ECON 753 Assignment 2 – Count Model

Paola Gabriela Villa Paro

In 1969, the popular magazine *Psychology Today* published a 101-question survey on extramarital affairs. Professor Ray Fair (1978) extracted a sample of 601 observations on men and women who were currently in their first marriage and analyzed their responses to the survey. He used the "Tobit" model as his estimation framework for this study. The dependent variable is a count of the number of affairs, so instead of a Tobit, a standard Poisson model may be a better choice. Download the data set psychtoday.csv, and estimate the parameters to the model below using nonlinear least squares and maximum likelihood using the algorithms that I suggest below.

Data Description

- y - count data: number of affairs in the past year.
- $\boldsymbol{x}$ - constant term=1, age, number of years married, religiousness (1-5 scale), occupation (1-7 scale), self-rating of marriage (1-5 scale)

Assingment

The following is the data generating assumptions for the Poisson model, where j is the number of affairs:

$$Pr[y_i = j] = \frac{e^{-\lambda_i}\lambda_i^j}{j!} \tag{1}$$

$$log(\lambda_i) = \boldsymbol{x}_i'\boldsymbol{\beta} \tag{2}$$

$$E(y_i \mid x_i) = e^{\boldsymbol{x}_i'\boldsymbol{\beta}} \tag{3}$$

for some $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5)'$.

The log-likelihood function is:

$$\begin{aligned}
\ln L &= \sum_{i=1}^{n} \ln f\left(y_i \,|x_i\,, \beta\right) \\
&= \sum_{i=1}^{n} \ln \frac{e^{-\lambda_i}\lambda_i^j}{j!} \\
&= \sum_{i=1}^{n} \left[-\lambda_i + y_i \ln\lambda_i - \ln y_i!\right] \\
&= \sum_{i=1}^{n} \left[-e^{\mathbf{x}_i{}'\beta} + y_i x_i'\beta - \ln y_i!\right]
\end{aligned}$$

The residual sum of squares is:

$$S\left(\beta\right) = \sum_{i=1}^{n}\left(y_i - e^{\beta' x_i}\right)^2$$

1. Estimate the parameter vector β using maximum likelihood. Use as the starting value a vector of zeros. Use four algorithms

1. Quasi-Newton with BFGS and a numerical derivative.
2. Quasi-Newton with BFGS and an analytical derivative.
3. Nelder-Mead.
4. The BHHH algorithm we went over in class. (This is the only one you should code up yourself. You can use packages for the others.)

Report, in a table, the estimated parameters, number of iterations, number of function evaluations, and time, for each method.

Answer

1. Quasi-Newton with BFGS and a numerical derivative.

$$ln(L) = \sum_{i=1}^{n} [-\lambda_i + y_i \ln \lambda_i - \ln y_i!]$$

where $\lambda_i = exp(\mathbf{x}_i\prime\beta)$

2. Quasi-Newton with BFGS and an analytical derivative.

Here we calculate the analytical derivative of the Log-likelihood:

$$\frac{\partial lnL(\beta)}{\partial \beta} = \sum_{i=1}^{n} (y_i - \lambda_i \mathbf{x}_i)$$

3. Nelder-Mead.

Evaluate the objective function at the vertices of a simplex and iteratively update the simplex to converge to the minimum.

4. We went over the BHHH algorithm in class. (This is the only one you should code up yourself. You can use packages for the others.)

Let the score function be the gradient of the log-likelihood function with respect to the parameters β, evaluated for each observation.

$$g_i(\beta) = \frac{\partial lnL_i(\beta)}{\partial \beta}$$

The information matrix is the negative of the expected value of the Hessian of the Log-likelihood function.

$$I(\beta) = -E\left[\frac{\partial^2 lnL(\beta)}{\partial \beta \partial \beta'}\right]$$

This information matrix can also be expressed as the expectation of the outer product of the score function:

$$I(\beta) = E[g_i(\beta)g_i(\beta)']$$

The BHHH method approximates this information matrix using the sample average of the outer product of the score function

$$H(\beta) \simeq \sum_{i=1}^{n} [g_i(\beta) g_i(\beta)']$$

Finally, we use the standard update rule.

The following table shows a comparison table between all four algorithms.

- All models show similar estimates, therefore the difference relies in the performance metrics as I explain above.
- We can see that QNN, and QNA are quite efficient in terms of iterations in comparison to NM and BHHH.
- In terms of time, BHHH is superior to all other models , and QNN is the most expensive one.

