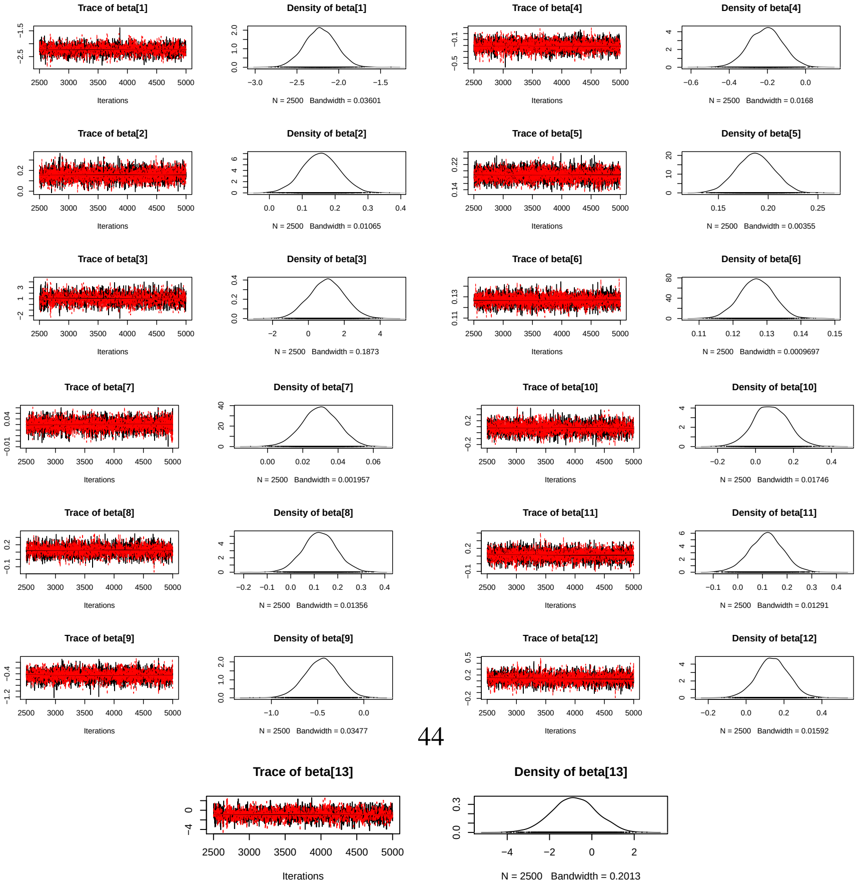

Figure 2 Checking model convergence using the package coda, through inspection of the simulated posterior density plots with trace plots of regressor variables obtained by HMC

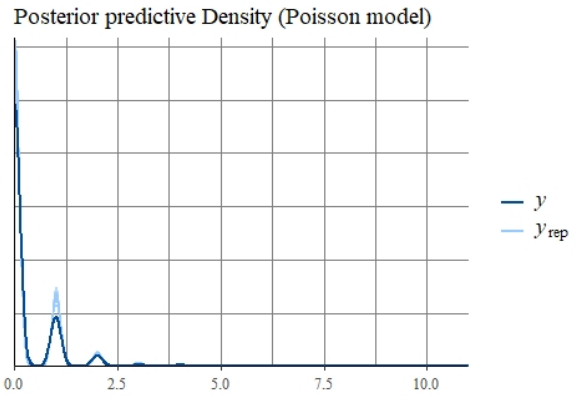

Figure 3 Posterior predictive density plot of Poisson regression model and the plot shows that the posterior predictive density fits the data well

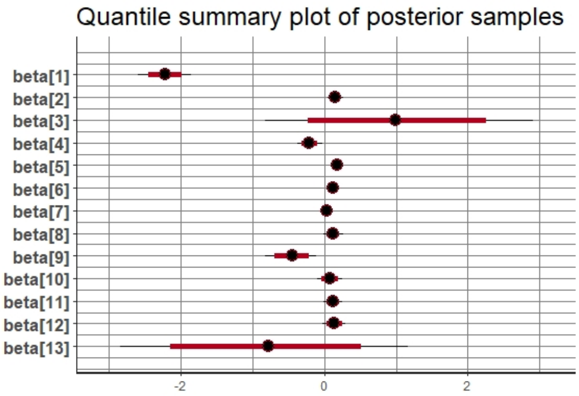

Figure 4 Caterpillar plot of Poisson regression model shows that 95% credible intervals of the parameters do not include zero value so the parameters are statistically significant. Otherwise, they are insignificant

