C3M2_peer_reviewed

June 24, 2023

1 C3M2: Peer Reviewed Assignment

1.0.1 Outline:

The objectives for this assignment:

1. Apply Poisson Regression to real data.
2. Learn and practice working with and interpreting Poisson Regression Models.
3. Understand deviance and how to conduct hypothesis tests with Poisson Regression.
4. Recognize when a model shows signs of overdispersion.

General tips:

1. Read the questions carefully to understand what is being asked.
2. This work will be reviewed by another human, so make sure that you are clear and concise in what your explanations and answers.

[1]:
```
# Load the required packages
library(MASS)
```

2 Problem 1: Poisson Estimators

Let $Y_1, ..., Y_n \overset{i}{\sim} Poisson(\lambda_i)$. Show that, if $\eta_i = \beta_0$, then the maximum likelihood estimator of λ_i is $\widehat{\lambda}_i = \bar{Y}$, for all $i = 1, ..., n$.

First we find the log-likelihood function:

$$({}_0) = \sum_{i=1}^{n}[y_i\eta - e^{\eta} - \log(y_i!)] = \sum_{i=1}^{n}[y_i\beta_0 - e^{\beta_0} - \log(y_i!)]$$

In order to find the maximum likelihood estimator (MLE), we aim to maximize the log-likelihood function with respect to ${}_0$. The value of ${}_0$ that maximizes the log-likelihood function is the MLE. Additionally, to find the MLE of , we use the relationship that equals e raised to the power of ${}_0$.

To determine the MLE, we differentiate the log-likelihood function ( ) with respect to ${}_0$, set the derivative equal to zero, and solve for ${}_0$.

$$\frac{d\ell(\beta_0)}{d\beta_0} = \sum_{i=1}^{n}[y_i\beta_0 - e^{\beta_0} - \log(y_i!)] = \sum_{i=1}^{n} y_i - \sum_{i=1}^{n} e^{\hat{\beta}_0} = 0$$

$$\sum_{i=1}^{n} y_i = \sum_{i=1}^{n} e^{\hat{\beta}_0} \implies \sum_{i=1}^{n} y_i = ne^{\hat{\beta}_0} \implies \frac{1}{n}\sum_{i=1}^{n} y_i = e^{\hat{\beta}_0} \implies \bar{y} = e^{\hat{\beta}_0}$$

$$\implies \hat{\beta}_0 = \log(\bar{y})$$

$$\hat{\lambda} = e^{\hat{\beta}_0} = e^{\log(\bar{y})} = \bar{y}$$

3 Problem 2: Ships data

The ships dataset gives the number of damage incidents and aggregate months of service for different types of ships broken down by year of construction and period of operation.

The code below splits the data into a training set (80% of the data) and a test set (the remaining 20%).

[41]:
```
data(ships)
ships = ships[ships$service != 0,]
ships$year = as.factor(ships$year)
ships$period = as.factor(ships$period)

set.seed(1111)
n = floor(0.8 * nrow(ships))
index = sample(seq_len(nrow(ships)), size = n)

train = ships[index, ]
test = ships[-index, ]
head(train)
summary(train)
```

A data.frame: 6 × 5

	type	year	period	service	incidents
	<fct>	<fct>	<fct>	<int>	<int>
29	D	70	60	349	2
6	A	70	75	3353	18
38	E	70	75	2161	12
40	E	75	75	542	1
17	C	60	60	1179	1
32	D	75	75	2051	4

```
 type   year    period      service           incidents
 A:7    60:7    60:13    Min.   :   63.0   Min.   : 0.000
 B:4    65:7    75:14    1st Qu.:  318.5   1st Qu.: 0.500
 C:6    70:9             Median : 1095.0   Median : 3.000
 D:7    75:4             Mean   : 4284.9   Mean   : 8.889
```

```
 E:3                    3rd Qu.: 2106.0   3rd Qu.:11.000  
                        Max.   :44882.0   Max.   :58.000  
```

3.0.1 2. (a) Poisson Regression Fitting

Use the training set to develop an appropriate regression model for `incidents`, using `type`, `period`, and `year` as predictors (HINT: is this a count model or a rate model?).

