Poisson Response - Categorical Data

Michael Winton

June 6, 2018

Poisson Responses

Poisson responses can be used to model discrete "counts" in a fixed time. The model assumes (1) all counts taken on some process have the same underlying intensity, and (2) the period of time for each observation is constant. The Poisson PMF is:

$$P(Y = y) = \frac{e^{-\mu}\mu^y}{y!}$$

Y is a random variable representing number of occurrences of an event, y is a possible outcome, and the parameter $\mu > 0$ (always, because it's a count). $E(Y) = \mu$ and also $var(Y) = \mu$. This is a nice model, but in reality, we often have actual variability $> \mu$; this is called *overdispersion.*

The shape of the Poisson distribution actually approaches that of a normal distribution as μ grows.

Similar to normal variables, linear combinations of Poisson random variables are also Poisson random variables. If our Y observations $(Y_1, Y_2, ...Y_n)$ are independent with distribution $Po(\mu_k)$, then we also have:

$$\sum_{k=1}^{n} Y_k \sim Po(\sum_{k=1}^{n} \mu_k)$$

If they all have the same mean, then this becomes $Po(n\mu)$.

The likelihood function is:

$$L(\mu|y_1, ...y_n) = \prod_{k=1}^{n} \frac{e^{-\mu}\mu^{y_k}}{y_k!}$$

The MLE for μ is the sample mean:

$$\hat{\mu} = \frac{\sum_{k=1}^{n} y_k}{n}$$

The MLE for variance is:

$$var(\hat{\mu}) = \frac{\hat{\mu}}{n}$$

Confidence Intervals

We can use the Wald test statistic with the Poisson MLE for variance. Wald test statistic:

$$Z_0 = \frac{\hat{\mu} - \mu_0}{\sqrt{\hat{\mu}/n}}$$

Wald CI:

$$\hat{\mu} \pm Z_{1-\alpha/2}\sqrt{\hat{\mu}/n}$$

The (Wilson) Score test statistic is derived from hypothesis tests of $H_0 : \mu = \mu_0$ vs. $H_a : \mu \neq \mu_0$, which is tested with the following Score statistic:

$$Z_0 = \frac{\hat{\mu} - \mu_0}{\sqrt{\mu_0/n}}$$

Inverting the Score statistic gives us the Score CI:

$$\left(\hat{\mu} + \frac{Z^2_{1-\alpha/2}}{2n}\right) \pm Z_{1-\alpha/2}\sqrt{\frac{\hat{\mu} + (Z^2_{1-\alpha/2})/4n}{n}}$$

Compared to tests based on normal distributions, using these Poisson-based methods (ie. using μ/n) allows for shorter CIs than using s^2, especially for smaller samples. They can be problematic, though, if the process generating the counts doesn't have constant intensity. In those cases, $\hat{var}(\hat{\mu}) = \hat{\mu}/n$ underestimates the true variability of the mean count, and s^2 better estimates variability due to *all* sources. In those cases t-distribution tests and CIs are more robust to deviations from the Poisson assumption.

The LR interval doesn't have a closed-form solution, and therefore isn't generally used. There is an exact interval, though:

$$\frac{\chi^2_{2n\hat{\mu},\alpha/2}}{2n} < \mu < \frac{\chi^2_{2n\hat{\mu}+1,1-\alpha/2}}{2n}$$

Choosing among these: Wald is not very good (true confidence level is usually less than stated). Exact CI tends to be excessively wide. Score CI is a good compromise.

Poisson Regression Models

To perform a regression, we use the Poisson Regression Model. This is a generalized linear model with a log-link, which guarantees $\mu > 0$.

$$log(\mu) = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p$$

In R, we use `glm(...)` with `family=poisson(link=log)`.

We no longer talk about log(odds). A c-unit increase in X has a multiplicative effect.

$$\frac{\mu(x+c)}{\mu(x)} = exp(c\beta_1)$$

The interpretation is that the percentage change in the *mean* response resulting from a c-unit change in x is $100(e^{c\beta_1} - 1)$, and is not dependent on the value of x. If other variables are in the model, we say "holding all other variables constant". For example, if $e^{c\beta_1} = 1.17$, this corresponds to a 17% increase in the response.

