Project 1

Generalized Linear Models 2020

Lene Tillerli Omdal LeneTOm@stud.ntnu.no 10022
Arne Rustad ArneIR@stud.ntnu.no 10028

02 oktober, 2020

1)

a)

Each observation y_i is Poisson distributed with expectation λ_i, $y_i \sim \text{poisson}(\lambda_i)$. The probability density function of y_i is

$$f(y_i|\lambda_i) = \frac{\lambda^{y_i}}{y_i!}e^{-\lambda_i}.$$

Using the canonical link function we get the following relationship between $E[y_i] = \mu_i$ and the linear predictor $x_i^T\beta = \eta_i$

$$\ln(\mu_i) = \ln(\lambda_i) = x_i^T\beta.$$

This gives the likelihood function

$$\begin{aligned}\mathrm{L}(\beta|y_i, x_i) &= \prod_{i=1}^n f(y_i|\lambda_i) = \prod_{i=1}^n f(y_i|\beta, x_i)\\ &= \prod_{i=1}^n \frac{(e^{x_i^T\beta})^{y_i}}{y_i!}e^{-e^{x_i^T\beta}},\end{aligned}$$

and the log likelihood function

$$\begin{aligned}\mathrm{l}(\beta|y_i, x_i) = \ln(\mathrm{L}(\beta|y_i, x_i)) &= \sum_{i=1}^n \ln(f(y_i|\lambda_i))\\ &= \sum_{i=1}^n -\ln(y_i!) + y_i x_i^T\beta - e^{x_i^T\beta}.\end{aligned}$$

The score vector becomes

$$\begin{aligned}\mathrm{s}(\beta) = \frac{\partial \mathrm{l}(\beta)}{\partial \beta} &= \sum_{i=1}^{n} y_i x_i - e^{x_i^T\beta} \cdot \frac{\partial}{\partial \beta} x_i^T \beta \\ &= \sum_{i=1}^{n} (y_i - e^{x_i^T\beta}) x_i \\ &= \sum_{i=1}^{n} (y_i - \lambda_i) x_i.\end{aligned}$$

In matrix notation this is

$$\mathrm{s}(\beta) = X^T(Y - \lambda),$$

where

$$X = \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_n^T \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix}$$

and

$$\lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{bmatrix} = \begin{bmatrix} e^{x_1^T\beta} \\ e^{x_2^T\beta} \\ \vdots \\ e^{x_n^T\beta} \end{bmatrix}.$$

The observed Fisher information is given as

$$\mathrm{H}(\beta) = -\frac{\partial^2 \mathrm{l}(\beta)}{\partial \beta \partial \beta^T} = -\frac{\partial \mathrm{s}(\beta)}{\partial \beta^T}.$$

Inserting the obtained score function results in

$$\begin{aligned}\mathrm{H}(\beta) &= -\frac{\partial}{\partial \beta^T} \sum_{i=1}^{n} \left(y_i - e^{x_i^T\beta}\right) x_i \\ &= \sum_{i=1}^{n} e^{x_i^T\beta} x_i \frac{\partial}{\partial \beta^T} (x_i^T \beta) \\ &= \sum_{i=1}^{n} e^{x_i^T\beta} x_i x_i^T,\end{aligned}$$

given in matrix notation this is

$$H(\beta) = X^T \mathrm{diag}(\lambda) X.$$

The expected Fisher information then becomes

$$\begin{aligned}\mathrm{F}(\beta) &= E\big[\mathrm{H}(\beta)\big] \\ &= \sum_{i=1}^{n} e^{x_i^T \beta} x_i x_i^T = X^T \mathrm{diag}(\lambda) X \\ &= \mathrm{H}(\beta).\end{aligned}$$

