TMA4315 Generalized linear models H2018

Module 6: Categorical regression

Mette Langaas, Department of Mathematical Sciences, NTNU, with contibutions from Ingeborg G. Hem

25.10.2018 [PL], 26.10.2018 [IL]

Contents

(Latest changes: 13.12: typo with w_i for multivariate exp fam, 27.10 a few typos corrected).

Overview

Learning material

This topic is *new* on the reading list this year.

- Textbook: Fahrmeir et al (2013): Chapter 6 (not p 344-345 nominal models and latent utility models, not 6.3.2 Sequential model, and not category specific varables on page 344-3458).
- Classnotes 25.10.2018

Topics

- multinomial random component
- nominal vs. ordinal response
- ungrouped and grouped data
- multivariate exponential family
- nominal response and logit models
- ordinal reponse and logit models - based on a latent model
- likelihood inference

Jump to interactive.

Categorical random component

We consider a situation where our random variable (response) is given as one of $c+1$ possible categories (where we will look at category $c+1$ as the reference category).

The categories will either be

- Unordered: *nominal response variable*. Example: food types in alligator example.
- Ordered: *ordered response variable*. Example: degrees of mental impairment.

Assumptions:

- *Independent* observation pairs $(\mathbf{Y}_i, \mathbf{x}_i)$.
- π_{ir}: probability that the response is category r for subject i.
- $\sum_{s=1}^{c+1} \pi_{is} = 1$ for all i, so that $\pi_{i,c+1} = 1 - \sum_{s=1}^{c} \pi_{is}$. So, we have c probabilities to estimate.
- Further, the covariate vector $\mathbf{x}_i$ consists of the same measurements for each response category (that is, not different covariate types that are measured for each response category - which in our textbook is written as *independent of the response category*).

When coding the response variable we use a dummy variable coding with c elements (the $c+1$ category is the reference level). This means that if we have that $\pi_{ir} = 1$ then $\mathbf{y}_i = (0, 0, \ldots, 0, 1, 0, \ldots, 0)$ with a value of 1 in the rth element of $\mathbf{y}_i$. If observation i comes from category $c+1$ we have $\mathbf{y}_i = (0, 0, \ldots, 0)$.

Categorical regression

is modelling and estimating the probabilites $\pi_{ir} = P(Y_i = r) = P(Y_{ir} = 1)$ as a function of the covariates $\mathbf{x}_i$. The modelling is done differently for nominal (unordered) and ordered categories, but both rely upon the multinomial distribution.

The multinomial distribution

Probability mass function for one observation:

$$f(\mathbf{y}) = \pi_1^{y_1}\pi_2^{y_2}\cdots\pi_c^{y_c}(1-\pi_1-\pi_2-\cdots-\pi_c)^{1-y_1-y_2-\cdots-y_c}$$

where then $\mathbf{y} = (y_1, y_2, \ldots, y_c)$ and $y_r = 1$ if the observation comes from the rth category.

If we then have m independent trials then $\mathbf{y} = (y_1, y_2, \ldots, y_c)$ is summed over our m responses, so that y_r is the number of observations where the response is from the rth category.

$$f(\mathbf{y}) = \frac{m!}{y_1!\cdots y_c!(m-y_1-\cdots-y_c)!}\pi_1^{y_1}\pi_2^{y_2}\cdots\pi_c^{y_c}(1-\pi_1-\pi_2-\cdots-\pi_c)^{m-y_1-y_2-\cdots-y_c}$$

The mean and the covariance matrix of the random vector $\mathbf{Y}$ are given by:

$$\mathrm{E}(\mathbf{Y}) = m\boldsymbol{\pi} = \begin{pmatrix} m\pi_1 \\ m\pi_2 \\ \vdots \\ m\pi_c \end{pmatrix}$$

$$\mathrm{Cov}(\mathbf{Y}) = m\begin{pmatrix} \pi_1(1-\pi_1) & -\pi_1\pi_2 & \cdots & -\pi_1\pi_c \\ -\pi_2\pi_1 & \pi_2(1-\pi_2) & \cdots & -\pi_2\pi_c \\ \vdots & \vdots & \ddots & \vdots \\ -\pi_c\pi_1 & -\pi_c\pi_2 & \cdots & \pi_c(1-\pi_c) \end{pmatrix}$$

Q: what about $\mathrm{E}(Y_{c+1})$ and $\mathrm{Cov}(Y_1, Y_{c+1})$?

Finally, if we look at $\bar{Y}_r = \frac{1}{m}Y_r$ then $\bar{\mathbf{Y}} = \frac{1}{m}\mathbf{Y}$ follows a scaled multinomial distribution $\bar{\mathbf{Y}} \sim \frac{1}{m}M(m, \boldsymbol{\pi})$ with $\mathrm{E}(\bar{\mathbf{Y}}) = \boldsymbol{\pi}$ and $\mathrm{Cov}(\bar{\mathbf{Y}}) = \frac{1}{m^2}\mathrm{Cov}(\mathbf{Y})$.

Data

Ungrouped data

$$\mathbf{Y} = \begin{pmatrix} Y_{11} & Y_{12} & \cdots & Y_{1c} \\ Y_{21} & Y_{22} & \cdots & Y_{2c} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{n1} & Y_{n2} & \cdots & Y_{nc} \end{pmatrix}$$

and $\mathbf{X}$ is an $n \times p$ matrix as usual.

Grouped data

As for the binomial case we look at the number of occurences with a group - that is, one covariate pattern.

$$\mathbf{Y} = \begin{pmatrix} Y_{11} & Y_{12} & \cdots & Y_{1c} \\ Y_{21} & Y_{22} & \cdots & Y_{2c} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{G1} & Y_{G2} & \cdots & Y_{Gc} \end{pmatrix}$$

The notation here is that we have n_i observation for each covariate pattern (group) i for $i = 1, \ldots, G$. This will replace the m used for the multinomial distribution above.

Regression with nominal responses

nominal=unordered

Agresti (2015, p203): "The model treats the response variable as nominal scale in the following sense: if the model holds and the outcome categories are permuted in any way, the model still holds with the corresponding permuatation of the effects."

This is a generalization of the binary logit model with $P(Y = 1)$ vs $P(Y = 0)$, to c models of π_{ir} vs $\pi_{i,c+1}$ for $r = 1, \ldots, c$.

The models can be written using log ratios:

$$\ln(\frac{\pi_{ir}}{\pi_{i,c+1}}) = \mathbf{x}_i^T \boldsymbol{\beta}_r$$

Remark: $\boldsymbol{\beta}_r$ is the $p \times 1$ coefficient vector for the rth response

Using this we may also look at the log ratio for any two probabilites π_{ia} and π_{ib}:

$$\ln(\frac{\pi_{ia}}{\pi_{ib}}) = \ln(\frac{\pi_{ia}}{\pi_{i,c+1}}) - \ln(\frac{\pi_{ib}}{\pi_{i,c+1}}) = \mathbf{x}_i^T (\boldsymbol{\beta}_a - \boldsymbol{\beta}_b)$$

Alternatively, we may write out the model for the probabilites:

$$P(Y_i = r) = \pi_{ir} = \frac{\exp(\mathbf{x}_i^T \boldsymbol{\beta}_r)}{1 + \sum_{s=1}^{c} \exp(\mathbf{x}_i^T \boldsymbol{\beta}_s)}$$

$$P(Y_i = c+1) = \pi_{i,c+1} = 1 - \pi_{i1} - \cdots \pi_{ic} = \frac{1}{1 + \sum_{s=1}^{c} \exp(\mathbf{x}_i^T \boldsymbol{\beta}_s)}$$

Multivariate GLM

This is a multivariate GLM and the multinomial distribution is *a multivariate exponential family.*

$$f(\mathbf{y}_i, \boldsymbol{\theta}_i, \phi) = \exp(\frac{\mathbf{y}_i^T \boldsymbol{\theta}_i - b(\boldsymbol{\theta}_i)}{\phi} w_i + c(\mathbf{y}_i, \phi, w_i))$$

where $\boldsymbol{\theta}$ has dimension c.

