Project 2 - TMA4315

martin.o.berild@ntnu.no
10014

yaolin.ge@ntnu.no
10026

October 30, 2020

Initially, we load the R-package `VGAM` to fit *Vector Generalized Linear and Addtive Models*, and `ggplot` as our plotting tool of choice.

```
library(VGAM)
library(ggplot2)
```

In this project, we will use multinomial regression on a dataset containing marital status, etnicity and age of individuals in New Zealand. The data is loaded in the code below, and the print display seven individuals.

```
attach(marital.nz)
head(marital.nz)
```

```
##   age ethnicity          mstatus
## 1  29  European           Single
## 2  55  European  Married/Partnered
## 3  44  European  Married/Partnered
## 4  53  European Divorced/Separated
## 5  45  European  Married/Partnered
## 7  30  European           Single
```

There are a total of four catagories for marital status (`mstatus`); `Single`, `Married/Partnered`, `Divorced` and `Widowed`. And we will try to predict the probabilty of each based on the age. A multinomial regression model with a linear effect of `age` is fit to the data as

```
 mod1 <- vglm(mstatus ~ age, multinomial)
 summary(mod1)
```

```
##
## Call:
## vglm(formula = mstatus ~ age, family = multinomial)
##
## Pearson residuals:
##                         Min      1Q   Median       3Q    Max
## log(mu[,1]/mu[,4]) -11.75 -0.1441 -0.13965 -0.13372  5.706
## log(mu[,2]/mu[,4]) -13.53  0.2871  0.31147  0.40939  1.212
## log(mu[,3]/mu[,4]) -12.47 -0.2364 -0.09098 -0.02037 82.311
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept):1  6.753157   0.515150   13.11   <2e-16 ***
## (Intercept):2  9.531824   0.482073   19.77   <2e-16 ***
## (Intercept):3 13.121214   0.513771   25.54   <2e-16 ***
## age:1         -0.099335   0.008043  -12.35   <2e-16 ***
## age:2         -0.102873   0.007100  -14.49   <2e-16 ***
```

```
## age:3           -0.252080    0.008955   -28.15    <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Names of linear predictors: log(mu[,1]/mu[,4]), log(mu[,2]/mu[,4]),
## log(mu[,3]/mu[,4])
##
## Residual deviance: 6822.79 on 18153 degrees of freedom
##
## Log-likelihood: -3411.395 on 18153 degrees of freedom
##
## Number of Fisher scoring iterations: 7
##
## Warning: Hauck-Donner effect detected in the following estimate(s):
## '(Intercept):2', 'age:3'
##
##
## Reference group is level  4  of the response
```

The `mstatus` is a categorical nominal response $Y \in \{1, 2, 3, 4\}$ modelled with the linear effect of a numerical covariate `age`. Given a catoregory r of the response, the probability of the subject having the rth `mstatus` is given by

$$\pi_r = P(Y = r), \quad r = 1, 2, 3, 4.$$

The response is usually reformulated to a vector $\boldsymbol{y}$ of $c = 3$ dummy variables:

$$y_r = \begin{cases} 1, & Y = r \\ 0, & \text{otherwise} \end{cases} \qquad r = 1, 2, 3.$$

where the 4th category is a *refrence* category, i.e. $y_4 = 1 - y_1 - y_2 - y_3$. The multinomial distribution for m independent trials is given by

$$
\begin{aligned}
f(\boldsymbol{y}|\boldsymbol{\pi}) &= \frac{m!}{y_1! \cdot y_2! \cdot y_3! \cdot (1 - y_1 - y_2 - y_3)!} \pi_1^{y_1} \cdot \pi_2^{y_2} \cdot \pi_3^{y_3} \cdot (1 - \pi_1 - \pi_2 - \pi_3)^{1 - y_1 - y_2 - y_3} \\
&= \mathcal{M}(m, \boldsymbol{\pi}),
\end{aligned}
$$

where $\mathcal{M}(m, \boldsymbol{\beta})$ is the multinomial probability mass function with m trials and probabilities $\boldsymbol{\pi} = (\pi_1, \pi_2, \pi_3)$. Using the linear predictor $\eta_{i,r} = \boldsymbol{x}_i^T \boldsymbol{\beta}_r$ and considering a logit model, we have the probabilities:

$$\pi_{i,r} = \frac{\exp(\boldsymbol{x}_i^T \boldsymbol{\beta}_r)}{\sum_{s=1}^{c+1} \exp(\boldsymbol{x}_i^T \boldsymbol{\beta}_s)}. \tag{1}$$

