A Primer on Cumulative Link Models with the ordinal2 Package

Rune Haubo B Christensen

May 16, 2011

Abstract

In this primer cumulative link models are introduced using the ordinal2 package. This is work in progress.

Contents

1 Introduction

Ordered categorical data, or simply *ordinal* data, are commonplace in scientific disciplines where humans are used as measurement instruments. Examples include school gradings, ratings of preference in consumer studies, degree of tumor involvement in MR images and animal fitness in field ecology. Cumulative link models are a powerful model class for such data since observations are treated rightfully as categorical, the ordered nature is exploited and the flexible regression framework allows in-depth analyses.

The name *cumulative link models* is adopted from Agresti (2002), but the models are also known as *ordinal regression models* although that term is sometimes also used for other regression models for ordinal responses such as *continuation ratio models* (see e.g., Agresti, 2002). Other aliases are *ordered logit models* and *ordered probit models* (Greene and Hensher, 2010) for the logit and probit link functions. Further, the cumulative link model with a logit link is widely known as the *proportional odds model* due to McCullagh (1980), also with a log-log link[1], the model is known as *proportional hazards model* for grouped survival times.

Ordinal response variables can be analyzed with omnibus Pearson χ^2 tests, base-line logit models or log-linear models. This corresponds to assuming that the response variable is nominal and information about the ordering of the categories will be ignored. Alternatively numbers can be attached to the response categories, e.g., $1, 2, \ldots, J$ and the resulting scores can be analyzed by conventional linear regression and ANOVA models. This approach is in a sense over-confident since the data are assumed to contain more information than they actually do. Observations on an ordinal scale are classified in ordered categories, but the distance between the categories is generally unknown. By using linear models the choice of scoring impose assumptions about the distance between the response categories. Further, standard errors and tests from linear models rest on the assumption that the response, conditional on the explanatory variables, is normally distributed (equivalently the residuals are assumed to be normally distributed). This cannot be the case since the scores are discrete and responses beyond the end categories are not possible. If there are many responses in the end categories, there will most likely be variance heterogeneity to which F and t tests can be rather sensitive. If there are many response categories and the response does not pile up in the end categories, we may expect tests from linear models to be accurate enough, but any bias and optimism is hard to quantify.

Cumulative link models provide the regression framework familiar from linear models while treating the response rightfully as categorical. While cumulative link models are not the only type of ordinal regression model, they are by far the most popular class of ordinal regression models.

Common to the application of methods for nominal responses and linear models to ordinal responses is that interpretation of effects on the ordinal response scale is ackward. For example, linear models will eventually give predictions outside the possible range and statements such as "the response increase 1.2 units with each degree increase in temperature" will only be approximately valid in a restricted range of the response variable.

In this document cumulative link models are described for modeling ordinal response variables and the ordinal2 package is introduced. Our focus will be ...

Example 1 (The wine data): As an example of the data set with an ordinal response varible consider the wine data from Randall (1989) available in the object `wine` in package ordinal2, cf.

[1] or is it the complementary log-log link?

Table 1: Wine data from Randall (1989).

		Least—Most bitter				
Temperature	Contact	1	2	3	4	5
cold	no	4	9	5	0	0
cold	yes	1	7	8	2	0
warm	no	0	5	8	3	2
warm	yes	0	1	5	7	5

Table 1. 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 cruching 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. In table X we have aggregated data over bottles and judges for simplicity, but these variables will be considered later. Initially we only assume that, say, category 4 is larger than 3, but not that the distance between 2 and 3 is half the distance between 2 and 4, for example. The main objective is to examine the effect of contact and temperature on the perceived bitterness of wine. □

2 Cumulative link models

A cumulative link model is a model for an ordinal response variable, Y_i that can fall in $j = 1, \ldots, J$ categories.[2] Then Y_i follows a multinomial distribution with parameter $\boldsymbol{\pi}$ where π_{ij} denote the probability that the ith observation falls in response category j. We define the cumulative probabilities as[3]

$$\gamma_{ij} = P(Y_i \leq j) = \pi_{i1} + \ldots + \pi_{ij} \ . \tag{1}$$

Initially we will consider the logit link. The logit function is defined as $\text{logit}(\pi) = \log[\pi/(1-\pi)]$ and cumulative logits are defined as:

$$\text{logit}(\gamma_{ij}) = \text{logit}(P(Y_i \leq j)) = \log \frac{P(Y_i \leq j)}{1 - P(Y_i \leq j)} \quad j = 1, \ldots, J-1 \tag{2}$$

so that the cumulative logits are defined for all but the last category.[4]

Example 2: For fixed j the cumulative logit model (3) is just a logistic regression model where the binomial response is divided into those observations falling in category j or less, and those falling in a higher category than j. This is a valid analysis approach, but it does not use all the information in the data and the (arbitrary) choice of j also needs to be made. To improve on this we could make all the ordinary logistic regressions for $j = 1, \ldots, J-1$, but it is not easy to draw inference from a collection of models and it would be nice if a plausible data generating mechanism could be summarized in a single model. This is also likely to increase the power of hypothesis tests for the parameters. The cumulative logit model is exactly the model that combines all these ordinary logistic regressions into one. If the cumulative model is correct, the

[2] where $J \geq 2$. If $J = 2$ binomial models also apply, and in fact the cumulative link model is in this situation identical to a generalized linear model for a binomial response.

[3] we have suppressed the conditioning on the covariate vector, $\boldsymbol{x}_i$, so we have that $\gamma_{ij} = \gamma_j(\boldsymbol{x}_i)$ and $P(Y_i \leq j) = P(Y \leq j|\boldsymbol{x}_i)$.

[4] since for $j = J$ the denominator would be $1 - P(Y_i \leq J) = 1 - 1 = 0$ and thus the fraction is not defined.

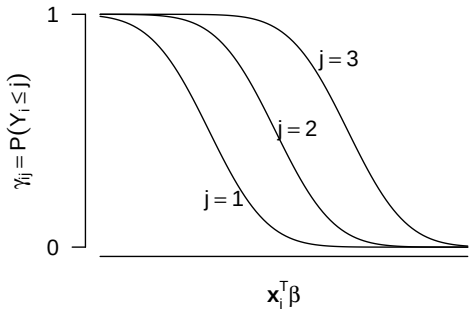

Figure 1: Illustration of a cumulative link model with four response categories.

regression parameter estimates from the individual logistic regression models will be similar and approximately identical to the regression parameter estimates from the cumulative probit model. Thus the cumulative link model saves the estimation of $J-2$ sets of $\boldsymbol{\beta}$—this is where the power gain comes from. Further, we also expect the intercept parameter estimates from the logistic regression models to be approximately the same as estimates of $\{\theta_j\}$ from the cumulative link model. □

A cumulative link model with a logit link, or simply *cumulative logit model* is a regression model for cumulative logits:

$$\text{logit}(\gamma_{ij}) = \theta_j - \boldsymbol{x}_i^T\boldsymbol{\beta} \tag{3}$$

where $\boldsymbol{x}_i$ is a vector of explanatory variables for the ith observation and $\boldsymbol{\beta}$ is the corresponding set of regression parameters. The $\{\theta_j\}$ parameters provide each cumulative logit (for each j) with its own intercept. A key point is that the regression part $\boldsymbol{x}_i^T\boldsymbol{\beta}$ is independent of j, so $\boldsymbol{\beta}$ has the same effect for each of the $J-1$ cumulative logits. Note that $\boldsymbol{x}_i^T\boldsymbol{\beta}$ does not contain an intercept, since the $\{\theta_j\}$ act as intercepts. The cumulative logit model is illustrated in Fig. 1 for data with four response categories. For small values of $\boldsymbol{x}_i^T\boldsymbol{\beta}$ the response is likely to fall in the first category and for large values of $\boldsymbol{x}_i^T\boldsymbol{\beta}$ the response is likely to fall in the last category. The horizontal displacements of the curves are given by the values of $\{\theta_j\}$.

Some sources write the cumulative logit model, (3) with a plus on the right-hand-side, but there are two good reasons for the minus. First, it means that the larger the value of $\boldsymbol{x}_i^T\boldsymbol{\beta}$, the higher the probability of the response falling in a category at the upper end of the response scale. Thus $\boldsymbol{\beta}$ has the same direction of effect as the regression parameter in an ordinary linear regression or ANOVA model. The second reason is related to the latent variable interpretation of cumulative link models that we will consider in section 2.6.

