Tutorial 13: Generalised Linear Models II: 2024-01-15: 10.15 - 11.45

1. **Regression Models for Count Data** Consider the data set `RecreationDemand` from **AER**. This is data on the number of recreational boating trips in Lake Somerville, Texas, in 1980, based on a survey administered to 2 000 registered boat users in 23 counties in eastern Texas. The dependent variable is `trips` and we want to regress on it all the other variables; quality ranking factor (`quality`), whether or not engaged in water skiing (`ski`), household income (`income`), whether the individual paid a user fee (`userfee`) and three cost variables (`costC, costS, costH`) representing opportunity costs.

   ```
   > data("RecreationDemand")
   > rd_pois <- glm(trips~.,data=RecreationDemand, family=poisson)
   > coeftest(rd_pois)
   ```

   Note that all the regression variables seem highly significant (except for `CostC` - expenditure when visiting Lake Conroe).

2. **Overdispersion** We continue with the previous example, about `RecreationDemand`. The Poisson distribution has the property that mean and variance are equal. This is called *equidispersion*. Many data sets exhibit *overdispersion*, where the variance is greater than the mean. Let

   $$\text{Var}(Y_i|x_i) = (1+\alpha)\mu_i = \text{dispersion}.\mu_i.$$

   The package **AER** provides a function `dispersiontest()` to test equidispersion ($\alpha = 0$) versus overdispersion ($\alpha > 0$). If the argument `trafo` is specified, the test is formulated in terms of the parameter α. Try

   ```
   > dispersiontest(rd_pois)
   ```

   and

   ```
   > dispersiontest(rd_pois,trafo=2)
   ```

   Look up the meaning of the commands and interpret the results. Both indicate overdispersion. The meaning of `trafo` here is that it considers a model

   $$\text{Var}(Y) = \mu + \alpha\mu^2$$

   (here `trafo` is 2). It concludes that the α parameter is significant and estimates it as 1.32.

3. **Negative Binomial** Now consider fitting a negative binomial model to the `RecreationDemand` data. This may be carried out using the `glm.nb()` command from the package **MASS**.

```
> library("MASS")
> rd_nb <- glm.nb(trips~.,data=RecreationDemand)
> coeftest(rd_nb)
```

The *shape* parameter estimate is $\widehat{\theta} = 0.7293$, suggesting considerable overdispersion. Recall the probability function for negative binomial:

$$p(y;\mu,\theta) = \frac{\Gamma(\theta+y)}{\Gamma(\theta)y!}\frac{\mu^y\theta^\theta}{(\mu+\theta)^{y+\theta}} \qquad y = 0,1,2,\ldots$$

where $\mu > 0$ and $\theta > 0$. It is written like this, for comparison with Poisson. Its variance is

$$\mathrm{Var}(y;\mu,\theta) = \mu + \frac{1}{\theta}\mu^2,$$

so that $\theta = +\infty$ is associated with equidispersion. We may now compare the log likelihood values:

```
> logLik(rd_pois)
> logLik(rd_nb)
```

The output suggests that the negative binomial model gives a better fit. From this, compute the Akaike Information Criterion for both models.

4. **Attendance behaviour of high school juniors at two schools** Predictors of the number of days of absence include the type of programme in which the student is enrolled and a standardized test in math.

   The data set `nb_data.dta` (found in the course directory) contains attendance data for 314 high school juniors from two urban high schools. The response variable of interest is days absent, which is `daysabs`. The variable `math` gives the standardised maths score for each student. The variable `prog` is a three-level nominal variable indicating the type of instructional program in which the student is enrolled.

   Let us look at the data. It is always a good idea to start with descriptive statistics and plots.

```
dat <-
read.dta("https://www.mimuw.edu.pl/~noble/courses/
MultivariateStatistics/data/nb_data.dta")

dat <- within(dat, {
    prog <- factor(prog, levels = 1:3, labels = c("General",
"Academic", "Vocational"))
```

```
    id <- factor(id)
})

summary(dat)
ggplot(dat, aes(daysabs, fill = prog)) + geom_histogram(binwidth = 1)
+ facet_grid(prog ~
    ., margins = TRUE, scales = "free")
```

Each variable has 314 valid observations and their distributions seem quite reasonable. The unconditional mean of our outcome variable is much lower than its variance.