Calculate the mean squared prediction error (MSPE) for the test set. Display your results.

[45]:
```
# Your Code Here
pmod = glm(incidents ~ type + period + year, data = train, family = "poisson")
summary(pmod)

pred = predict(pmod, data = test, type = "response")
MSPE = mean((test$incidents - pred)^2)
print(paste("MSPE is:", MSPE))
```

```
Call:
glm(formula = incidents ~ type + period + year, family = "poisson",
    data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.5195  -1.9574  -0.6758   1.2328   3.6134

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.30210    0.21854   5.958 2.55e-09 ***
typeB        1.94896    0.18848  10.341  < 2e-16 ***
typeC       -1.24711    0.35247  -3.538 0.000403 ***
typeD       -0.90446    0.28746  -3.146 0.001653 **
typeE       -0.03192    0.29550  -0.108 0.913977
period75     0.36477    0.15470   2.358 0.018376 *
year65       0.47608    0.19885   2.394 0.016658 *
year70       0.30682    0.19665   1.560 0.118714
year75       0.18777    0.32593   0.576 0.564548
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 460.55  on 26  degrees of freedom
Residual deviance: 125.51  on 18  degrees of freedom
AIC: 215.39

Number of Fisher Scoring iterations: 6
```

```
Warning message in test$incidents - pred:
"longer object length is not a multiple of shorter object length"
```

```
[1] "MSPE is: 368.277662340249"
```

After fitting the model using Poisson regression, we obtained a residual deviance of 125.51. Additionally, the Mean Squared Prediction Error (MSPE) for this model is approximately 368.28.

3.0.2 2. (b) Poisson Regression Model Selection

Do we really need all of these predictors? Construct a new regression model leaving out `year` and calculate the MSE for this second model.

Decide which model is better. Explain why you chose the model that you did.

[46]:

```r
# Your Code Here
pmod_cut = glm(incidents ~ type + period, data = train, family = "poisson")
summary(pmod_cut)

pred_cut = predict(pmod_cut, data = test, type = "response")
MSPE = mean((test$incidents - pred_cut)^2)
print(paste("MSPE is:", MSPE))
```

```
Call:
glm(formula = incidents ~ type + period, family = "poisson",
    data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.4173  -1.9661  -0.7655   0.7212   3.7014

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.56346    0.17901   8.734  < 2e-16 ***
typeB        1.97434    0.18015  10.960  < 2e-16 ***
typeC       -1.25509    0.35199  -3.566 0.000363 ***
typeD       -0.90446    0.28746  -3.146 0.001653 **
typeE       -0.03345    0.28196  -0.119 0.905556
period75     0.37074    0.13774   2.692 0.007110 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 460.55  on 26  degrees of freedom
Residual deviance: 131.62  on 21  degrees of freedom
```

```
AIC: 215.5

Number of Fisher Scoring iterations: 6



Warning message in test$incidents - pred_cut:
"longer object length is not a multiple of shorter object length"
```

```
[1] "MSPE is: 370.87861538894"
```

[47]:
```
# Can compare nested poisson models with a chi-squared
pchisq(pmod_cut$deviance-pmod$deviance, df=pmod_cut$df.residual-pmod$df.
↪residual, lower.tail=FALSE)
```

0.106286940714784

The chi-squared test yielded a p-value of 0.106, which is greater than the significance level of 0.05. Therefore, we fail to reject the null hypothesis at = 0.05, suggesting that the reduced model may be sufficient. If our primary goal is prediction, it appears that the reduced model performs slightly better than the full model.

3.0.3 2. (c) Deviance

How do we determine if our model is explaining anything? With linear regression, we had a F-test, but we can't do that for Poisson Regression. If we want to check if our model is better than the null model, then we're going to have to check directly. In particular, we need to compare the deviances of the models to see if they're significantly different.

Conduct two χ^2 tests (using the deviance). Let $\alpha = 0.05$:

1. Test the adequacy of null model.
2. Test the adequacy of your chosen model agaisnt the saturated model (the model fit to all predictors).

