Project 1

TMA4315 GLM H2020

10013 - siljeanf@stud.ntnu.no

01 October, 2020

Problem 1

1a) Parameter estimation with MLE

Let y be Poisson distributed,

$$f(y) = \frac{\lambda^y}{y!}e^{-\lambda} \text{ for } y = 0, 1, 2, ...$$

with a canonical link function: $\eta_i = ln(\lambda_i)$ or $\lambda_i = \exp(\eta_i)$
and systematic component $\eta_i = x_i^T\beta$.

Likelihood

$$L(\beta) = \prod_{i=1}^{n} f(y_i;\beta) = \prod_{i=1}^{n} \frac{\lambda_i^{y_i}}{y_i!}\exp(-\lambda_i)$$

log likelihood

$$l(\beta) = \ln L(\beta) = \sum_{i=1}^{n} \ln(f(y_i;\beta)) = \sum_{i=1}^{n} [y_i \ln(\lambda_i) - \lambda_i - \ln(y!)]$$

Score function

$$\begin{aligned}
s(\beta) &= \frac{\partial l(\beta)}{\partial\beta} \\
&= \sum_{i=1}^{n} \frac{\partial l_i(\beta)}{\partial\beta} \\
&= \sum_{i=1}^{n} \frac{\partial l_i(\beta)}{\partial\eta_i}\cdot\frac{\partial\eta_i}{\partial\beta} \\
&= \sum_{i=1}^{n} \frac{\partial[y_i\eta_i - \exp(\eta_i) + \ln(y!)]}{\partial\eta_i}\cdot\frac{\partial[\mathbf{x}_i^T\beta]}{\partial\beta} \\
&= \sum_{i=1}^{n} [y_i - \exp(\eta_i)]\cdot\mathbf{x}_i \\
&= \sum_{i=1}^{n} (y_i - \lambda_i)\mathbf{x}_i
\end{aligned}$$

Fisher information

$$\begin{aligned}
F(\beta) &= \mathrm{Cov}(s(\beta)) \\
&= \sum_{i=1}^{n} E(s_i(\beta)s_i(\beta)^T) \\
&= \sum_{i=1}^{n} E((Y_i - \lambda_i)\mathbf{x}_i(Y_i - \lambda_i)\mathbf{x}_i^T) \\
&= \sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^T E((Y_i - \lambda_i)^2) \\
&= \sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^T \lambda_i \\
&= X^T W X
\end{aligned}$$

Notice that $E((Y_i - \lambda_i)^2) = \mathrm{Var}(Y_i) = \lambda$ is the variance of Y_i

Observed Fisher information matrix $H(\beta)$

$$\begin{aligned}
H(\beta) &= -\frac{\partial^2 l(\beta)}{\partial\beta\partial\beta^T} \\
&= -\frac{\partial s(\beta)}{\partial\beta^T} \\
&= \frac{\partial}{\partial\beta^T}\left[\sum_{i=1}^{n}(\lambda_i - y_i)\mathbf{x}_i\right] \\
&= \sum_{i=1}^{n} \frac{\partial}{\partial\beta^T}[\mathbf{x}_i\lambda_i - \mathbf{x}_i y_i] \\
&= \sum_{i=1}^{n} \frac{\partial}{\partial\beta^T}\mathbf{x}_i\lambda_i \\
&= \sum_{i=1}^{n} \mathbf{x}_i\frac{\partial\lambda_i}{\partial\eta_i}\frac{\partial\eta_i}{\partial\beta^T} \\
&= \sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^T\lambda_i \\
&= X^T W X,
\end{aligned}$$

where W is the diagonal matrix with λ_i as diagonal elements.

The derivatives in the chain rule are derived as following: $\frac{\partial\eta_i}{\partial\beta^T} = \frac{\partial\mathbf{x}_i^T\beta}{\partial\beta^T} = \left(\frac{\partial\mathbf{x}_i^T\beta}{\partial\beta}\right)^T = \mathbf{x}_i^T$ and $\frac{\partial\lambda_i}{\partial\eta_i} = \frac{\partial\exp(\eta_i)}{\partial\eta_i} = \exp(\eta_i) = \lambda_i$

We note that the Observed Fisher information matrix equal the expected Fisher information matrix for canonical choice of link function.

1b) Fisher Scoring Algorithm

Input: formula: a model formula.
data: a data frame containing the data.
start: an optional vector containing starting values defaulting to 0.