However, this model is non-identifiable because adding a constant to the linear predictor would yield the equivalent probabilities. Therefore, as mentioned earlier, one of the categories is used as *reference* category s.t. $\boldsymbol{\beta}_r = \boldsymbol{\beta}_r - \boldsymbol{\beta}_{c+1}$, where $c + 1 = 4$ in this model. The resulting probabilities are

$$\pi_{i,r} = \frac{1}{1 + \sum_{s=1}^{c} \exp(\boldsymbol{x}_i^T \boldsymbol{\beta}_s)} \begin{cases} \exp(\boldsymbol{x}_i^T \boldsymbol{\beta}_r), & r = 1, 2, \ldots, c \\ 1, & \text{otherwise} \end{cases}. \tag{2}$$

An alternative representation is the logarithmic odds or relative risk between category r and the reference category $c + 1$ as

$$\log \frac{\pi_{i,r}}{\pi_{i,c+1}} = \boldsymbol{x}_i^T \boldsymbol{\beta}_r. \tag{3}$$

This logarithmic odds is also the output of a `predict()` call using the model object `mod1`, and is presented in the `Pearson residuals` in the `summary()` output above. Note that with this representation a positive

effect β_r only implies that the odds of category r increase relative the reference category, and not that the probability of r increase by itself.

In general, the interpretation of odd ratio is the magnitude of change in the odds between two categories r_1 and r_2 for a individual i by a unit change in the jth element of the covariates, $x_{i,j}$, as

$$\boldsymbol{x}_i^* = \boldsymbol{x}_i + (0,\ldots,1,\ldots,0)^T,$$

which yields the odds ratio:

$$\frac{\pi_{i,r_1}^*/\pi_{i,r_2}^*}{\pi_{i,r_1}/\pi_{i,r_2}} = \frac{\exp[\boldsymbol{x}_i^{*T}(\boldsymbol{\beta}_{r_1} - \boldsymbol{\beta}_{r_2})]}{\exp[\boldsymbol{x}_i^T(\boldsymbol{\beta}_{r_1} - \boldsymbol{\beta}_{r_2})]} = \exp[(\boldsymbol{x}_i^* - \boldsymbol{x}_i)^T(\boldsymbol{\beta}_{r_1} - \boldsymbol{\beta}_{r_2})] = \exp(\boldsymbol{\beta}_{r_1,j} - \boldsymbol{\beta}_{r_2,j}).$$

A similar interpretation can be found for the odds between category r and the complementary of r (or simply not in the category r), and is formulized as

$$\frac{\pi_{i,r}}{1 - \pi_{i,r}} = \frac{\exp(\boldsymbol{x}_i^T\boldsymbol{\beta}_r)}{1 + \sum_{s \in \mathcal{S}}\exp(\boldsymbol{x}_i^T\boldsymbol{\beta}_s)},$$

where $\mathcal{S} = \{s \in [1,c] \setminus r\}$ is the set containing all categories but r and the reference.

Observe that $\pi_r(\boldsymbol{x}) \propto \exp(\boldsymbol{x}^T\boldsymbol{\beta}_r)$ with the denominator being the a normalizing constant, and that the effects $\boldsymbol{\beta}_r$ includes a interecept $\beta_{0,r}$ and the effect of age $\beta_{\text{age},r}$ for category r. Therefore, as $x > 0$ the function $\pi(\boldsymbol{x})$ is a monotonic function only for a intercept $\beta_{0,r} = 0$, and otherwise it's slope is changing from decreasing to increase or oppositley depending on the sign of the effect of age.

We could also formulate the multinomial logistic regression as a latent utility model:

$$u_r = \eta_r + \epsilon_r,$$

where u_r is the utility, η_r the linear predictor and ϵ_r some error of the rth category. Furthermore, we assume that the error follow a standard Gumbel distribution, $\epsilon \sim \text{Gumbel}(0,1)$. The aim is to find the probabilities of choosing $Y = r$ for all r or find the probability that the rth utility, $u_{i,r}$ is larger than all other utilities for a given subject i. The differece between two identically distributed Gumbel distributions follows a logistic distibution and, thus, the probabilities π_r in the latent utility model is found similar previous formulation using Equation (1).