Multivariate GLM-set-up

1. $\mathbf{Y}_i$ is multinomial with

$$\boldsymbol{\mu}_i = \mathrm{E}(\mathbf{Y}_i) = \boldsymbol{\pi}_i = \begin{pmatrix} \pi_{i1} \\ \pi_{i2} \\ \vdots \\ \pi_{i,c+1} \end{pmatrix}$$

Remark: if grouped data we instead look at $\bar{\mathbf{Y}}_i \sim \frac{1}{n_i} M(n_i, \pi_i)$ so that the mean is $\boldsymbol{\pi}_i$

---

2. Linear predictor is now a $c \times 1$ vector:

$$\boldsymbol{\eta}_i = \begin{pmatrix} \eta_{i1} \\ \eta_{i2} \\ \vdots \\ \eta_{i,c} \end{pmatrix} = \begin{pmatrix} \mathbf{x}_i^T \boldsymbol{\beta}_1 \\ \mathbf{x}_i^T \boldsymbol{\beta}_2 \\ \vdots \\ \mathbf{x}_i^T \boldsymbol{\beta}_c \end{pmatrix}$$

---

3. Link functions (c of those): $\mathbf{g}(\boldsymbol{\mu}_i) = \boldsymbol{\eta}_i$ where for the *nominal logit data model* element r (for $r = 1, \ldots, c$) of $\mathbf{g}$ is

$$g_r(\boldsymbol{\mu}_i) = \ln(\frac{\mu_{ir}}{1 - \mu_{i1} - \cdots - \mu_{ic}}) = \ln(\frac{\pi_{ir}}{1 - \pi_{i1} - \cdots - \pi_{ic}})$$

We also define response functions ($\mathbf{h}$) with elements h_r given by $\pi_{ir} = h_r(\eta_{i1}, \eta_{i2}, \ldots, \eta_{ic})$, and we have for the *nominal data model*

$$\pi_{ir} = h_r((\eta_{i1}, \eta_{i2}, \ldots, \eta_{ic}) = \frac{\exp(\eta_{ir})}{1 + \sum_{s=1}^{c} \exp(\eta_{is})}$$

It turns out that the reference category logits are the canonical links for the multinomial distribution GLM.

In this case, as for the univariate eksponential family GLM the loglikelihood is concave with an unique maximum (if it exists) and the expected and observed Fisher information matrices are equal.

As before, we find maximum likelihood parameter estimates from the Fisher scoring or Newton Raphson method.

Remember: now we have $p \times c$ parameters to estimate — p for each category c. All of these coefficients may either be put into a long vector (length $p \cdot c$) — which might be easiest to understand for the estimation, or into a matrix of dimension $p \times c$ — might be easier for viewing.

Likelihood

(grouped data)

With the notation that $\boldsymbol{\beta}$ is a long vector with the coefficients for the c categories stacked upon eachother.

$$L(\boldsymbol{\beta}) = \Pi_{i=1}^{G} f(\mathbf{y}_i \mid \boldsymbol{\pi})$$

where f is the multinomial distribution function.

Loglikelihood

$$l(\boldsymbol{\beta}) \propto \sum_{i=1}^{G} \sum_{s=1}^{c+1} y_{is} \ln(\pi_{is})$$

where we remember that $y_{i,c+1} = n_i - y_{i1} - \cdots - y_{ic}$, and $1 - \pi_{i1} - \cdots \pi_{ic}$.

(This formula is also correct for the ordinal model of the next section.) General formulas for the score function and expected Fisher information matrix follow later.

Deviance

The derivation used for the binary GLM model generalizes directly ot the multinomial GLM. The fitted probabilities are $\hat{\pi}_{ij}$ (group i and category j) and the saturated model (grouped data) is $n_i \tilde{\pi}_{ij} = y_{ij}$.

$$D = 2 \sum_{i=1}^{G} \sum_{s=1}^{c+1} y_{is} \ln(\frac{y_{is}}{n_i \hat{\pi}_{is}})$$

The asymptotic distribution is as before χ^2 with "the number of groups times number of categories minus 1 (Gc)" minus "the number of covariates (cp)", giving $Gc - cp = c(G - p)$ degrees of freedom for the nominal model.

The deviance can be used for model check with grouped data (G groups with n_i observations), but can be used to compare nested unsaturated models also for individual (ungrouped) data, with again an asymptotic χ^2 distribution with the difference of number of parameters between the two models.

This formula is also correct for the ordinal model of the next section, except that the number of parameters estimated differ. For the ordinal model to come next we have to estimate $k + c$ instead of p parameters, so the formula for degrees of freedom for the deviance for the ordinal model is $Gc - k - c$.

Alligators example

Example and data are taken from Agresti (2015, pages 217-219).

Research question: what is the factors influencing the primary food choice of alligators?

Data are from 219 captured alligators from four lakes in Florida, where the stomack contents of the alligators were investigated. The weight of different types of food was measured, and then the primary food choice (highest weight) was noted. The primary choice is given as y1:y5 below. In addition the size of the alligator (non-adult or adult) was registered.

- lake: each of the 4 lakes in Florida (1:4)
- size: non-adult=the size of the alligator (0: 2.3 meters or smaller) and adult=(1: larger than 2.3 meters)
- y1: fish
- y2: inverterbrate
- y3: reptile
- y4: bird
- y5: other

