# TMA4315: Project 1

Jim Totland, Martin Gudahl Tufte

9/8/2021

Litt usikker på hva slags notasjon vi skal bruke, f. eks. boldface for vektorer eller ikke? Bare si ifra hvis du vil ha noe spesifikit:)

## Problem 1

### a)

Since the response variables $y_i \sim \text{Bernoulli}(p_i)$, where $p_i = \Pr(y_i = 1 \mid x_i) = \Phi(x_i^T \beta)$, the conditional mean is given by $\mathrm{E} y_i = p_i$, which is connected to the covariates via the following relationship:

$$x_i^T \beta =: \eta_i = \Phi^{-1}(p_i),$$

which implies that $p_i = \Phi(\eta_i)$. This results in the likelihood function

$$\begin{aligned} L(\beta) &= \prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{1-y_i} \\ &= \prod_{i=1}^{n} \Phi(\eta_i)^{y_i}(1-\Phi(\eta_i))^{1-y_i}. \end{aligned}$$

Thus, the log-likelihood becomes

$$l(\beta) := \ln(L(\beta)) = \sum_{i=1}^{n} y_i \ln(\Phi(\eta_i)) + (1-y_i)\ln(1-\Phi(\eta_i)) = \sum_{i=1}^{n} l_i(\beta).$$

To find the score function, we calculate

$$\begin{aligned} \frac{\partial l_i(\beta)}{\partial \beta} &= \frac{y_i}{\Phi(\eta_i)} \frac{\partial \Phi(\eta_i)}{\partial \beta} - \frac{1-y_i}{1-\Phi(\eta_i)} \frac{\partial \Phi(\eta_i)}{\partial \beta} \\ &= \frac{y_i}{\Phi(\eta_i)} \phi(\eta_i) x_i - \frac{1-y_i}{1-\Phi(\eta_i)} \phi(\eta_i) x_i \\ &= \frac{y_i(1-\Phi(\eta_i)) - (1-y_i)\Phi(\eta_i)}{\Phi(\eta_i)(1-\Phi(\eta_i))} \phi(\eta_i) x_i \\ &= \frac{y_i - \Phi(\eta_i)}{\Phi(\eta_i)(1-\Phi(\eta_i))} \phi(\eta_i) x_i. \end{aligned}$$

Consequently, the score function is given by

$$s(\beta) = \sum_{i=1}^{n} \frac{y_i - \Phi(\eta_i)}{\Phi(\eta_i)(1-\Phi(\eta_i))} \phi(\eta_i) x_i.$$

Next, we find the expected Fisher information, $F(\beta)$. We find it by using the result

$$\begin{aligned}
F(\beta) = \text{Var}(s(\beta)) &= \text{Var}\left(\sum_{i=1}^{n} \frac{y_i - \Phi(\eta_i)}{\Phi(\eta_i)(1-\Phi(\eta_i))}\phi(\eta_i)x_i\right) \\
&= \sum_{i=1}^{n} \underbrace{\left[\frac{\phi(\eta_i)}{\Phi(\eta_i)(1-\Phi(\eta_i))}\right]^2}_{=:\xi_i} \text{Var}(y_i x_i) = \sum_{i=1}^{n} \xi_i x_i \text{Var}(y_i) x_i^T \\
&= \sum_{i=1}^{n} \xi_i p_i (1-p_i) x_i x_i^T \\
&= \sum_{i=1}^{n} \frac{\phi(\eta_i)^2}{\Phi(\eta_i)(1-\Phi(\eta_i))} x_i x_i^T,
\end{aligned}$$

Where in the third equality we have used that the $y_i$'s are independent. The expected Fisher information can also be verified to have this expression by the relationship

$$F(\beta) = \sum_{i=1}^{n} \frac{h'(\eta_i)^2}{\text{Var}(y_i)} x_i x_i^T,$$

where $h'(\eta_i) = \Phi'(\eta_i) = \phi(\eta_i)$ and $\text{Var}(y_i) = p_i(1-p_i) = \Phi(\eta_i)(1-\Phi(\eta_i))$.