Let us continue with our description of the variables in this dataset. The table produced by the command below shows the average numbers of days absent by programme type and seems to suggest that programme type is a good candidate for predicting the number of days absent, our outcome variable, because the mean value of the outcome appears to vary by `prog`. The variances within each level of `prog` are higher than the means within each level. These are the conditional means and variances. These differences suggest that over-dispersion is present and that a Negative Binomial model would be appropriate.

```
with(dat, tapply(daysabs, prog, function(x) {
    sprintf("M (SD) = %1.2f (%1.2f)", mean(x), sd(x))
}))
```

The following are some analysis methods that may be considered. Some of the methods listed are quite reasonable, while others have either fallen out of favor or have limitations.

- **Negative binomial regression** Negative binomial regression can be used for over-dispersed count data, that is when the conditional variance exceeds the conditional mean. It can be considered as a generalisation of Poisson regression since it has the same mean structure as Poisson regression and it has an extra parameter to model the over-dispersion. If the conditional distribution of the outcome variable is over-dispersed, the confidence intervals for the Negative binomial regression are likely to be wider as compared to those from a Poisson regression model.
- **Poisson regression** Poisson regression is often used for modeling count data. Poisson regression has a number of extensions useful for count models.
- **Zero-inflated regression model** Zero-inflated models attempt to account for excess zeros. In other words, two kinds of zeros are thought to exist in the data, 'true zeros' and 'excess zeros'. Zero-inflated models estimate two equations simultaneously, one for the count model and one for the excess zeros.

- **OLS regression** Count outcome variables are sometimes log-transformed and analysed using OLS regression. Many issues arise with this approach, including loss of data due to undefined values generated by taking the log of zero (which is undefined), as well as the lack of capacity to model the dispersion.

Negative binomial regression analysis

The `glm.nb` function from the MASS package estimates a negative binomial regression.

```
summary(m1 <- glm.nb(daysabs ~ math + prog, data = dat))
```

R first displays the call and the *deviance residuals*. Next, we see the regression coefficients for each of the variables, along with standard errors, z-scores, and p-values. The variable `math` has a coefficient of -0.006, which is statistically significant. This means that for each one-unit increase in math, the expected log count of the number of days absent decreases by 0.006. The indicator variable shown as **progAcademic** is the expected difference in log count between group 2 and the reference group (prog=1). The expected log count for level 2 of prog is 0.44 lower than the expected log count for level 1. The indicator variable for **progVocational** is the expected difference in log count between group 3 and the reference group.The expected log count for level 3 of prog is 1.28 lower than the expected log count for level 1. To determine if `prog` itself, overall, is statistically significant, we can compare a model with and without `prog`. The reason it is important to fit separate models, is that unless we do, the overdispersion parameter is held constant.

```
m2 <- update(m1, . ~ . - prog)
anova(m1, m2)
```

- The two degree-of-freedom chi-square test indicates that `prog` is a statistically significant predictor of `daysabs`.
- The null deviance is calculated from an intercept-only model with 313 degrees of freedom. Then we see the residual deviance, the deviance from the full model. We are also shown the AIC and 2*log likelihood.
- The theta parameter shown is the dispersion parameter.

Checking model assumption The negative binomial model assumes the conditional means are not equal to the conditional variances. This inequality is captured by estimating a dispersion parameter (not shown in the output) that is held constant in a Poisson model. Thus, the Poisson model is actually nested in the negative binomial model. We can then use a likelihood ratio test to compare these two and test this model assumption. To do this, we will run our model as a Poisson.

```
m3 <- glm(daysabs ~ math + prog, family = "poisson", data = dat)
pchisq(2 * (logLik(m1) - logLik(m3)), df = 1, lower.tail = FALSE)
```

In this example the associated chi-squared value estimated from 2*(logLik(m1) – logLik(m3)) is 926.03 with one degree of freedom. This strongly suggests the negative binomial model, estimating the dispersion parameter, is more appropriate than the Poisson model.