b)

```
vec2diag = function(vec) {
    m = diag(length(vec))
    for (i in 1:length(vec)) {
        m[i, i] = vec[i]
    }
    return(m)
}

diag2vec = function(m) {
    vec = rep(0, nrow(m))
    for (i in 1:nrow(m)) {
        vec[i] = m[i, i]
    }
    return(vec)
}
```

```
myglm = function(formula, data, start = rep(0, ncol(model.matrix(formula, data))),
    epsilon = 1e-08) {
    n = nrow(data)
    X = model.matrix(formula, data)
    Y = matrix(data$y)
    beta = matrix(start)
    delta.beta = epsilon + 1
    counter = 0
    while (Norm(delta.beta) > epsilon) {
        lambda = matrix(exp(X %*% beta))
        F.beta = t(X) %*% vec2diag(lambda) %*% X
        s.beta = t(X) %*% (Y - lambda)
        delta.beta = solve(F.beta, s.beta)
        beta = beta + delta.beta
        counter = counter + 1
    }
    logY = ifelse(Y == 0, 1, log(Y))
    deviance = 2 * sum(Y * logY - Y - Y * log(lambda) + lambda)
    vcov = solve(F.beta)
    std.errors = sqrt(diag2vec(vcov))
    Coefficients = matrix(c(beta, std.errors), nrow = length(beta))
    colnames(Coefficients) = c("Estimate", "Std. Error")
    rownames(Coefficients) = rownames(beta)
    return(list(Coefficients = Coefficients, Deviance = deviance, vcov = vcov,
        X = X, p = length(beta), n = length(Y)))
}
```

c)

We test our function by comparing the results it gives on simulated data, to those obtained using the built-in functions glm and vcov.

```
n = 1000
real_beta1 = 0.1
real_beta0 = 1
t = runif(n, 0, 10)
y = rpois(n, exp(real_beta0 + real_beta1 * t))
test.data = data.frame(y = y, t = t)
mymodel = myglm(y ~ t, test.data)
model_compare = glm(y ~ t, family = "poisson", data = test.data)
```

Comparing the coefficients and std. errors:

```
mymodel$Coefficients
```

```
##               Estimate  Std. Error
## (Intercept) 1.00870950 0.034038918
## t           0.09831261 0.005118018
```

```
summary(model_compare)$coefficients
```

```
##               Estimate  Std. Error  z value      Pr(>|z|)
## (Intercept) 1.00870951 0.034038001 29.63480 5.324056e-193
## t           0.09831261 0.005117925 19.20947  3.084176e-82
```

Comparing the deviance:

```
mymodel$Deviance
```

```
## [1] 1034.105
```

```
model_compare$deviance
```

```
## [1] 1034.105
```

Comparing the covariance matrix of the coefficients:

```
mymodel$vcov
```

```
##              (Intercept)             t
## (Intercept)  0.001158648 -1.576250e-04
## t           -0.000157625  2.619411e-05
```

```r
vcov(model_compare)
```

```
##               (Intercept)             t
## (Intercept)  0.0011585855 -1.576173e-04
## t           -0.0001576173  2.619316e-05
```

All of the elements appear to be computed correctly.

2)

a)

We interpret the parameter λ_0 as the expected number of fledglings produced by each female if she starts breeding at the optimal time. θ we interpret as the optimal timing for breeding, based on the timing of important food recurces. Lastly, we interpret ω as a parameter for how fast the expected number of fledlings produced by each female detoriates when initiating breeding at a non-optimal date. If ω is small then λ_i decreases rapidly once t_i differs from θ. If, however, ω is large, a difference between t_i and θ is less severe concerning the decrease of λ_i.

b)

The poisson distribution belongs to the exponential family and we choose to use the cannonical link function.

$$
\begin{aligned}
\mu_i &= \lambda_i = \lambda_0 e^{\frac{(t_i-\theta)^2}{2\omega^2}} \\
\ln(\mu_i) &= \ln(\lambda_0) - \frac{(t_i-\theta)^2}{2\omega^2} \\
\ln(\mu_i) &= \ln(\lambda_0) - \frac{\theta^2}{2\omega^2} + \frac{\theta}{\omega^2}t_i - \frac{1}{2\omega^2}t_i^2 \\
g(\mu_i) &= t_i^T\beta
\end{aligned}
$$

After reparametrization the expectation of μ_i is connected to a set of linear predictors through a link function, therefore this is a GLM . The relationship between the GLM-parameters β and ω, θ and λ_0 is defined by

$$t_i^T\beta = \ln(\lambda_0) - \frac{\theta^2}{2\omega^2} + \frac{\theta}{\omega^2}t_i - \frac{1}{2\omega^2}t_i^2.$$

This gives $\beta_0 = \ln(\lambda_0) - \frac{\theta^2}{2\omega^2}$, $\beta_1 = \frac{\theta}{\omega^2}$ and $\beta_2 = \frac{-1}{2\omega^2}$.