What conclusions should you draw from these tests?

[48]:
```
# Let's test if the model is better than the null model
chisq.rslt = with(train, sum((incidents - fitted(pmod))^2/fitted(pmod)))
# Testing chi_sq results
pchisq(chisq.rslt, df=pmod$df.residual, lower.tail=FALSE)

# Test against the saturated model
sat_model = glm(incidents~., train, family="poisson")
pchisq(pmod$deviance-sat_model$deviance, df=pmod$df.residual-sat_model$df.
↪residual, lower.tail=FALSE)
```

3.94928685926698e-17

1.3012895360747e-23

[49]:

```
# Test if the model is better than the null model

# Test chi_sq stat

# Test against the saturated model
```

Our model yields statistically significant results from both tests, indicating that it performs better than the null model. However, it falls short when compared to the saturated model, indicating that there is still room for improvement.

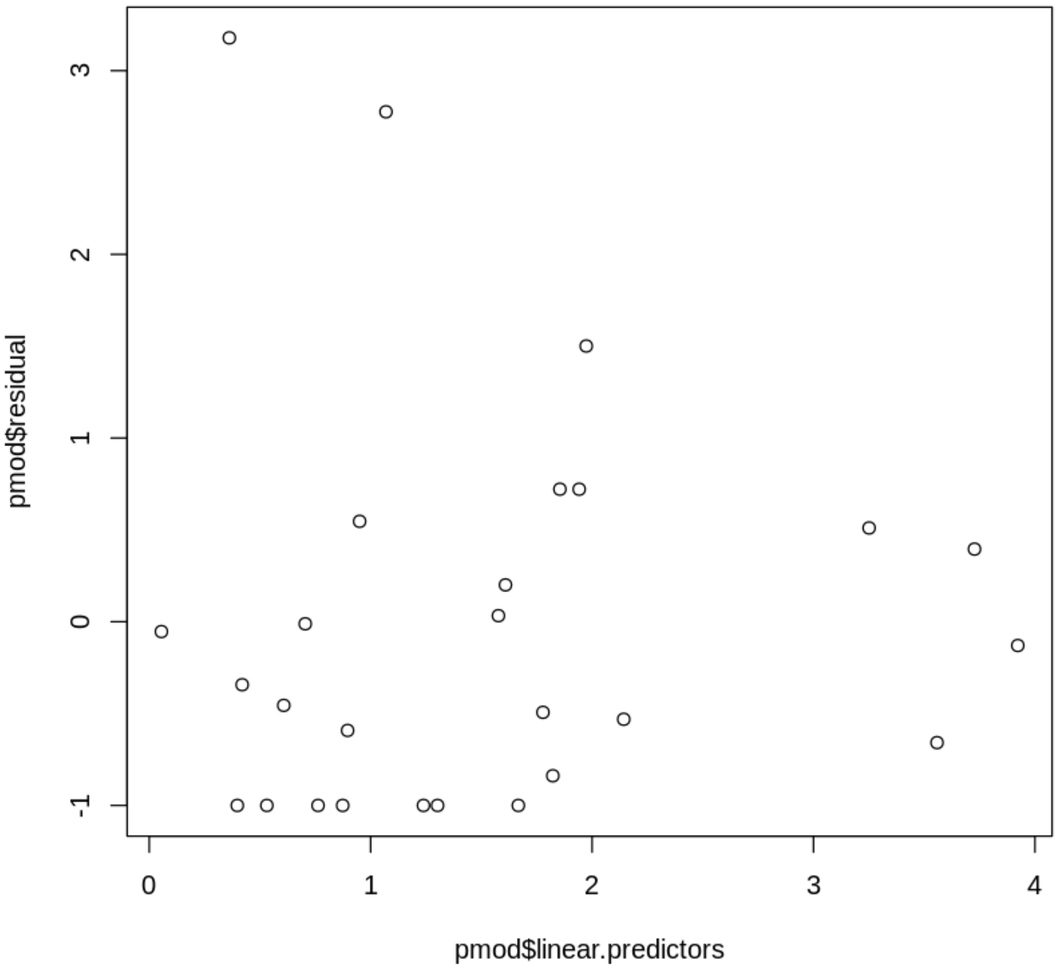
3.0.4 2. (d) Poisson Regression Visualizations

Just like with linear regression, we can use visualizations to assess the fit and appropriateness of our model. Is it maintaining the assumptions that it should be? Is there a discernable structure that isn't being accounted for? And, again like linear regression, it can be up to the user's interpretation what is an isn't a good model.