These data are grouped, and we let y1:fish be the reference category.

```
# data from Agresti (2015), section 6, with use of the VGAM packages
data = "http://www.stat.ufl.edu/~aa/glm/data/Alligators.dat"
ali = read.table(data, header = T)
ali
attach(ali)
```

```
##   lake size y1 y2 y3 y4 y5
## 1    1    1 23  4  2  2  8
## 2    1    0  7  0  1  3  5
## 3    2    1  5 11  1  0  3
## 4    2    0 13  8  6  1  0
## 5    3    1  5 11  2  1  5
## 6    3    0  8  7  6  3  5
## 7    4    1 16 19  1  2  3
## 8    4    0 17  1  0  1  3
```

```
y.data = cbind(y2, y3, y4, y5, y1)
y.data
dim(y.data)
x.data = model.matrix(~size + factor(lake), data = ali)
x.data
dim(x.data)
```

```
##      y2 y3 y4 y5 y1
## [1,]  4  2  2  8 23
## [2,]  0  1  3  5  7
## [3,] 11  1  0  3  5
## [4,]  8  6  1  0 13
## [5,] 11  2  1  5  5
## [6,]  7  6  3  5  8
## [7,] 19  1  2  3 16
## [8,]  1  0  1  3 17
## [1] 8 5
##   (Intercept) size factor(lake)2 factor(lake)3 factor(lake)4
## 1           1    1             0             0             0
## 2           1    0             0             0             0
## 3           1    1             1             0             0
## 4           1    0             1             0             0
## 5           1    1             0             1             0
## 6           1    0             0             1             0
## 7           1    1             0             0             1
## 8           1    0             0             0             1
## attr(,"assign")
## [1] 0 1 2 2 2
## attr(,"contrasts")
## attr(,"contrasts")$`factor(lake)`
## [1] "contr.treatment"
##
## [1] 8 5
```

```
# We use library VGAM:
library(VGAM)

# We fit a multinomial logit model with fish (y1) as the reference category:
fit.main = vglm(cbind(y2, y3, y4, y5, y1) ~ size + factor(lake), family = multinomial,
    data = ali)
summary(fit.main)
pchisq(deviance(fit.main), df.residual(fit.main), lower.tail = FALSE)
```

```
##
## Call:
## vglm(formula = cbind(y2, y3, y4, y5, y1) ~ size + factor(lake),
##     family = multinomial, data = ali)
##
##
## Pearson residuals:
##   log(mu[,1]/mu[,5]) log(mu[,2]/mu[,5]) log(mu[,3]/mu[,5])
## 1             0.0953           0.028205           -0.54130
## 2            -0.5082           0.003228            0.66646
## 3            -0.3693          -0.461102           -0.42005
## 4             0.4125           0.249983            0.19772
## 5            -0.5526          -0.191149            0.07215
## 6             0.6500           0.110694           -0.02784
## 7             0.6757           0.827737            0.79863
## 8            -1.3051          -0.802694           -0.69525
##   log(mu[,4]/mu[,5])
## 1            -0.7268
## 2             1.2589
## 3             1.8347
## 4            -1.3779
## 5             0.2790
## 6            -0.2828
## 7            -0.3081
## 8             0.4629
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept):1    -3.2074     0.6387  -5.021 5.13e-07 ***
## (Intercept):2    -2.0718     0.7067  -2.931 0.003373 **
## (Intercept):3    -1.3980     0.6085  -2.297 0.021601 *
## (Intercept):4    -1.0781     0.4709  -2.289 0.022061 *
## size:1            1.4582     0.3959   3.683 0.000231 ***
## size:2           -0.3513     0.5800  -0.606 0.544786
## size:3           -0.6307     0.6425  -0.982 0.326296
## size:4            0.3316     0.4482   0.740 0.459506
## factor(lake)2:1   2.5956     0.6597   3.934 8.34e-05 ***
## factor(lake)2:2   1.2161     0.7860   1.547 0.121824
## factor(lake)2:3  -1.3483     1.1635  -1.159 0.246529
## factor(lake)2:4  -0.8205     0.7296  -1.125 0.260713
## factor(lake)3:1   2.7803     0.6712   4.142 3.44e-05 ***
## factor(lake)3:2   1.6925     0.7804   2.169 0.030113 *
## factor(lake)3:3   0.3926     0.7818   0.502 0.615487
## factor(lake)3:4   0.6902     0.5597   1.233 0.217511
## factor(lake)4:1   1.6584     0.6129   2.706 0.006813 **
## factor(lake)4:2  -1.2428     1.1854  -1.048 0.294466
## factor(lake)4:3  -0.6951     0.7813  -0.890 0.373608
## factor(lake)4:4  -0.8262     0.5575  -1.482 0.138378
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of linear predictors:  4
##
## Names of linear predictors:
## log(mu[,1]/mu[,5]), log(mu[,2]/mu[,5]), log(mu[,3]/mu[,5]), log(mu[,4]/mu[,5])
##
## Residual deviance: 17.0798 on 12 degrees of freedom
##
## Log-likelihood: -47.5138 on 12 degrees of freedom
##
## Number of iterations: 5
##
## Warning: Hauck-Donner effect detected in the following estimate(s):
## '(Intercept):1'
##
## Reference group is level  5  of the response
## [1] 0.1466189
```

Q:

- Why is the number of degrees of freedom for the residual deviance 12? Hint: there are 8 covariate patterns, and we have 5 reponse categories.
- How can you interpret the coefficient for inverterbrate (y2) and size? Hint: we have y2,y3,y4,y5 as 1:4.

```
exp(coefficients(fit.main))
```

```
##   (Intercept):1   (Intercept):2   (Intercept):3   (Intercept):4
##       0.0404626       0.1259644       0.2471007       0.3402497
##          size:1          size:2          size:3          size:4
##       4.2982356       0.7037987       0.5322405       1.3931262
## factor(lake)2:1 factor(lake)2:2 factor(lake)2:3 factor(lake)2:4
##      13.4043318       3.3739877       0.2596748       0.4401925
## factor(lake)3:1 factor(lake)3:2 factor(lake)3:3 factor(lake)3:4
##      16.1245576       5.4329197       1.4808988       1.9940595
## factor(lake)4:1 factor(lake)4:2 factor(lake)4:3 factor(lake)4:4
##       5.2506853       0.2885818       0.4990158       0.4377111
```

Testing out other models, and comparing with LRT-test - by using deviances for different models.

```
# Fit model with only lake:
fit.lake = vglm(cbind(y2, y3, y4, y5, y1) ~ factor(lake), family = multinomial,
    data = ali)
summary(fit.lake)
# Test effect of size (no anova command is available)
G2 = deviance(fit.lake) - deviance(fit.main)
G2
df.diff = df.residual(fit.lake) - df.residual(fit.main)
df.diff
1 - pchisq(G2, df.diff)
# Size has a significant effect
```

```
##
## Call:
## vglm(formula = cbind(y2, y3, y4, y5, y1) ~ factor(lake), family = multinomial,
##     data = ali)
##
##
## Pearson residuals:
##   log(mu[,1]/mu[,5]) log(mu[,2]/mu[,5]) log(mu[,3]/mu[,5])
## 1             0.6180            -0.1545            -0.8981
## 2            -0.9649             0.2413             1.4021
## 3             1.4225            -0.8590            -0.4978
## 4            -1.2022             0.7260             0.4208
## 5             1.1855            -0.6985            -0.4768
## 6            -1.0784             0.6355             0.4338
## 7             2.0262             0.5839             0.2555
## 8            -2.7660            -0.7971            -0.3487
##   log(mu[,4]/mu[,5])
## 1            -0.4975
## 2             0.7767
## 3             1.7751
## 4            -1.5003
## 5             0.3849
## 6            -0.3501
## 7            -0.1850
## 8             0.2525
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept):1    -2.0149     0.5323  -3.785 0.000153 ***
## (Intercept):2    -2.3026     0.6055  -3.803 0.000143 ***
## (Intercept):3    -1.7918     0.4830  -3.709 0.000208 ***
## (Intercept):4    -0.8362     0.3320  -2.518 0.011787 *
## factor(lake)2:1   2.0690     0.6257   3.307 0.000944 ***
## factor(lake)2:2   1.3581     0.7517   1.807 0.070810 .
## factor(lake)2:3  -1.0986     1.1353  -0.968 0.333199
## factor(lake)2:4  -0.9555     0.7065  -1.352 0.176228
## factor(lake)3:1   2.3403     0.6448   3.629 0.000284 ***
## factor(lake)3:2   1.8171     0.7540   2.410 0.015963 *
## factor(lake)3:3   0.6131     0.7485   0.819 0.412725
## factor(lake)3:4   0.5739     0.5359   1.071 0.284216
## factor(lake)4:1   1.5141     0.6030   2.511 0.012042 *
## factor(lake)4:2  -1.1939     1.1819  -1.010 0.312427
## factor(lake)4:3  -0.6061     0.7726  -0.785 0.432746
## factor(lake)4:4  -0.8685     0.5543  -1.567 0.117138
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of linear predictors:  4
##
## Names of linear predictors:
## log(mu[,1]/mu[,5]), log(mu[,2]/mu[,5]), log(mu[,3]/mu[,5]), log(mu[,4]/mu[,5])
##
## Residual deviance: 38.1672 on 16 degrees of freedom
##
## Log-likelihood: -58.0575 on 16 degrees of freedom
##
## Number of iterations: 5
##
## No Hauck-Donner effect found in any of the estimates
##
## Reference group is level  5  of the response
## [1] 21.08741
## [1] 4
## [1] 0.0003042796
```

```r
# Fit model with only size:
fit.size = vglm(cbind(y2, y3, y4, y5, y1) ~ size, family = multinomial, data = ali)
summary(fit.size)

# Test effect of lake
G2 = deviance(fit.size) - deviance(fit.main)
G2
df.diff = df.residual(fit.size) - df.residual(fit.main)
df.diff
1 - pchisq(G2, df.diff)
# Lake has a significant effect
```

```
##
## Call:
## vglm(formula = cbind(y2, y3, y4, y5, y1) ~ size, family = multinomial,
##     data = ali)
##
##
## Pearson residuals:
##                        Min      1Q   Median     3Q   Max
## log(mu[,1]/mu[,5]) -3.414 -1.6814  1.18344 1.3708 1.786
## log(mu[,2]/mu[,5]) -2.183 -0.6923 -0.03712 1.0781 1.360
## log(mu[,3]/mu[,5]) -1.029 -0.7234  0.12037 0.3921 1.447
## log(mu[,4]/mu[,5]) -1.925 -0.6523  0.26477 0.9089 1.923
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept):1  -1.0341     0.2911  -3.553 0.000381 ***
## (Intercept):2  -1.2417     0.3149  -3.944 8.03e-05 ***
## (Intercept):3  -1.7272     0.3837  -4.502 6.74e-06 ***
## (Intercept):4  -1.2417     0.3149  -3.944 8.03e-05 ***
## size:1          0.9489     0.3569   2.659 0.007837 **
## size:2         -0.8583     0.5350  -1.604 0.108626
## size:3         -0.5552     0.6063  -0.916 0.359864
## size:4          0.2943     0.4150   0.709 0.478129
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of linear predictors:  4
##
## Names of linear predictors:
## log(mu[,1]/mu[,5]), log(mu[,2]/mu[,5]), log(mu[,3]/mu[,5]), log(mu[,4]/mu[,5])
##
## Residual deviance: 66.2129 on 24 degrees of freedom
##
## Log-likelihood: -72.0803 on 24 degrees of freedom
##
## Number of iterations: 4
##
## No Hauck-Donner effect found in any of the estimates
##
## Reference group is level  5  of the response
## [1] 49.13308
## [1] 12
## [1] 1.982524e-06
```