Thus far, we have just assumed that the linear effect `age` is statistically significant in modelling the `mstatus`; however, we would like perform a test to confirm our assumption. A such test could be a *Likelihood-ratio test* (LRT), which compares the goodness-of-fit in the *null* model (only a intercept) and the fitted model `mod1` with a linear effect of `age`. The test uses the fact that the difference in log-likelihood beteen the two models is chi-squared distributed as

$$-2[\ell(\hat{\boldsymbol{\beta}}_n) - \ell(\hat{\boldsymbol{\beta}}_a)] \sim \mathcal{X}_{p_n - p_a}^2,$$

where the index n defines the *null* model and a the alternative model, $\ell()$ is the log-likelihood, $\hat{\boldsymbol{\beta}}$ are the fitted parameters of the respective models, and p is the number of parameters in each model. Thus, the degrees of freedom of the chi-squared distribution is the difference in number of parameters in the two models. Given a significance level let's say $\alpha = 0.05$, we have the critical point with 3 degrees of freedom $q_{\alpha=0.05,\nu=3} \simeq 7.81$ and the likelihood ration, (or deviance if you want), is $\text{LR} = -2[\ell(\hat{\boldsymbol{\beta}}_n) - \ell(\hat{\boldsymbol{\beta}}_a)] \simeq 1600.9$. Since $\text{LR} >> q_{\alpha=0.05,\nu=3}$ the model with the linear effect of `age` is much better than the *null*-model, i.e. the linear effect of `age` is statistically significant. We could also have use `anova()` to show this as presented in the code below.

```
mod0 <- vglm(mstatus ~ 1, multinomial)
anova(mod0, mod1, test="LRT", type="I")

## Analysis of Deviance Table
##
```

```
## Model 1: mstatus ~ 1
## Model 2: mstatus ~ age
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1     18156     8423.7