Output: coefficents: A matrix containing the maximum likelihood estimates of each parameter in the first column and estimates of their standard errors based on the final Fisher information matrix.
deviance: The deviance of the model, that is twice the difference in log likelihood between the fitted and the saturated model.
vcov: The estimated variance matrix of the coefficients.

Notice: We will use Fisher scoring algorithm to estimate β from the MLE.

$$\beta^{(t+1)} = \beta^{(t)} + F(\beta^{(t)})^{-1} s(\beta^{(t)})$$

Therefore the expected Fisher information matrix F needs to be invertible. As the parameters $\lambda_i = e^{x_i^T \beta} > 0$ for all i and the design matrix X has full rank, p we know F is invertible.

We notice that standard error estimates are the the square root of diagonal elements of the inverse Fisher information matrix. This matrix again equals the variance of β.

```r
myglm <- function(formula, data, start=0) {
  y <- data$y
  X <- model.matrix(formula, data) #design matrix
  beta <- rep(0, ncol(X)) #beta vector

  repeat {
    eta <- as.vector(X%*% beta) #eta vector
    lambda <- exp(eta)
    W <- diag(lambda) #diagonal matrix of lambdas
    score <- apply((y-lambda)*X,2,sum)
    Fisherinfo <- t(X) %*% W %*% X

  if (sum(score^2)<1e-10) #break point
    break()

    #find coefficient estimate and std.error
    beta <- beta + solve(Fisherinfo) %*% score
    F.inv <- solve(Fisherinfo)
    std.error <- sqrt(diag(F.inv))
    #matrix of beta and std.error:
    coeff = cbind(beta, std.error)
    colnames(coeff) <- c("Estimate", "Std.Error")
  }
  # find deviance
  log.cand <- sum(y*log(lambda)-lambda) #removed y as it cancels!
  log.sat <- sum(ifelse(y==0,0,y*log(y) ) - y ) #set y=lambda
  deviance = 2 * (log.sat - log.cand)

  return (list(coef=coeff, deviance=deviance, vcov = F.inv))
 }
```

1c) Test

Load the dataset for problem 2 in order to test the myglm function in 1b compared to the glm function in R.

```r
#data from problem 2 (called data)
load(url("https://www.math.ntnu.no/emner/TMA4315/2020h/hoge-veluwe.Rdata"))
formula = data$y~data$t+I(data$t^2)
```

```r
#myglm function from 1b
mod <- myglm(formula, data)
mod
```

```
## $coef
##                  Estimate   Std.Error
## (Intercept)  1.420130462 0.282433135
## data$t       0.085183057 0.034053808
## I(data$t^2) -0.003298608 0.001019461
## 
## $deviance
## [1] 277.4613
## 
## $vcov
##               (Intercept)       data$t   I(data$t^2)
## (Intercept)  0.0797684757 -0.009308502  0.0002550173
## data$t      -0.0093085017  0.001159662 -0.0000336900
## I(data$t^2)  0.0002550173 -0.000033690  0.0000010393
```

```r
#glm function
testMod <- glm(formula = formula, family = poisson(link = log))
summary(testMod)
```

```
## 
## Call:
## glm(formula = formula, family = poisson(link = log))
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -3.7797  -0.7788   0.3430   0.7438   2.2312  
## 
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)    
## (Intercept)  1.420130   0.282427   5.028 4.95e-07 ***
## data$t       0.085183   0.034053   2.502  0.01237 *  
## I(data$t^2) -0.003299   0.001019  -3.236  0.00121 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for poisson family taken to be 1)
## 
##     Null deviance: 300.11  on 134  degrees of freedom
## Residual deviance: 277.46  on 132  degrees of freedom
## AIC: 740.67
## 
## Number of Fisher Scoring iterations: 5
```

```r
vcov(testMod)
```

```
##               (Intercept)        data$t   I(data$t^2)
## (Intercept)  0.0797650582 -9.307971e-03  2.549981e-04
## data$t      -0.0093079707  1.159579e-03 -3.368697e-05
## I(data$t^2)  0.0002549981 -3.368697e-05  1.039189e-06
```

Problem 2

y = number of fledglings leaving the nest per female t = date for when the female lays the first egg $n = 135$

Assumption: The number of fledglings produced by each female follows a Poisson distribution.

The expected number of fledglings by each female depends on t_i via a Gaussian function with parameters,λ_0, θ, ω.

$$E[y_i] = \lambda_i = \lambda_0 \exp(-\frac{(t_i - \theta)^2}{2\omega^2})$$

2a) Interpret parameters

λ_0 is the height of the curve's peak, which in this example is the maximum number of fledglings, y_{max}.