2.1 Fitting cumulative link models with `clm` from package ordinal2

Cumulative link models can be fitted with `clm` from package ordinal2. The function takes the following arguments:

```
clm(formula, data, weights, start, subset, doFit = TRUE, na.action,
  contrasts, model = TRUE, control, link = c("logit", "probit",
```

```
"cloglog", "loglog", "cauchit"), threshold = c("flexible",
"symmetric", "equidistant"), ...)
```

Most arguments are standard and well-known from `lm` and `glm`, so they will not be introduced. The `formula` argument is of the form `response ~ covariates` and specifies the linear predictor. The `response` should be an *ordered factor* (see `help(factor)`) with levels corresponding to the response categories. A number of link functions are available and the logit link is the default. The `doFit` and `threshold` arguments will be introduced in later sections. For further information about the arguments see the help page for `clm`[5].

Example 3: In this example we fit a cumulative logit model to the wine data presented in example 1 with `clm` from package ordinal2. A cumulative logit model that includes additive effects of temperature and contact is fitted and summarized with

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

formula: rating ~ contact + temp
data:    wine

 link  threshold nobs logLik AIC    niter max.grad cond.H
 logit flexible  72   -86.49 184.98 7(0)  4.01e-12 2.7e+01

Coefficients:
           Estimate Std. Error z value Pr(>|z|)
contactyes   1.5278     0.4766   3.205  0.00135 **
tempwarm     2.5031     0.5287   4.735 2.19e-06 ***
---
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 summary provides basic information about the model fit. There are two coefficient tables: one for the regression variables and one for the thresholds or cutpoints. Often the thresholds are not of primary interest, but they are an integral part of the model. It is not relevant to test whether the thresholds are equal to zero, so no p-values are provided for this test. The condition number of the Hessian is a measure of how identifiable the model is; large values, say larger than `1e4` indicate that the model may be ill defined. From this model it appears that contact and high temperature both lead to higher probabilities of observations in the high categories as we would also expect from examining Table 1.

The Wald tests provided by `summary` indicate that both contact and temperature effects are strong. More accurate likelihood ratio tests can be obtained using the `drop1` and `add1` methods (equivalently `dropterm` or `addterm`). The Wald tests are marginal tests so the test of e.g., `temp` is measuring the effect of temperature while *controling* for the effect of contact. The equivalent likelihood ratio tests are provided by the `drop`-methods:

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

Single term deletions
```

[5]Typing `?clm` or `help(clm)` in the command prompt should display the help page for `clm`.

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

In this case the likelihood ratio tests are slightly more significant than the Wald tests. We could also have tested the effect of the variables while *ignoring* the effect of the other variable. For this test we use the `add`-methods:

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

Single term additions

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

where we used the `scope` argument to indicate which terms to include in the model formula. These tests are a little less significant than the tests controlling for the effect of the other variable.

Conventional symmetric so-called Wald confidence intervals for the parameters are available as

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

               2.5 %   97.5 %
contactyes 0.5936345 2.461961
tempwarm   1.4669081 3.539296
```

More accurate profile likelihood confidence intervals are also available and these are discussed in section 5. □

2.2 Odds ratios and proportional odds

The odds ratio of the event $Y \leq j$ at $\boldsymbol{x}_1$ relative to the same event at $\boldsymbol{x}_2$ is

$$\text{OR} = \frac{\gamma_j(\boldsymbol{x}_1)/[1-\gamma_j(\boldsymbol{x}_1)]}{\gamma_j(\boldsymbol{x}_2)/[1-\gamma_j(\boldsymbol{x}_2)]} = \frac{\exp(\theta_j - \boldsymbol{x}_1^T\boldsymbol{\beta})}{\exp(\theta_j - \boldsymbol{x}_2^T\boldsymbol{\beta})} = \exp[(\boldsymbol{x}_2^T - \boldsymbol{x}_1^T)\boldsymbol{\beta}]$$

which is independent of j. Thus the cumulative odds ratio is proportional to the distance between $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ which made McCullagh (1980) call the cumulative logit model a *proportional odds model*. If x represent a treatment variable with two levels (e.g., placebo and treatment), then $x_2 - x_1 = 1$ and the odds ratio is $\exp(-\beta_{\text{treatment}})$. Similarly the odds ratio of the event $Y \geq j$ is $\exp(\beta_{\text{treatment}})$.

Confidence intervals for the odds ratios are obtained by transforming the limits of confidence intervals for $\boldsymbol{\beta}$, which will lead to asymmetric confindence intervals for the odds ratios.

Table 2: Summary of various link functions

Name	logit	probit	log-log	clog-log[a]	cauchit
Distribution	logistic	Normal	Gumbel (max)[b]	Gumbel (min)[b]	Cauchy[c]
Shape	symmetric	symmetric	right skew	left skew	kurtotic
Link function (F^{-1})	$\log[\gamma/(1-\gamma)]$	$\Phi^{-1}(\gamma)$	$-\log[-\log(\gamma)]$	$\log[-\log(1-\gamma)]$	$\tan[\pi(\gamma-0.5)]$
Inverse link (F)	$1/[1+\exp(\eta)]$	$\Phi(\eta)$	$\exp(-\exp(-\eta))$	$1-\exp[-\exp(\eta)]$	$\arctan(\eta)/\pi+0.5$
Density ($f=F'$)	$\exp(-\eta)/[1+\exp(-\eta)]^2$	$\phi(\eta)$	$\exp(-\exp(-\eta)-\eta)$	$\exp[-\exp(\eta)+\eta]$	$1/[\pi(1+\eta^2)]$

[a]: the *complementary log-log* link
[b]: the Gumbel distribution is also known as the extreme value (type I) distribution for extreme mimina or maxima. It is also sometimes refered to as the Weibull (or log-Weibull) distribution (`http://en.wikipedia.org/wiki/Gumbel_distribution`).
[c]: the Cauchy distribution is a t-distribution with one df

Symmetric confidence intervals constructed from the standard error of the odds ratios will not be appropriate and should be avoided.

Example 4: The (cumulative) odds ratio of `rating` $\geq j$ (for all $j = 1, \ldots, J-1$) for contact and temperature are

```
> round(exp(fm1$beta), 1)

contactyes   tempwarm
       4.6       12.2
```

attesting to the strong effects of contact and temperature. Asymmetric confidence intervals for the odds ratios based on the Wald statistic are:

```
> round(exp(confint(fm1, type = "Wald")), 1)

           2.5 % 97.5 %
contactyes   1.8   11.7
tempwarm     4.3   34.4
```

□

2.3 Link functions

Cumulative link models are not formally a member of the class of (univariate) generalized linear models[6] (McCullagh and Nelder, 1989), but they share many similarities with generalized linear models. Notably a link function and a linear predictor ($\eta_{ij} = \theta_j - \boldsymbol{x}_i^T\boldsymbol{\beta}$) needs to be specified as in generalized linear models while the response distribution is just the multinomial. Fahrmeir and Tutz (2001) argues that cumulative link models are members of a class of multivariate generalized linear models. In addition to the logit link other choices are the probit, cauchit, log-log and clog-log links. These are summarized in Table 2. The cumulative link model may be written as

$$F^{-1}(\gamma_{ij}) = \theta_j - \boldsymbol{x}_i^T\boldsymbol{\beta} \tag{4}$$

where F^{-1} is the link function—the motivation for this particular notation will be given in section 2.6.

[6]the distribution of the response, the multinomial, is not a member of the (univariate) exponential family of distributions.

Table 3: Income distribution (percentages) in the Northeast US adopted from McCullagh (1980).

Year	Income						
	0-3	3-5	5-7	7-10	10-12	12-15	15+
1960	6.50	8.20	11.30	23.50	15.60	12.70	22.20
1970	4.30	6.00	7.70	13.20	10.50	16.30	42.10

The probit link is often used when the model is interpreted with reference to a latent variable, cf. section 2.6. When the response variable represent grouped duration or survival times the complementary log-log link is often used. This leads to the propotional hazard model for grouped responses:

$$-\log\{1-\gamma_j(\boldsymbol{x}_i)\} = \exp(\theta_j - \boldsymbol{x}_i^T\boldsymbol{\beta})$$

or equivalently

$$\log[-\log\{1-\gamma_j(\boldsymbol{x}_i)\}] = \theta_j - \boldsymbol{x}_i^T\boldsymbol{\beta}\ . \tag{5}$$

Here $1-\gamma_j(\boldsymbol{x}_i)$ is the probability or survival beyond category j given $\boldsymbol{x}_i$. The proportional hazards model has the property that

$$\log\{\gamma_j(\boldsymbol{x}_1)\} = \exp[(\boldsymbol{x}_2^T - \boldsymbol{x}_1^T)\boldsymbol{\beta}]\log\{\gamma_j(\boldsymbol{x}_2)\}\ .$$

If the log-log link is used on the response categories in the reverse order, this is equivalent to using the c-log-log link on the response in the original order. This reverses the sign of $\boldsymbol{\beta}$ as well as the sign and order of $\{\theta_j\}$ while the likelihood and standard errors remain unchanged.

