A Tutorial on fitting Cumulative Link Models with the ordinal Package

Rune Haubo B Christensen

September 13, 2011

Abstract

It is shown by example how a cumulative link mixed model is fitted with the `clm` function in package ordinal. Model interpretation and inference is briefly discussed.

Contents

Table 1: Ratings of the bitterness of some white wines. Data are adopted from Randall (1989).

Temperature	Contact	Bottle	Judge 1	2	3	4	5	6	7	8	9
cold	no	1	2	1	2	3	2	3	1	2	1
cold	no	2	3	2	3	2	3	2	1	2	2
cold	yes	3	3	1	3	3	4	3	2	2	3
cold	yes	4	4	3	2	2	3	2	2	3	2
warm	no	5	4	2	5	3	3	2	2	3	3
warm	no	6	4	3	5	2	3	4	3	3	2
warm	yes	7	5	5	4	5	3	5	2	3	4
warm	yes	8	5	4	4	3	3	4	3	4	4

1 Introduction

We will consider the data on the bitterness of wine from Randall (1989) presented in Table 1 and available as the object wine in package ordinal. The data were also analyzed with mixed effects models by Tutz and Hennevogl (1996). The following gives an impression of the wine data object:

```
> data(wine)
> head(wine)

  response rating temp contact bottle judge
1       36      2 cold      no      1     1
2       48      3 cold      no      2     1
3       47      3 cold     yes      3     1
4       67      4 cold     yes      4     1
5       77      4 warm      no      5     1
6       60      4 warm      no      6     1

> str(wine)

'data.frame':       72 obs. of  6 variables:
 $ response: num  36 48 47 67 77 60 83 90 17 22 ...
 $ rating  : Ord.factor w/ 5 levels "1"<"2"<"3"<"4"<..: 2 3 3 4 4 4 5 5 1 2 ...
 $ temp    : Factor w/ 2 levels "cold","warm": 1 1 1 1 2 2 2 2 1 1 ...
 $ contact : Factor w/ 2 levels "no","yes": 1 1 2 2 1 1 2 2 1 1 ...
 $ bottle  : Factor w/ 8 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 1 2 ...
 $ judge   : Factor w/ 9 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 2 2 ...
```

The data represent a factorial experiment on factors determining the bitterness of wine with 1 = "least bitter" and 5 = "most bitter". Two treatment factors (temperature and contact) each have two levels. Temperature and contact between juice and skins can be controlled when crushing grapes during wine production. Nine judges each assessed wine from two bottles from each of the four treatment conditions, hence there are 72 observations in all. For more information see the manual entry for the wine data: help(wine).

The intention with this tutorial is not to explain cumulative link models in details, rather the main aim is to briefly cover the main functions and methods in the ordinal package to analyze ordinal data. A more thorough introduction to cumulative link models and the ordinal package is given in Christensen (2011); a book length treatment of ordinal data

2

analysis is given by Agresti (2010) although not related to the ordinal package.

2 Fitting Cumulative Link Models

We will fit the following cumulative link model to the wine data:

$$\text{logit}(P(Y_i \leq j)) = \theta_j - \beta_1(\texttt{temp}_i) - \beta_2(\texttt{contact}_i)$$
$$i = 1, \ldots, n, \quad j = 1, \ldots, J-1 \tag{1}$$

This is a model for the cumulative probability of the ith rating falling in the jth category or below, where i index all observations ($n = 72$) and $j = 1, \ldots, J$ index the response categories ($J = 5$). The $theta_j$ parameter is the intercept for the jth cumulative logit $\text{logit}(P(Y_i \leq j))$; they are known as threshold parameters, intercepts or cut-points.

This model is also known as the *proportional odds model*, a *cumulative logit model*, and an *ordered logit model*.

We fit this cumulative link model by maximum likelihood with the `clm` function in package ordinal. Here we save the fitted `clm` model in the object `fm1` (short for `fitted model 1`) and `print` the model by simply typing its name:

```
> fm1 <- clm(rating ~ temp + contact, data = wine)
> fm1

formula: rating ~ temp + contact
data:    wine

 link  threshold nobs logLik AIC     niter max.grad
 logit flexible  72   -86.49 184.98 6(0)  4.01e-12

Coefficients:
  tempwarm contactyes
     2.503      1.528

Threshold coefficients:
   1|2    2|3    3|4    4|5
-1.344  1.251  3.467  5.006
```

Additional information is provided with the `summary` method:

```
> summary(fm1)

formula: rating ~ temp + contact
data:    wine

 link  threshold nobs logLik AIC     niter max.grad cond.H
 logit flexible  72   -86.49 184.98 6(0)  4.01e-12 2.7e+01

Coefficients:
          Estimate Std. Error z value Pr(>|z|)
tempwarm    2.5031     0.5287   4.735 2.19e-06 ***
contactyes  1.5278     0.4766   3.205  0.00135 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Threshold coefficients:
    Estimate Std. Error z value
1|2  -1.3444    0.5171  -2.600
2|3   1.2508    0.4379   2.857
3|4   3.4669    0.5978   5.800
4|5   5.0064    0.7309   6.850
```