Inference and CIs

Standard approaches to inference in MLE are used for Poisson regression, too. We can derive Wald CIs and tests for regression parameters (and functions of them) by appealing to the large-sample normal approximation for the distributions of the $\hat{\beta}$ MLEs. Likelihood Ratio intervals are better than Wald CIs.

```
crab <- read.csv("HorseshoeCrabs.csv")
head(crab)
```

```
##   Color Spine Width Weight Sat
## 1     2     3  28.3   3.05   8
## 2     3     3  26.0   2.60   4
## 3     3     3  25.6   2.15   0
## 4     4     2  21.0   1.85   0
## 5     2     3  29.0   3.00   1
## 6     1     2  25.0   2.30   3
```

```
table(crab$Sat)  # table is a good way to look at count data
```

```
##
##  0  1  2  3  4  5  6  7  8  9 10 11 12 14 15
## 62 16  9 19 19 15 13  4  6  3  3  1  1  1  1
```

```
crab_model <- glm(formula = Sat ~ Width, family = poisson(link = log),
    data = crab)
summary(crab_model)
```

```
##
## Call:
## glm(formula = Sat ~ Width, family = poisson(link = log), data = crab)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8526  -1.9884  -0.4933   1.0970   4.9221
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.30476    0.54224  -6.095  1.1e-09 ***
## Width        0.16405    0.01997   8.216  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 632.79  on 172  degrees of freedom
## Residual deviance: 567.88  on 171  degrees of freedom
## AIC: 927.18
##
## Number of Fisher Scoring iterations: 6
```

This tells us the regression equation is:

$$log(\hat{\mu}) = -3.305 + 0.164Width$$

The Wald test tells us that the coefficient on Width is significant. We can also do a LRT to see whether coefficient on Width is significant:

```
library(car)
Anova(crab_model)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: Sat
##       LR Chisq Df Pr(>Chisq)
## Width   64.913  1  7.828e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

And we can check the profile LR interval for the coefficients:

```
confint(crab_model, level = 0.95)
```

```
## Waiting for profiling to be done...
```

```
##                  2.5 %     97.5 %
## (Intercept) -4.3662326 -2.2406858
## Width        0.1247244  0.2029871
```

Let's see the effect of a c-unit change on $\hat{\mu}$:

```
c <- 1
as.numeric(exp(c * coef(crab_model)[2]))
```

```
## [1] 1.178267
```

This tells us that a 1-unit increase in c results in a 17.8% increase in satellites.

Next let's calculate the Wald CI for $Width = 23$. As before, we calculate CI for the linear predictor, and then exponentiate it:

```
wald_ci <- function(x, model, alpha = 0.05) {
    lp_hat <- predict(model, newdata = data.frame(Width = x), type = "link",
        se = TRUE)  # need 'link' to get LP
    lower <- exp(lp_hat$fit - qnorm(1 - alpha/2) * lp_hat$se)
    upper <- exp(lp_hat$fit + qnorm(1 - alpha/2) * lp_hat$se)
    list(mu_hat = exp(lp_hat$fit), se = lp_hat$se, lower = lower, upper = upper)
}

data.frame(wald_ci(23, crab_model, 0.05))
```

```
##     mu_hat         se    lower    upper
## 1 1.597244 0.09260231 1.332135 1.915114
```

Now let's calculate Profile Likelihood interval:

```r
library(mcprofile)
```

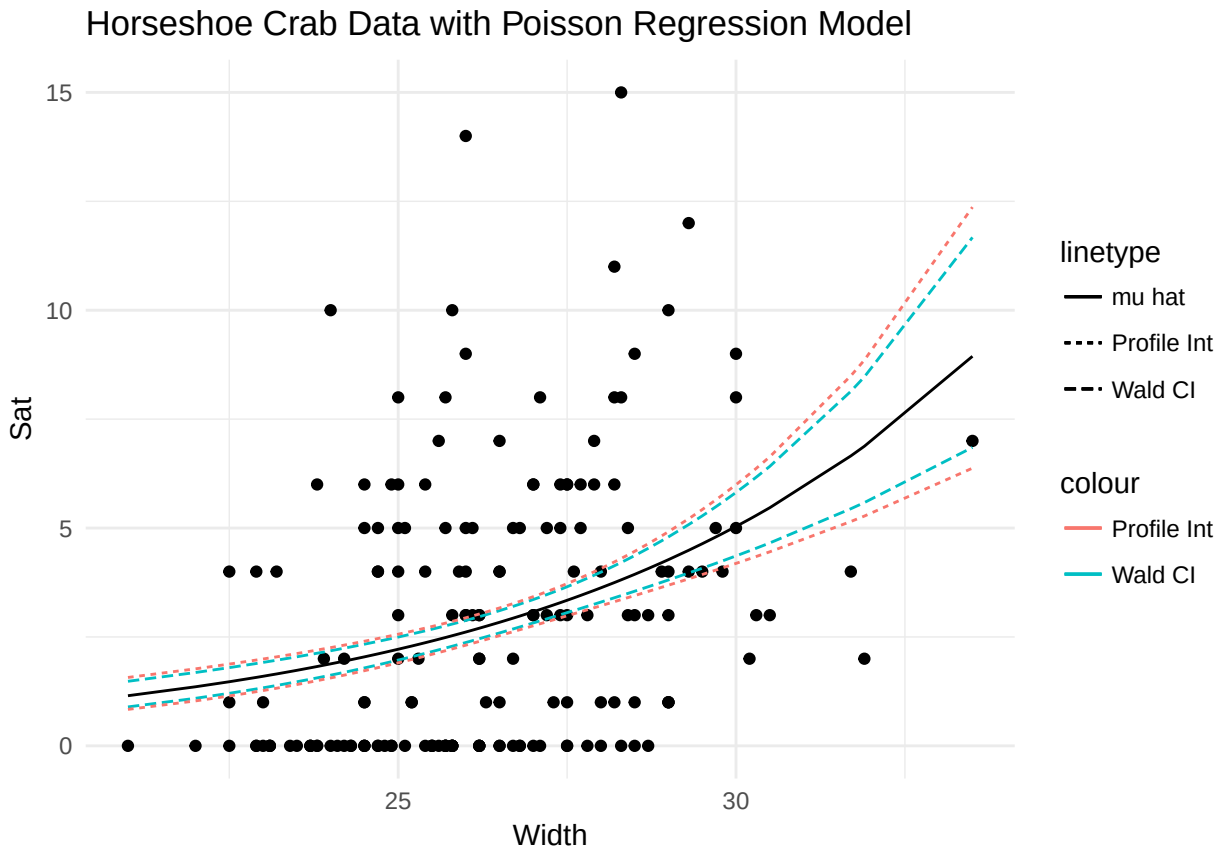
```
## Loading required package: ggplot2
```