In addition to the standard links in Table 2, flexible link functions are available for `clm` in package ordinal and these are described in section XX.

Example 5: McCullagh (1980) present data on income distribution in the Northeast US reproduced in Table 3 and available in package ordinal2 as the object `income`. The unit of the income groups are thousands of (constant) 1973 US dollars. The numbers in the body of the table are percentages of the population summing to 100 in each row[7], so these are not the orginal observations. The uncertainty of parameter estimates depends on the sample size, which is unknown here, so we will not consider hypothesis tests. Rather the most important systematic component is an upward shift in the income distribution from 1960 to 1970 which can be estimated from a cumulative link model. This is possible since the parameter estimates themselves only depend on the relative proportions and not the absolute numbers.

McCullagh considers which of the logit or cloglog links best fit the data in a model with an additive effect of `year`. He concludes that a the complementary log-log link corresponding to a right-skew distribution is a good choice. We can compare the relative merit of the links by comparing the value of the log-likelihood of models with different link functions:

```
> links <- c("logit", "probit", "cloglog", "loglog", "cauchit")
> sapply(links, function(link) {
     clm(income ~ year, data = income, weights = pct, link = link)$logLik
 })

    logit    probit   cloglog    loglog   cauchit
-353.3589 -353.8036 -352.8980 -355.6028 -352.8434
```

[7] save rounding error

The cauchit link attains the highest log-likelihood closely followed by the complementary log-log link. This indicates that a symmetric heavy tailed distribution such as the cauchy provides an even slightly better description of these data than a right skew distribution.

Adopting the complementary log-log link we can summarize the connection between the income in the two years by the following: If $p_{1960}(x)$ is proportion of the population with an income larger than $\$x$ in 1960 and $p_{1970}(x)$ is the equivalent in 1970, then approximately

$$\begin{aligned} \log p_{1960}(x) &= \exp(\hat{\beta}) \log p_{1970}(x) \\ &= \exp(0.568) \log p_{1970}(x) \end{aligned}$$ □

2.4 Maximum likelihood estimation of cumulative link models

Cumulative link models are usually estimated by maximum likelihood (ML) and this is also the criterion used in package ordinal2. The log-likelihood function (ignoring additive constants) can be written as

$$\ell(\boldsymbol{\theta}, \boldsymbol{\beta}; \boldsymbol{y}) = \sum_{i=1}^{n} w_i \log \pi_i \tag{6}$$

where i index all scalar observations (not multinomial vector observations), w_i are potential case weights and π_i is the probability of the ith observation falling in the response category that it did, i.e., π_i are the non-zero elements of $\pi_{ij}\mathrm{I}(Y_i = j)$. Here $\mathrm{I}(\cdot)$ is the indicator function being 1 if its argument is true and zero otherwise. The ML estimates of the parameters; $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\beta}}$ are those values of $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ that maximize the log-likelihood function in (6).

Not all data sets can be summarized in a table like Table 1. If a continuous variable takes a unique value for each observation, each row of the resulting table would contain a single 1 and zeroes for the rest. In this case all $\{w_i\}$ are one unless the observations are weighted for some other reason. If the data can be summarized as in Table 1, a multinomial observation vector such as $[3, 1, 2]$ can be fitted using $\boldsymbol{y} = [1, 1, 1, 2, 3, 3]$ with $\boldsymbol{w} = [1, 1, 1, 1, 1, 1]$ or by using $\boldsymbol{y} = [1, 2, 3]$ with $\boldsymbol{w} = [3, 1, 2]$. The latter construction is considerably more computationally efficient (an therefore faster) since the log-likelihood function contains three rather than six terms and the design matrix, $\boldsymbol{X}$ will have three rather than six rows.

The details of the actual algorithm by which the likelihood function is optimized is deferred to a later section.

According to standard likelihood theory, the variance-covariance matrix of the parameters can be obtained as the inverse of the observed Fisher information matrix. This matrix is given by the negative Hessian of the log-likelihood function[8] evaluated at the maximum likelihood estimates. Standard errors can be obtained as the square root of the diagonal of the variance-covariance matrix.

Let $\boldsymbol{\alpha} = [\boldsymbol{\theta}, \boldsymbol{\beta}]$ denote the full set of parameters. The Hessian matrix is then given as the second order derivative of the log-likelihood function evaluated at the ML estimates:

$$\boldsymbol{H} = \left. \frac{\partial^2 \ell(\boldsymbol{\alpha}; \boldsymbol{y})}{\partial \boldsymbol{\alpha} \partial \boldsymbol{\alpha}^T} \right|_{\boldsymbol{\alpha} = \hat{\boldsymbol{\alpha}}} . \tag{7}$$

[8] equivalently the Hessian of the negative log-likelihood function.

The observed Fisher information matrix is then $\boldsymbol{I}(\hat{\boldsymbol{\alpha}}) = -\boldsymbol{H}$ and the standard errors are given by

$$\text{se}(\hat{\boldsymbol{\alpha}}) = \sqrt{\text{diag}[\boldsymbol{I}(\hat{\boldsymbol{\alpha}})^{-1}]} = \sqrt{\text{diag}[-\boldsymbol{H}(\hat{\boldsymbol{\alpha}})^{-1}]}. \tag{8}$$

Another general way to obtain the variance-covariance matrix of the parameters is to use the expected Fisher information matrix. The choice of whether to use the observed or the expected Fisher information matrix is often dictated by the fitting algorithm: re-weighted least squares methods often produce the expected Fisher information matrix as a by-product of the algorithm, and Newton-Raphson algorithms (such as the one used for `clm` in ordinal2) similarly produce the observed Fisher information matrix. Efron and Hinkley (1978) considered the choice of observed versus expected Fisher information and argued that the observed information contains relevant information thus it is prefered over the expected information.

Pratt (1981) and Burridge (1981) showed (seemingly independent of each other) that the log-likelihood function of cumulative link models with the link functions considered in Table 2, except for the cauchit link, is concave. This means that there is a unique global optimum so there is no risk of convergence to a local optimum. It also means that the step of a Newton-Raphson algorithm is quarantied to be in the direction of a higher likelihood although the step may be too larger to cause an increase in the likelihood. Succesively halfing the step whenever this happens effectively ensures convergence.

Notably the log likelihood of cumulative cauchit models is not guarantied to be concave, so convergence problems may occur with the Newton-Raphson algorithm. Using the estimates from a cumulative probit models as starting values seems to be a widely successful approach.

Observe also that the concavity property does not extend to cumulative link models with scale effects, but that structured thresholds (cf. section 2.7) are included.

2.5 Deviance and model comparison

2.5.1 Model comparison with likelihood ratio tests

A general way to compare models is by means of the likelihood ratio statistic. Consider two models, m_0 and m_1, where m_0 is a submodel of model m_1, that is, m_0 is simpler than m_1 and m_0 is *nested* in m_1. The likelihood ratio statistic for the comparison of m_0 and m_1 is

$$LR = -2(\ell_0 - \ell_1) \tag{9}$$

where ℓ_0 is the log-likelihood of m_0 and ℓ_1 is the log-likelihood of m_1. The likelihood ratio statistic measures the evidence in the data for the extra complexity in m_1 relative to m_0. The likelihood ratio statistic asymptotically follows a χ^2 distribution with degrees of freedom equal to the difference in the number of parameter of m_0 and m_1. The likelihood ratio test is generally more accurate than Wald tests. Cumulative link models can be compared by means of likelihood ratio tests with the `anova` method.