The primary result is the coefficient table with parameter estimates, standard errors and Wald (or normal) based p-values for tests of the parameters being zero. The maximum likelihood estimates of the parameters are:

$$\hat{\beta}_1 = 2.50, \quad \hat{\beta}_2 = 1.53, \quad \{\hat{\theta}_j\} = \{-1.34, \ 1.25, \ 3.47, \ 5.01\}. \tag{2}$$

The number of Newton-Raphson iterations is given below `niter` with the number of step-halvings in parenthesis. `max.grad` is the maximum absolute gradient of the log-likelihood function with respect to the parameters. A small absolute gradient is a necessary condition for convergence of the model. The iterative procedure will declare convergence whenever the maximum absolute gradient is below

```
> clm.control()$gradTol
```

```
[1] 1e-06
```

which may be altered — see `help(clm.control)`.

The condition number of the Hessian (`cond.H`) measures the empirical identifiability of the model. High numbers, say larger than 10^4 or 10^6 indicate that the model is ill defined. This could indicate that the model can be simplified, that possibly some parameters are not identifiable, and that optimization of the model can be difficult. In this case the condition number of the Hessian does not indicate a problem with the model.

The coefficients for `temp` and `contact` are positive indicating that higher temperature and more contact increase the bitterness of wine, i.e., rating in higher categories is more likely. The odds ratio of the event $Y \geq j$ is $\exp(\beta_{\text{treatment}})$, thus the odds ratio of bitterness being rated in category j or above at warm relative to cold temperatures is

```
> exp(coef(fm1)[5])
```

```
tempwarm
12.22034
```

The p-values for the location coefficients provided by the `summary` method are based on the so-called Wald statistic. More accurate tests are provided by likelihood ratio tests. These can be obtained with the `anova` method, for example, the likelihood ratio test of `contact` is

```
> fm2 <- clm(rating ~ temp, data = wine)
> anova(fm2, fm1)
```

```
Likelihood ratio tests of cumulative link models:

    formula:                link: threshold:
fm2 rating ~ temp          logit flexible
fm1 rating ~ temp + contact logit flexible

    no.par    AIC  logLik LR.stat df Pr(>Chisq)
fm2      5 194.03 -92.013
```

```
fm1     6 184.98 -86.492  11.043  1  0.0008902 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

which in this case produces a slightly lower *p*-value. Equivalently we can use `drop1` to obtain likelihood ratio tests of the explanatory variables while *controlling* for the remaining variables:

```
> drop1(fm1, test = "Chi")

Single term deletions

Model:
rating ~ temp + contact
        Df    AIC    LRT   Pr(Chi)
<none>     184.98
temp     1 209.91 26.928 2.112e-07 ***
contact  1 194.03 11.043 0.0008902 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Likelihood ratio tests of the explanatory variables while *ignoring* the remaining variables are provided by the `add1` method:

```
> fm0 <- clm(rating ~ 1, data = wine)
> add1(fm0, scope = ~temp + contact, test = "Chi")

Single term additions

Model:
rating ~ 1
        Df    AIC    LRT   Pr(Chi)
<none>     215.44
temp     1 194.03 23.4113 1.308e-06 ***
contact  1 209.91  7.5263   0.00608 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In this case these latter tests are not as strong as the tests controlling for the other variable.

Confidence intervals are provided by the `confint` method:

```
> confint(fm1)

              2.5 %    97.5 %
tempwarm   1.5097627 3.595225
contactyes 0.6157925 2.492404
```

These are based on the profile likelihood function and generally fairly accurate. Less accurate, but simple and symmetric confidence intervals based on the standard errors of the parameters (so-called Wald confidence intervals) can be obtained with

```
> confint(fm1, type = "Wald")

              2.5 %     97.5 %
1|2       -2.3578848 -0.330882
2|3        0.3925794  2.109038
3|4        2.2952980  4.638476
4|5        3.5738541  6.438954
tempwarm   1.4669081  3.539296
```

```
contactyes  0.5936345  2.461961
```

In addition to the logit link, which is the default, the probit, log-log, complementary log-log and cauchit links are also available. For instance, a proportional hazards model for grouped survival times is fitted using the complementary log-log link:

```
> fm.cll <- clm(rating ~ contact + temp, data = wine, link = "cloglog")
```

The cumulative link model in (1) assume that the thresholds, $\{\theta_j\}$ are constant for all values of the remaining explanatory variables, here `temp` and `contact`. This is generally referred to as the *proportional odds assumption* or *equal slopes assumption*. We can relax that assumption in two general ways: with nominal effects and scale effects which we will now discuss in turn.

3   Nominal Effects

The CLM in (1) specifies a structure in which the regression parameters, $\boldsymbol{\beta}$ are not allowed to vary with j. Nominal effects relax this assumption by allowing one or more regression parameters to vary with j. In the following model we allow the regression parameter for `contact` to vary with j:

$$\text{logit}(P(Y_i \leq j)) = \theta_j - \beta_1(\texttt{temp}_i) - \beta_{2j}(\texttt{contact}_i) \atop i = 1, \ldots, n, \quad j = 1, \ldots, J - 1 \tag{3}$$

This means that there is one estimate of β_2 for each j. This model is specified as follows with `clm`:

```
> fm.nom <- clm(rating ~ temp, nominal = ~contact, data = wine)
> summary(fm.nom)

formula: rating ~ temp
nominal: ~contact
data:    wine

 link  threshold nobs logLik AIC    niter max.grad cond.H
 logit flexible  72   -86.21 190.42 6(0)  1.64e-10 4.8e+01


Coefficients:
         Estimate Std. Error z value Pr(>|z|)
tempwarm    2.519      0.535   4.708  2.5e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Threshold coefficients:
               Estimate Std. Error z value
1|2.(Intercept)  -1.3230     0.5623  -2.353
2|3.(Intercept)   1.2464     0.4748   2.625
3|4.(Intercept)   3.5500     0.6560   5.411
4|5.(Intercept)   4.6602     0.8604   5.416
1|2.contactyes   -1.6151     1.1618  -1.390
2|3.contactyes   -1.5116     0.5906  -2.559
3|4.contactyes   -1.6748     0.6488  -2.581
4|5.contactyes   -1.0506     0.8965  -1.172
```

As can be seen from the output of `summary` there is no regression coefficient estimated for `contact`, but there are two sets of threshold parameters estimated.

The first five threshold parameters have `.(Intercept)` appended their names indicating that these are the estimates of θ_j. The following five threshold parameters have `.contactyes` appended their name indicating that these parameters are differences between the threshold parameters at the two levels of contact. This interpretation corresponds to the default treatment contrasts; if other types of contrasts are specified, the interpretation is a little different. As can be seen from the output, the effect of `contact` is almost constant across thresholds and around 1.5 corresponding to the estimate from `fm1` on page 4, so probably there is not much evidence that the effect of `contact` varies with j

We can perform a likelihood ratio test of the equal slopes or proportional odds assumption for `contact` by comparing the likelihoods of models (1) and (3) as follows:

```
> anova(fm1, fm.nom)

Likelihood ratio tests of cumulative link models:

       formula:                   nominal: link: threshold:
fm1    rating ~ temp + contact ~1        logit flexible
fm.nom rating ~ temp           ~contact logit flexible

       no.par   AIC  logLik LR.stat df Pr(>Chisq)
fm1         6 184.98 -86.492
fm.nom      9 190.42 -86.209  0.5667  3     0.904
```

There is only little difference in the log-likelihoods of the two models, so the test is insignificant. There is therefore no evidence that the proportional odds assumption is violated for `contact`.

It is not possible to estimate both β_2 and β_{2j} in the same model. Consequently variables that appear in `nominal` cannot enter elsewhere as well. For instance not all parameters are identifiable in the following model:

```
> fm.nom2 <- clm(rating ~ temp + contact, nominal = ~contact, data = wine)
```

We are made aware of this when summarizing or printing the model:

```
> summary(fm.nom2)

formula: rating ~ temp + contact
nominal: ~contact
data:    wine

 link  threshold nobs logLik AIC    niter max.grad cond.H
 logit flexible  72   -86.21 190.42 6(0)  1.64e-10 4.8e+01

Coefficients: (1 not defined because of singularities)
         Estimate Std. Error z value Pr(>|z|)
tempwarm    2.519      0.535   4.708 2.5e-06 ***
contactyes     NA         NA      NA      NA
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Threshold coefficients:
               Estimate Std. Error z value
1|2.(Intercept)  -1.3230     0.5623  -2.353
```