We can get the confidence intervals for the coefficients by profiling the likelihood function.

```
(est <- cbind(Estimate = coef(m1), confint(m1)))
```

We might be interested in looking at incident rate ratios rather than coefficients. To do this, we can exponentiate our model coefficients. The same applies to the confidence intervals.

```
exp(est)
```

The output indicates that the incident rate for prog = 2 is 0.64 times the incident rate for the reference group (prog = 1). Likewise, the incident rate for prog = 3 is 0.28 times the incident rate for the reference group holding the other variables constant. The percent change in the incident rate of daysabs is a 1% decrease for every unit increase in math.

The form of the model equation for negative binomial regression is the same as that for Poisson regression. The log of the expected outcome is predicted with a linear combination of the predictors:

$$ln(\widehat{daysabs}_i) = Intercept + b_1 I(prog_i = 2) + b_2 I(prog_i = 3) + b_3 math_i$$

where $I(prog_i = j)$ is an indicator function such that if $prog_i = j$ is equal to 1 and otherwise is equal to 0, for $j \in \{2, 3\}$. Therefore,

$$\widehat{daysabs}_i = e^{Intercept + b_1 I(prog_i=2) + b_2 I(prog_i=3) + b_3 math_i} = e^{Intercept} e^{b_1 I(prog_i=2)} e^{b_2 I(prog_i=3)} e^{b_3 math_i}$$

The coefficients have an additive effect in the $ln(y)$ scale and the IRR have a multiplicative effect in the y scale. The dispersion parameter in negative binomial regression does not affect the expected counts, but it does affect the estimated variance of the expected counts.

For assistance in further understanding the model, we can look at predicted counts for various levels of our predictors. Below we create new datasets with values of math and prog and then use the predict command to calculate the predicted number of events.

First, we can look at predicted counts for each value of prog while holding math at its mean. To do this, we create a new dataset with the combinations of prog and math for which we would like to find predicted values, then use the predict command.

```
newdata1 <- data.frame(math = mean(dat$math), prog = factor(1:3,
levels = 1:3,
    labels = levels(dat$prog)))
newdata1$phat <- predict(m1, newdata1, type = "response")
newdata1
```

In the output above, we see that the predicted number of events (e.g., days absent) for a general program is about 10.24, holding math at its mean. The predicted number of events for an academic program is lower at 6.59, and the predicted number of events for a vocational program is about 2.85.

Below we will obtain the mean predicted number of events for values of math across its entire range for each level of prog and graph these.

```
newdata2 <- data.frame(
  math = rep(seq(from = min(dat$math), to = max(dat$math), length.out
= 100), 3),
  prog = factor(rep(1:3, each = 100), levels = 1:3, labels =
  levels(dat$prog)))

newdata2 <- cbind(newdata2, predict(m1, newdata2, type = "link",
se.fit=TRUE))
newdata2 <- within(newdata2, {
  DaysAbsent <- exp(fit)
  LL <- exp(fit - 1.96 * se.fit)
  UL <- exp(fit + 1.96 * se.fit)
})

ggplot(newdata2, aes(math, DaysAbsent)) +
  geom_ribbon(aes(ymin = LL, ymax = UL, fill = prog), alpha = .25) +
  geom_line(aes(colour = prog), size = 2) +
  labs(x = "Math Score", y = "Predicted Days Absent")
```

The graph shows the expected count across the range of math scores, for each type of program along with 95 percent confidence intervals. Note that the lines are not straight because this is a log linear model, and what is plotted are the expected values, not the log of the expected values.

Things to consider

- It is not recommended that negative binomial models be applied to small samples.

- One common cause of over-dispersion is excess zeros by an additional data generating process. In this situation, zero-inflated model should be considered.
- If the data generating process does not allow for any 0's (such as the number of days spent in the hospital), then a zero-truncated model may be more appropriate.
- Count data often have an exposure variable, which indicates the number of times the event could have happened. This variable should be incorporated into the negative binomial regression model with the use of the `offset` option. See the `glm` documentation for details.
- The outcome variable in a negative binomial regression cannot have negative numbers.
- You will need to use the `m1$resid` command to obtain the residuals from our model to check other assumptions of the negative binomial model.

5. Consider the `DoctorVisits` data in the **AER** package. Use a Poisson regression for the number of visits. Is the Poisson model satisfactory? If not, where are the problems and what can be done about them?