Q: explain what is presented below, in particular "what is the probability that the main food source is fish given size=0 and lake=1"?

```
library(knitr)
# Fitted values for main effect model 'fit.main':
fitted = data.frame(fitted(fit.main), lake = ali$lake, size = ali$size)
kable(fitted)
```

y2	y3	y4	y5	y1	lake	size
0.0930988	0.0474566	0.0704015	0.2537396	0.5353035	1	1
0.0230717	0.0718246	0.1408963	0.1940096	0.5701978	1	0
0.6018967	0.0772276	0.0088175	0.0538721	0.2581861	2	1
0.2486452	0.1948374	0.0294161	0.0686628	0.4584385	2	0
0.5168385	0.0887672	0.0358947	0.1742005	0.1842990	3	1
0.1929612	0.2023995	0.1082251	0.2006616	0.2957525	3	0
0.4128558	0.0115665	0.0296712	0.0938024	0.4521040	4	1
0.1396778	0.0238987	0.0810674	0.0979136	0.6574425	4	0

Regression with ordinal responses

(we will only consider cumulative models - and not sequential models)

An unobservable latent variable U_i drives the observed category Y_i.

$$Y_i = r \Leftrightarrow \theta_{r-1} \leq U_i \leq \theta_r$$

where these θs are our unobservable thresholds, and the thresholds are monotonely increasing, $-\infty = \theta_0 < \theta_1 < \cdots < \theta_{c+1} = \infty$.

We further assume that the latent variables are dependent on our covariates through

$$U_i = -\mathbf{x}_i^T \boldsymbol{\beta} + \varepsilon_i$$

where we have a new random variable that has cumulative distribution function (cdf) F. No intercept is included due to identifiability issuse (shift in intercept would produce the same effect as negative shift in threshold).

We get rid of the latent variable U_i by considereing

$$\begin{aligned} P(Y_i \leq r) = P(U_i \leq \theta_r) = P(-\mathbf{x}_i^T \boldsymbol{\beta} + \varepsilon_i \leq \theta_r) \\ = P(\varepsilon_i \leq \theta_r + \mathbf{x}_i^T \boldsymbol{\beta}) = F(\theta_r + \mathbf{x}_i^T \boldsymbol{\beta}) \end{aligned}$$

Observe that the final expression does not include the latent variable U_i, but includes the unknown threshold and k regression parameters.

Different choices of F will give different models, and we will only consider F to be the cdf for the logistic distribution. (Another popular choice is the cdf of the standard normal distribution.)

$$P(Y_i \leq r) = \frac{\exp(\theta_r + \mathbf{x}_i^T \boldsymbol{\beta})}{1 + \exp(\theta_r + \mathbf{x}_i^T \boldsymbol{\beta})}$$

which also can be written as

$$\ln(\frac{P(Y_i \leq r)}{P(Y_i > r)}) = \theta_r + \mathbf{x}_i^T \boldsymbol{\beta}$$

Our model is a proportional odds model, in the sense that the cumulative odds are proportional across categories

$$\frac{\frac{P(Y_i \leq r | \mathbf{x}_i)}{P(Y_i > r | \mathbf{x}_i)}}{\frac{P(Y_i \leq r | \mathbf{x}_i^*)}{P(Y_i > r | \mathbf{x}_i^*)}} = \exp((\mathbf{x}_i - \mathbf{x}_i^*)^T \boldsymbol{\beta})$$

Observe that this is independent of r.

If the only change from $\mathbf{x}$ to $\mathbf{x}^*$ is that one covariate (say covariate k) change with one unit - then $\exp((\mathbf{x}_i - \mathbf{x}_i^*)^T \boldsymbol{\beta}) = \exp(\beta_k)$, and the proportional odds model makes us able to explain what β_k means.

Response function

What is the response function here?

$$\pi_{i1} = F(\eta_{i1})$$
$$\pi_{ir} = F(\eta_{ir}) - F(\eta_{i,r-1})$$

where $\eta_{ir} = \theta_r + \mathbf{x}_i^T \boldsymbol{\beta}$, and F is the logistic cdf.

Plotting the cumulative probabilites

We have five categories, where the fifth is the reference category.

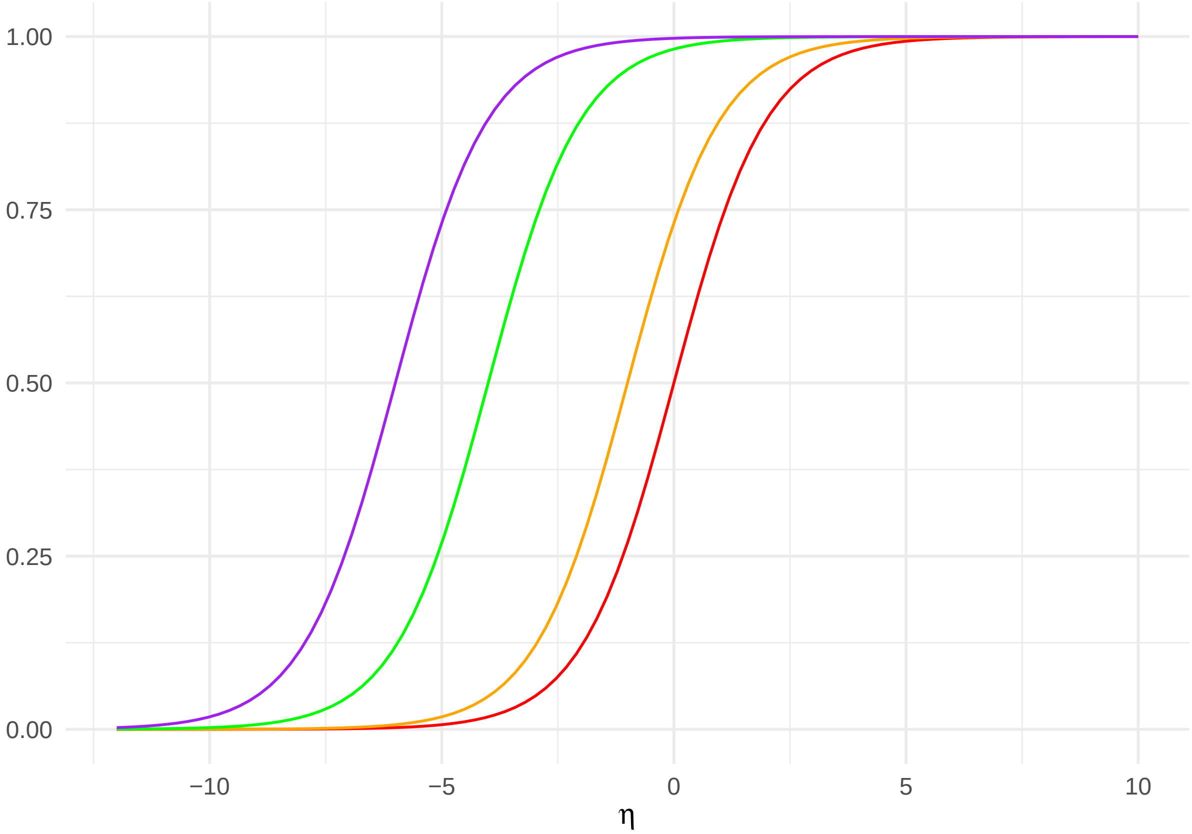
True parameters

- $\theta_1 = 0$, $\theta_2 = 1$, $\theta_3 = 4$ and $\theta_4 = 6$ and
- one covariate with parameter $\beta = 1$.