Plot the deviance residuals against the linear predictor η. Interpret this plot.

[50]:

```
# Your Code Here
plot(x=pmod$linear.predictors, y=pmod$residual)
```



From the plot we can see that there are 2 potential outliers on the y-axis near 3. The plot overall is looking good whitout outliers.

3.0.5 2. (e) Overdispersion

For linear regression, the variance of the data is controlled through the standard deviation σ, which is independent of the other parameters like the mean μ. However, some GLMs do not have this independence, which can lead to a problem called overdispersion. Overdispersion occurs when the observed data's variance is higher than expected, if the model is correct.

For Poisson Regression, we expect that the mean of the data should equal the variance. If overdispersion is present, then the assumptions of the model are not being met and we can not trust its output (or our beloved p-values)!

Explore the two models fit in the beginning of this question for evidence of overdisperion. If you find evidence of overdispersion, you do not need to fix it (but it would be useful for you to know how to). Describe your process and conclusions.

[51]:
```
# Your Code Here
summary(pmod)
summary(pmod_cut)
```

```
Call:
glm(formula = incidents ~ type + period + year, family = "poisson",
    data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.5195  -1.9574  -0.6758   1.2328   3.6134

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.30210    0.21854   5.958 2.55e-09 ***
typeB        1.94896    0.18848  10.341  < 2e-16 ***
typeC       -1.24711    0.35247  -3.538 0.000403 ***
typeD       -0.90446    0.28746  -3.146 0.001653 **
typeE       -0.03192    0.29550  -0.108 0.913977
period75     0.36477    0.15470   2.358 0.018376 *
year65       0.47608    0.19885   2.394 0.016658 *
year70       0.30682    0.19665   1.560 0.118714
year75       0.18777    0.32593   0.576 0.564548
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 460.55  on 26  degrees of freedom
Residual deviance: 125.51  on 18  degrees of freedom
AIC: 215.39

Number of Fisher Scoring iterations: 6



Call:
glm(formula = incidents ~ type + period, family = "poisson",
    data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.4173  -1.9661  -0.7655   0.7212   3.7014
```

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)    
(Intercept)  1.56346    0.17901   8.734  < 2e-16 ***
typeB        1.97434    0.18015  10.960  < 2e-16 ***
typeC       -1.25509    0.35199  -3.566 0.000363 ***
typeD       -0.90446    0.28746  -3.146 0.001653 ** 
typeE       -0.03345    0.28196  -0.119 0.905556    
period75     0.37074    0.13774   2.692 0.007110 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 460.55  on 26  degrees of freedom
Residual deviance: 131.62  on 21  degrees of freedom
AIC: 215.5

Number of Fisher Scoring iterations: 6
```

When the Residual deviance significantly exceeds the degrees of freedom, it indicates the presence of overdispersion. In the case of both models, this criterion is satisfied, suggesting that both models exhibit overdispersion. Overdispersion refers to a situation where the observed variation in the data is greater than what can be accounted for by the assumed model.

The presence of overdispersion implies that the models may not fully capture the underlying variability in the data. It suggests that there might be additional sources of variation or unaccounted factors influencing the response variable. To address overdispersion, alternative modeling approaches such as using generalized linear models (GLMs) with appropriate distributional assumptions (e.g., negative binomial) or incorporating random effects might be considered.

By acknowledging the presence of overdispersion in both models, it becomes crucial to assess the impact of this phenomenon on the model's predictions and the validity of the statistical inferences drawn from the models. Additionally, exploring potential sources of overdispersion and refining the modeling approach accordingly can lead to more accurate and reliable results.

[ ]:
```

```