```
2|3.(Intercept)   1.2464     0.4748   2.625
3|4.(Intercept)   3.5500     0.6560   5.411
4|5.(Intercept)   4.6602     0.8604   5.416
1|2.contactyes   -1.6151     1.1618  -1.390
2|3.contactyes   -1.5116     0.5906  -2.559
3|4.contactyes   -1.6748     0.6488  -2.581
4|5.contactyes   -1.0506     0.8965  -1.172
```

4 Scale Effects

Scale effects are usually motivated from the latent variable interpretation of a CLM. Assume the following model for a latent variable:

$$S_i = \alpha^* + \boldsymbol{x}_i^T \boldsymbol{\beta}^* + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^{*^2}) \tag{4}$$

If the ordinal variable Y_i is observed such that $Y_i = j$ is recorded if $\theta_{j-1}^* < S_i \le \theta_j^*$, where

$$-\infty \equiv \theta_0^* < \theta_1^* < \ldots < \theta_{J-1}^* < \theta_J^* \equiv \infty \tag{5}$$

then we have the cumulative link model for Y_i:

$$P(Y_i \le j) = \Phi(\theta_j - \boldsymbol{x}_i^T \boldsymbol{\beta})$$

where we have used $\theta_j = (\theta_j^* + \alpha^*)/\sigma^*$ and $\boldsymbol{\beta} = \boldsymbol{\beta}^*/\sigma^*$ (parameters with a "*" exist on the latent scale, while those without are identifiable), and Φ is the inverse probit link and denotes the standard normal CDF. Other assumptions on the distribution of the latent variable, S_i lead to other link functions, e.g., a logistic distribution for the latent variable corresponds to a logit link.

If the scale (or dispersion) of the latent distribution is described by a log-linear model such that $\log(\sigma_i) = \boldsymbol{z}_i^T \boldsymbol{\zeta}$ (equivalently $\sigma_i = \exp(\boldsymbol{z}_i^T \boldsymbol{\zeta})$; also note that $\boldsymbol{z}_i^T \boldsymbol{\zeta}$ is a linear model just like $\boldsymbol{x}_i^T \boldsymbol{\beta}$), then the resulting CLM reads (for more details, see e.g., Christensen (2011) or Agresti (2010)):

$$P(Y_i \le j) = \Phi\left(\frac{\theta_j - \boldsymbol{x}_i^T \boldsymbol{\beta}}{\sigma_i}\right) \tag{6}$$

The conventional link functions in cumulative link models correspond to distributions for the latent variable that are members of the location-scale family of distributions (cf. http://en.wikipedia.org/wiki/Location-scale_family). They have the common form $F(\mu, \sigma)$, where μ denotes the location of the distribution and σ denotes the scale of the distribution. For instance in the normal distribution (probit link) μ is the mean, and σ is the spread, while in the logistic distribution (logit link), μ is the mean and $\sigma\pi/\sqrt{3}$ is the spread (cf. http://en.wikipedia.org/wiki/Logistic_distribution).

Thus allowing for scale effects corresponds to modelling not only the location of the latent distribution, but also the scale. Just as the absolute location (α^*) is not identifiable, the absolute scale (σ^*) is not identifiable either in the CLM, however *differences* in location modelled with $\boldsymbol{x}_i^T \boldsymbol{\beta}$ and *ratios* of scale modelled with $\exp(\boldsymbol{z}_i^T \boldsymbol{\zeta})$ are identifiable.

Now we turn to our running example and fit a model where we allow the scale of the latent distribution to depend on temperature:

$$\text{logit}(P(Y_i \le j)) = \frac{\theta_j - \beta_1(\texttt{temp}_i) - \beta_2(\texttt{contact}_i)}{\exp(\zeta_1(\texttt{temp}_i))} \tag{7}$$

$$i = 1, \ldots, n, \quad j = 1, \ldots, J - 1$$

We can estimate this model with

```
> fm.sca <- clm(rating ~ temp + contact, scale = ~temp, data = wine)
> summary(fm.sca)

formula: rating ~ temp + contact
scale:   ~temp
data:    wine

 link  threshold nobs logLik AIC    niter max.grad cond.H
 logit flexible  72   -86.44 186.88 8(0)  5.25e-09 1.0e+02


Coefficients:
           Estimate Std. Error z value Pr(>|z|)
tempwarm     2.6294     0.6860   3.833 0.000127 ***
contactyes   1.5878     0.5301   2.995 0.002743 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

log-scale coefficients:
         Estimate Std. Error z value Pr(>|z|)
tempwarm  0.09536    0.29414   0.324    0.746

Threshold coefficients:
    Estimate Std. Error z value
1|2  -1.3520     0.5223  -2.588
2|3   1.2730     0.4533   2.808
3|4   3.6170     0.7774   4.653
4|5   5.2982     1.2027   4.405
```

Notice that both location and scale effects of `temp` are identifiable. Also notice that the scale coefficient for `temp` is given on the log-scale, where the Wald test is more appropriate. The absolute scale of the latent distribution is not estimable, but we can estimate the scale at warm conditions relative to cold conditions. Therefore the estimate of κ in the relation $\sigma_{warm} = \kappa \sigma_{cold}$ is given by

```
> exp(fm.sca$zeta)

tempwarm
1.100054
```

However, the scale difference is not significant in this case as judged by the p-value in the summary output. `confint` and `anova` apply with no change to models with scale, but `drop1`, `add1` and `step` methods will only drop or add terms to the (location) `formula` and not to `scale`.

5   Structured Thresholds

In section 3 we relaxed the assumption that regression parameters have the same effect across all thresholds. In this section we will instead impose additional restrictions on the thresholds. In the following model we require that the thresholds, θ_j are equidistant or equally spaced ($\theta_j - \theta_{j-1}$ = constant for $j = 2, \ldots, J - 1$):