θ is the position of the center of the peak, in other words the optimal initiate breeding time, $E(t)$.

ω controls the width of the "bell", which can be interpreted as the spread or variance of the number of fledglings.

2b) Reparameterized model

The reparametrized model is a GLM because:

1) We have a random component y_i which is Poisson distributed.
2) We have a systematic component $\eta_i = t_i^T \beta$
3) We have a canonical link function $\eta_i = ln(\lambda_i)$ giving the following relation between the GLM parameters contained in $\boldsymbol{\beta}$ and (λ_0, θ, ω):

$$\lambda_i = \exp\left(log(\lambda_0) - \frac{(t_i - \theta)^2}{2\omega^2}\right) = \exp(\eta_i) = \exp(t_i \boldsymbol{\beta}) = \exp(\beta_0 + \beta_1 t_i + \beta_2 t_i^2)$$

2c) Fit the model

This has already been done in problem 1b. Below is the printout once again.

```
mod
```

```
## $coef
##                 Estimate   Std.Error
## (Intercept)  1.420130462 0.282433135
## data$t       0.085183057 0.034053808
## I(data$t^2) -0.003298608 0.001019461
##
## $deviance
## [1] 277.4613
##
## $vcov
##               (Intercept)       data$t   I(data$t^2)
## (Intercept)  0.0797684757 -0.009308502  0.0002550173
## data$t      -0.0093085017  0.001159662 -0.0000336900
## I(data$t^2)  0.0002550173 -0.000033690  0.0000010393
```

2d) hypothesis test

In order to check if there is evidence of quadratic effect of t we will perform the following hypothesis test, H_0: $\beta_2 = 0$ H_1: $\beta_2 \neq 0$

Let‘s choose the Likelihood Ratio Test, LRT, as we already know the deviance from the function in problem 1. We will perform the test using the difference between two deviances, which in this case is a model A with the quadratic effect of t and a model B without this term.

```
#hypothesis test using LRT, comparing deviances
modA <- mod #with beta2
modB <- myglm(data$y~t, data) #model without beta2

deviancediff <- modB$deviance- mod$deviance

#calculate pvalue
pvalue <- 1-pchisq(deviancediff, 1)
pvalue
```

```
## [1] 0.0005524157
```

Since the p-value is 5.5241571×10^{-4} which is smaller then the usual significance level of 0.05 we reject H_0, hence the quadratic effect of t significant.

2e) Goodness-of-fit

Using the observed deviance, we will test the goodness-of-fit of the fitted model (with the quadratic effect of t) with the following hypotheses:

H_0: candidate model is good H_1: candidate model is not good

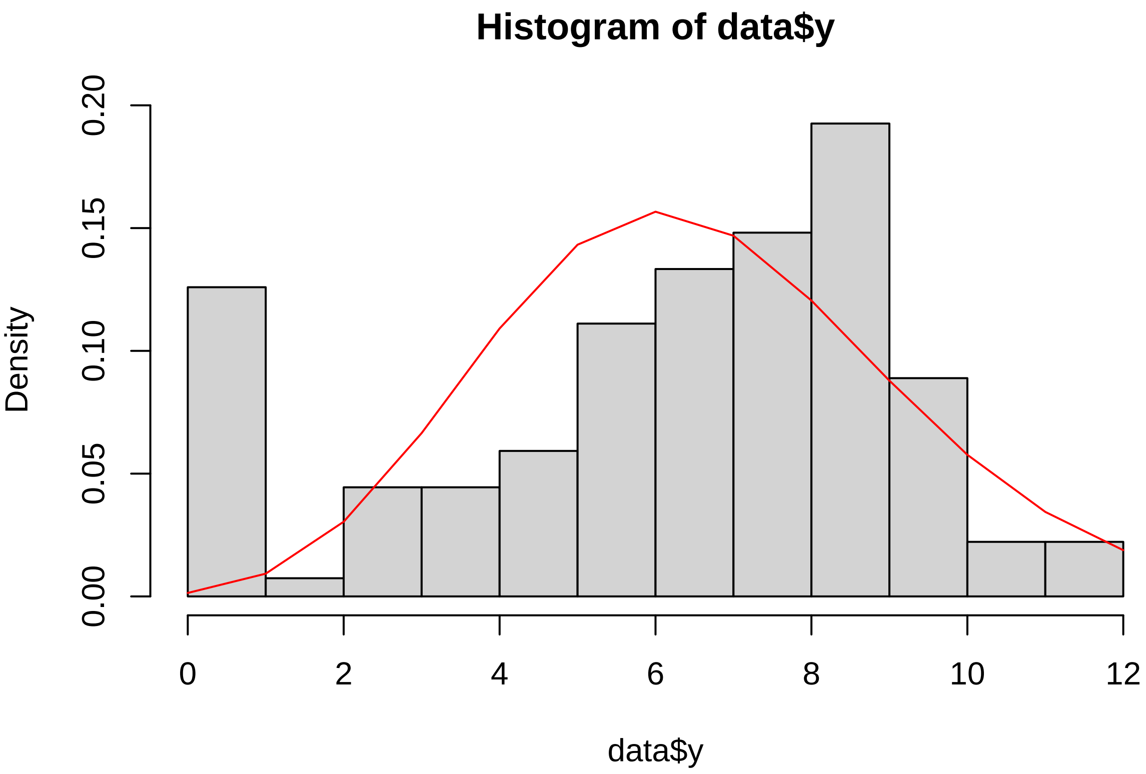
```
#deviance test for model, pvalue
pvalue <- 1 - pchisq(mod$deviance, (length(data$t)-length(mod$coef)) )
pvalue
```

```
## [1] 6.913359e-13
```