## 2     18153     6822.8  3   1600.9 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Here, the p-value is much lower than the significance level of 0.05 and we observe that the deviance is similar to our previous calculations.

To obtain predictions of the probabilties of the belonging to the respective categories given a age between 16 and 88 can be found rewriting Equation (3) using Equation (2). The different probabilities can be visualized in the figure below, where the probabilities functions $\pi_r(x) = P(Y = r)$ is plotted against the covariate `age`, x.

```r
ymp1 = predict(mod1,newdata = data.frame(age=seq(16,88)))
ymp1.df = as.data.frame(matrix(cbind(seq(16,88),exp(ymp1)*(1/(1 + rowSums(exp(ymp1)))),
                              (1/(1 + rowSums(exp(ymp1))))),ncol=5,
                        dimnames = list(c(),c("age","pi1","pi2","pi3","pi4"))))
```
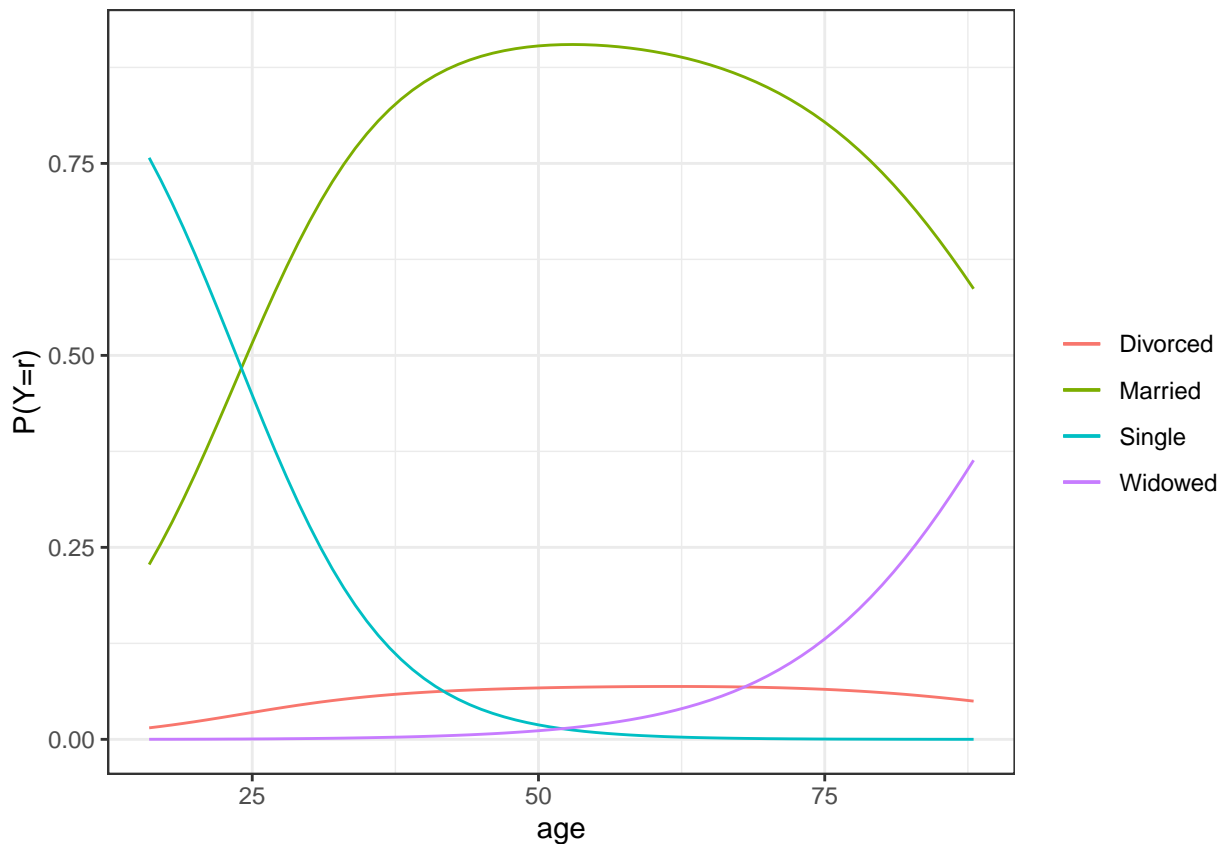

The number of `Single` is monotonically decreasing as the age gets higher, which is reasonable since people generally get `Married` or find a partner during their life. And since a individual can't go back to being single because of the `Divorced/Seperated` category, it can't start increasing again as age increases.

The probability of getting `Married` however is increasing until age is $\simeq 50$, where it starts to decrease. This could be explained by individual getting `Divorced` or `Seperated` after some time together, or that their partner dies resulting in being placed in the `Widowed` category.

The `Divorced`category is close to horisontal but is slightly concave with a maximum around age 65, and

lastly, the widow is monotonically increasing. Both of these functions have a reasonable behavior as the number of proportion of `Widowed` should go up because of the correlation between `age`and `mortality`.

Thus far we have only considered a linear effect of `age`; however, models that regard a linear effect of higher degrees of `age` might yield a better approximation. We will now fit different multinomial logit models with degrees of `age` up to four as covariates and, then, perform model selection based on the *Akaike information criterion* (AIC). In the code below the models are fit and the probabilities of `age` betwen 16 to 88 is predicted.