```
> fm.equi <- clm(rating ~ temp + contact, data = wine, threshold = "equidistant")
> summary(fm.equi)

formula: rating ~ temp + contact
data:    wine

 link  threshold   nobs logLik AIC    niter max.grad cond.H
 logit equidistant 72   -87.86 183.73 5(0)  4.80e-07 3.2e+01

Coefficients:
          Estimate Std. Error z value Pr(>|z|)
tempwarm    2.4632     0.5164    4.77 1.84e-06 ***
contactyes  1.5080     0.4712    3.20  0.00137 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Threshold coefficients:
            Estimate Std. Error z value
threshold.1  -1.0010     0.3978  -2.517
spacing       2.1229     0.2455   8.646
```

The parameters determining the thresholds are now the first threshold and the spacing among consecutive thresholds. The mapping to this parameterization is stored in the transpose of the Jacobian (`tJac`) component of the model fit. This makes it possible to get the thresholds imposed by the equidistance structure with

```
> c(with(fm.equi, tJac %*% alpha))
```

```
[1] -1.001044  1.121892  3.244828  5.367764
```

The following shows that the distance between consecutive thresholds in `fm1` is very close to the `spacing` parameter from `fm.equi`:

```
> mean(diff(fm1$alpha))
```

```
[1] 2.116929
```

The gain in imposing additional restrictions on the thresholds is the use of fewer parameters. Whether the restrictions are warranted by the data can be tested in a likelihood ratio test:

```
> anova(fm1, fm.equi)
```

```
Likelihood ratio tests of cumulative link models:

        formula:                   link: threshold:
fm.equi rating ~ temp + contact logit equidistant
fm1     rating ~ temp + contact logit flexible

        no.par  AIC   logLik LR.stat df Pr(>Chisq)
fm.equi      4 183.73 -87.865
fm1          6 184.98 -86.492 2.7454  2     0.2534
```

In this case the test is non-significant, so there is no considerable loss of fit at the gain of saving two parameters, hence we may retain the model with equally spaced thresholds.

6 Predictions

Fitted values are extracted with e.g., `fitted(fm1)` and produce fitted probabilities, i.e., the ith fitted probability is the probability that the ith observation falls in the response category that it did. The predictions of which response class the observations would be most likely to fall in given the model are obtained with:

```
> predict(fm1, type = "class")

$fit
 [1] 2 2 3 3 3 3 4 4 2 2 3 3 3 3 4 4 2 2 3 3 3 3 4 4 2 2 3 3 3 3 4 4 2 2 3 3 3 3
[39] 4 4 2 2 3 3 3 3 4 4 2 2 3 3 3 3 4 4 2 2 3 3 3 3 4 4 2 2 3 3 3 3 4 4
Levels: 1 2 3 4 5
```

Say we just wanted the predictions for the four combinations of `temp` and `contact`. The probability that an observation falls in each of the five response categories based on the fitted model is given by:

```
> newData <- expand.grid(temp = levels(wine$temp), contact = levels(wine$contact))
> cbind(newData, predict(fm1, newdata = newData)$fit)

  temp contact          1          2         3          4          5
1 cold      no 0.206790132 0.57064970 0.1922909 0.02361882 0.00665041
2 warm      no 0.020887709 0.20141572 0.5015755 0.20049402 0.07562701
3 cold     yes 0.053546010 0.37764614 0.4430599 0.09582084 0.02992711
4 warm     yes 0.004608274 0.05380128 0.3042099 0.36359581 0.27378469
```

Standard errors and confidence intervals of predictions are also available, for example, the predictions, standard errors and 95% confidence intervals are given by (illustrated here using `do.call` for the first six observations):

```
> head(do.call("cbind", predict(fm1, se.fit = TRUE, interval = TRUE)))

           fit       se.fit        lwr        upr
[1,] 0.57064970 0.08683884 0.39887109 0.7269447
[2,] 0.19229094 0.06388672 0.09609419 0.3477399
[3,] 0.44305990 0.07939754 0.29746543 0.5991420
[4,] 0.09582084 0.04257593 0.03887676 0.2173139
[5,] 0.20049402 0.06761012 0.09886604 0.3643505
[6,] 0.20049402 0.06761012 0.09886604 0.3643505
```

The confidence level can be set with the `level` argument and other types of predictions are available with the `type` argument.

7 Infinite Parameter Estimates

If we attempt to test the proportional odds assumption for `temp`, some peculiarities show up:

```
> fm.nom2 <- clm(rating ~ contact, nominal = ~temp, data = wine)
> summary(fm.nom2)

formula: rating ~ contact
nominal: ~temp
data:    wine
```

```
 link  threshold nobs logLik AIC    niter max.grad cond.H
 logit flexible  72   -84.90 187.81 16(0) 6.45e-07 2.8e+08


