Implementation Project: Bayesian Negative Binomial Regression

Bayesian Econometrics

Zachary Kiefer

June 14, 2018

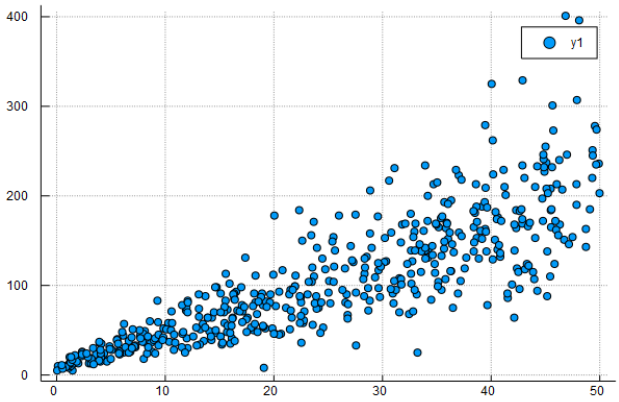
1 Introduction

Negative binomial (NB) regression presents an alternative to OLS regression and similar methods when working with count (i.e. discrete) data, and particularly when the data exhibits heteroskedasticity of a form where the conditional variance of the dependent variable is proportional to its conditional mean, as in Figure 1. Although this can also be achieved through methods such as Poisson regression, NB regression does not suffer from the restriction that the mean of the error distribution be equal to its variance[1], which can make it difficult to apply Poisson regression to data which does not exhibit this property. Additionally, the negative binomial distribution is naturally suited for modeling certain economic processes, such as the amount of time a worker remains unemployed or the number of bargaining periods before an agreement among several parties is reached.

This paper will cover a method for estimating an NB regression model using a Bayesian framework, with emphasis on doing so in a computationally efficient manner.

[1]The negative binomial distribution can be considered an overdispersed Poisson distribution, in that it allows for the variance to be greater than the mean.

Figure 1
Data Suitable for Negative Binomial Regression



2 Negative Binomial Regression

To begin, consider OLS regression: a fundamental assumption of OLS is that the observed data is produced by a data-generating process of the form:

$$y_i = x_i\beta + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2) \tag{1}$$

This could alternately be expressed in the form:

$$y_i \sim N(x_i\beta, \sigma^2) \tag{2}$$

That is, each observation of the dependent variable y is drawn from an i.i.d. normal distribution, centered on $x_i\beta$ with variance σ^2.

Similarly, in a negative binomial regression, we make the assumption that each observation of y is drawn from an i.i.d. negative binomial distribution with mean $\mu_i = x_i\beta$ and dispersion parameter r.

2.1 Negative Binomial Distribution

Consider a simple experiment in which the probability of success is denoted p. Now suppose that this experiment is repeated until it has failed a certain number of times, denoted r. The negative binomial distribution describes the probability that the experiment will

achieve k successes by the time it accumulates r failures.

2.2 Probability Density Function

The pdf of the negative binomial distribution, following the parameterization described above, is:

$$P(k|p,r) = \binom{k+r-1}{k} (1-p)^r p^k \tag{3}$$

In this parameterization, the mean of the distribution is $m = \frac{pr}{1-p}$. Because we wish to describe a data-generating process in which the mean of an NB distribution is defined by the data, it becomes more convenient to use an alternate parameterization using this m and the previously-defined r. Solving for p in the equation for m, we find that $p = \frac{m}{m+r}$, giving us the pdf for this alternative parameterization:

$$P(k|m,r) = \frac{\Gamma(k+r)}{k!\Gamma(r)} \left(\frac{r}{r+m}\right)^r \left(\frac{m}{r+m}\right)^k \tag{4}$$

2.3 Likelihood Function

Given a set of observed data $Y = [y_1, y_2, \ldots, y_N]'$ and $X = [x_1', x_2', \ldots, x_N']'$, in which $x_i = [x_{i1}, x_{i2}, \ldots, x_{iK}]$, and parameters β and r, the likelihood of having observed this data is:

$$P(Y|\beta,r) = \prod_{i=1}^{N} P(y_i|\beta,r) \tag{5}$$

For ease of calculation, we will resort to the log-likelihood function:

$$P(Y|\beta, r) = \sum_{i=1}^{N} logP(y_i|\beta, r) \tag{6}$$

$$= \sum_{i=1}^{N} \log\Gamma(r + y_i) - \log(y_i!) - \log\Gamma(r) + y_i \log(\frac{m_i}{r + m_i}) + r\log(\frac{r}{r + m_i}) \tag{7}$$

in which $m_i = x_i\beta$.

3 Posterior Probability Sampling

Because the posterior of a negative binomial regression is not conducive to analytical solutions, we shall instead sample the posterior distribution using a Metropolis-Hastings algorithm. This will require calculation of the acceptance ratio,

$$\alpha = \min\left(\frac{P(Y|\theta^*)}{P(Y|\theta^g)}\frac{P(\theta^*)}{P(\theta^g)}\frac{q(\theta^g|\theta^*)}{q(\theta^*|\theta^g)}, 1\right) \tag{8}$$

in which $\theta = \{\beta, r\}$.