The graph shows the cumulative probability $P(Y \leq r)$ for $r = 1$ (red), $r = 2$ (organge), $r = 3$ (green), $r = 4$ (purple).

Observe the parallell lines.

What would $P(Y \leq 5)$ be? Why is this missing from the plot?



Mental health data example

Example and data are taken from Agresti (2015, pages 219-223).

Research question: understand mental health issues.

The data comes from a random sample of size 40 of adult residents of Alachua County, Florida, USA.

- Mental impairment Y: 1=well, 2=mild symptom formation, 3=moderate symptom formation, 4=impaired.
- Life event index (x_1): compsite measure of the number and severity of important life events within the last three years (birth, new job, divorce, death in the family, . . .)
- SES (x_2): socioeconomic index, 1=high, 0=low.

These data are ungrouped (but could be grouped). In the original study several other explanatory variables were studied.

```r
# Read mental health data from the web:
library(knitr)
data = "http://www.stat.ufl.edu/~aa/glm/data/Mental.dat"
mental = read.table(data, header = T)
colnames(mental)
apply(mental, 2, table)
# kable(mental)
```

```
## [1] "impair" "ses"    "life"
## $impair
##
##  1  2  3  4
## 12 12  7  9
##
## $ses
##
##  0  1
## 18 22
##
## $life
##
## 0 1 2 3 4 5 6 7 8 9
## 2 5 4 8 5 4 2 2 4 4
```

```r
library(VGAM)
# We fit a cumulative logit model with main effects of 'ses' and 'life':
fit.imp = vglm(impair ~ life + ses, family = cumulative(parallel = T), data = mental)
# parallell=T gives proportional odds structure - only intercepts differ
summary(fit.imp)
```

```
##
## Call:
## vglm(formula = impair ~ life + ses, family = cumulative(parallel = T),
##     data = mental)
##
##
## Pearson residuals:
```

```
##                     Min      1Q  Median     3Q   Max
## logit(P[Y<=1]) -1.568 -0.7048 -0.2102 0.8070 2.713
## logit(P[Y<=2]) -2.328 -0.4666  0.2657 0.6904 1.615
## logit(P[Y<=3]) -3.688  0.1198  0.2039 0.4194 1.892
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept):1  -0.2819     0.6231  -0.452  0.65096
## (Intercept):2   1.2128     0.6511   1.863  0.06251 .
## (Intercept):3   2.2094     0.7171   3.081  0.00206 **
## life           -0.3189     0.1194  -2.670  0.00759 **
## ses             1.1112     0.6143   1.809  0.07045 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of linear predictors:  3
##
## Names of linear predictors:
## logit(P[Y<=1]), logit(P[Y<=2]), logit(P[Y<=3])
##
## Residual deviance: 99.0979 on 115 degrees of freedom
##
## Log-likelihood: -49.5489 on 115 degrees of freedom
##
## Number of iterations: 5
##
## No Hauck-Donner effect found in any of the estimates
##
## Exponentiated coefficients:
##      life       ses
## 0.7269742 3.0380707
```

The ML fit for this model can be written as

$$\text{logit}(\hat{P}(y_i \leq r)) = \hat{\theta}_r + 0.319x_{i1} + 1.111x_{i2}$$

Q: give an interpretation of this model!

Remember:

- Life event index (x_1): compsite measure of the number and severity of important life events within the last three years (birth, new job, divorce, death in the family, . . .)
- SES (x_2): socioeconomic index, 1=high, 0=low.

Q: How can you interpret the last line below? Why is it exp(CI(beta)) and not CI(exp(beta))?

```r
exp(confint(fit.imp))
```

```
##                   2.5 %     97.5 %
## (Intercept):1 0.2224503  2.5581328
## (Intercept):2 0.9385968 12.0489557
## (Intercept):3 2.2342109 37.1467162
## life          0.5752574  0.9187045
## ses           0.9114465 10.1266209
```

Q: How are these predictions calculated? What is the interpretation?

```
fitted = data.frame(fitted(fit.imp), ses = mental$ses, life = mental$life)
fitted[c(6, 18, 10), ]  #0,7 not fitted

xs = cbind(c(2, 7, 2, 7), c(0, 0, 1, 1))
coeff = coefficients(fit.imp)
linpreds = cbind(coeff[1] + xs %*% coeff[4:5], coeff[2] + xs %*% coeff[4:5],
    coeff[3] + xs %*% coeff[4:5])
cprobs = exp(linpreds)/(1 + exp(linpreds))
cprobs
pprobs = cbind(cprobs[, 1], cprobs[, 2] - cprobs[, 1], cprobs[, 3] - cprobs[,
    2], 1 - cprobs[, 3])
pprobs
```

```
##           X1        X2         X3         X4 ses life
## 6  0.2850362 0.3548973 0.18808559 0.17198084   0    2
## 18 0.5477558 0.2959810 0.09227184 0.06399141   1    2
## 10 0.1973858 0.3255923 0.22513134 0.25189056   1    7
##            [,1]      [,2]      [,3]
## [1,] 0.28503623 0.6399336 0.8280192
## [2,] 0.07488691 0.2651744 0.4943332
## [3,] 0.54775576 0.8437368 0.9360086
## [4,] 0.19738576 0.5229781 0.7481094
##            [,1]      [,2]       [,3]       [,4]
## [1,] 0.28503623 0.3548973 0.18808559 0.17198084
## [2,] 0.07488691 0.1902875 0.22915882 0.50566678
## [3,] 0.54775576 0.2959810 0.09227184 0.06399141
## [4,] 0.19738576 0.3255923 0.22513134 0.25189056
```