6. Stan Modeling for Negative Binomial Regression Model

To implement the negative binomial regression model within Bayesian framework. In R, tools for negative binomial regression are provided by the MASS package (Venables and Ripley , 2002). The functions glm.nb and negative.binomial can be used for estimating the negative binomial GLMs, which are available in the MASS package. When fitting the negative binomial regression in Stan, we use NB2. NB2 is the traditional parameterization of the negative binomial model. To see how is it works, let's again revisit the DoctorVisits data. Here, the Stan code defining a model consists of different named blocks using rstan package.

```
library("rstan")
stan_code="
data {
int<lower=0> N;//the number of observations
int<lower=0> K;//the number of covariates
matrix[N,K] X;//matrix of covariates with N rows and
K columns
vector[N] y;//observed times
}
// parameters block
parameters {
vector[K] beta;
real phi;//the overdispersion parameter
}
//transformed parameters block
transformed parameters {
vector[N] mu;//the linear predictor
mu <- exp(X*beta); //using the log link
}
// model block
model {
beta ~ student_t(2,0,5);//prior for regression coefficients
y ~ neg_binomial_2(mu,phi);//likelihood
}
// generated quantities block
generated quantities{
real log_lik;
log_lik=neg_binomial_2_lpmf(y|mu,phi);
}"
```

The whole code blocks are saved as stan_code that is used afterwards. The same DoctorVisits data object (dat) prepared for Poisson regression model is applied here for fitting negative binomial regression model. But, only model matrix X is different here which can be extracted by the function model.matrix() from a fitted object fitglm using the MASS package.

```
library(MASS)
fitglm<-glm.nb(visits ~ .+I(age^2), data = DoctorVisits)
X<-model.matrix(fitglm)
```

Model fitting

To fit the negative binomial regression model under the Bayesian framework and to simulate from the posterior distribution using the function `stan` is called from the `rstan` package, and its results are assigned with the object `Fitstannb`.

```
Fitstannb<-stan(model_code=stan_code,data=dat,iter=
5000,chains=3)
print(Fitstannb,c("beta"),digits=3,prob=c(.025,.5,.975)))
// compute looic and waic using loo package
```

```
library(loo)
log_lik_2 <- extract_log_lik(Fitstannb,merge_chains=FALSE)
r_eff2 <- relative_eff(exp(log_lik_2))
loo_2 <- loo(log_lik_2, r_eff = r_eff2, cores = 2)
print(loo_2)
waic2 <- waic(log_lik_2)
print(waic2)
compare(waic1,waic2)
```

Table 2 Summary of the simulated results of fitted negative binomial regression model obtained by HMC algorithm using the function stan

Parameters	mean	sd	2.5%	50%	97.5%	n_eff	Rhat
beta[1]	-2.188	0.233	-2.655	-2.184	-1.735	2427	1.002
beta[2]	0.218	0.069	0.084	0.218	0.357	4129	0.999
beta[3]	-0.262	1.264	-2.710	-0.268	2.208	2235	1.005
beta[4]	-0.141	0.110	-0.356	-0.142	0.074	4155	1.001
beta[5]	0.215	0.024	0.168	0.215	0.261	4195	1.003
beta[6]	0.144	0.008	0.129	0.144	0.160	4872	1.000
beta[7]	0.038	0.014	0.011	0.038	0.065	4595	1.000
beta[8]	0.119	0.086	-0.047	0.119	0.291	3485	0.999
beta[9]	-0.502	0.205	-0.902	-0.500	-0.104	4559	1.001
beta[10]	0.148	0.118	-0.083	0.147	0.381	3732	0.999
beta[11]	0.099	0.079	-0.056	0.098	0.257	3746	1.001
beta[12]	0.189	0.106	-0.015	0.189	0.402	3539	1.001
beta[13]	0.659	1.385	-2.083	0.644	3.272	2233	1.004