Again, since the pvalue is $6.9133588 \times 10^{-13}$, which is very small compared to the usual significance level we reject the null hypothesis and the model fit is not good. This can be cause by fitting the wrong model, missing covariate, or overdispersion (i.e. the data show more variability than we would expect for a Poisson distributed response).

We will now examine the distribution of the y more closely in order to discuss how the assumptions of the model may be violated.

```
hist(data$y, prob=TRUE)
lines(0:max(data$y), dpois(0:max(data$y), mean(data$y)), col = 'red')
```



From the histogram above we notice that the column for $y = 0$ is much greater than what we expect for a Poission distributed response (see red line). In other words, the number of birds in the population that produces zero fledglings is too high compared to what we normally expect from a Poisson distributed response. From this we can conclude that assumption we did about the response being Poisson distributed fails.

2f) MLE for parameterized model

From problem 1 we found that the MLE for the parameter vector $\boldsymbol{\beta}$ is $\hat{\boldsymbol{\beta}}$. We now have a parameterization of $\boldsymbol{\beta}$ to (λ_0, θ, ω), call this $g(\boldsymbol{\beta})$. Then by defintion, the MLE for $g(\boldsymbol{\beta})$ is $g(\widehat{\boldsymbol{\beta}})$ as the MLE is invariant with respect to transformations of the parameter. Therefore we can estimate the parameters θ and ω using the estimation of $\boldsymbol{\beta}$.

First, we use the relations between each parameter in $\boldsymbol{\beta}$ and (λ_0, θ, ω) found in problem 2b to find an expression for each parameter in in $\boldsymbol{\beta}$. Recall that $\eta = ln(\lambda_i) = t_i\boldsymbol{\beta}$.

$$\beta_0 = \log(\lambda_0) - \frac{\theta^2}{2\omega^2}\beta_1 = \frac{\theta}{\omega^2}\beta_2 = -\frac{1}{2\omega^2}$$

Giving the following expressions for the parameters we wanted to estimate:

$$\omega = g(\hat{\beta}_1, \hat{\beta}_2) = \sqrt{-\frac{1}{2\hat{\beta}_2}}\theta = h(\hat{\beta}_1, \hat{\beta}_2) = \hat{\beta}_1\omega^2 = \frac{\hat{\beta}_1}{-2\hat{\beta}_2}$$

We can now use the estimation of the parameters in the fitted model from 2c in order to calculate an estimation for ω and θ.

```
coeflist <- mod$coef

beta1 <-coeflist[2,1]
beta2 <-coeflist[3,1]
```

```
omega = sqrt(-1/(2*beta2))
theta = beta1/(-2*beta2)
```

The estimation of ω is 12.3117455, while the estimation of θ is 12.9119692.