c)

```r
load(url("https://www.math.ntnu.no/emner/TMA4315/2020h/hoge-veluwe.Rdata"))
df.birds = data
model = myglm(y ~ t + I(t^2), data = df.birds)
```

d)

```
model.wo.t2 = myglm(y ~ t, data = df.birds)

compare.nested.models = function(model0, model1) {
    LRT = model0$Deviance - model1$Deviance
    r = model1$p - model0$p
    probLRT = pchisq(LRT, df = r, lower.tail = FALSE)
    if (probLRT == 0)
        probLRT = "<1e^-16"
    m = matrix(c(LRT, probLRT), nrow = 1)
    colnames(m) = c("LRT", "Pr(>Chi)")
    return(m)
}
lrt.test = compare.nested.models(model.wo.t2, model)
lrt.test
```

```
##           LRT     Pr(>Chi)
## [1,] 11.92985 0.0005524157
```

We perform a hypothesis test where the null hypothesis is that there is no quadratic effect of t, against the alternative hypothesis that there exist a non-neglectable quadratic effect of t. The likehood-ratio test gives that if H_0 is correct it is only a 5.5241571×10^{-4} probability to observe equally or more extreme results. Based on this we reject the null hypothesis and conclude that there is evidence of a quadratic effect of t.

e)

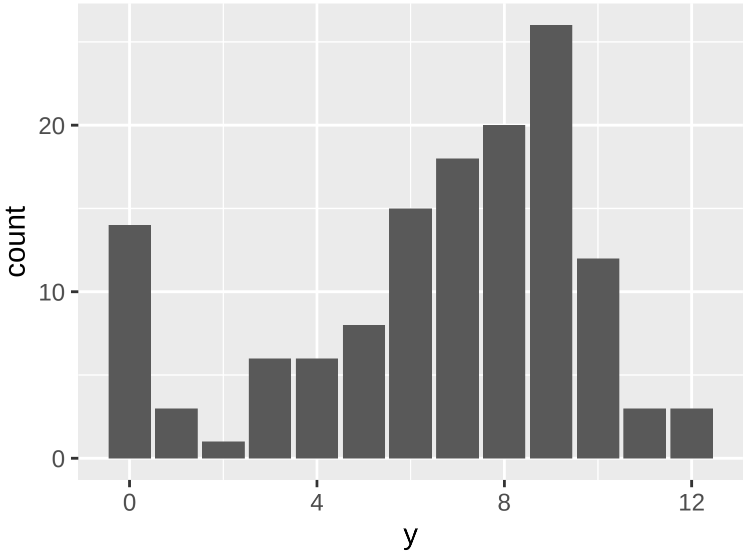
```
goodness.of.fit = function(model) {
    prob = pchisq(model$Deviance, model$n - model$p, lower.tail = FALSE)
    m = matrix(c(model$Deviance, prob), ncol = 2)
    colnames(m) = c("Deviance", "Pr(>Chi)")
    rownames(m) = "Model"
    return(m)
}

goodness.of.fit(model)
```

```
##       Deviance     Pr(>Chi)
## Model 277.4613 2.210642e-12
```

The model fits the data quite well, at least according to the goodness of fit test using deviance shown above. Plotting a histogram with y_i as parameter we get

```
ggplot(df.birds, aes(y)) + geom_bar()
```



It is difficult to assess from the plot above if the model assumptions are violated, since the histogram shows obersvations from a mix of different poission models (different expectation and variance). Therefore to conclude we take a closer look at the distribution of λ_i-values.

```
beta = model$Coefficients[, 1]
X = model$X
lambda = exp(X %*% beta)
lambda.min = min(lambda)
numb.lambdas.less.than.4 = length(lambda[lambda <= 4])
```

The smallest λ value is $\lambda_{min} = 1.89$. Additionally, out of 135 observations, we only estimate 5 observations to be drawn from a poisson distribution with expectation less than or equal to 4. Taking this into acccount, it is highly unlikely to observe 14 cases of $y_i = 0$ if the assumption of each y_i being poisson distributed is satisfied. However, even though the assumption of each y_i being poission distributed might be violated, it still appears that the model has good overall predictive ability. The model just underestimates the chance of $y_i = 0$.