```r
ymp0 = predict(mod0,newdata = data.frame(age=seq(16,88)))
ymp0.df = as.data.frame(matrix(cbind(seq(16,88),exp(ymp0)*(1/(1 + rowSums(exp(ymp0)))),
                                     (1/(1 + rowSums(exp(ymp0)))))),ncol=5,
                          dimnames = list(c(),c("age","pi1","pi2","pi3","pi4"))))
mod2 <- vglm(mstatus ~ poly(age,2), multinomial)
ymp2 = predict(mod2,newdata = data.frame(age=seq(16,88)))
ymp2.df = as.data.frame(matrix(cbind(seq(16,88),exp(ymp2)*(1/(1 + rowSums(exp(ymp2)))),
                                     (1/(1 + rowSums(exp(ymp2)))))),ncol=5,
                          dimnames = list(c(),c("age","pi1","pi2","pi3","pi4"))))
mod3 <- vglm(mstatus ~ poly(age,3), multinomial)
ymp3 = predict(mod3,newdata = data.frame(age=seq(16,88)))
ymp3.df = as.data.frame(matrix(cbind(seq(16,88),exp(ymp3)*(1/(1 + rowSums(exp(ymp3)))),
                                     (1/(1 + rowSums(exp(ymp3)))))),ncol=5,
                          dimnames = list(c(),c("age","pi1","pi2","pi3","pi4"))))
mod4 <- vglm(mstatus ~ poly(age,4), multinomial)
ymp4 = predict(mod4,newdata = data.frame(age=seq(16,88)))
ymp4.df = as.data.frame(matrix(cbind(seq(16,88),exp(ymp4)*(1/(1 + rowSums(exp(ymp4)))),
                                     (1/(1 + rowSums(exp(ymp4)))))),ncol=5,
                          dimnames = list(c(),c("age","pi1","pi2","pi3","pi4"))))
```

Using these predicted probabilities the figure below is constructed to visualized the dissimilarities between the models (presented in different colors). The associated categories is highlighted on the y-axis.
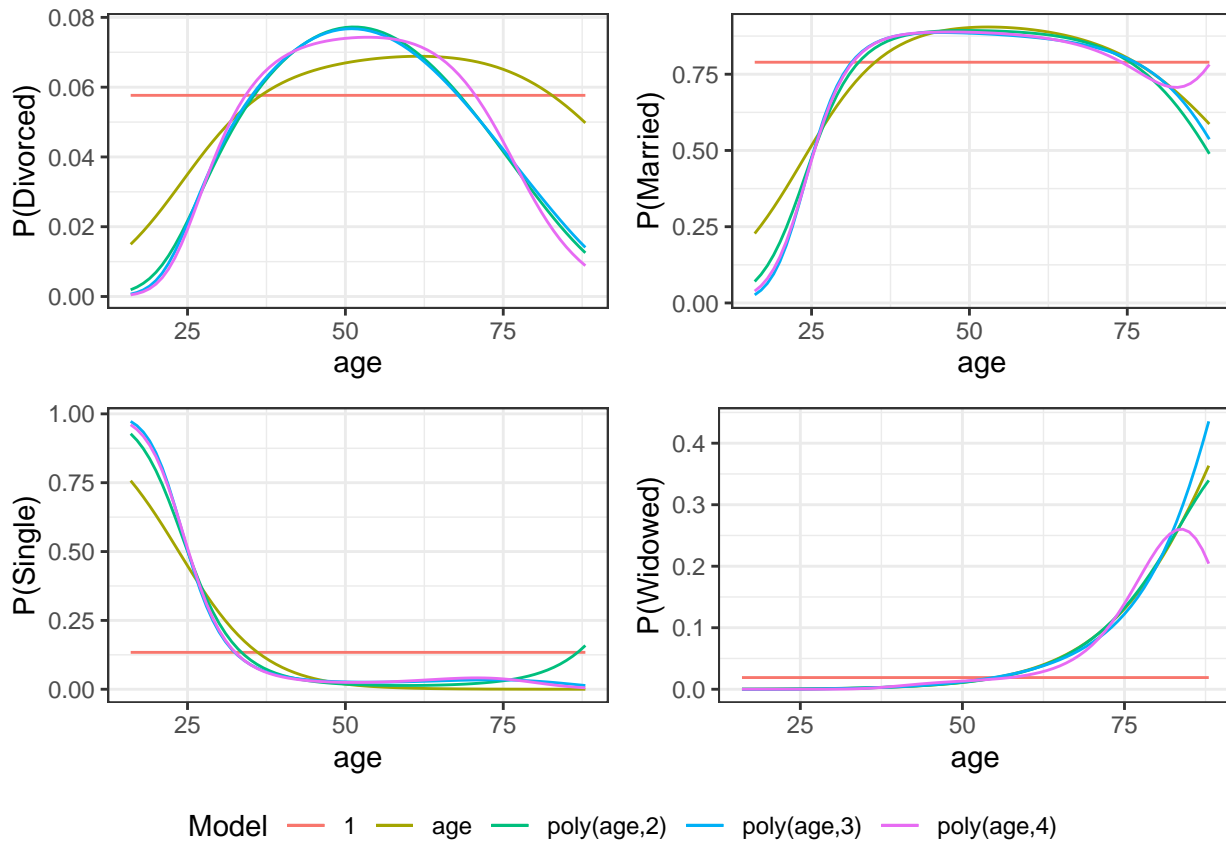
As previously mentioned the model selection will be based on minimizing the AIC which is calculated as

$$\text{AIC} = 2 \cdot k - 2 \cdot \ln(\hat{\text{L}}).$$

Here, k is the number of parameters in the model and $\hat{\text{L}}$ is the likelihood. The AIC then tries to asses the goodness-of-fit through the likelihood but also considers the complexity or number of parameters of the model as a penalty. For example in the model with a 4th degree polynomial of `age` we have $k = 3 \cdot 5$ parameters; 3 from the number of categories $c = 3$ using the $r = 4$ as reference, and 5 for the number of terms in a 4th degree polynomial. In R we can use the built in function `AIC()` to calculate the AIC for a `VGAM` object. The AIC values for the respective models is printed below.

```
AIC(mod0)
```

```
## [1] 8429.723
```

```
AIC(mod1)
```

```
## [1] 6834.79
```

```
AIC(mod2)
```

```
## [1] 6583.123
```

```
AIC(mod3)
```

```
## [1] 6555.048
```

```
AIC(mod4)
```

```
## [1] 6552.665
```

Here, we observe that the most complex model, a 4th degree polynomial of `age`, has the highest AIC score. Considering the predicted probabilities in the above figures, the best model seem to overfit the data as seen

by the tails for high values of `age` in `Married` and `Widowed` turn alot compared to the other models. In other words, the model might perform bad on new unseen data.