Poisson Regression

The Training Data

Office workers at a large insurance company are randomly assigned to one of 3 computer use training programmes, and their number of calls to IT support during the following month is recorded. Additional information on each worker includes years of experience and score on a computer literacy test (out of 100). It is reasonable to model calls to IT support as a Poisson process, and the question is whether training programme affects the rate of the process.

Could test H_0: $\lambda_1=\lambda_2=\lambda_3$ with a likelihood ratio test, but ...

```
> train = 
read.table("http://fisher.utstat.utoronto.ca/~brunner/appliedf12/data/train
ing.data")
> train[1:4,]
  Program Experience Score Support
1       A       3.92    60       6
2       A       5.83    64       3
3       A       0.92    51       8
4       A       8.50    58       2
> attach(train)
> table(Support)
Support
 0  1  2  3  4  5  6  7  8  9 10 11 12 
 6 27 42 61 70 39 23 17  9  2  2  1  1 
> aggregate(Support,by=list(Program),FUN=mean)
  Group.1    x
1       A 4.07
2       B 3.47
3       C 4.05
> aggregate(Support,by=list(Program),FUN=length)
  Group.1   x
1       A 100
2       B 100
3       C 100
> 
```

```
> model1 = glm(Support ~ Program, family=poisson)
> summary(model1)

Call:
glm(formula = Support ~ Program, family = poisson)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.8531  -0.6319  -0.0348   0.4552   3.1765  

Coefficients:
             Estimate Std. Error z value Pr(>|z|)    
(Intercept)  1.403643   0.049567  28.318   <2e-16 ***
ProgramB    -0.159488   0.073066  -2.183   0.0291 *  
ProgramC    -0.004926   0.070185  -0.070   0.9440    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 330.39  on 299  degrees of freedom
Residual deviance: 324.26  on 297  degrees of freedom
AIC: 1250.2

Number of Fisher Scoring iterations: 4

> anova(model1,test="Chisq") # Overall likelihood ratio test
Analysis of Deviance Table

Model: poisson, link: log

Response: Support

Terms added sequentially (first to last)


        Df Deviance Resid. Df Resid. Dev Pr(>Chi)  
NULL                      299     330.39           
Program  2    6.122       297     324.26  0.04684 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> # Include covariates
> model2 = glm(Support ~ Score+Experience+Program, family=poisson)
> summary(model2)

Call:
glm(formula = Support ~ Score + Experience + Program, family = poisson)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.9625  -0.6957  -0.1018   0.5362   2.9386  

Coefficients:
             Estimate Std. Error z value Pr(>|z|)    
(Intercept)  1.992744   0.159223  12.515  < 2e-16 ***
Score       -0.009205   0.003019  -3.049  0.00230 ** 
Experience  -0.028014   0.010317  -2.715  0.00662 ** 
ProgramB    -0.170519   0.073163  -2.331  0.01977 *  
ProgramC    -0.007833   0.070218  -0.112  0.91118    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 330.39  on 299  degrees of freedom
Residual deviance: 305.90  on 295  degrees of freedom
AIC: 1235.8

Number of Fisher Scoring iterations: 4

> anova(model2,test="Chisq") # Sequential
Analysis of Deviance Table

Model: poisson, link: log

Response: Support

Terms added sequentially (first to last)


           Df Deviance Resid. Df Resid. Dev Pr(>Chi)   
NULL                         299     330.39            
Score       1   9.9766       298     320.41 0.001585 **
Experience  1   7.6333       297     312.78 0.005730 **
Program     2   6.8767       295     305.90 0.032118 * 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> # Wald test for program
> 
> WaldTest = function(L,thetahat,Vn,h=0) # H0: L theta = h
+ # Note Vn is the asymptotic covariance matrix, so it's the
+ # Consistent estimator divided by n. For true Wald tests
+ # based on numerical MLEs, just use the inverse of the Hessian.
+      {
+      WaldTest = numeric(3)
+      names(WaldTest) = c("W","df","p-value")
+      r = dim(L)[1]
+      W = t(L%*%thetahat-h) %*% solve(L%*%Vn%*%t(L)) %*%
+           (L%*%thetahat-h)
+      W = as.numeric(W)
+      pval = 1-pchisq(W,r)
+      WaldTest[1] = W; WaldTest[2] = r; WaldTest[3] = pval
+      WaldTest
+      } # End function WaldTest
> 
> Lprog = rbind(c(0,0,0,1,0),
+               c(0,0,0,0,1) )
> WaldTest(L=Lprog,thetahat=model2$coefficients,Vn=vcov(model2))
         W         df    p-value 
6.73350088 2.00000000 0.03450157 
> # Compare G^2 = 6.8767, df=2, p=0.032118
```