Q: What do you see here, and what is the formula for this matrix?

```
vcov(fit.imp)
```

```
##               (Intercept):1 (Intercept):2 (Intercept):3        life
## (Intercept):1    0.38819709    0.32992954    0.32615019 -0.04231112
## (Intercept):2    0.32992954    0.42395851    0.40844529 -0.05427393
## (Intercept):3    0.32615019    0.40844529    0.51423495 -0.06185757
## life            -0.04231112   -0.05427393   -0.06185757  0.01426291
## ses             -0.15615761   -0.11440293   -0.09055667 -0.01855824
##                       ses
## (Intercept):1 -0.15615761
## (Intercept):2 -0.11440293
## (Intercept):3 -0.09055667
## life          -0.01855824
## ses            0.37732634
```

```
# We consider a model with interaction between 'ses' and 'life':
fit.int = vglm(impair ~ life + ses + life:ses, family = cumulative(parallel = T),
    data = mental)
```

```r
summary(fit.int)
```

```
##
## Call:
## vglm(formula = impair ~ life + ses + life:ses, family = cumulative(parallel = T),
##     data = mental)
##
##
## Pearson residuals:
##                    Min      1Q  Median     3Q   Max
## logit(P[Y<=1]) -1.393 -0.7139 -0.2172 0.9084 2.262
## logit(P[Y<=2]) -2.758 -0.4862  0.2781 0.7218 1.797
## logit(P[Y<=3]) -3.364  0.1347  0.2062 0.3795 2.344
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept):1  0.09807    0.81102   0.121  0.90375
## (Intercept):2  1.59248    0.83717   1.902  0.05714 .
## (Intercept):3  2.60660    0.90966   2.865  0.00416 **
## life          -0.42045    0.19031  -2.209  0.02715 *
## ses            0.37090    1.13022   0.328  0.74279
## life:ses       0.18131    0.23611   0.768  0.44255
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of linear predictors:  3
##
## Names of linear predictors:
## logit(P[Y<=1]), logit(P[Y<=2]), logit(P[Y<=3])
##
## Residual deviance: 98.5044 on 114 degrees of freedom
##
## Log-likelihood: -49.2522 on 114 degrees of freedom
##
## Number of iterations: 5
##
## No Hauck-Donner effect found in any of the estimates
##
## Exponentiated coefficients:
##      life       ses  life:ses
## 0.6567529 1.4490350 1.1987822
```

```r
# And test if there is a significant effect of interaction:
G2 = deviance(fit.imp) - deviance(fit.int)
df.diff = df.residual(fit.imp) - df.residual(fit.int)
1 - pchisq(G2, df.diff)
# The effect of interaction is not significant
```

```
## [1] 0.4410848
```

```r
# We consider a model where the effect of the covariates may differ between
# the cumulative logits - so not parallell lines for the cdfs
fit.nopar = vglm(impair ~ life + ses, family = cumulative, data = mental)
summary(fit.nopar)
```

```
# The change in the deviance compared to the model 'fit.imp' is
# 99.0979-96.7486=2.3493 with df.diff=115-111=4, which is not significant

# So model 'fit.imp'seems fine.
```

```
##
## Call:
## vglm(formula = impair ~ life + ses, family = cumulative, data = mental)
##
##
## Pearson residuals:
##                    Min      1Q  Median     3Q   Max
## logit(P[Y<=1]) -1.509 -0.6707 -0.2126 0.8310 2.790
## logit(P[Y<=2]) -2.583 -0.6027  0.2487 0.6277 1.793
## logit(P[Y<=3]) -3.841  0.1128  0.1849 0.4095 2.063
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept):1  -0.1930     0.7387  -0.261   0.7938
## (Intercept):2   0.8278     0.7036   1.176   0.2394
## (Intercept):3   2.8049     0.9615      NA       NA
## life:1         -0.3182     0.1597  -1.993   0.0463 *
## life:2         -0.2739     0.1372  -1.997   0.0458 *
## life:3         -0.3964     0.1592  -2.490   0.0128 *
## ses:1           0.9732     0.7720   1.261   0.2074
## ses:2           1.4962     0.7460   2.006   0.0449 *
## ses:3           0.7518     0.8358   0.899   0.3684
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of linear predictors:  3
##
## Names of linear predictors:
## logit(P[Y<=1]), logit(P[Y<=2]), logit(P[Y<=3])
##
## Residual deviance: 96.7486 on 111 degrees of freedom
##
## Log-likelihood: -48.3743 on 111 degrees of freedom
##
## Number of iterations: 14
##
## Warning: Hauck-Donner effect detected in the following estimate(s):
## '(Intercept):3'
##
## Exponentiated coefficients:
##    life:1    life:2    life:3     ses:1     ses:2     ses:3
## 0.7274592 0.7604194 0.6727572 2.6465169 4.4645713 2.1207587
```

Why not use MLR instead of ordinal regression?

Based on Agresti (2015, p 214-216)

To use MLR the ordinal categories need to be replaced with numerical values, and we then need to assume a normal error structure. The following are questions to be answered and possible limitation to be assumed for using MLR instead of ordinal regression:

- how to translate ordered categories into numerical scores?
- is it better with an ordinal variable with some range than a single numerical number?
- MLR will not give probabilities for each response category
- variability in the response may be dependent on the category, for MLR we assume homoscedasticity

Likelihood inference

We use the notation that $\boldsymbol{\beta}$ is a long vector with all regression parameters. The content of this vector is slightly different for our two models, with intercept and k covariate effects for each response category for the nominal model - and with c thresholds but the same k-dimensional $\boldsymbol{\beta}$ vector for all categories.

Full matrix versions (over all i) can be found in our textbook, page 345-346.

Loglikelihood

We have seen that the loglikelihood is:

$$l(\boldsymbol{\beta}) \propto \sum_{i=1}^{n} \sum_{s=1}^{c+1} y_{is} \ln(\pi_{is})$$

where we remember that $y_{i,c+1} = n_i - y_{i1} - \cdots - y_{ic}$, and $1 - \pi_{i1} - \cdots \pi_{ic}$.

Design matrix and coefficient vector

The design matrix $\mathbf{X}$ and coefficient vector are different for our nominal logit model and our ordinal cumulative model.

Nominal logit model

$$\mathbf{X}_i = \text{diag}(\mathbf{x}_i^T) = \begin{pmatrix} \mathbf{x}_i^T & 0 & \cdots & 0 \\ 0 & \mathbf{x}_i^T & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{x}_i^T \end{pmatrix}$$

where the 0s are $1 \times p$ vectors. The dimension of the design matrix for covariate pattern i is $c \times c \cdot p$.

The vector of coefficients has dimension $c \cdot p \times 1$.

$$\boldsymbol{\beta} = \begin{pmatrix} \boldsymbol{\beta}_1 \\ \boldsymbol{\beta}_2 \\ \vdots \\ \boldsymbol{\beta}_c \end{pmatrix}$$

Ordinal cumulative model

$$\mathbf{X}_i = \begin{pmatrix} 1 & 0 & \cdots & 0 & \mathbf{x}_i^T \\ 0 & 1 & \cdots & 0 & \mathbf{x}_i^T \\ \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & \cdots & 1 & \mathbf{x}_i^T \end{pmatrix}$$

he dimension of the design matrix for covariate pattern i is $c \times (c+k)$

The vector of coefficients has dimension $(c+k) \times 1$ (where $p = k+1$), and now the thresholds replace the intercept and are put first in the vector, and the effects of the covariates are the same for all categories.

$$\boldsymbol{\beta} = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_c \\ \boldsymbol{\beta} \end{pmatrix}$$

Score function

$$\mathbf{s}(\boldsymbol{\beta}) = \sum_{i=1}^{G} \mathbf{X}_i^T \mathbf{D}_i \boldsymbol{\Sigma}_i^{-1} (\mathbf{y}_i - n_i \boldsymbol{\pi}_i)$$

where

- $\mathbf{D}_i = \frac{\partial \boldsymbol{h}(\boldsymbol{\eta})}{\partial \boldsymbol{\eta}}|_{\boldsymbol{\eta}=\boldsymbol{\eta}_i}$ has dimension $c \times c$
- $\boldsymbol{\Sigma}_i = \text{Cov}(\mathbf{Y}_i)$

Fisher information

The dimension of the matrix is $cp \times cp$ for the nominal case and $(c+k) \times (c+k)$ for the ordinal case studied.

$$F(\boldsymbol{\beta}) = \sum_{i=1}^{G} \mathbf{X}_i^T \mathbf{W}_i \mathbf{X}_i$$

where $\mathbf{W}_i$ is given as $\mathbf{D}_i \boldsymbol{\Sigma}_i^{-1} \mathbf{D}_i^T$.

Finding the ML estimate

As in modules 1-5 we find the ML estimate by the Fisher scoring or Newton Raphson method.

Asymptotic distribution

As in modules 1-5 the ML estimator $\hat{\boldsymbol{\beta}}$ asymptotically follows a multivariate normal distribution with unbiased mean and covariance matrix given by the inverse of the expected Fisher information matrix.

Summing up

Interactive session

Problem 1: Exam 2006, problem 1 (ordinal model)

Table 1 shows the results from a study where two injection plans for the the neuroleptic preparation perphenazine decanoate have been compared (from P. Knudsen, L. B. Hansen, K. Højholdt, N. E. Larsen, Acta Psychiatrica Scandinavica, 1985).