## b) The expected Fisher information is given by

$$F(\beta) = \sum_{i=1}^{n} \frac{\phi(\eta_i)^2}{\Phi(\eta_i)(1-\Phi(\eta_i))} x_i x_i^T = x^T W x,$$

where $W = \text{diag}\left(\frac{\phi(\eta_i)^2}{\Phi(\eta_i)(1-\Phi(\eta_i))}\right)$.

The Fisher scoring algorithm states that the next iterate is given by

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

Inserting the expected Fisher information and the score function we get

$$\beta^{(t+1)} = (x^T W^{(t)} x)^{-1} x^T W^{(t)} \tilde{y}^{(t)},$$

where the working response vector $\tilde{y}^{(t)}$ has element $i$ given by

$$\tilde{y}_i^{(t)} = x_i^T \beta^{(t)} + \frac{y_i - h(x_i^T \beta(t))}{h'(x_i^T \beta(t))} = \eta_i^{(t)} + \frac{y_i - \Phi(\eta_i^{(t)})}{\phi(\eta_i^{(t)})}.$$

Implementing myglm in R:

```
Phi <- function(x) return (pnorm(x))
phi <- function(x) return (dnorm(x))


myglm <- function(formula, data, start = NULL){

  # response variable
  resp <- all.vars(formula)[1]
  y <- as.matrix( data[resp] )
```

```r
  # model matrix
  X <- model.matrix(formula, data)
  n <- dim(X)[1]
  p <- dim(X)[2]


  # starting beta
  if (is.null(start)){
    beta = rep(0, p)
  }
  else {
    beta = start
  }


  # Fisher scoring algorithm
  max_iter <- 50
  tol <- 1e-10
  iter <- 0
  rel.err <- Inf

  while (rel.err > tol & iter < max_iter){
    # calculate eta, y tilde, W
    eta <- X %*% beta
    y.tilde <- eta + (y - Phi(eta)) / (phi(eta))
    W <- diag( as.vector(phi(eta)^2 / (Phi(eta)*(1-Phi(eta)))), n, n )


    # update beta
    A <- t(X) %*% W %*% X
    b <- t(X) %*% W %*% y.tilde
    beta.new <- solve(A, b)

    iter <- iter + 1
    rel.err <- max(abs(beta.new - beta) / abs(beta.new))
    beta <- beta.new
  }




  # remains to find the coefficients matrix, deviance and estimated variance matrix

  coeff <- 1
  deviance <- 1
  vcov <- 1

  return (beta)
}

#beta <- myglm(menarche ~ age, juul.girl)
```

```r
#beta

#X <- model.matrix(menarche ~ age, juul.girl)
```

### c)

```r
# probability
x = runif(1000, 0, 1)
# draw n bernoulli with prob x
y <- rbinom(1000, 1, x)

df <- data.frame(y, x)



### fit using glm
model <- glm(y ~ poly(x,2), family = binomial(link = "probit"), data = df)

# beta
model$coefficients
```

```
## (Intercept) poly(x, 2)1 poly(x, 2)2
## -0.07806477 28.42962567  1.07999646
```

```r
# vcov
vcov(model)
```

```
##               (Intercept)   poly(x, 2)1 poly(x, 2)2
## (Intercept)  0.0022194112 -0.0002758148  0.02036958
## poly(x, 2)1 -0.0002758148  2.6475022407  0.06215754
## poly(x, 2)2  0.0203695792  0.0621575350  2.60350461
```

```r
# deviance
# ...

### fit using myglm
beta <- myglm(y ~ poly(x,2), data = df)

# beta
t(beta)
```

```
##      (Intercept) poly(x, 2)1 poly(x, 2)2
## [1,] -0.07806366    28.42966    1.080062
```

```r
# vcov
# ...

# deviance
# ...
```

## Problem 2

### a)

```r
#install.packages("ISwR")
library(ISwR) # Install the package if needed
data(juul)
juul$menarche <- juul$menarche - 1
juul.girl <- subset(juul, age>8 & age<20 & complete.cases(menarche))
```

```r
#model <- glm(menarche ~ age, family=familytype(link="probit"), data= juul.girl)
```