Example 6: Consider the additive model for the wine data in example 3 with a main effect of temperature and contact. We can use the likelihood ratio test to assess whether the interaction between these factors are supported by the data:

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

```
Likelihood ratio tests of cumulative link models:

  formula:                link: threshold:
1 rating ~ contact + temp logit flexible
2 rating ~ contact * temp logit flexible
  no.par    AIC  logLik LR.stat df Pr(>Chisq)
1      6 184.98 -86.492
2      7 186.83 -86.416  0.1514  1     0.6972
```

The likelihood ratio statistic is small in this case and compared to a χ^2 distribution with 1 df, the p-value turns out insignifcant. We conclude that the interaction is not supported by the data. □

2.5.2 Deviance and ANODE tables

In linear models ANOVA tables and F-tests are based on the decomposition of sums of squares. The concept of sums of squares does not make much sense for categorical observations, but a more general measure called the *deviance* is defined for generalized linear models and contingency tables[9]. The deviance can be used in much the same way to compare nested models and to make a so-called analysis of deviance (ANODE) table. The deviance is closely related to sums of squares for linear models (McCullagh and Nelder, 1989).

The deviance is defined as minus twice the difference between the log-likelihoods of a *full* (or *saturated*) model and a reduced model:

$$D = -2(\ell_{\text{reduced}} - \ell_{\text{full}}) \tag{10}$$

The full model has a parameter for each observation and describes the data perfectly while the reduced model provides a more concise description of the data with fewer parameters.

A special reduced model is the *null model* which describes no other structure in the data than what is implied by the design. The corresponding deviance is known as the *null deviance* and analogous to the total sums of squares for linear models. The null deviance is therefore also denoted the *total deviance*. The *residual deviance* is a concept similar to a residual sums of squares and simply defined as

$$D_{\text{resid}} = D_{\text{total}} - D_{\text{reduced}} \tag{11}$$

A *difference in deviance* between two nested models is identical to the likelihood ratio statistic for the comparison of these models. Thus the deviance difference, just like the likelihood ratio statistic, asymptotically follows a χ^2-distribution with degrees of freedom equal to the difference in the number of parameters in the two models. In fact the deviance in (10) is just the likelihood ratio statistic for the comparison of the full and reduced models.

The likelihood of reduced models are available from fits of cumulative link models, but since it is not always easy to express the full model as a cumulative link model, the log-likelihood of the full model has to be obtained in another way. For a two-way table like Table 1 indexed by h (rows) and j (columns), the log-likelihood of the full model (comparable to the likelihood in (6)) is given by

$$\ell_{\text{full}} = \sum_h \sum_j w_{hj} \log \hat{\pi}_{hj} \tag{12}$$

[9] i.e., for likelihood based models for contingency tables

Table 4: ANODE table for the data in Table 1.

Source	df	deviance	p-value
Total	12	39.407	< 0.001
Treatment	3	34.606	< 0.001
Temperature, T	1	26.928	< 0.001
Contact, C	1	11.043	< 0.001
Interaction, $T \times C$	1	0.1514	0.6972
Residual	9	4.8012	0.8513

where $\hat{\pi}_{hj} = w_{hj}/w_{h\cdot}$, w_{hj} is the count in the (h, j)th cell and $w_{h\cdot}$ is the sum in row h.

Example 7: We can get the likelihood of the full model for the wine data in Table 1 with

```
> tab <- with(wine, table(temp:contact, rating))
> pi.hat <- tab/rowSums(tab)
> (ll.full <- sum(tab * ifelse(pi.hat > 0, log(pi.hat), 0)))
```

```
[1] -84.01558
```

The total deviance (10) for the wine data is given by

```
> fm0 <- clm(rating ~ 1, data = wine)
> ll.null <- fm0$logLik
> (Deviance <- -2 * (ll.null - ll.full))
```

```
[1] 39.407
```

□

Example 8: An ANODE table for the wine data in Table 1 is presented in Table 4 where the total deviance is brocken up into model deviance (due to treatments) and residual deviance. Further, the treatment deviance is described by contributions from main effects and interaction. Observe that the deviances for the main effects and interaction do not add up to the deviance for Treatment as the corresponding sums of squares would have in a analogous linear model (ANOVA)[10]. The deviances for these terms can instead be interpreted as likelihood ratio tests of nested models: the deviance for the interaction term is the likelihood ratio statistics of the interaction controlling for the main effects, and the deviances for the main effects are the likelihood ratio statistics for these terms while controlling for the other main effect and ignoring the interaction term. As is clear from Table 4, there are significant treatment differences and these seem to describe the data well since the residual deviance is insignificant—the latter is a goodness of fit test for the cumulative logit model describing treatment differences. Further, the treatment differences are well captured by the main effects and there is no indication of an important interaction. □

The terminology can be a bit confusing in this area. Sometimes any difference in deviance between two nested models, i.e., a likelihood ratio statistic is denoted a deviance and sometimes any quantity that is proportional to minus twice the log-likelihood of a model is denoted the deviance of that model.

2.5.3 Goodness of fit tests with the deviance

The deviance can be used to test the goodness of fit of a particular reduced model. The deviance asymptotically follows as χ^2 distribution with degrees of freedom equal to the dif-

[10] This holds for orthogonal designs including balanced and complete tables like Table 1.

Table 5: Table of the wine data similar to Table 1, but including bottle in the tabulation.

			Least—Most bitter				
Temperature	Contact	Bottle	1	2	3	4	5
cold	no	1	3	4	2	0	0
cold	no	2	1	5	3	0	0
cold	yes	3	1	2	5	1	0
cold	yes	4	0	5	3	1	0
warm	no	5	0	3	4	1	1
warm	no	6	0	2	4	2	1
warm	yes	7	0	1	2	2	4
warm	yes	8	0	0	3	5	1

ference in the number of parameters between the two models. The asymptotics are generally good if the expected frequencies under the reduced model are not too small and as a general rule they should all be at least five. This provides a goodness of fit test of the reduced model. The expectation of a random variable that follows a χ^2-distribution is equal to the degrees of freedom of the distribution, so as a rule of thumb, if the deviance in (10) is about the same size as the difference in the number of parameters, there is not evidence of lack of fit.

One problem with the deviance for a particular (reduced) model is that it depends on which model is considered the full model, i.e., how the total deviance is calculated, which often derives from the tabulation of the data. Observe that differences in deviance for nested models are independent of the likelihood of a full model, so deviance differences are insensitive to this choice. Collett (2002) recommends that the data are aggregated as much as possible when evaluating deviances and goodness of fit tests are performed.

Example 9: In the presentation of the wine data in example 1 and Table 1, the data were aggregated over judges and bottles. Had we included `bottle` in the tabulation of the data we would have arrived at Table 5. A full model for the data in Table 1 has $(5-1)(4-1) = 12$ degrees of freedom while a full model for Table 5 has $(5-1)(8-1) = 28$ degrees of freedom and a different deviance.

If it is decided that bottle is not an important variable, Collett's recommendation is that we base the residual deviance on a full model defined from Table 1 rather than Table 5. □

2.6 Latent variable motivation for cumulative link models

A cumulative link model can be motivated by assuming an underlying continuous latent variable, S with cumulative distribution function, F. The ordinal response variable, Y_i is then observed in category j if S_i is between the thresholds $\theta^*_{j-1} < S_i \leq \theta^*_j$ where

$$-\infty \equiv \theta^*_0 < \theta^*_1 < \ldots < \theta^*_{J-1} < \theta^*_J \equiv \infty$$

divide the real line on which S lives into $J+1$ intervals. The situation is illustrated in Fig. 2 where a probit link and $J = 4$ is adopted. The three thresholds, $\theta_1, \theta_2, \theta_3$ divide the area under the curve into four parts each of which represent the probability of a response falling in the four response categories. The thresholds are fixed on the scale, but the location of the latent distribution, and therefore also the four areas under the curve, changes with $\boldsymbol{x}_i$.

Better figure here - see Agresti 2002, p. 278

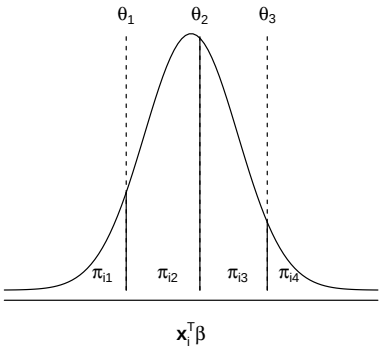

Figure 2: Illustration of a cumulative link model in terms of the latent distribution.

