Extended Negative Binomial 2 Regression

Julian Granna
Universität Innsbruck

Abstract

The **enbin** package (https://R-Forge.R-project.org/projects/uibk-rprog-2017/) fits negative binomial (NB2) regression models allowing for a non-constant θ using analytical gradient based maximum likelihood estimation. An overview of the underlying model and its implementation in the package is provided, along with some illustrations.

Keywords: negative binomial, NB2, count data, R.

1. Introduction

In accordance with Winkelmann (2013), negative binomial models account for unobserved heterogeneity in the data. The problem of possible unobserved heteogeneity in the data can be shown formally as derived by Schmetterer (1978): The Poisson parameter may be expressed as

$$\tilde{\lambda}_i = exp(x_i'\beta + \epsilon_i), \tag{1}$$

where ϵ_i gives the unobserved heterogeneity. $\tilde{\lambda}_i$ can now be rewritten as

$$\tilde{\lambda}_i = exp(x_i'\beta)\ exp(\epsilon_i) = exp(x_i'\beta)u_i = \lambda_i u_i. \tag{2}$$

Now, the mean and variance can be derived as

$$\mathsf{E}(y_i|x_i) = \mathsf{E}_u(\tilde{\lambda}_i|x_i) = exp(x_i'\beta)\ \mathsf{E}(u_i|x_i) = \lambda_i, \tag{3}$$

$$Var(y_i|x_i) = \mathsf{E}_u(\tilde{\lambda}_i|x_i) + Var(\tilde{\lambda}_i|x_i) = \lambda_i \sigma_u^2 \lambda_i^2. \tag{4}$$

With $\sigma_u^2 > 0$, it follows that $Var(y_i|x_i) > \mathsf{E}(y_i|x_i)$. Negative binomial models can be applied to assess this issue. In this application, a negative binomial 2 model (NB2) is employed with the conditional expectation function

$$\mathsf{E}(y_i|x_i) = exp(x_i'\beta) = exp(\eta_{\mu,i}) \tag{5}$$

and scale function

$$Var(y_i|x_i) = \mu_i + \alpha \cdot \mu_i^2, \tag{6}$$

where α could be taken as constant with $\alpha = \theta^{-1}$. This package also allows for a non-constant θ_i with

$$\theta_i = exp(z_i'\gamma) = \eta_{\theta,i}. \tag{7}$$

This feature provides the major improvement towards other packages involving NB2 models.

2. Implementation

The main model fitting function `enbin()` uses a formula-based interface and returns an (S3) object of class `enbin`:

```
enbin(formula, data, subset, na.action,
  model = TRUE, y = TRUE, x = FALSE,
  control = enbin_control(...), ...)
```

The underlying workhorse function, which is usually not called, is `enbin_fit()`. It features a matrix interface and returns an unclassed list.

Various S3 methods are provided, see Table 1.

Method	Description
`print()`	Simple printed display with coefficients
`summary()`	Standard regression summary; returns `summary.enbin` object (with `print()` method)
`coef()`	Extract coefficients
`vcov()`	Associated covariance matrix
`predict()`	(Different types of) predictions for new data
`fitted()`	Fitted values for observed data
`residuals()`	Extract (different types of) residuals
`terms()`	Extract terms
`model.matrix()`	Extract model matrix (or matrices)
`nobs()`	Extract number of observations
`logLik()`	Extract fitted log-likelihood
`bread()`	Extract bread for **sandwich** covariance
`estfun()`	Extract estimating functions (= gradient contributions) for **sandwich** covariances
`getSummary()`	Extract summary statistics for `mtable()`

Table 1: S3 methods provided in **enbin**.

These included methods allow for a broad variety of utilities to work automatically, e.g., `AIC()`, `BIC()`, `coeftest()` (**lmtest**), `lrtest()` (**lmtest**), `waldtest()` (**lmtest**), `linearHypothesis()` (**car**), `mtable()` (**memisc**), etc.

3. Illustration and Replication

To show the usefulness of the package in practice, the `enbin()`-function is applied to the `RecreationDemand` dataset from the **AER**-package. At first, a negative binomial model is computed employing the `glm.nb()`-function from the **MASS**-package and its output is compared with the one from the `enbin`-package to assess its accuracy:

```
R> library(MASS)
R> data("RecreationDemand", package = "AER")
R> m1 <- glm.nb(trips ~ ., data = RecreationDemand)
R> summary(m1)
```

```
Call:
glm.nb(formula = trips ~ ., data = RecreationDemand, init.theta = 0.7292568331,
    link = log)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.9727  -0.6256  -0.4619  -0.2897   5.0494

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.121936   0.214303  -5.235 1.65e-07 ***
quality      0.721999   0.040117  17.998  < 2e-16 ***
skiyes       0.612139   0.150303   4.073 4.65e-05 ***
income      -0.026059   0.042453  -0.614    0.539
userfeeyes   0.669168   0.353021   1.896    0.058 .
costC        0.048009   0.009185   5.227 1.72e-07 ***
costS       -0.092691   0.006653 -13.931  < 2e-16 ***
costH        0.038836   0.007751   5.011 5.42e-07 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

(Dispersion parameter for Negative Binomial(0.7293) family taken to be 1)

    Null deviance: 1244.61  on 658  degrees of freedom
Residual deviance:  425.42  on 651  degrees of freedom
AIC: 1669.1

Number of Fisher Scoring iterations: 1


              Theta:  0.7293
          Std. Err.:  0.0747

 2 x log-likelihood:  -1651.1150
```