A group of 19 psycotic patients was given injections every second week, while another group of 19 patients was given injections every third week. The patients were studied for six months, and the effect of the treatment was evaluated in the end. Clinical evaluations was done using a six-point scale calles CGI (Clinical Global Impression), where a higher score means a worse state for the patient.

The 12 rows in Table 1 corresponds to 12 different combinations of the three explanatory variables x_1, x_2 and x_3:

$$x_1 = \begin{cases} 0, \text{ if injections are given every second week} \\ 1, \text{ if injections are given every third week} \end{cases} \quad x_2 = \begin{cases} 0, \text{ if patient is female} \\ 1, \text{ if patient is male} \end{cases} \quad x_3 = \text{ CGI at beginning of treatment (i}$$

Table 2: Table 1: Data for neuroleptic treatment

Interval (x1)	Sex (x2)	Initial CGI (x3)	Final CGI 0 (y0)	Final CGI 1 (y1)	Final CGI 2 (y2)
2	F	2	1	0	0
2	F	3	3	1	0
2	F	4	0	1	0
2	M	3	4	4	1
2	M	4	0	2	1
2	M	5	0	0	1
3	F	2	1	0	0
3	F	3	2	1	0
3	F	4	1	2	0
3	M	2	3	1	0
3	M	3	0	5	0
3	M	4	0	3	0

The corresponding responses are counts for each combination of explanatory variables:

$$y_0 = \text{ number with (CGI} = 0\text{) after treatment} y_1 = \text{ number with (CGI} = 1\text{) after treatment} y_2 = \text{ number with (CGI} = 2\text{) a}$$

Note that no patients had final CGI above 2 after the treatment.. We use y_2 as the reference category. Assume that the CGI for a patient with covariate vector $\mathbf{x} = (x_1, x_2, x_3)$ has response value j, $j = 0, 1, 2$, with probabilities

$$\pi_j = \text{Prob(CGI} = j|\mathbf{x}) \text{ for } j = 0, 1, 2.$$

The response $\mathbf{y} = (y_0, y_1, y_2)$ for a row in the table is assumed to come from a multinomial distribution vector $\mathbf{Y} = (Y_0, Y_1, Y_2)$ with probability vector $\pi = (\pi_0, \pi_1, \pi_2)$, and π depends on $\mathbf{x}$. Note that x_3 is numeric, not cathegorical.

a)

- Write down the proportional odds model for these data, and discuss it briefly. Assume there are no interactions between x_1, x_2 and x_3. Remember that y_2 is the reference category.
- Express π_j, $j = 0, 1, 2$ as functions of the θ's and β's in the model and $\mathbf{x}$.

b)

- $\text{Prob}(\text{CGI} \leq j|\mathbf{x})/\text{Prob}(\text{CGI} > j|\mathbf{x})$ for $j = 0, 1$ is called the *cumulative odds ratios* for a patient with covariate vector $\mathbf{x}$. Show that if initial CGI increases with 1 in the model from a), then the cumulative odds ratios will be multiplied by e^{β_3}. Here β_3 is the coefficient belonging to x_3 in the linear predictor of the model.
- Interpret the value e^{β_3}.
- Interpret also the values e^{β_1} and e^{β_2}.

c)

- Describe the saturated model for these data. How many free parameters does it have? (Remark: how many "parameters" can be estimated.)
- How would you calculate the deviance for the model from a)? (Just explain using words, no calculations necessary!)
- How many degrees of freedom does the deviance have here? Give reasons for your answer.

Below you can see the deviance for all proportional odds models that contain the variable x_3 (initial CGI). The formulas work in the same way as for the `lm` and `glm` formulas.

Model	Deviance
cbind(y0, y1, y2) ~ x3	17.68
cbind(y0, y1, y2) ~ x1 + x3	17.66
cbind(y0, y1, y2) ~ x2 + x3	10.64
cbind(y0, y1, y2) ~ x1 * x3	14.43
cbind(y0, y1, y2) ~ x2 * x3	10.63
cbind(y0, y1, y2) ~ x1 + x2 + x3	10.56
cbind(y0, y1, y2) ~ x1 * x2 + x3	10.33
cbind(y0, y1, y2) ~ x1 * x3 + x2	8.52
cbind(y0, y1, y2) ~ x1 + x2 * x3	10.51
cbind(y0, y1, y2) ~ x1 * x2 + x1 * x3	8.33
cbind(y0, y1, y2) ~ x1 * x2 + x2 * x3	10.31
cbind(y0, y1, y2) ~ x1 * x3 + x2 * x3	8.47
cbind(y0, y1, y2) ~ x1 * x2 + x1 * x3 + x2 * x3	8.28
cbind(y0, y1, y2) ~ x1 * x2 * x3	8.28

d)

- New: What do we mean by the formula `cbind(y0, y1, y2) ~ x1 + x2*x3`? OBS: Ask Mette/Ingeborg if you are not sure before moving on!

- Describe the model that corresponds to `x1*x2 + x1*x3`. How many parameters are in this model? How many degrees of freedom for the deviance?
- A statistician has picked the models `x2 + x3`, `x1 + x2 + x3`, `x1*x2 + x3` and `x1*x2 + x1*x3` as candidates for "the best model". Which of these models would you choose based on the deviances? Reason using hypothesis testing (you have to choose one model for the null-hypothesis, which?).

e)

Below you see (a slightly edited) `R`-summary of the `x1 + x2 + x3` model. Assume we still use the model from a).

```
Call:
vglm(formula = x, family = cumulative(parallel = TRUE), data = data2)


Pearson residuals:
                  Min       1Q   Median     3Q   Max
logit(P[Y<=1]) -1.294 -0.33737 -0.08605 0.1788 1.211
logit(P[Y<=2]) -1.442 -0.08222  0.12144 0.2428 1.100

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept):1   8.0355     2.5079   3.204  0.00136 **
(Intercept):2  12.4324     3.1752   3.916 9.02e-05 ***
x1             -0.2199     0.7561  -0.291  0.77114
x2M            -2.1576     0.8875  -2.431  0.01506 *
x3             -2.2725     0.6985  -3.253  0.00114 **
---

Number of linear predictors:  2

Names of linear predictors: logit(P[Y<=1]), logit(P[Y<=2])

Residual deviance: 10.5552 on ?? degrees of freedom

Log-likelihood: -11.6754 on ?? degrees of freedom
```

And this is the estimated covariace matrix for the estimators:

```
##               (Intercept):1 (Intercept):2      x1     x2M      x3
## (Intercept):1        6.2898        7.6696 -0.6459 -1.1575 -1.6602
## (Intercept):2        7.6696       10.0816 -0.7281 -1.4997 -2.0776
## x1                  -0.6459       -0.7281  0.5716  0.0921  0.0986
## x2M                 -1.1575       -1.4997  0.0921  0.7877  0.1982
## x3                  -1.6602       -2.0776  0.0986  0.1982  0.4879
```

- Use the `R`-summary to estimate e^{β_k} for $k = 1, 2, 3$.
- Find an approximate confidence interval for e^{β_1}. Comment on this considering the model choice you made in d).
- Estimate the probability to get final CGI equal to 0 for a female with injections every second week and initial CGI equal to 5.
- Explain how you can find an estimated standard deviation for this estimate (you do not need to do all calculations). Hint: You need Taylor expansions here!

Problem 2: More alligators (nominal model)

We will analyses an extended version of the alligators data, where also the gender of the alligator is included.