A normal linear model for the latent variable is

$$S_i = \alpha + \boldsymbol{x}_i^T \boldsymbol{\beta}^* + \varepsilon_i \ , \quad \varepsilon_i \sim N(0, \sigma^2) \tag{13}$$

where $\{\varepsilon_i\}$ are random disturbances and α is the intercept, i.e., the mean value of S_i when $\boldsymbol{x}_i$ correspond to a reference level for factors and to zero for continuous covariates. Equivalently we could write: $S_i \sim N(\alpha + \boldsymbol{x}_i^T \boldsymbol{\beta}^*, \sigma^2)$.

The cumulative probability of an observation falling in category j or below is then:

$$\gamma_{ij} = P(Y_i \leq j) = P(S_i \leq \theta_j^*) = P\left(Z_i \leq \frac{\theta_j^* - \alpha - \boldsymbol{x}_i^T \boldsymbol{\beta}^*}{\sigma}\right) = \Phi\left(\frac{\theta_j^* - \alpha - \boldsymbol{x}_i^T \boldsymbol{\beta}^*}{\sigma}\right) \tag{14}$$

where $Z_i = (S_i - \alpha - \boldsymbol{x}_i^T \boldsymbol{\beta}^*)/\sigma \sim N(0,1)$ and Φ is the standard normal CDF.

Since the absolute location and scale of the latent variable, α and σ respectively, are not identifiable from ordinal observations, an identifiable model is

$$\gamma_{ij} = \Phi(\theta_j - \boldsymbol{x}_i^T \boldsymbol{\beta}), \tag{15}$$

with identifiable parameter functions:

$$\theta_j = (\theta_j^* - \alpha)/\sigma \quad \text{and} \quad \boldsymbol{\beta} = \boldsymbol{\beta}^*/\sigma \ . \tag{16}$$

Observe how the minus in (15) entered naturally such that a positive β means a shift of the latent distribution in a positive direction.

Model (15) is exactly a cumulative link model with a probit link. Other distributional assumptions for S correspond to other link functions. In general assuming that the cumulative distribution funtion of S is F corresponds to assuming the link function is F^{-1}, cf. Table 2.

Some expositions of the latent variable motivation for cumulative link models get around the identifiability problem by introducing resctrictions on α and σ, usually $\alpha = 0$ and $\sigma = 1$ are chosen, which leads to the same definition of the threshold and regression parameters that we use here. However, it seems misleading to introduce restrictions on unidentifiable parameters. If observations really arize from a continuous latent variable, α and σ are real

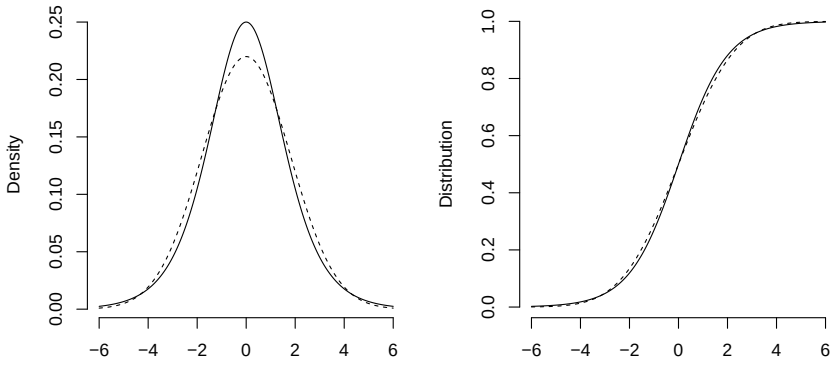

Figure 3: Left: densities. Right: distributions of logistic (solid) and normal (dashed) distributions with mean zero and variance $\pi^2/3$ which corresponds to the standard form for the logistic distribution.

unknown parameters and it makes little sense to restrict them to take certain values. This draws focus from the appropriate *relative* signal-to-ratio interpretation of the parameters evident from (16).

The standard form of the logistic distribution has mean zero and variance $\pi^2/3$. The logistic distribution is symmetric and shows a some resemblence with a normal distribution with the same mean and variance in the central part of the distribution; the tails of the logistic distribution are a little heavier than the tails of the normal distribution. In Fig. 3 the normal and logistic distributions are compared with variance $\pi^2/3$. Therefore, to a reasonable approximation, the parameters of logit and probit models are related in the following way:

$$\theta_j^{\text{probit}} \approx \theta_j^{\text{logit}}/(\pi/\sqrt{3}) \quad \text{and} \quad \boldsymbol{\beta}^{\text{probit}} \approx \boldsymbol{\beta}^{\text{logit}}/(\pi/\sqrt{3}) \;, \tag{17}$$

where $\pi/\sqrt{3} \approx 1.81$

Example 10: Considering once again the wine data the coefficients from logit and probit models with additive effects of temperature and contact are

```
> fm1 <- clm(rating ~ contact + temp, data = wine, link = "logit")
> fm2 <- clm(rating ~ contact + temp, data = wine, link = "probit")
> structure(rbind(coef(fm1), coef(fm2)), dimnames = list(c("logit",
     "probit"), names(coef(fm1))))

             1|2       2|3      3|4      4|5 contactyes tempwarm
logit  -1.3443834 1.2508088 3.466887 5.006404  1.5277977 2.503102
probit -0.7732627 0.7360215 2.044680 2.941345  0.8677435 1.499375
```

In comparison the approximate probit estimates using (17) are

```
> coef(fm1)/(pi/sqrt(3))

       1|2        2|3        3|4        4|5 contactyes   tempwarm
-0.7411974  0.6896070  1.9113949  2.7601753  0.8423190  1.3800325
```

These estimates are a great deal closer to the real probit estimates than the unscaled logit estimates. The average difference between the probit and approximate probit estimates being -0.079. □

2.6.1 More on parameter interpretation

Observe that the regression parameter in cumulative link models, cf. (16) are signal-to-noise ratios. This means that adding a covariate to a cumulative link model that reduces the residual noise in the corresponding latent model will increase the signal-to-noise ratios. Thus adding a covariate will (often) increase the coefficients of the other covariates in the cumulative link model. This is different from linear models, where (in orthogonal designs) adding a covariate does not alter the value of the other coefficients[11]. Bauer (2009), extending work by Winship and Mare (1984) suggests a way to rescale the coefficients such they are comparable in size during model development. See also Fielding (2004).

Example 11: Consider the estimate of `temp` in models for the wine data ignoring and controlling for `contact`, respectively:

```
> coef(clm(rating ~ temp, data = wine, link = "probit"))["tempwarm"]

tempwarm
 1.37229

> coef(clm(rating ~ temp + contact, data = wine, link = "probit"))["tempwarm"]

tempwarm
1.499375
```

and observe that the estimate of `temp` is larger when controlling for `contact`. In comparison the equivalent estimates in linear models are not affected—here we use the observed scores for illustration:

```
> coef(lm(as.numeric(rating) ~ temp, data = wine))["tempwarm"]

tempwarm
1.166667

> coef(lm(as.numeric(rating) ~ contact + temp, data = wine))["tempwarm"]

tempwarm
1.166667
```

In this case the coefficients are exactly identical, but in designs that are not orthogonal and observed studies with correlated covariates they will only be approximately the same. □

Regardless of how the threshold parameters discretize the scale of the latent variable, the regression paramaters $\boldsymbol{\beta}$ have the same interpretation. Thus $\boldsymbol{\beta}$ have the same meaning whether the ordinal variable is measured in, say, five or six categories. Further, the nature of the model interpretation will not chance if two or more categories are amalgamated, while parameter estimates will, of course, not be completely identical. This means that regression parameter estimates can be compared (to the extent that the noise level is the same) across studies where response scales with a different number of response categories are adopted. In comparison, for linear models used on scores, it is not so simple to just combine two scores, and parameter estimates from different linear models are not directly comparable.

If the latent variable, S_i is approximated by scores assigned to the response variable, denote this variable Y_i^*, then a linear model for Y_i^* can provide approximate estimates of $\boldsymbol{\beta}$ by applying (16) for cumulative probit models[12]. The quality of the estimates rest on a number

[11] but the same thing happens in other generalized linear models, e.g., binomial and Poisson models, where the variance is determined by the mean.