As the variable `income` is not significantly different from Zero, another model is fit, where the variable is left out.

```
R> m2 <- glm.nb(trips ~ . - income, data = RecreationDemand)
R> summary(m2)

Call:
glm.nb(formula = trips ~ . - income, data = RecreationDemand,
    init.theta = 0.7263941439, link = log)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.9745  -0.6335  -0.4626  -0.2812   5.1072

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.206658   0.165071  -7.310 2.67e-13 ***
quality      0.723457   0.040176  18.007  < 2e-16 ***
skiyes       0.599777   0.147323   4.071 4.68e-05 ***
userfeeyes   0.668006   0.353546   1.889   0.0588 .
costC        0.047652   0.009210   5.174 2.29e-07 ***
costS       -0.093291   0.006629 -14.074  < 2e-16 ***
costH        0.039536   0.007737   5.110 3.23e-07 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

(Dispersion parameter for Negative Binomial(0.7264) family taken to be 1)

    Null deviance: 1241.78  on 658  degrees of freedom
Residual deviance:  424.83  on 652  degrees of freedom
AIC: 1667.4

Number of Fisher Scoring iterations: 1


              Theta:  0.7264
          Std. Err.:  0.0743

 2 x log-likelihood:  -1651.4460

R> library(lmtest)
R> lrtest(m2, m1)

Likelihood ratio test

Model 1: trips ~ (quality + ski + income + userfee + costC + costS + costH) -
    income
```

```
Model 2: trips ~ quality + ski + income + userfee + costC + costS + costH
  #Df  LogLik Df  Chisq Pr(>Chisq)
1   8 -825.72
2   9 -825.56  1 0.3309     0.5651
```

The likelihood ratio test also does not reject the null hypothesis, so `income` is not considered subsequently. One could further investigate, which variables should possibly be excluded (such as `userfees`), but this is neglected here, as it is not of special interest. $\theta = 0.7264$ indicates significant unobserved heterogeneity in the data. To compare the `summary`-output of `glm.nb()` from **MASS** with this package's output, the same model is fit utilizing the `enbin()`-function from `enbin`:

```
R> library(enbin)
R> m3 <- enbin(trips ~ . - income, data = RecreationDemand)
R> summary(m3)

Call:
enbin(formula = trips ~ . - income, data = RecreationDemand)

Standardized residuals:
       Min         1Q     Median         3Q        Max
-5536.7830    -0.7707    -0.1871    -0.0854   113.3092

Coefficients (location model with log link):
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.206658   0.166481  -7.248 4.23e-13 ***
quality      0.723457   0.045417  15.929  < 2e-16 ***
skiyes       0.599777   0.149153   4.021 5.79e-05 ***
userfeeyes   0.668007   0.361934   1.846 0.064942 .
costC        0.047652   0.015972   2.983 0.002850 **
costS       -0.093291   0.008243 -11.318  < 2e-16 ***
costH        0.039536   0.011666   3.389 0.000702 ***

Coefficients (scale model with log link):
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.3197     0.1057  -3.024  0.00249 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log-likelihood: -825.7 on 8 Df
Number of iterations in BFGS optimization: 17
```

It is apparent that the estimated coefficients match the ones obtained by `glm.nb`. Further, the intercept in the model is also clearly significant and in the univariate scale model, the constant θ can be computed by taking $exp(-0.320) = 0.726$, which is due to the log link in the scale model.

Now, in order to point out the major impovement of this package, another model is fit, where the scale depends on covariates as well:

```
R> m4 <- enbin(trips ~ . - income | . - income, data = RecreationDemand)
R> summary(m4)

Call:
enbin(formula = trips ~ . - income | . - income, data = RecreationDemand)

Standardized residuals:
          Min            1Q        Median            3Q           Max
-1878570.6205       -8.4139       -3.7922       -0.6384      445.7305

Coefficients (location model with log link):
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.353134   0.346859  -1.018  0.30863
quality      0.460530   0.081098   5.679 1.36e-08 ***
skiyes       0.563404   0.135109   4.170 3.05e-05 ***
userfeeyes   0.666164   0.214216   3.110  0.00187 **
costC        0.052663   0.011209   4.698 2.62e-06 ***
costS       -0.071861   0.008730  -8.231  < 2e-16 ***
costH        0.013280   0.006374   2.083  0.03722 *

Coefficients (scale model with log link):
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.406096   0.426104  -7.994 1.31e-15 ***
quality      0.845400   0.115232   7.337 2.19e-13 ***
skiyes      -0.450438   0.242429  -1.858 0.063166 .
userfeeyes   1.137410   0.525802   2.163 0.030526 *
costC       -0.078090   0.022261  -3.508 0.000452 ***
costS        0.016792   0.008195   2.049 0.040464 *
costH        0.070411   0.020283   3.471 0.000518 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log-likelihood:  -787 on 14 Df
Number of iterations in BFGS optimization: 30

R> AIC(m3, m4)

   df      AIC
m3  8 1667.446
m4 14 1602.006
```

```
R> BIC(m3, m4)

   df      BIC
m3  8 1703.372
m4 14 1664.876

R> lrtest(m3, m4)

Likelihood ratio test

Model 1: trips ~ . - income
Model 2: trips ~ . - income | . - income
  #Df  LogLik Df Chisq Pr(>Chisq)
1   8 -825.72
2  14 -787.00  6 77.44  1.206e-14 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

As can be seen in the output, letting the scale depend on covariates proves to be useful in terms of the regarded model selection criteria. Both AIC and BIC prefer the less restrictive variant of the model. The same holds for the likelihood ratio test.

References

Schmetterer L (1978). *Introduction to mathematical statistics*, volume 4. Macmillan.

Winkelmann R (2013). *Econometric Analysis of Count Data*. Springer Science & Business Media.

Affiliation:
Julian Granna
Department of Statistics
Faculty of Economics and Statistics
Universität Innsbruck
Universitätsstr. 15
6020 Innsbruck, Austria
E-mail: julian.granna@uibk.ac.at