In the data below the following column names are given:

- lake: each of the 4 lakes in Florida (1:4)
- gender: gender of alligator (0:) and (1:) – not given in data file, what do you think?
- size: the size of the alligator (0: 2.3 meters or smaller) and (1: larger than 2.3 meters)
- y1: fish
- y2: inverterbrate
- y3: reptile
- y4: bird
- y5: other

a) Investigate different models and select the best. Call this the best model.
b) Assess the model fit of this best model.
c) Interpret effects and perform inference for the best model.

```
library(VGAM)
data2 = "http://www.stat.ufl.edu/~aa/glm/data/Alligators2.dat"
ali2 = read.table(data2, header = T)
ali2
```

```
##    lake gender size y1 y2 y3 y4 y5
## 1     1      1    1  7  1  0  0  5
## 2     1      1    0  4  0  0  1  2
## 3     1      0    1 16  3  2  2  3
## 4     1      0    0  3  0  1  2  3
## 5     2      1    1  2  2  0  0  1
## 6     2      1    0 13  7  6  0  0
## 7     2      0    1  0  1  0  1  0
## 8     2      0    0  3  9  1  0  2
## 9     3      1    1  3  7  1  0  1
## 10    3      1    0  8  6  6  3  5
## 11    3      0    1  2  4  1  1  4
## 12    3      0    0  0  1  0  0  0
## 13    4      1    1 13 10  0  2  2
## 14    4      1    0  9  0  0  1  2
## 15    4      0    1  3  9  1  0  1
## 16    4      0    0  8  1  0  0  1
```

```
colnames(ali2)
```

```
## [1] "lake"   "gender" "size"   "y1"     "y2"     "y3"     "y4"     "y5"
```

```
fit = vglm(cbind(y2, y3, y4, y5, y1) ~ factor(lake) + factor(size) + factor(gender),
    family = multinomial, data = ali2)

# 6 possible models to investigate: only lake, only gender, only size,
# lake+gender, lake+size, size+gender
fit.lake = vglm(cbind(y2, y3, y4, y5, y1) ~ factor(lake), family = multinomial,
    data = ali2)
fit.size = vglm(cbind(y2, y3, y4, y5, y1) ~ factor(size), family = multinomial,
    data = ali2)
fit.gender = vglm(cbind(y2, y3, y4, y5, y1) ~ factor(gender), family = multinomial,
    data = ali2)
fit.lake.size = vglm(cbind(y2, y3, y4, y5, y1) ~ factor(lake) + factor(size),
    family = multinomial, data = ali2)
fit.lake.gender = vglm(cbind(y2, y3, y4, y5, y1) ~ factor(lake) + factor(gender),
    family = multinomial, data = ali2)
fit.gender.size = vglm(cbind(y2, y3, y4, y5, y1) ~ factor(size) + factor(gender),
    family = multinomial, data = ali2)
all = list(fit = fit, fit.lake = fit.lake, fit.size = fit.size, fit.gender = fit.gender,
    fit.lake.size = fit.lake.size, fit.lake.gender = fit.lake.gender, fit.gender.size = fit.gender.size)
lapply(all, AIC)
```

```
## $fit
## [1] 199.6089
##
## $fit.lake
## [1] 201.9464
##
## $fit.size
## [1] 221.7073
##
## $fit.gender
## [1] 227.0375
##
## $fit.lake.size
## [1] 196.089
##
## $fit.lake.gender
## [1] 204.2444
##
## $fit.gender.size
## [1] 228.4214
```

```
# you may also look at deviance tests if you prefer that to AIC

# what is best? toggle to match your choice
best = fit
summary(best)
```

```
##
## Call:
## vglm(formula = cbind(y2, y3, y4, y5, y1) ~ factor(lake) + factor(size) +
##     factor(gender), family = multinomial, data = ali2)
##
##
## Pearson residuals:
##                          Min      1Q   Median     3Q   Max
## log(mu[,1]/mu[,5]) -1.5637 -0.7657 -0.20492 0.4958 1.561
## log(mu[,2]/mu[,5]) -0.7771 -0.6221 -0.34204 0.2740 2.224
## log(mu[,3]/mu[,5]) -1.0918 -0.6256 -0.24678 0.3566 7.269
## log(mu[,4]/mu[,5]) -1.6323 -0.3743 -0.06655 0.8940 1.503
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept):1     -2.78138    0.66023  -4.213 2.52e-05 ***
## (Intercept):2     -1.94388    0.78487  -2.477  0.01326 *
## (Intercept):3     -1.41471    0.69227  -2.044  0.04100 *
## (Intercept):4     -0.78063    0.50933  -1.533  0.12536    
## factor(lake)2:1    3.14631    0.71871   4.378 1.20e-05 ***
## factor(lake)2:2    1.25128    0.84769   1.476  0.13992    
## factor(lake)2:3   -1.10312    1.20848  -0.913  0.36134    
## factor(lake)2:4   -0.79033    0.77202  -1.024  0.30597    
## factor(lake)3:1    3.04532    0.70100   4.344 1.40e-05 ***
## factor(lake)3:2    1.84442    0.83227   2.216  0.02668 *  
## factor(lake)3:3    0.76742    0.84320   0.910  0.36276    
## factor(lake)3:4    0.76354    0.59707   1.279  0.20097    
## factor(lake)4:1    1.87573    0.62803   2.987  0.00282 ** 
## factor(lake)4:2   -1.15467    1.19439  -0.967  0.33367    
## factor(lake)4:3   -0.50671    0.79598  -0.637  0.52439    
## factor(lake)4:4   -0.76568    0.57046  -1.342  0.17953    
## factor(size)1:1    1.24700    0.42426   2.939  0.00329 ** 
## factor(size)1:2   -0.35772    0.65755  -0.544  0.58643    
## factor(size)1:3   -0.28685    0.65065  -0.441  0.65931    
## factor(size)1:4    0.09454    0.46435   0.204  0.83867    
## factor(gender)1:1 -0.78010    0.37880  -2.059  0.03946 *  
## factor(gender)1:2 -0.32559    0.60986  -0.534  0.59343    
## factor(gender)1:3 -0.52430    0.66647  -0.787  0.43147    
## factor(gender)1:4 -0.33231    0.46161  -0.720  0.47159    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Number of linear predictors:  4 
## 
## Names of linear predictors: 
## log(mu[,1]/mu[,5]), log(mu[,2]/mu[,5]), log(mu[,3]/mu[,5]), log(mu[,4]/mu[,5])
## 
## Residual deviance: 55.2285 on 40 degrees of freedom
## 
## Log-likelihood: -75.8045 on 40 degrees of freedom
## 
## Number of iterations: 6 
## 
## No Hauck-Donner effect found in any of the estimates
## 
## Reference group is level  5  of the response
```

```
pchisq(deviance(best), df.residual(best), lower.tail = FALSE)
```

```
## [1] 0.05511724
```

```
confint(best)
```

```
##                        2.5 %      97.5 %
## (Intercept):1     -4.0754134 -1.48734693
## (Intercept):2     -3.4821892 -0.40556893
## (Intercept):3     -2.7715358 -0.05787731
## (Intercept):4     -1.7789047  0.21764692
## factor(lake)2:1    1.7376626  4.55495618
## factor(lake)2:2   -0.4101634  2.91273242
## factor(lake)2:3   -3.4717099  1.26546012
## factor(lake)2:4   -2.3034619  0.72280210
## factor(lake)3:1    1.6713798  4.41925474
## factor(lake)3:2    0.2131968  3.47564154
## factor(lake)3:3   -0.8852275  2.42007378
## factor(lake)3:4   -0.4067037  1.93378524
## factor(lake)4:1    0.6448037  3.10665210
## factor(lake)4:2   -3.4956349  1.18628645
## factor(lake)4:3   -2.0668098  1.05338460
## factor(lake)4:4   -1.8837510  0.35239661
## factor(size)1:1    0.4154726  2.07853673
## factor(size)1:2   -1.6464982  0.93105746
## factor(size)1:3   -1.5620902  0.98839240
## factor(size)1:4   -0.8155718  1.00464755
## factor(gender)1:1 -1.5225425 -0.03766570
## factor(gender)1:2 -1.5208982  0.86971475
## factor(gender)1:3 -1.8305675  0.78196426
## factor(gender)1:4 -1.2370402  0.57242028
```

Exam questions

None found at NTNU or UiO - except the IL-problem.

R packages

```r
install.packages(c("VGAM", "ggplot2", "statmod", "knitr"))
```

Further reading

- A. Agresti (2015): "Foundations of Linear and Generalized Linear Models." Chapter 6. Wiley.