[12] these approximate regression parameters could be used as starting values for an iterative algorithm to find the ML estimates of $\boldsymbol{\beta}$, but we have not found it worth the trouble in our Newton algorithm

of aspects:

- The scores assigned to the ordinal response variable should be structurally equivalent to the thresholds, $\boldsymbol{\theta}^*$ that generate Y_i from S_i. In particular, if the (equidistant) numbers $1, \ldots, J$ are the scores assigned to the response categories, the thresholds, $\boldsymbol{\theta}^*$ are also assumed to be equidistant.
- The distribution of Y_i^* should not deviate too much from a bell-shaped curve; especially there should not be too many observations in the end categories
- By appeal to the central limit theorem the coarsening of S_i into Y_i^* will "average out" such that bias due to coarsening is probably small.

This approximate estimation scheme extends to other latent variable distributions than the normal where linear models are exchanged with the appropriate location-scale models, cf. Table 2.

Example 12: Consider the following linear model for the rating scores of the wine data, cf. Table 1:

$$Y_i^* = \alpha + \beta_1 \texttt{temp}_i + \beta_2 \texttt{contact}_i + \varepsilon_i \quad \varepsilon_i \ \sim N(0, \sigma_\varepsilon^2)$$

The relative parameter estimates, $\tilde{\beta}$ are

```
> lm1 <- lm(as.numeric(rating) ~ contact + temp, data = wine)
> sd.lm1 <- summary(lm1)$sigma
> coef(lm1)[-1]/sd.lm1
```

```
contactyes   tempwarm
  0.791107   1.384437
```

which should be compared with the estimates from the corresponding cumulative probit model:

```
> fm1 <- clm(rating ~ contact + temp, data = wine, link = "probit")
> coef(fm1)[-(1:4)]
```

```
contactyes   tempwarm
 0.8677435  1.4993746
```

The relative estimates from the linear model are a lower than the cumulative probit estimates, which is a consequence of the fact that the assumptions for the linear model are not fulfilled. In particular the distance between the thresholds is not equidistant:

```
> diff(coef(fm1)[1:4])
```

```
      2|3       3|4       4|5
1.5092842 1.3086590 0.8966645
```

while the distribution is probably sufficiently bell-shaped, cf. Fig 4.

□

2.7 Structured thresholds

In this section we will motivate and describe structures on the thresholds in cumulative link models. Three options are available in `clm` using the `threshold` argument: flexible, symmetric and equidistant thresholds. The default option is `"flexible"`, which corresponds

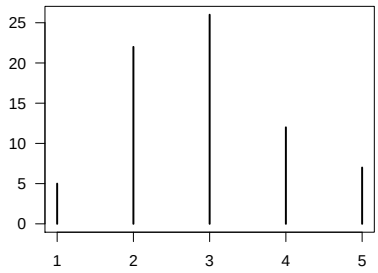

Figure 4: Histogram of the ratings in the wine data, cf. Table 1.

Table 6: Symmetric thresholds with six response categories use the three parameters a, b and c.

θ_1	θ_2	θ_3	θ_4	θ_5
$-b+c$	$-a+c$	c	$a+c$	$b+c$

to the conventional ordered, but otherwise unstructured thresholds. The `"symmetric"` option restricts the thresholds to be symmetric while the `"equidistant"` option restricts the thresholds to be equally spaced.

2.7.1 Symmetric thresholds

The basic cumulative link model assumed that the thresholds are constant for all values of $\boldsymbol{x}_i^T\boldsymbol{\beta}$, that they are ordered and finite but otherwise without structure. In questionaire type response scales, the question is often of the form "how much do you agree with *statement*" with response categories ranging from "completely agree" to "completely disagree" in addition to a number of intermediate categories possibly with appropriate ancoring words. In this situation the response scale is meant to be perceived as being symmetric, thus, for example, the end categories are equally far from the central category/categories. Thus, in the analysis of such data it can be relevant to restrict the thresholds to be symmetric or at least test the hypothesis of symmetric thresholds against the more general alternative requiring only that the thresholds are ordered in the conventional cumulative link model. An example with six response categories and five thresholds is given in Table 6 where the central threshold, θ_3 maps to c while a and b are *spacing*s determining the distance to the remaining thresholds. Symmetric thresholds is a parsimonious alternative since three rather than five parameters are required to determine the thresholds in this case. Naturally at least four response categories, i.e., three thresholds are required for the symmetric thresholds to use less parameters than the general alternative. With an even number of thresholds, we use a parameterization with two central thresholds as shown in Table 7.

Example 13: I am missing some good data to use here. □

Table 7: Symmetric thresholds with seven response categories use the four parameters, a, b, c and d.

θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
$-b+c$	$-a+c$	c	d	$a+d$	$b+d$

2.7.2 Equidistant thresholds

Ordinal data sometimes arize when the intensity of some perception is rated on an ordinal response scale. An example of such a scale is the ratings of the bitterness of wine described in example 1. In such cases it is natural to hypothesize that the thresholds are equally spaced, or equidistant as we shall denote this structure. Equidistant thresholds use only two parameters and our parameterization can be described by the following mapping:

$$\theta_j = a + b(j-1), \quad \text{for} \quad j = 1, \ldots, J-1 \tag{18}$$

such that $\theta_1 = a$ is the first threshold and b denotes the distance between adjacent thresholds.

Example 14: In example 3 we fitted a model for the wine data (cf. Table 1) with additive effects of temperature and contact while only restricting the thresholds to be suitably ordered. For convenience this model fit is repeated here:

```
> fm1 <- clm(rating ~ temp + contact, data = wine)
> 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 7(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 succesive distances between the thresholds in this model are

```
> diff(fm1$alpha)

     2|3      3|4      4|5
2.595192 2.216078 1.539517
```

so the distance between the thresholds seems to be decreasing. However, the standard errors of the thresholds are about half the size of the distances, so their position is not that well determined. A model where the thresholds are restricted to be equally spaced is fitted with

```
> fm2 <- clm(rating ~ temp + contact, data = wine, threshold = "equidistant")
> summary(fm2)
```

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

 link  threshold   nobs logLik AIC    niter max.grad cond.H
 logit equidistant 72   -87.86 183.73 6(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
```

so here $\hat{\theta}_1 = \hat{a} = -1.001$ and $\hat{b} = 2.123$ in the parameterization of (18). We can test the assumption of equidistant thresholds against the flexible alternative with a likelihood ratio test:

```
> anova(fm1, fm2)

Likelihood ratio tests of cumulative link models:

  formula:                link: threshold:
1 rating ~ temp + contact logit equidistant
2 rating ~ temp + contact logit flexible
  no.par    AIC  logLik LR.stat df Pr(>Chisq)
1      4 183.73 -87.865
2      6 184.98 -86.492  2.7454  2     0.2534
```

so the p-value is $p = 0.253$ not providing much evidence against equidistant thresholds. □

2.8 *Matrix representation of cumulative link models

Just as a linear model can be expressed as $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, cumulative link models can also be expressed in terms of matrices. The matrix representation of cumualtive link models is a little more complicated than it is for linear models and (univariate) generalized linear models because of the special role of the threshold parameters.

The matrix representation of cumulative link models is important in order to efficiently evaluate the log-likelihood and the first and second derivatives of the log-likelihood function with respect to the parameters. These are important in order to efficiently optimize the likelihood function when fitting cumulative link models.

3 Maximum likelihood estimation of cumulative link models

We will distinguish between cumulative link models of the form (4) and more involved models with at least one of the following complications: random effects, scale effects, nominal effects

and flexible link functions. Structured thresholds are allowed in first class of models, which we denote standard cumulative link models.

In the optimization literature it is conventional to *minimize* an objective function rather than to *maximize* it. As our objective function in this section we therefore adopt the *negative* log-likelihood function.

For standard cumulative link models, we will describe a Newton-Raphson algorithm. While for the general cumulative link models we will use a general purpose optimizer of the quasi-Newton type. General purpose optimizers can be used for standard cumulative link models, but the Newton algorithm has some advantages here.