Coefficients:
          Estimate Std. Error z value Pr(>|z|)
contactyes   1.4652     0.4688   3.125  0.00178 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


Threshold coefficients:
               Estimate Std. Error z value
1|2.(Intercept)  -1.2656     0.5149  -2.458
2|3.(Intercept)   1.1040     0.4388   2.516
3|4.(Intercept)   3.7657     0.8046   4.680
4|5.(Intercept)  19.8964  2145.3568   0.009
1|2.tempwarm    -16.0954  1245.0152  -0.013
2|3.tempwarm     -2.1530     0.5898  -3.650
3|4.tempwarm     -2.8733     0.8174  -3.515
4|5.tempwarm    -17.5500  2145.3569  -0.008
```

Several of the threshold coefficients are extremely large with huge standard errors. Also the condition number of the Hessian is very large and a larger number of iterations was used all indicating that something is not well-behaved. The problem is that the the ML estimates of some of the threshold parameters are at (plus/minus) infinity. `clm` is not able to detect this and just stops iterating when the likelihood function is flat enough for the termination criterion to be satisfied, i.e., when the maximum absolute gradient is small enough.

Even though some parameter estimates are not at (plus/minus) infinity while they should have been, the remaining parameters are accurately determined and the value of the log-likelihood is also accurately determined. This means that likelihood ratio tests are still available, for example, it is still possible to test the proportional odds assumption for `temp`:

```
> anova(fm1, fm.nom2)

Likelihood ratio tests of cumulative link models:

       formula:                  nominal: link: threshold:
fm1     rating ~ temp + contact ~1        logit flexible
fm.nom2 rating ~ contact        ~temp     logit flexible


       no.par   AIC  logLik LR.stat df Pr(>Chisq)
fm1         6 184.98 -86.492
fm.nom2     9 187.81 -84.904   3.175  3     0.3654
```

8   Unidentified parameters

In the following example (using another data set) one parameter is not identifiable:
```
> data(soup)
> fm.soup <- clm(SURENESS ~ PRODID * DAY, data = soup)
> summary(fm.soup)
```

```
formula: SURENESS ~ PRODID * DAY
data:     soup

 link  threshold nobs logLik   AIC     niter max.grad cond.H
 logit flexible  1847 -2672.08 5374.16 6(1)  1.95e-13 9.4e+02

Coefficients: (1 not defined because of singularities)
           Estimate Std. Error z value Pr(>|z|)
PRODID2      0.6665     0.2146   3.106  0.00189 **
PRODID3      1.2418     0.1784   6.959 3.42e-12 ***
PRODID4      0.6678     0.2197   3.040  0.00237 **
PRODID5      1.1194     0.2400   4.663 3.11e-06 ***
PRODID6      1.3503     0.2337   5.779 7.53e-09 ***
DAY2        -0.4134     0.1298  -3.186  0.00144 **
PRODID2:DAY2 0.4390     0.2590   1.695  0.09006 .
PRODID3:DAY2     NA         NA      NA       NA
PRODID4:DAY2 0.3308     0.3056   1.083  0.27892
PRODID5:DAY2 0.3871     0.3248   1.192  0.23329
PRODID6:DAY2 0.5067     0.3350   1.513  0.13030
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Threshold coefficients:
    Estimate Std. Error z value
1|2 -1.63086    0.10740 -15.184
2|3 -0.64177    0.09682  -6.629
3|4 -0.31295    0.09546  -3.278
4|5 -0.05644    0.09508  -0.594
5|6  0.61692    0.09640   6.399
```

The undefined parameter shows up as NA in the coefficient table. The source of the singularity is revealed in the following table:

```
> with(soup, table(DAY, PRODID))

   PRODID
DAY   1   2   3  4  5  6
  1 369  94 184 93 88 93
  2 370 275   0 92 97 92
```

which shows that the third PRODID was not presented at the second day at all. The design matrix will in this case be column rank deficient (also referred to as singular). This is detected by clm using the drop.coef function from ordinal. The following illustrates that the column rank of the design matrix is less than its number of columns:

```
> mm <- model.matrix(~PRODID * DAY, data = soup)
> ncol(mm)

[1] 12

> qr(mm, LAPACK = FALSE)$rank

[1] 11
```

A similar type of rank deficiency occurs when variables in nominal are also present in formula or scale as illustrated in section 3.

9   Assessment of Model Convergence

The maximum likelihood estimates of the parameters in cumulative link models do not have closed form expressions, so iterative methods have to be applied to fit the models. Such iterative methods can fail to converge simply because an optimum cannot be found or because the parameter estimates are not determined accurately enough.

An optimum has been found if the maximum absolute gradient is small and if the condition number of the Hessian (defined here as the ratio of the largest to the smallest eigenvalues of the Hessian evaluated at convergence) is positive and not very large, say smaller than 10^4 or 10^6. The maximum absolute gradient (`max.grad`) and the condition number of the Hessian (`cond.H`) are reported by the summary method, for an example see page 4. A large condition number of the Hessian is not necessarily a problem, but it can be. A small condition number of the Hessian on the other hand is a good insurance that an optimum has been reached. Thus the maximum absolute gradient and the condition number of the Hessian can be used to check if the optimization has reached a well-defined optimum.

To determine the accuracy of the parameter estimates we use the `convergence` method:

```
> convergence(fm1)

 nobs logLik niter max.grad cond.H  logLik.Error
 72   -86.49 6(0)  4.01e-12 2.7e+01 -1.42e-14


          Estimate Std.Err  Gradient      Error Cor.Dec Sig.Dig
1|2         -1.344  0.5171  2.06e-12  3.09e-13      12      13
2|3          1.251  0.4379  2.12e-12 -2.42e-13      12      13
3|4          3.467  0.5978 -4.01e-12 -9.32e-13      11      12
4|5          5.006  0.7309 -7.36e-14 -9.21e-13      11      12
tempwarm     2.503  0.5287 -4.54e-13 -6.33e-13      11      12
contactyes   1.528  0.4766  5.57e-14 -2.95e-13      12      13


Eigen values of Hessian:
21.7090 18.5615 10.3914  5.2093  4.0955  0.8163
```

The most important information is the number of correct decimals (`Cor.Dec`) and the number of significant digits (`Sig.Dig`) with which the parameters are determined. In this case all parameters are very accurately determined, so there is no reason to lower the convergence tolerance. The `logLik.error` shows that the error in the reported value of the log-likelihood is in the order of 10^{-14}, which is by far small enough that likelihood ratio tests based on this model are accurate.

The convergence properties of the fitted model may be illustrated by plotting slices of the log-likelihood function for the parameters. The following code produce the slices in Figure 1.

```
> slice.fm1 <- slice(fm1, lambda = 5)
> par(mfrow = c(2, 3))
> plot(slice.fm1)
```

The slices illustrates the log-likelihood function plotted as a function each parameter in turn while the remaining parameters are fixed at the ML estimates. The `lambda` argument controls how far from the ML estimates the slices should be computed; it can be interpreted as a multiplier in curvature units, where a curvature unit is similar to a standard error.

For an inspection of the log-likelihood function closer to the optimum we can use a smaller
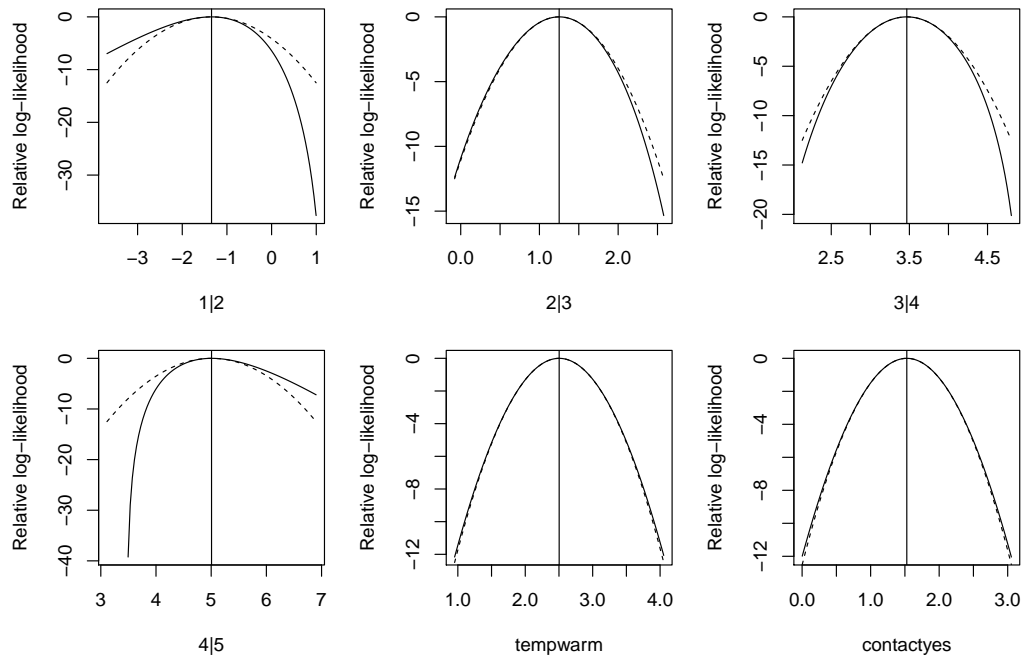
Figure 1: Slices of the (negative) log-likelihood function for parameters in a model for the bitterness-of-wine data. Dashed lines indicate quadratic approximations to the log-likelihood function and vertical bars indicate maximum likelihood estimates.
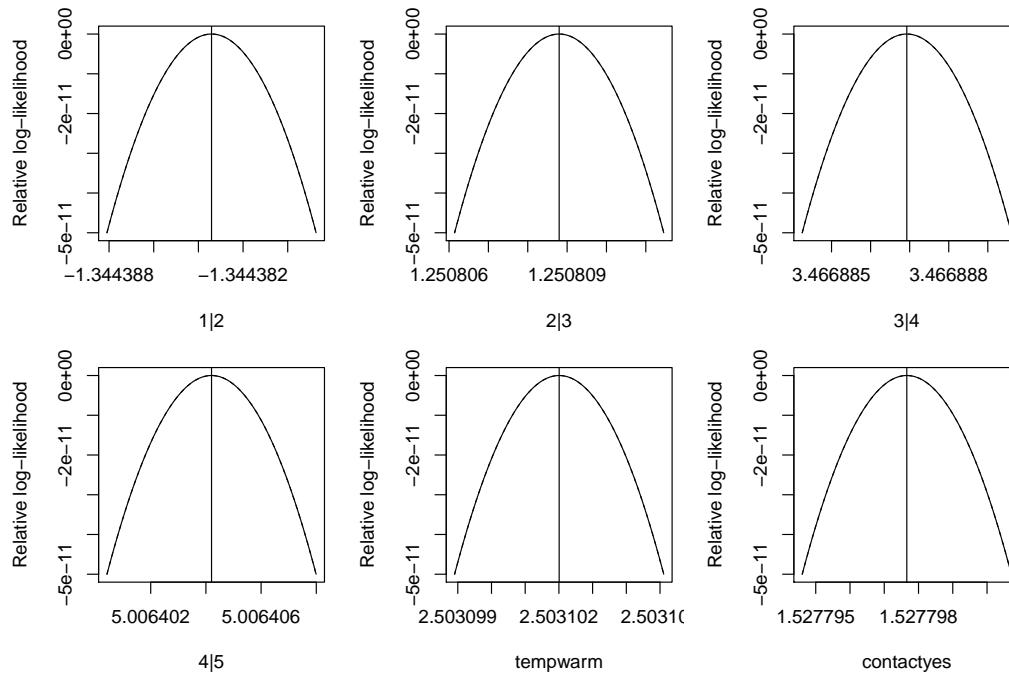
15

Figure 2: Slices of the log-likelihood function for parameters in a model for the bitterness-of-wine data very close to the MLEs. Dashed lines indicate quadratic approximations to the log-likelihood function and vertical bars the indicate maximum likelihood estimates.

lambda:

```
> slice2.fm1 <- slice(fm1, lambda = 1e-05)
> par(mfrow = c(2, 3))
> plot(slice2.fm1)
```

The resulting figure is shown in Fig. 2.

10   Profile Likelihood

The profile likelihood can be used for several things. Two of the most important objectives are to provide accurate likelihood confidence intervals and to illustrate effects of parameters in the fitted model.

Confidence intervals based on the profile likelihood were already obtained in section 2 and will not be treated any further here.

The effects of contact and temp can be illustrated with

```
> pr1 <- profile(fm1, alpha = 1e-04)
> plot(pr1)
```

and provided in Figure 3. The alpha argument is the significance level controling how far from the maximum likelihood estimate the likelihood function should be profiled. Learn more about the arguments to profile with help(profile.clm). From the relative profile
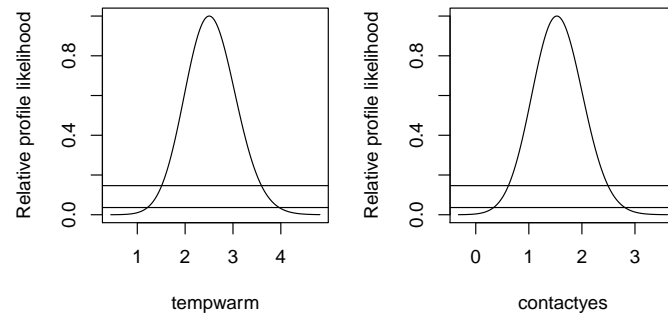
16

Figure 3: Relative profile likelihoods for the regression parameters in the Wine study. Horizontal lines indicate 95% and 99% confidence bounds.

likelihood for `tempwarm` we see that parameter values between 1 and 4 are reasonably well supported by the data, and values outside this range has little likelihood. Values between 2 and 3 are very well supported by the data and all have high likelihood.

References

Agresti, A. (2010). *Analysis of ordinal categorical data* (2nd ed.). Wiley.

Christensen, R. H. B. (2011). *Analysis of ordinal data with cumulative link models — estimation with the ordinal package.* R-package version 2011.09-13.

Randall, J. (1989). The analysis of sensory data by generalised linear model. *Biometrical journal 7*, pp. 781–793.

Tutz, G. and W. Hennevogl (1996). Random effects in ordinal regression models. *Computational Statistics & Data Analysis 22*, pp. 537–557.