Delta method Now we will use the delta method in order to find their standard errors based on asymptotic theory. First to find the variance of ω, then for θ. \$\$

$$
\begin{aligned}
\mathrm{Var}(\omega) &= \left(\frac{\partial g}{\partial \hat{\beta}_1}\right)^2 \mathrm{Var}(\hat{\beta}_1) + \left(\frac{\partial g}{\partial \hat{\beta}_2}\right)^2 \mathrm{Var}(\hat{\beta}_2) + \frac{\partial g}{\partial \hat{\beta}_1}\frac{\partial g}{\partial \hat{\beta}_2}\mathrm{Cov}(\hat{\beta}_1, \hat{\beta}_2) && (1)\\
&= \left(\frac{1}{\sqrt{2\hat{\beta}_2^3}}\right)^2 \mathrm{Var}(\hat{\beta}_2) && (2)\\
&= \frac{1}{2\hat{\beta}_2^3}\mathrm{Var}(\hat{\beta}_2) && (3)
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{Var}(\theta) &= \left(\frac{\partial h}{\partial \hat{\beta}_1}\right)^2 \mathrm{Var}(\hat{\beta}_1) + \left(\frac{\partial h}{\partial \hat{\beta}_2}\right)^2 \mathrm{Var}(\hat{\beta}_2) + \frac{\partial h}{\partial \hat{\beta}_1}\frac{\partial h}{\partial \hat{\beta}_2}\mathrm{Cov}(\hat{\beta}_1, \hat{\beta}_2) && (4)\\
&= \left(-\frac{1}{2\hat{\beta}_2}\right)^2 \mathrm{Var}(\hat{\beta}_1) + \left(\frac{\hat{\beta}_1}{2\hat{\beta}_2^2}\right)^2 \mathrm{Var}(\hat{\beta}_2) + 2\left(\frac{-1}{2\hat{\beta}_2}\right)\left(\frac{\hat{\beta}_1}{2\hat{\beta}_2^2}\right)\mathrm{Cov}(\hat{\beta}_1, \hat{\beta}_2) && (5)
\end{aligned}
$$

\$\$ Now we are ready to estimate the variances numerically,

```
#find variances and covariances of beta vector
covmatrix <- mod$vcov

varb1 <-covmatrix[2,2]
varb2 <-covmatrix[3,3]
covb1b2 <- covmatrix[2:3,2:3]
covb1b2
```

```
##                   data$t I(data$t^2)
## data$t       0.001159662 -3.3690e-05
## I(data$t^2) -0.000033690  1.0393e-06
```

```
#gradients for g and h
grad_g <- c(0, (-1/(2*beta2))^(3/2) )
grad_h <- c(-1/(2*beta2), (beta1/(2*beta2^2)) )

sd_omega <- sqrt(t(grad_g) %*% covb1b2 %*% grad_g)
sd_theta <- sqrt(t(grad_h) %*% covb1b2 %*% grad_h)
```

The standard deviation for ω is 1.9025208
The standard deviation for θ is 1.609376

2g)

In order to check if the mean value of t in this population is significantly different from the estimated optimal date based on the fitted model we will do a hypothesis test with the following hypotheses:

$H_0 : \theta = \mu$ $H_1 : \theta \neq \mu$

As we know that λ_i is normal distributed with mean θ and variance ω we perform a t-test and calculate the p-value from this distribution. We use the mean of the t-values in the data to estimate μ.

```
n <- 135
tbar <- mean(data$t)
Z <- tbar-theta/(omega/sqrt(n))
pvalue <- 2*pnorm(-abs(Z))
pvalue
```

```
## [1] 0.0001756826
```

As the p-value is 1.7568263×10^{-4} which is smaller then the usual significance level, the null hypothesis is false, meaning $\theta \neq \mu$ and the mean value of t in the population is therefore significantly different from the estimated optimal date based on the fitted model.

Problem 3

Parametric bootstraps resample a known distribution function, whose parameters are estimated from the sample data in problem 2. Below, we estimate the variance by doing

```
#set up
beta <- myglm(formula, data = data)$coef[,1]
X <- model.matrix(formula, data)
eta <- as.vector(X%*%beta) #eta vector
lambda <- exp(eta)
#matrix of estimated betahat from each bootstrap sample
beta_hat <- matrix(0, nrow = 1000, ncol = 3)

#bootstrapping
for (i in 1:1000){
  data$y <- rpois(135,lambda) #new estimate of y from a poisson distribution
  beta_hat[i,] <- myglm(data$y~data$t+I(data$t^2), data = data)$coef[,1]
}

# variance of beta_hat
var(beta_hat)

#covariance matrix from fitted model in problem 2
mod$vcov
```

We notice that the estimated variance from bootstrapping and from the Expected Fisher information are quite similar. From this we can conclude that the asymptotic result is a good approximation even for finite sample size, $n = 135$ in this case.