3.1 Priors

For simplicity, we can use an uninformative prior which places equal weight upon all possible values of θ. This can be thought of as a continuous uniform distribution, over an arbitrarily large support. This allows the $P(\theta^*)$ and $P(\theta^g)$ factors to cancel out.

3.2 Proposal Distribution

We use an essentially symmetric proposal distribution, again for simplicity, as it allows the proposal probabilities to cancel out. β^* can be proposed using a multivariate normal distribution of appropriate dimensionality, centered on β^g and having an arbitrary variance-covariance matrix[2].

[2]This matrix can be calibrated to result in a desirable acceptance rate.

r, on the other hand, must necessarily be positive, and thus we propose it (independently of β) using a normal distribution which has been left-truncated at 0.

3.3 Acceptance Ratio

With both the prior and proposal factors canceling out, the acceptance probability becomes:

$$\alpha = \min\left(\frac{P(Y|\theta^*)}{P(Y|\theta^g)}, 1\right) \tag{9}$$

The likelihood ratio in this formula can be constructed by first employing the log-likelihood functions defined above:

$$\begin{aligned}
\log\left(\frac{P(Y|\theta^*)}{P(Y|\theta^g)}\right) &= \log P(Y|\theta^*) - \log P(Y|\theta^g) \\
&= \sum_{i=1}^{N}\left[\log\Gamma(r^* + y_i) - \log(y_i!) - \log\Gamma(r^*) + y_i \log\left(\frac{m_i^*}{r^* + m_i^*}\right) + r^* \log\left(\frac{r^*}{r^* + m_i^*}\right)\right] \\
&\quad - \sum_{i=1}^{N}\left[\log\Gamma(r^g + y_i) - \log(y_i!) - \log\Gamma(r^g) + y_i \log\left(\frac{m_i^g}{r^g + m_i^g}\right) + r^g \log(\frac{r^g}{r^g + m_i^g})\right]
\end{aligned}$$

The terms containing only y_i cancel, since this is not parameter-dependent. Other terms can be combined, leaving us with:

$$\begin{aligned}
\log\left(\frac{P(Y|\theta^*)}{P(Y|\theta^g)}\right) = \sum_{i=1}^{N}\Bigg[&\log\Gamma(r^* + y_i) - \log\Gamma(r^g + y_i) + r^* \log\left(\frac{r^*}{r^* + m_i^*}\right) - r^g \log\left(\frac{r^g}{r^g + m_i^g}\right) \\
&+ y_i \log\left(\frac{m_i^*}{m_i^g}\frac{r^g + m_i^g}{r^* + m_i^*}\right)\Bigg] + N\left[\log\Gamma(r^g) - \log\Gamma(r^*)\right]
\end{aligned}$$

This logged likelihood ratio (distinct from a log-likelihood ratio) can then be used to generate the acceptance ratio.

3.4 Notes on Computational Considerations

When implementing the Metropolis-Hastings algorithm as outlined above, certain complications may present themselves. Common concerns, and remedies to them, are presented here:

3.4.1 Informative Priors and Non-Symmetric Proposals

If an informative prior or a non-symmetric proposal distribution is desired, it can be implemented with minimal additional complexity, and without altering the calculation of the logged likelihood ratio. The example code accompanying this writeup already incorporates functions to emulate the prior and proposal components of the acceptance ratio: since this code is implemented using an uninformative prior and symmetric proposal, both functions return 1 by default, but they can easily be altered to perform more informative calculations.

3.4.2 Requirement for Positive Means

It is not necessary for any particular element of β to be positive, but β as a whole must be such that $x_i\beta > 0 \ \forall \ i$. In practice, this may not be an issue, as the acceptance ratio becomes small when $x_i\beta$ approaches 0, making it unlikely that the random-walk proposals will result in it becoming negative. If this requirement does become an issue (e.g. if the proposal distributions have large variance, or if the algorithm is allowed to run for a large number of iterations), it may be beneficial to instead specify $\log m_i = x_i\beta$, which removes the need to alter any proposal distributions.

3.4.3 Efficient Calculation of Acceptance Ratio

An effective method for computing the logged likelihood ratio is to exploit vectorized functions, a feature of many mathematical software packages. Each of the five terms inside the sum in the log-likelihood ratio can be rapidly computed as a vector of values, each element corresponding to an observation of the data. These five vectors can be added together, then summed element-wise, to produce the value of the sum. The sixth term (outside of the sum) can then be added as well.

3.4.4 Efficient Computation of log-Gamma Functions

While the log-Gamma function can be computed in a brute-force manner, i.e. by first computing the Gamma function, then taking its logarithm, most mathematical software will have a dedicated log-Gamma function. Aside from being more computationally efficient, the dedicated log-Gamma function is less prone to rounding errors which may result from taking the Gamma function of large values.

3.4.5 Determination of Acceptance Decision

Finally, once the logged likelihood ratio is computed, it is straightforward to compute α and compare this value with a value drawn from the $U(0,1)$ distribution to make the acceptance decision. However, this is not necessary: a mathematically equivalent approach is to draw a value of the random variable $Z \sim Exp(1)$, and accept the proposal if $-Z$ is less than the logged likelihood ratio.