Table 1: Comparison Table between Linear Models

Parameter String	**QNN** Float64	**QNA** Float64	**NM** Float64	**BHHH** Float64
Constant	2.52532	2.52532	2.52533	2.52532
Age	-0.0321881	-0.0321881	-0.0321883	-0.0321881
Years Married	0.115591	0.115591	0.115591	0.115591
Religiousness	-0.35358	-0.35358	-0.35358	-0.35358
Occupation	0.0811116	0.0811116	0.0811117	0.0811116
Marriage Rating	-0.408781	-0.408781	-0.40878	-0.408781
Number of Iterations	15.0	15.0	382.0	187.0
Function Evaluations	92.0	91.0	638.0	187.0
Elapsed Time (seconds)	0.337935	0.09553	0.0130792	0.007

2. Report the eigenvalues for the Hessian approximation for the BHHH MLE method from the last question. Report the eigenvalues for the initial Hessian approximation, and the Hessian at the estimated parameters.
Answer

We want to know about the eigenvalues because they tell us about the curvature off the likelihood function, the next two tables present the Hessians and the Eigenvalues at those two different points of analysis.

Table 2: Hessian Approximation and Eigenvalues at Initial Parameters

Hessian Matrix					
6653.0	231118.5	71710.003	18109.0	29157.0	21197.0
231118.5	8.49574825e6	2.688158821e6	644124.5	1.0262485e6	735435.0
71710.003	2.688158821e6	933406.846251	203231.635	316181.303	216276.805
18109.0	644124.5	203231.635	57705.0	78173.0	55744.0
29157.0	1.0262485e6	316181.303	78173.0	147245.0	96182.0
21197.0	735435.0	216276.805	55744.0	96182.0	78859.0
Eigenvalues					
210.754593	8697.939167	9360.633156	22906.405326	82795.705309	9.5956456587e6

Table 3: Hessian Approximation and Eigenvalues at Estimated Parameters

Hessian Matrix					
5564.659877	194505.182749	58461.828529	15285.397898	24458.248339	18423.87056
194505.182749	7.266426866794e6	2.233104832432e6	550225.093979	871687.173087	641304.35046
58461.828529	2.233104832432e6	757302.674063	168185.224116	261443.273523	183930.013779
15285.397898	550225.093979	168185.224116	48747.791287	66515.416974	49049.036736
24458.248339	871687.173087	261443.273523	66515.416974	123751.862538	83288.729374
18423.87056	641304.35046	183930.013779	49049.036736	83288.729374	70706.503569
Eigenvalues					
188.750356	7045.260631	8748.788959	19740.531081	70613.283976	8.166163743124e6

3. Now estimate the model using the NLLS method we went over in class, starting from the same initial point. Report the results in the table as well.

Answer

I am following the algorithm in class: 1. Initialize θ_0 2. Define Jacobian

$$J(\theta) = \begin{pmatrix} \frac{\partial f_1}{\partial \theta_1} \cdots & \frac{\partial f_1}{\partial \theta_k} \\ \cdots & \cdots \\ \frac{\partial f_N}{\partial \theta_1} \cdots & \frac{\partial f_N}{\partial \theta_k} \end{pmatrix}$$

3. The gradient of $S(\theta) = \sum_N J(\theta)^\top f(\theta)$. 4. The Hessian is then

$$\mathcal{H}(\theta) = J(\theta)^\top J(\theta) + \sum_N f_i(\theta) \frac{\partial^2 f_i}{\partial \theta \partial \theta^T}$$

5. Approximate the Hessian by dropping the second term (to guarentee positive definite matrix).

$$\mathcal{H}(\theta) \simeq J(\theta)^\top J(\theta)$$

6. Search step is then

$$d = -\left[J(\theta)^\top J(\theta)\right]^{-1} J(\theta)^\top f(\theta)$$

7. Repeat until convergence

The following Table include an additional last column showing the results from the Non-Liner Algorithm.

- The non-linear model NLLS is efficient in iterations and similar to QNN, and QNA
- in terms of time efficiency, NLLS is also more convenient in comparison to the rest of models.
- A potential drawback can be the fact that some estimates with the NLLS are not that close to the rest of the models. It's close but the other models are almost identical terms of beta estimates.

Table 4: Comparison between Linear and Non-Linear Models

Parameter	**QNN**	**QNA**	**NM**	**BHHH**	**NLLS**
String	Float64	Float64	Float64	Float64	Float64
Constant	2.52532	2.52532	2.52533	2.52532	2.5069
Age	-0.0321881	-0.0321881	-0.0321883	-0.0321881	-0.038329
Years Married	0.115591	0.115591	0.115591	0.115591	0.113949
Religiousness	-0.35358	-0.35358	-0.35358	-0.35358	-0.279351
Occupation	0.0811116	0.0811116	0.0811117	0.0811116	0.0683522
Marriage Rating	-0.408781	-0.408781	-0.40878	-0.408781	-0.368998
Number of Iterations	15.0	15.0	382.0	187.0	14.0
Function Evaluations	92.0	91.0	638.0	187.0	14.0
Elapsed Time (seconds)	0.337935	0.09553	0.0130792	0.007	0.0