3.1 A Newton algorithm for standard cumulative link models

Let $\boldsymbol{\psi}^{(i)}$ denote the vector of parameters at iteration i. The Newton-Raphson procedure works by computing

$$\boldsymbol{\psi}^{(i+1)} = \boldsymbol{\psi}^{(i)} - \boldsymbol{H}(\boldsymbol{\psi}^{(i)}; \boldsymbol{y})^{-1} \boldsymbol{g}(\boldsymbol{\psi}^{(i)}; \boldsymbol{y})$$

where $\boldsymbol{H}(\boldsymbol{\psi}^{(i)}; \boldsymbol{y}) = -\ell''_{\psi\psi}(\boldsymbol{\psi}; \boldsymbol{y})$ is the Hessian, i.e. the second order derivative of the negative log-likelihood function with respect to the parameters, $\boldsymbol{\psi}^{(i)}$ and $\boldsymbol{g}(\boldsymbol{\psi}^{(i)}; \boldsymbol{y}) = -\ell'_{\psi}(\boldsymbol{\psi}; \boldsymbol{y})$ is the gradient. Each new parameter vector $\boldsymbol{\psi}^{(i+1)}$ is obtained by computing the step; $\boldsymbol{\Lambda}^{(i)} = \boldsymbol{H}(\boldsymbol{\psi}^{(i)}; \boldsymbol{y})^{-1} \boldsymbol{g}(\boldsymbol{\psi}^{(i)}; \boldsymbol{y})$ with the Hessian and gradient evaluated the current parameters; $\boldsymbol{\psi}^{(i)}$. To evaluate the log-likelihood, the gradient and Hessian efficiently we express the cumulative link model in the following matrix notation:

The Newton algorithm can be motivated by approximating the objective function with a second order Taylor approximation and then solving this.

3.2 Motivation for the Newton algorithm for standard cumulative link models

The convergence rate of a Newton algorithm is said to be quadratic, which is the best we can hope for. This means that the Newton algorithm has to take fewer steps than many other algorithms to obtain the same accuracy in the estimates. The quadratic convergence rate is attained when the shape of the objective function is close to a parabola, i.e., the objective function behaves as a quadratic function. This is often the case for nicely behaved likelihood functions, especially near the optimum.

The speedyness of an optimization algorithm not only depends on the convergence rate of the algorithm, but also on how much computation has to be done at each iteration. This is exactly the reason that the Newton algorithm is not always the fastest algorithm even though it has the best convergence rate. For instance, the Newton algorithm requires that both gradient and Hessian are computed at each iteration. In a quasi Newton algorithm only the gradient needs to be computed while an internal approximation of the Hessian is updated at each iteration. If evaluation of the Hessian requires many computations, a quasi Newton algorithm may well be faster than a genuine Newton algorithm.

In standard cumulative link models both gradient and Hessians can be computed relatively easily and with a moderate amount of computations.

A general problem with the Newton algorithm is that even though the Newton step[13] may be in the right direction, it can be too large to actually cause a reduction in the objective function. The algorithm is said to over-shoot. An effective modification is to successively half the length of the step until the new coefficients cause a reduction in the objective function. This is adopted in the the ordinal2 package.

In cumulative link models a full Newton step may cause the ordering of the threshold coefficients to be changed. In this case we define the negative log-likelihood to be infinity. This causes the algorithm to successively half the step length and eventually the step is small enough that the new threshold coefficients have the right ordering and the negative log-likelihood is reduced.

As mentioned in section 2.4, Pratt (1981) and Burridge (1981) showed (seemingly independent of each other) that the log-likelihood function of cumulative link models with the link functions considered in Table 2, except for the cauchit link, is concave. This means that there is a unique global optimum so there is no risk of convergence to a local optimum. It also means that the step of a Newton-Raphson algorithm is quarantied to be in the direction of a higher likelihood.

In conclusion the Newton algorithm works well for cumulative link models because 1) the log-likelihood function is approximately quadratic, 2) the Hessian does not require too many computations and 3) the Newton step is guarantied to be in the right direction.

4 Assessing the likelihood and model convergence

Write section example based

Closed form expressions for the ML estimates of cumulative link models do not exist, so iterative algorithms have to be used to fit cumulative link models. Such algorithms usually produce a sequence of coefficients that get closer and closer to the true coefficients such that each new set of coefficients attain a higher log-likelihood than the previous set. At some point the algorithm terminates and returns the coefficients.

A rather general requirement is that the numerical error in the parameter estimates should be small relative to their statistical uncertainty. Also, for likelihood ratio tests, we want the log-likelihood for the models involved to be determined precise enough that likelihood ratio tests of nested models are accurate.

In this section we will consider how to quantify the numerical error in parameter estimates, in standard errors and in the log-likelihood.

Models are used for different purposes with different requirements to the degree of accuracy. We will discuss which requirements will be reasonable and attainable in different settings.

Before we move on, lets introduce some terminology: We refer to a *coefficient* as a parameter estimate and use *true coefficient* to denote the correct and exact coefficient or ML parameter estimate. Thus a true coefficient is not the same as a true parameter. Estimating a coefficient accurately means that the difference between the coefficient estimate and the true coefficient is ssmall, i.e., that the numerical error in the coefficient estimate is small. On the other hand, if a parameter is estimated accurately, it means that the statistical uncertainty of that parameter (for instance measured by its standard error) is small.

[13] the step is a vector; there is a step for each parameter.

We distinguish between convergence and termination. Termination is the point at which the optimizer stops iterating. Successful termination is a binary variable: either the optimizer terminated without errors or it didn't. A model has converged if the coefficients are close enough to the true coefficients. An optimizer can terminate with a model fit that has not converged.

4.1 What is convergence?

In finite computer arithmetic with double precision, numbers are only be represented with around 16 digits. This means that we cannot hope to know model coefficients or the log-likelihood to any higher precision than that. Hoever, even this limit is optimistic due to the numerical errors that occur when adding and multiplying numbers with finite precision.

But who needs to know model coefficients more accurately than with, say, six digits? The real problem here is not the coefficients are *correct* or not, but whether they are *sufficiently accurate*. The same consideration applies to the log-likelihood and standard errors.

A basic problem is that we want to estimate the coefficients with high enough precision, but we also do not want to perform unnessecary computations, so we want the optimization to stop when the estimates are accurate enough. This means that the optimization does not (necessarily) stop due to lack of progress, but because it is instructed not ro proceed from this point.

We can only quantify the distance from our current coefficient values to the true coefficients if we have more precise coefficients. If the optimizer stops from lack of progress, it is hard to obtain more precise coefficients. If the optimizer stops because the termination criteria are met, then we can tighten the termination criteria and ask the optimizer to proceed with additional iterations.

Example 15: In `clm` the termination criterion is the maximum absolute gradient, `gradTol` with default value `1e-6`. If we increase this value the optimizer will take fewer steps:

```
> fm1 <- clm(rating ~ contact + temp, data = wine, gradTol = 0.001)
> fm1$niter

outer inner
    6     0
```

If we decrease `gradTol` the optimizer will take additional steps:

```
> fm2 <- clm(rating ~ contact + temp, data = wine, gradTol = 1e-12)
> fm2$niter

outer inner
    8     0
```

The difference here. □

The likelihood function of a cumulative link models approximately has the shape of a downward facing parabola. The closer to the optimum the closer the shape. Looking at the likelihood function, it is clear that for a coefficient to be close to the true coefficient, we must require that the gradient, i.e., the tangent line to the likelihood function at the coefficient is close to zero. A small gradient is also satisfied at saddle points, so we must also require that the likelihood function is downward facing. This is satisfied if the negative Hessian is positive definit, which means that all the eigen values of the negative Hessian should

be positive. These requirements are generally known as the KKT conditions. While this ensures that *a* maximum has been reached, they do not say whether a local optimum has been reached rather than the global optimum. However, X and Y showed that the likelihood function is concave if the parameters are identified, so there can be no local optimum that is also not the global optimum.

The Newton (-Raphson) optimization algorithm is a very speedy algorithm for concave and approximately quadratic objective functions. The rate of convergence is said to be quadratic, which, in terms of the coefficients, means that the number of correct digits roughly doubles with each iteration. This behavior is generally seen when the first couple of digits have been correctly determined. Thus, when the

4.2 Assessment of model convergence

Once we have fitted a cumulative link model we may ask accurate the parameter estimates returned by `clm` are. To assess convergence we can use the the `convergence` function, but for a brief look, we already get some initial information from the summary. Consider, for example the following model for the wine data that we already considered in example ZZ:

```
> fm1 <- clm(rating ~ contact, dat = wine)
> summary(fm1)

formula: rating ~ contact
data:    wine

 link  threshold nobs logLik AIC    niter max.grad cond.H
 logit flexible  72   -99.96 209.91 6(0)  1.67e-07 1.7e+01

Coefficients:
           Estimate Std. Error z value Pr(>|z|)
contactyes   1.2070     0.4499   2.683   0.0073 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Threshold coefficients:
    Estimate Std. Error z value
1|2 -2.13933    0.48981  -4.368
2|3  0.04257    0.32063   0.133
3|4  1.71449    0.38637   4.437
4|5  2.97875    0.50207   5.933
```

The last three entries of the first table displays information about the estimation of the model. The `niter` entry displays the number of iterations of the Newton-Raphson algorithm used to estimate the model with the number of iterations in the inner loop where step-halfings take place in parentheses. In this case seven iterations were taken from the starting values and the full Newton step was taken in all iterations. The starting values are available as `fm1$start`. We are also informed that the maximum absolute gradient is 1.67e-07. Since the gradient is small, the optimization terminated at a so-called *stationary point*. The `cond.H` entry of the table contains the *condition number* of the Hessian of the negative log-likelihood. Since this number is positive, we know that the point at which the optimization terminated is not only a stationary point, it is also a maximum of the log-likelihood function. In fact, it is also a global maximum and not just a local optimum because the log-likelihood function

is concave as described in section 2.4.

