Prof. Dr. M. Kateri,
Dr. W. Herff,
L. Kaufmann, M.Sc.

RWTH Aachen, SS 2022
Release date: July 4th, 2022
Solution: Week starting from July 11th, 2022

Applied Data Analysis

R-Laboratory 11

Count Data - Penalized Regression

Useful packages and functions:

- `addmargins()`
- `rstandard()`
- `chisq.test()`
- `vcd`
- `vcd::mosaic()`
- `logistf`
- `glmnet`
- `glmnet: cv.glmnet`
- `ISLR`

Remark (Quasi-Likelihood Approach)
An alternative to modelling count data using a negative binomial GLM or a zero inflated GLM for taking possible sources of overdispersion into account is given by the *quasi-likelihood* (QL) approach. Assume a random sample $Y_1, \dots, Y_n$, $n \in \mathbb{N}$, with $\mu_i = \mathsf{E}(Y_i)$, $i = 1, \dots, n$. For a GLM with link function g and a linear predictor for the mean $g(\mu_i) = \eta_i = \sum_{j=1}^{p} x_{ij}\beta_j$, $i = 1, \dots, n$, the QL estimation approach, instead of assuming a distribution for the response and deriving the variance $v(\mu_i) = \mathrm{Var}(Y_i)$, $i = 1, \dots, n$, directly assumes an appropriate variance function $v(\mu_i)$ to formulate the estimating equations

$$\sum_{i=1}^{n} \frac{(Y_i - \mu_i)x_{ij}}{v(\mu_i)} \cdot \frac{\partial \mu_i}{\partial \eta_i} = 0, \quad j = 1, \dots, p. \tag{1}$$

Any solution $\hat{\beta}_1, \dots, \hat{\beta}_p$ of (1) is called a quasi–likelihood estimate of $\beta_1, \dots, \beta_p$. While assuming $Y_i \sim \mathcal{P}(\mu_i)$ leads to $v(\mu_i) = \mu_i$ in ML estimation, we can also set $v(\mu_i) = \phi\mu_i$, where $\phi > 1$ represents overdispersion and allow a robust estimation of the standard errors. Notice that if it seems to be appropriate, we can also set any other relationship like $v(\mu_i) = \phi$ (as in the normal linear model with $\phi = \sigma^2$) or $v(\mu_i) = \phi\mu_i^2$, if even an inflated quadratic relation seems appropriate.

Task 35

Load the dataset *Basketball.csv* into your workspace. Let $y_1, \dots, y_{514}$ be realizations of independent random variables, where y_i is representing the number of field goals (`FG`) to the ith player, $i = 1, \dots, 514$. Further, denote with x_i the number of games `G` of the ith player, $i = 1, \dots, 514$. Assume that y_i is the realization of a Poisson distributed random variable $Y_i \sim \mathcal{P}(\mu_i)$ and that the GLM

$$\log(\mu_i) = \alpha + \beta x_i, \quad i = 1, \dots, 514,$$

with unknown $\alpha, \beta \in \mathbb{R}$, holds. Estimate the parameters α and β using the data set and the R-function `glm` with parameter `family=poisson(link='log')`. Create a scatterplot

$(x_1, y_1), \ldots, (x_{514}, y_{514})$ of `G` against `FG`.

Denote with $\hat{\alpha}, \hat{\beta}$ the MLEs of α and β. Draw a sample of the random variables $Y_i^* \sim \mathcal{P}(\exp(\hat{\alpha} + \hat{\beta} x_i))$, $i = 1, \ldots, 514$, which are conditionally independent given $\hat{\alpha}$ and $\hat{\beta}$. Create a scatterplot $(x_1, y_1^*), \ldots, (x_{514}, y_{514}^*)$ of the simulated sample. Compare both scatterplots. Repeat this a few times to get a better feeling of how scatterplot behaves, if the Poisson GLM truly holds. Is the assumed GLM a plausible model?

Task 36

Load the dataset *Crabs.dat* into your workspace and construct a model for the horseshoe crab satellite counts, using the QL approach and weight, color and spine condition as possible explanatory variables. Compare with the results obtained from zero-inflated GLMs in the example II.6.18 of the lecture.

Hint: You can solve the equations (1) using the `glm` function with `family=quasi(link, variance)`. Since we want to model overdispersed Poisson data, `link='log'` is appropriate for holding the connection. Possible values for the variance are for example `variance='mu'` for $v(\mu_i) = \phi \mu_i$, $i = 1, \ldots, n$ or `variance='mu^2'` for $v(\mu_i) = \phi \mu_i^2$, $i = 1, \ldots, n$.

Task 37

In Remark II.3.30, the penalized likelihood approach of Firth was mentioned (Firth, Biometrika 1993). The idea of Firth's logistic regression is to take the logistic model

$$\pi_i = (1 + \exp(-\sum_{r=1}^{k} x_{ir} \beta_r)) \tag{2}$$

and replace the score equations $\sum_{i=1}^{n} (y_i - \pi_i) x_{ir} = 0$ by modified score equations

$$\sum_{i=1}^{n} (y_i - \pi_i + h_i(1/2 - \pi_i)) x_{ir} = 0 \tag{3}$$

for $r = 1, ..., k$. Here, h_i is the i-th diagonal element of the hat matrix. Having that, the Firth-type estimates $\hat{\beta}$ are computed by solving the modified score equations until convergence is attained.