To prove the claim that it is highly unlikely that each y_i is poission distributed with expectation λ_i, we want to find an upper limit for the probability of observing 14 or more cases of $y_i = 0$ given that y_i is poisson distributed. To do this we divide the observations into two groups. The first containing the y_is with $\lambda_i \leq 4$, and the second containing the ones with $\lambda_i > 4$. Since we only need an upper limit we assume all five observations in the first group was $y_i = 0$. Then we can study the probability of the second group getting $14 - 5 = 9$ observations equal to zero. Again, since we only need an upper bound, we assume each observation in second group has $\lambda_i = 4$. Then the probability of at least 9 $y_i = 0$ among the 130 observations in the second group, can be written as

$$\sum_{i=9}^{130} \frac{130}{i! \cdot (130-i)!} e^{-4i} \left(1 - e^{-4}\right)^{130-i}.$$

```
p = 0
for (i in 9:130) {
    k = factorial(130)/(factorial(i) * factorial(130 - i)) * exp(-4 * i) * (1 -
        exp(-4 * (130 - i)))
    p = p + k
}
```

Calculating this probability we get an upper bound for the probability of observing something at least as extreme as we have done, given the assumption of $y_i \sim \text{poisson}(\lambda_i)$. This upper bound is 0.0065138. Consequently, it is highly likely that our assumption of poisson distribution is violated.

f)

We can now easily find the maximum likelihood estimates of θ and ω since we have the maximum likelihood estimates of the β coefficents and we have three equations describing the relationship between λ_0, θ, ω and β. Solving two of these equations gives expressions for $\hat{\theta}$ and $\hat{\omega}$.

$$\hat{\omega}(\hat{\beta}) = \sqrt{\frac{-1}{2\hat{\beta}_2}}$$

$$\hat{\theta}(\hat{\beta}) = \hat{\beta}_1\hat{\omega}^2 = \frac{-\hat{\beta}_1}{2\hat{\beta}_2}$$

```
omega.hat = sqrt(-1/(2 * beta[3]))
theta.hat = beta[2] * omega.hat^2
```

The estimates are $\hat{\omega} = 12.312$ and $\hat{\theta} = 12.912$.

Using the Delta method we can find the variance of $\hat{\omega}(\hat{\beta})$ and $\hat{\theta}(\hat{\beta})$. Let G be a function of $\hat{\beta}$ then

$$\text{Var}[G(\hat{\beta})] = \nabla G(\hat{\beta})^T \text{Cov}[\hat{\beta}] G(\hat{\beta}). \tag{1}$$

The gradient of $\hat{\omega}(\hat{\beta})$ is

$$\nabla\hat{\omega}(\hat{\beta})^T = \begin{bmatrix} 0 & 0 & (-2\hat{\beta}_2)^{-3/2} \end{bmatrix}.$$

The gradient of $\hat{\theta}(\hat{\beta})$ is

$$\nabla\hat{\theta}(\hat{\beta})^T = \begin{bmatrix} 0 & \frac{-1}{2\hat{\beta}_2} & \frac{\hat{\beta}_1}{2\hat{\beta}_2^2} \end{bmatrix}.$$

Inserting these gradients into (1) we compute the variance of $\hat{\omega}$ and $\hat{\theta}$

```
grad.omega = matrix(c(0, 0, (-2 * beta[3])^(-3/2)))
se.omega.hat = sqrt(t(grad.omega) %*% model$vcov %*% grad.omega)

grad.theta = matrix(c(0, -1/(2 * beta[3]), beta[2]/(2 * beta[3]^2)))
se.theta.hat = sqrt(t(grad.theta) %*% model$vcov %*% grad.theta)

coef = matrix(c(c(omega.hat, theta.hat), c(se.omega.hat, se.theta.hat)), ncol = 2)
colnames(coef) = c("Estimate", "Std. Error")
rownames(coef) = c("Omega", "Theta")
coef
```

```
##       Estimate Std. Error
## Omega 12.31175   1.902526
## Theta 12.91197   1.609386
```

Using the Delta method we get $\text{SE}[\hat{\omega}] = 1.903$ and $\text{SE}[\hat{\theta}] = 1.609$.

g)

We want to examine if the optimal breeding date, $\hat{\theta}$, differs from the mean breeding date of the population, μ_i. In task 2f we estimate the optimal breeding date as $\hat{\theta} = 12.91$. An estimator for μ is

$$\hat{\mu} = \sum_{i=1}^{n} t_i.$$

```r
mu.hat = mean(df.birds$t)
```

An estimate for μ is therefore $\hat{\mu} = 15.937037$. To decide if the mean value of t in this population is significantly different from the estimated optimal date we perform a hypothesis test. Choosing $\alpha = 0.05$ as our significance level.

$$\begin{aligned} \mathrm{H}_0 : &\quad \mu = \theta \\ \mathrm{H}_1 : &\quad \mu \neq \theta \end{aligned}$$