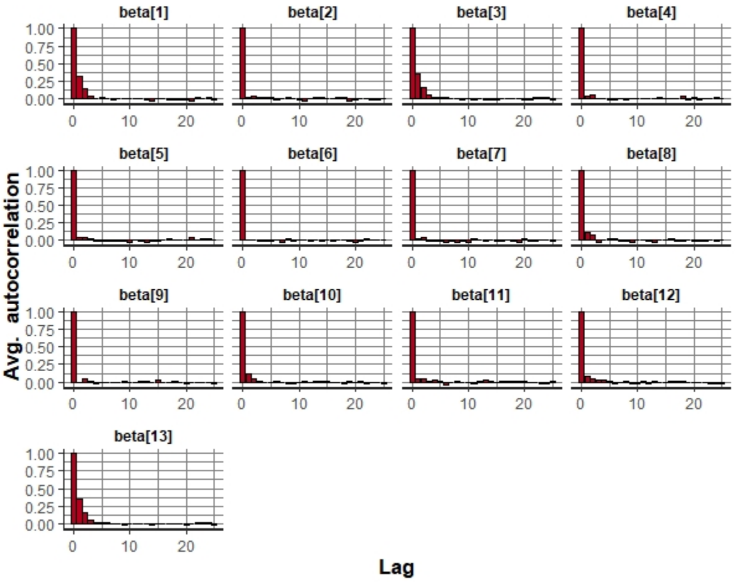

Figure 5 Autocorrelation plot of fitted negative binomial regression model using Stan shows that autocorrelation drops to values close to zero at around lags of 2

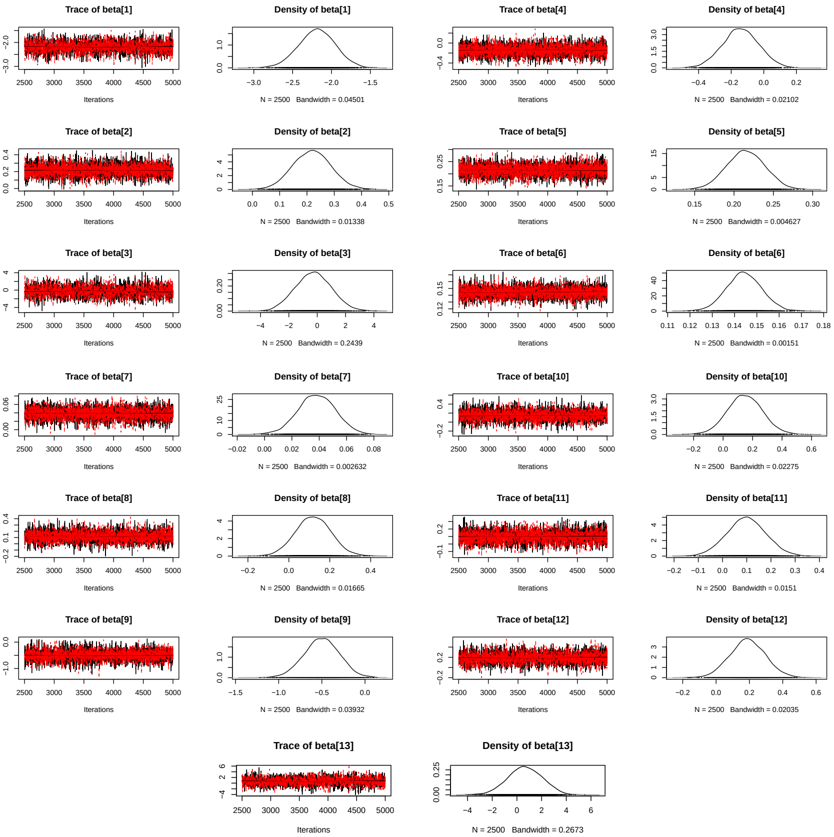

Figure 6 Trace of the Markov chains for model parameters as well as their posterior density plots using the package coda obtained by HMC

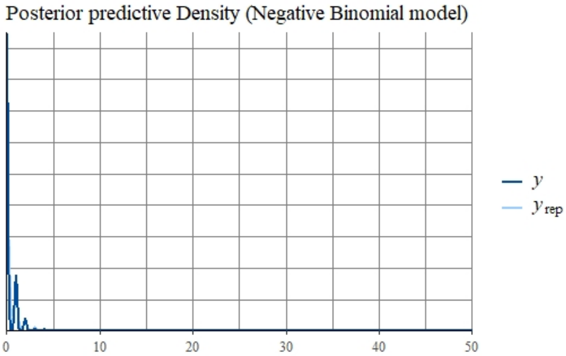

Figure 7 Posterior predictive density plot of negative binomial regression model shows that the posterior predictive density fits the data well