Back to the Multinomial Jobs Example

Students at a trade school are either employed in their field of study, employed outside their field of study, or unemployed. University administrators recognize that the percentage of students who are unemployed after graduation will vary depending upon economic conditions, but they claim that still, about twice as many students will be employed in a job related to their field of study, compared to those who get an unrelated job. To test this hypothesis, they select a random sample of 200 students from the most recent class, and observe 106 employed in a job related to their field of study, 74 employed in a job unrelated to their field of study, and 20 unemployed.

Under a multinomial model,

- Senseless H_0: $\pi_1=\pi_2=\pi_3$ yielded $G^2 = 65.6$, df=2
- H_0: $\pi_1=2\pi_2$, $G^2 = 4.739$, df=1

Maybe the numbers of students in each category are independent Poisson RVs. Recall that independent Poissons conditional on their total are multinomial, with

$$\pi_j = \frac{\lambda_j}{\sum_{i=1}^{c} \lambda_i}$$

So what we did earlier could be justified under the present assumptions.

```
> jobz = read.table(stdin()) # Read from standard input
0:     Job         Freq
1: 1   Related     106
2: 2   Unrelated    74
3: 3   Unemployed   20
4:
> # End with Ctrl-D on Unix (Mac) or Ctrl-Z on Windows
> jobz
        Job Freq
1    Related  106
2  Unrelated   74
3 Unemployed   20
```

```
> freq = jobz$Freq
> job = factor(jobz$Job)
> contrasts(job)
           Unemployed Unrelated
Related             0         0
Unemployed          1         0
Unrelated           0         1
```

Job Status	d_1	d_2	$\log \lambda = \beta_0 + \beta_1 d_1 + \beta_2 d_2$	λ
Related	0	0	β_0	$\exp(\beta_0)$
Unemployed	1	0	$\beta_0 + \beta_1$	$\exp(\beta_0)\ \exp(\beta_1)$
Unrelated	0	1	$\beta_0 + \beta_2$	$\exp(\beta_0)\ \exp(\beta_2)$

On average, we expect $\exp(\beta_1)$ times as many unemployed students as students with jobs related to their fields of study.

```
> full0 = glm(freq~job,family=poisson)
> summary(full0)

Call:
glm(formula = freq ~ job, family = poisson)

Deviance Residuals: 
[1]  0  0  0

Coefficients:
              Estimate Std. Error z value Pr(>|z|)    
(Intercept)    4.66344    0.09713  48.013  < 2e-16 ***
jobUnemployed -1.66771    0.24379  -6.841 7.88e-12 ***
jobUnrelated  -0.35937    0.15148  -2.372   0.0177 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1 

(Dispersion parameter for poisson family taken to be 1)

    Null deviance:  6.5598e+01  on 2  degrees of freedom
Residual deviance: -7.9936e-15  on 0  degrees of freedom
AIC: 23.489
```

```
Number of Fisher Scoring iterations: 3

> full0$null.deviance # LR test, compare G^2 = 65.6
[1] 65.59798
```

Better null hypothesis

$$
\begin{aligned}
H_0: \quad & \pi_1 = 2\pi_2 \\
\Leftrightarrow \quad & \frac{\lambda_1}{\lambda_1+\lambda_2+\lambda_3} = 2\frac{\lambda_2}{\lambda_1+\lambda_2+\lambda_3} \\
\Leftrightarrow \quad & \lambda_1 = 2\lambda_2 \\
\Leftrightarrow \quad & \log\lambda_1 = \log 2 + \log\lambda_2 \\
\Leftrightarrow \quad & \beta_0 = \log 2 + \beta_0 + \beta_1 \\
\Leftrightarrow \quad & \beta_1 = -\log 2
\end{aligned}
$$

```
> # Offset "can be used to specify an a priori known component
> # to be included in the linear predictor during fitting. This should
> # be NULL or a numeric vector of length either one or equal to the
> # number of cases."
> freq
[1] 106  74  20
> d1 = c(0,1,0)
> d2 = c(0,0,1)
> red0 = glm(freq ~ d2, offset=-log(2)*d1,family=poisson)
> anova(red0,full0,test='Chisq') # Compare 4.739
Analysis of Deviance Table

Model 1: freq ~ d2
Model 2: freq ~ job
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1         1     4.7395
2         0     0.0000  1   4.7395  0.02948 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```