```r
profile_ci <- function(x, model, alpha = 0.05) {
    matrix_vals <- cbind(1, x)
    K <- matrix(data = matrix_vals, nrow = length(x), ncol = 2)
    linear_combo <- mcprofile(model, CM = K)
    ci_logmu <- confint(linear_combo, level = 1 - alpha)
    exp(ci_logmu$confint)
}
data.frame(profile_ci(23, crab_model, 0.05))
```

```
##      lower    upper
## 1 1.328635 1.910106
```

Next, plot the data, fitted regression line, Wald CI, and Profile interval. (Profile interval takes ~20 seconds.)

```r
library(ggplot2)
ggplot(crab, aes(Width, Sat)) + ggtitle("Horseshoe Crab Data with Poisson Regression Model") +
    geom_point() + geom_line(aes(y = fitted(crab_model), linetype = "mu hat")) +
    geom_line(aes(Width, wald_ci(Width, crab_model, 0.05)$lower, color = "Wald CI",
        linetype = "Wald CI")) + geom_line(aes(Width, wald_ci(Width, crab_model,
    0.05)$upper, color = "Wald CI", linetype = "Wald CI")) + geom_line(aes(Width,
    profile_ci(Width, crab_model, 0.05)$lower, color = "Profile Int", linetype = "Profile Int")
    geom_line(aes(Width, profile_ci(Width, crab_model, 0.05)$upper, color = "Profile Int",
        linetype = "Profile Int")) + theme_minimal()
```



Remember, the model is trying to estimate the *average* number of satellites for a given width.

Categorical Explanatory Variables

Sometimes we may want to bin a continuous variable in order to treat it as categorical (e.g. age groups). When we have a categorical explanatory variable, we deal with it similarly to other regression models. If it have I levels, then we create $I-1$ indicator variables. To respect the base case, usually we omit the β_1 coefficient and write our equation as:

$$log(\mu) = \beta_0 + \beta_2 x_2 + \beta_3 x_3 + ...\beta_I x_I$$

Note that β_2 is the difference in log means between level 2 and level 1. As such, the coefficients are often referred to as *effect parameters* for categorical variable X. We can see that $\beta_2 = log(\mu_2) - log(\mu_1)$, which in turn tells us that $\mu_2/\mu_1 = e^{\beta_2 - 0}$. Similarly to compare any two levels, $\mu_i/\mu_{i'} = e^{\beta_i - \beta_{i'}}$.

With categorical explanatory variables, inference is usually about estimating the $\hat{\mu}$ means for each level i of X, and comparing them in some way. We can estimate means for each level as $\hat{\mu}_i = exp(\hat{\beta}_0 + \hat{\beta}_i)$.

A common test is $H_0 : \mu_1 = \mu_2 = ...\mu_i$, which is equivalent to $H_0 : \beta_2 = ...\beta_i = 0$. This is easy to do with a LRT using `Anova(...)`. To calculate LR confidence intervals for various linear combinations of parameters, use `mcprofile(...)`.