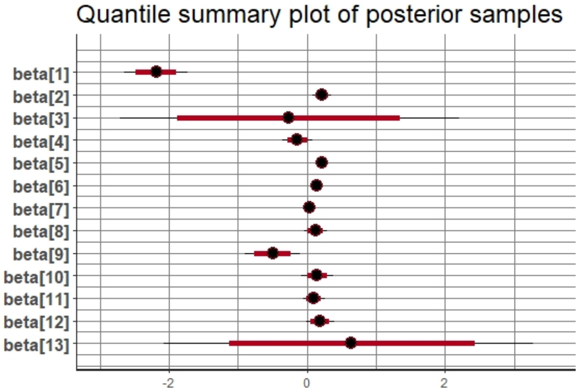

Figure 8 Caterpillar plot of negative binomial model

7. Model Comparison

In Bayesian inference, as well as in classical inference, it is always a crucial part to select the best model from among the several competitive models. Two information criteria Leave-One-Out Cross validation LOO and Widely Applicable or Watanabe Akaike Information Criterion WAIC (Vehtari et al., 2017; McElreath, 2020; Gelman et al., 2014) are usually used to compare the Bayesian fitted models and to select the better one than the others. A better fitted model is that which has smaller LOOIC (LOO Information Criterion) or WAIC than the others. The fitted negative binomial regression model has smaller LOOIC as well as WAIC can be seen in Table 3. Thus, negative binomial regression model fit the DoctorVisits data better than the Poisson regression model.

Table 3 Model comparison of Poisson and negative binomial regression model for DoctorVisits data. Both LOOIC and WAIC criterion support negative binomial regression model is a better choice as compared to Poisson regression model

Models	LOOIC	SE	WAIC	SE
Poisson regression	6750.4	169.1	6750.3	169.1
Negative binomial regression	6428.0	140.4	6427.9	140.4

8. Conclusions

This paper introduced how count data can be modeled using the Poisson regression and negative binomial regression within Bayesian framework. Analysis with logarithm link, that are commonly used in Poisson and negative binomial regression was studied. The implementation is made through rstan software package. The algorithm such as Hamiltonian Monte Carlo is used for simulations. Finally, we compared the models using loo package. Upon comparison with the results obtained through LOOIC and WAIC, it can be concluded that the negative binomial regression model fits the data better than the Poisson regression model. Thus, it is not easy to analyze these types of data by Classical (Frequentist) technique, whereas it is completely simple in the Bayesian framework using tools like R.

Acknowledgements

The authors would like to thank the referees and the Editor-in-Chief for their valuable comments and suggestions which led to considerable improvement in the manuscript.

References

Betancourt MJ. Generalizing the No-U-Turn sampler to Riemannian manifolds. 2013. Available from: https://doi.org/10.48550/arXiv.1304.1920.

Cameron AC, Trivedi PK, Milne F, Piggott J. A microeconometric model of the demand for health care and health insurance in Australia. Rev Econ Stud. 1988; 55(1): 85-106.

Carpenter B, Gelman A, Hoffman MD, Lee D, Goodrich B, Betancourt M, Brubaker M, Guo J, Li P, Riddell A. Stan: A probabilistic programming language. J Stat Softw. 2017; 76(1): 1-32.

Gelman A, Carlin JB, Stern HS, Dunson DB, Vehtari A, Rubin DB. Bayesian Data Analysis. New York: Chapman and Hall/CRC; 2014.

Hausman JA, Hall BH, Griliches Z. Econometric models for count data with an application to the patents-R and D relationship. Econometrica. 1984; 52(4): 909-938.

Hoffman MD, Gelman A. The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. J Mach Learn Res. 2014; 15(1): 1593-1623.

McCullagh P, Nelder JA. Generalized Linear Models. New York: Chapman and Hall/CRC; 1989.

McElreath R. Statistical rethinking: A Bayesian course with examples in R and Stan. Chapman and Hall/CRC; 2020.

Nelder JA,Wedderburn RWM. Generalized Linear Models. J R Stat Soc Ser A-G. 1972; 135: 370-384.

Ntzoufras I. Bayesian Modeling Using WinBUGS. New York: John Wiley and Sons; 2009.

Stan Development Team. RStan: the R interface to Stan. R package version 2.21.2. 2020. Available from: http://mc-stan.org/.

Vehtari A, Gelman A, Gabry J. Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. Stat Comput. 2017; 27(5): 1413-1432.

Venables WN, Ripley BD. Modern Applied Statistics with S. New York: Springer; 2002.

Wang X, Ryan YY, Faraway JJ. Bayesian Regression Modeling with INLA. Chapman and Hall/CRC; 2018.

Zellner A. An Introduction to Bayesian Inference in Econometrics. New York: Wiley; 1971.