We assume that each t_i is identically and independently distributed. Then we have by the central limit theorem that asymptotically $\hat{\mu}$ becomes normally distributed as $n \to \infty$. We also know that $\hat{\theta}$ is asymptotically normally distributed. Consequently $\hat{\mu} - \hat{\theta}$ is asymptotically normally distributed, since it is a linear combination of asymptotically normally distributed variables. From the definition of $\hat{\mu}$ it is clear that it is an unbiased estimator. We assume $\hat{\theta}$ is an unbiased estimator, and that $\hat{\theta}$ and $\hat{\mu}$ is independent of each other. This gives

$$\begin{aligned} \mathrm{E}[\hat{\mu} - \hat{\theta}] &= \mu - \theta \\ \mathrm{Var}[\hat{\mu} - \hat{\theta}] &= \mathrm{Var}[\hat{\mu}] + \mathrm{Var}[\hat{\theta}]. \end{aligned}$$

Since $\hat{\mu} - \hat{\theta}$ is asymptotically normally distributed and number of observations, $n = 135$, is quite large we have that

$$Z = \frac{(\hat{\mu} - \hat{\theta}) - (\mu - \theta)}{\mathrm{SE}[\hat{\mu} - \hat{\theta}]} \approx \frac{(\hat{\mu} - \hat{\theta}) - (\mu - \theta)}{\widehat{\mathrm{SE}}[\hat{\mu} - \hat{\theta}]}$$

is approximately standard normal distributed.

$$\begin{aligned} P(|Z| > z_\alpha) &= P(-z_\alpha \leq \frac{(\hat{\mu} - \hat{\theta}) - (\mu - \theta)}{\mathrm{SE}[\hat{\mu} - \hat{\theta}]}) \leq z_\alpha) \\ &= P((\hat{\mu} - \hat{\theta}) - z_\alpha \mathrm{SE}[\hat{\mu} - \hat{\theta}] \leq \mu - \theta \leq (\hat{\mu} - \hat{\theta}) + z_\alpha \mathrm{SE}[\hat{\mu} - \hat{\theta}]) \\ &\approx P((\hat{\mu} - \hat{\theta}) - z_\alpha \widehat{\mathrm{SE}}[\hat{\mu} - \hat{\theta}] \leq \mu - \theta \leq (\hat{\mu} - \hat{\theta}) + z_\alpha \widehat{\mathrm{SE}}[\hat{\mu} - \hat{\theta}]) \end{aligned}$$

```r
var.mu.hat = var(df.birds$t)/length(df.birds$t)
z.value = (mu.hat - theta.hat)/sqrt(var.mu.hat + se.theta.hat^2)
p.value = 2 * pnorm(z.value, lower.tail = F)
result.z.test = matrix(c(z.value, p.value), nrow = 1)
colnames(result.z.test) = c("Z-value", "Pr(>|Z|)")
rownames(result.z.test) = "H0: mu = theta"
result.z.test
```

```
##                   Z-value   Pr(>|Z|)
## H0: mu = theta 1.821029 0.06860242
```

Since the p-value is larger than the significance level $\alpha = 0.05$ we choose to not reject the null hypothesis. In other words we conclude that the mean breeding date for the population is not significantly different from the optimal breeding date.

3)

```
B = 1000
n = length(df.birds$y)
curr_data = df.birds
formel = formula(y ~ t + I(t^2))
X = model.matrix(formel, curr_data)
lambda = matrix(exp(X %*% beta))
betas = matrix(, nrow = length(beta), ncol = B)
for (i in 1:B) {
    y.sim = rpois(n, lambda)
    curr_data$y = y.sim
    fit = myglm(formel, curr_data, start = beta)
    betas[, i] = matrix(fit$Coefficients[, 1])
}

df.betas.bootstrap = data.frame(beta0 = betas[1, ], beta1 = betas[2, ], beta2 = betas[3,
    ])

var(df.betas.bootstrap)
```

```
##               beta0         beta1         beta2
## beta0  0.0810534429 -9.375393e-03  2.543442e-04
## beta1 -0.0093753927  1.160409e-03 -3.356127e-05
## beta2  0.0002543442 -3.356127e-05  1.038142e-06
```

```
model$vcov
```

```
##               (Intercept)             t        I(t^2)
## (Intercept)  0.0797693783 -9.308596e-03  2.550195e-04
## t           -0.0093085957  1.159672e-03 -3.369024e-05
## I(t^2)       0.0002550195 -3.369024e-05  1.039306e-06
```

The results from parametric bootstrapping are quite similar to those obtained from the expected Fisher information. This indicates that the asymptotic results is a good approximation even for finite sample sizes (n=135).