(a) Load the data *fungal.dat* into your workspace.

(b) Transform `center` as factor variable. Fit a logistic regression model predicting `my/m` based on the explanatory variables `treatment` and `center` and comment the resulting fit.

(c) Using the `logistf` package, fit a model with Firth's logistic regression predicting `my/m` based on the explanatory variables `treatment` and `center`. Compare the fit with the fit in (b).

Task 38

Load the dataset *prostate* from RWTH moodle into your workspace. You can find the description of the dataset in the dataset documentation in RWTH moodle.

(a) Split the data into test and training data where the training data contains all rows of *prostate* where `train == TRUE`.

(b) Fit a linear model which predicts `lpsa` based on the explanatory variables `lcavol, lweight, age, lbph, svi, lcp, gleason` and `ppg45`.

(c) Fit the penalized regression model with Lasso. Plot the resulting coefficient paths.

(d) Set the seed to 2021. Again, fit the penalized regression model with Lasso using cross-validation. What is the minimum value of λ ? What is the value of λ suggested by one-standard-error rule?

(e) Extract the corresponding coefficient estimates for the values of λ in (d).

```
#############
#
# Task 35
#
#############

BB=read.csv("Basketball.csv")
model=glm(FG~G,data=BB, family=poisson)
par(mfrow=c(2,2))
plot(BB$G,BB$FG) #scatterplot of G against FG
#curve(exp(model$coefficients[1]+x*model$coefficients[2]), from=0, to =80,add=TRUE,col='red')
for (i in 1:3){
	random.model.sample=rpois(length(model$fitted.values),model$fitted.values)
	plot(BB$G,random.model.sample)
}
par(mfrow=c(1,1))
```

```
# Task 36
crabs.data=read.table("Crabs.dat", sep="",dec=".",header = TRUE)
#crabs2.data=read.table("Crabs2.dat", sep="",dec=".",header = TRUE) #aus Vorlesung

#variance=mu
QL.glm.1<-glm(y~weight+color+spine,family=quasi(link='log',variance = 'mu'),data=crabs.data)
summary(QL.glm.1)

#variance=mu^2
QL.glm.2<-glm(y~weight+color+spine,family=quasi(link='log',variance = 'mu^2'),data=crabs.data)
summary(QL.glm.2)

# Example 6.1.18 - zero-inflated negative binomial glm
library(pscl)
ZINB.glm <- zeroinfl(y ~ weight | weight + color, dist="negbin",data=crabs.data)
summary(ZINB.glm)

# classical Negative Binomial glm
library("MASS")
NB.glm <- glm.nb(y~weight+color+spine,data=crabs.data)
summary(NB.glm)

# Classical Poisson glm
P.glm<-glm(y~weight+color+spine,family=poisson(link='log'),data=crabs.data)
summary(P.glm)

# some plots
plot(ZINB.glm$fitted.values,QL.glm.1$fitted.values) #fitted values for QL (var=mu) and Zero Infl NB
abline(0,1,col="red")

plot(ZINB.glm$fitted.values,QL.glm.2$fitted.values) #fitted values for QL (var=mu^2) and Zero Infl NB
abline(0,1,col="red")

plot(NB.glm$fitted.values,QL.glm.2$fitted.values) #fitted values for QL (var=mu^2) and NB
abline(0,1,col="red")

### Task 37
library(logistf)
#(a)
fungal <- read.csv("fungal.dat", sep="")

#(b)
fit<-glm(my/m~ treatment +factor(center),weights=m,family=binomial,data=fungal)
summary(fit)

#(c)
fit.pen<-logistf(my/m~ treatment +factor(center),weights=m,family=binomial,data=fungal)
summary(fit.pen)
```

```
############
#
### Task 38
#
###########
library("glmnet")

#(a)
prostate <- read.delim("prostate")

# split into training and test data
data=prostate
data.training = data[data$train==TRUE,]
data.test =data[-data$train==TRUE,]

#(b)
model1<-lm(lpsa~lcavol+lweight+age+lbph+svi+lcp+gleason+pgg45, data=data.training)
model1
#lcavol, lweight, svi show strongest fit

#(c)
model.lasso<-
glmnet(as.matrix(data.training[,c("lcavol","lweight","age","lbph","svi","lcp","gleason","pgg45")]),
                    y=data.training$lpsa,alpha=1,family="gaussian")
#glmnet expects matrix of predictors
plot(model.lasso)

#(d)
set.seed(2021)
cv<-cv.glmnet(as.matrix(data.training[,c("lcavol","lweight","age","lbph","svi","lcp","gleason","pgg45")]),
              y=data.training$lpsa,alpha=1,family="gaussian")

cv$lambda.min
cv$lambda.1se

#(e)
coef(glmnet(as.matrix(data.training[,c("lcavol","lweight","age","lbph","svi","lcp","gleason","pgg45")]),
            y=data.training$lpsa,alpha=1,family="gaussian",lambda=cv$lambda.min)) #lambda from
lambda.min
coef(glmnet(as.matrix(data.training[,c("lcavol","lweight","age","lbph","svi","lcp","gleason","pgg45")]),
            y=data.training$lpsa,alpha=1,family="gaussian",lambda=cv$lambda.1se))  #cv lambda.1se
```