The size of the condition number of the Hessian describes how well defined the model is. If this number is large, say, larger than 10^4 or 10^6, the model is ill-defined. If the condition number is large, it means that the log-likelihood function is flat in some direction indicating that the model is over parameterized and can be simplified with only a small reduction in likelihood. Not only does it have consequences for model interpretation, it also means that the likelihood function can be difficult to optimize. For more detailed information about the condition number see section XX.

We can obtain more detailed information about the convergence properties of the model fit returned by `clm` with the `convergence` function:

```
> convergence(fm1)

 nobs logLik niter max.grad cond.H
 72   -99.96 6(0)  1.67e-07 1.7e+01

           Estimate Std. Error  Gradient      Dist Rel. Dist
1|2        -2.13933     0.4898  1.67e-07  3.12e-08 -1.46e-08
2|3         0.04257     0.3206 -1.21e-07 -1.96e-09 -4.61e-08
3|4         1.71449     0.3864 -1.41e-09 -1.69e-09 -9.86e-10
4|5         2.97875     0.5021 -7.90e-11 -1.66e-09 -5.59e-10
contactyes  1.20695     0.4499 -9.27e-09 -1.74e-09 -1.44e-09

Eigen values of Hessian:
29.895 19.524 10.911  4.942  1.761
```

Some of the information from the `summary` is repeated here for convenience. The new thing is the table that in addition to the coefficients and their standard errors also contain three new columns. The first new column gives the gradients of the likelihood function with respect to the parameters at the optimum. These should all be small. The most important column is probably the one denoted `Abs. Dist` which contains the absolute distance from the coefficients to the true, or exact ML parameter estimates. Thus this column quantifies the accuracy with which the coefficients returned by `clm` are determined. In this case we see that the absolute distances are all less than 10^{-7}, so the coefficients reported by `clm` are accurate to at least seven digits: more than enough for all practical purposes.

Naturally, the `convergence` function does not by some magic method know the true coefficients, but it can determine an accurate approximation to them. The consequence is that the `Abs. Dist` values are only accurate when the fit is close to the optimum anyway, say when all `Abs. Dist` are less than 10^{-2}.

The `Rel. Dist` column gives the relative numerical errors in the coefficients, i.e., `Estimate / Abs. Dist`.

The `Dist` values are actually the values of the Newton step at convergence. Close to the optimum, the number of correct digits in the parameter estimates will approximately double at each iteration, so with already, say, four digits correctly determined, the Newton step equals the distance to the optimum to approximately four correct digits.

When reporting parameter estimates, the number of significant digits that we use should reflect the uncertainty in the parameter estimates. It makes little sense to report a parameter estimate with several digits, if the standard error is around the same size as the paramter estimate; this only gives a false impression of precision.

Rarely will the statistical uncertainty in coefficients be so small that we need more than a few digits of numerical precision in the coefficients. However, standard errors that reflect the statistical precision are evaluated at the current coefficients. Thus if there is numerical error in the coefficients, the standard errors may be inaccurate as well. Therefore we need the coefficients to be determined with small enough numerical error, that the standard errors are determined with good enough precision.

To further substantiate the determination of the precision in the parameter estimates, we will reconsider the model from above, but we will relax the termination criteria to get less precise estimate for the sake of illustration. The termination criteria in the Newton algorithm is the size of the maximum absolute gradient, `gradTol`. The default is to terminate if the maximum absolute gradient is less than 10^{-6}. Here we are satisfied if the maximum absolute gradient is 10^{-2}:

```
> fm2 <- clm(rating ~ contact, dat = wine, gradTol = 0.01)
> convergence(fm2)

 nobs logLik niter max.grad cond.H
 72   -99.96 5(0)  1.26e-03 1.7e+01

           Estimate Std. Error  Gradient      Dist Rel. Dist
1|2        -2.13910     0.4898  1.26e-03  2.33e-04 -1.09e-04
2|3         0.04256     0.3206 -7.50e-04 -1.47e-05 -3.45e-04
3|4         1.71446     0.3864 -1.69e-04 -2.73e-05 -1.59e-05
4|5         2.97872     0.5021 -5.00e-06 -2.66e-05 -8.94e-06
contactyes  1.20693     0.4499 -9.72e-05 -2.47e-05 -2.04e-05

Eigen values of Hessian:
29.896 19.525 10.911  4.943  1.761
```

the maximum absolute numerical error in the coefficients is around 2.3e-4. These numerical errors were obtained by

```
> (approx.error <- with(fm2, solve(Hessian, gradient)))

          1|2           2|3           3|4           4|5    contactyes
 2.334389e-04 -1.467784e-05 -2.732915e-05 -2.663627e-05 -2.466396e-05
```

so they are based on information already in the model object; no additional evaluations of gradient or Hessian are required. To verify these numbers are indeed the numerical errors in the estimates (to a good approximation), we refit the model asking for a very high degree of accuracy:

```
> fm3 <- clm(rating ~ contact, dat = wine, gradTol = 1e-14)
```

The numerical error in the estimates from fm2 compared to those from fm3 are:

```
> (true.error <- coef(fm2) - coef(fm3))

          1|2           2|3           3|4           4|5    contactyes
 2.334702e-04 -1.467980e-05 -2.733084e-05 -2.663793e-05 -2.466570e-05
```

which are all very close to our approximate estimates of the numerical accuracy. The absolute and relative error of our approximate error are:

```
> true.error - approx.error

          1|2           2|3           3|4           4|5    contactyes
 3.124015e-08 -1.964010e-09 -1.691223e-09 -1.664675e-09 -1.739181e-09
```

```
> (true.error - approx.error)/true.error
```

```
         1|2          2|3          3|4          4|5   contactyes
1.338079e-04 1.337899e-04 6.187964e-05 6.249265e-05 7.051012e-05
```

Thus the approximate errors are accurate to 3-4 significant digits while the absolute error or the approximate errors are all just very small. From the following we see that the more accurate model took two additional iterations:

```
> rbind(fm2$niter, fm3$niter)
```

```
     outer inner
[1,]     5     0
[2,]     7     0
```

We can also compare the values of the log-likelihood to see how accurately it is determined:

```
> (true.ll.err <- fm3$logLik - fm2$logLik)
```

```
[1] 1.55584e-07
```

so the log likelihood is determined accurately to six digits. If we correct the coefficients with the approximate numerical errors, then we can determine an approximate error in the value of the log-likelihood function:

```
> env2 <- update(fm2, doFit = FALSE)
> env2$par <- coef(fm2) - approx.error
> new.logLik <- -ordinal2:::clm.nll(env2)
> (approx.ll.err <- fm2$logLik - new.logLik)
```

```
[1] -1.55584e-07
```

This approximate error is identical to the true error above to the printed accuracy. In fact the error in our new approximate log-likelihood is

```
> fm3$logLik - new.logLik
```

```
[1] 0
```

so these two numbers are exactly identical in double precision:

```
> print(fm3$logLik, digits = 15)
```

```
[1] -99.9559109316376
```

```
> print(new.logLik, digits = 15)
```

```
[1] -99.9559109316376
```

The error in the standard errors are a little smaller (in absolute values) compared to the errors in the coefficients. Unfortunately they are all a little too small:

```
> coef(summary(fm2))[, 2] - coef(summary(fm3))[, 2]
```

```
          1|2           2|3           3|4           4|5    contactyes
-5.029645e-05 -4.757830e-06 -4.656283e-06 -3.608865e-06 -4.840441e-06
```

4.3 The slice method

Cumulative link models are non-linear models and in general the likelihood function non-linear models is not garantied to be uni-modal or nice and bell-shaped. In cumulative link

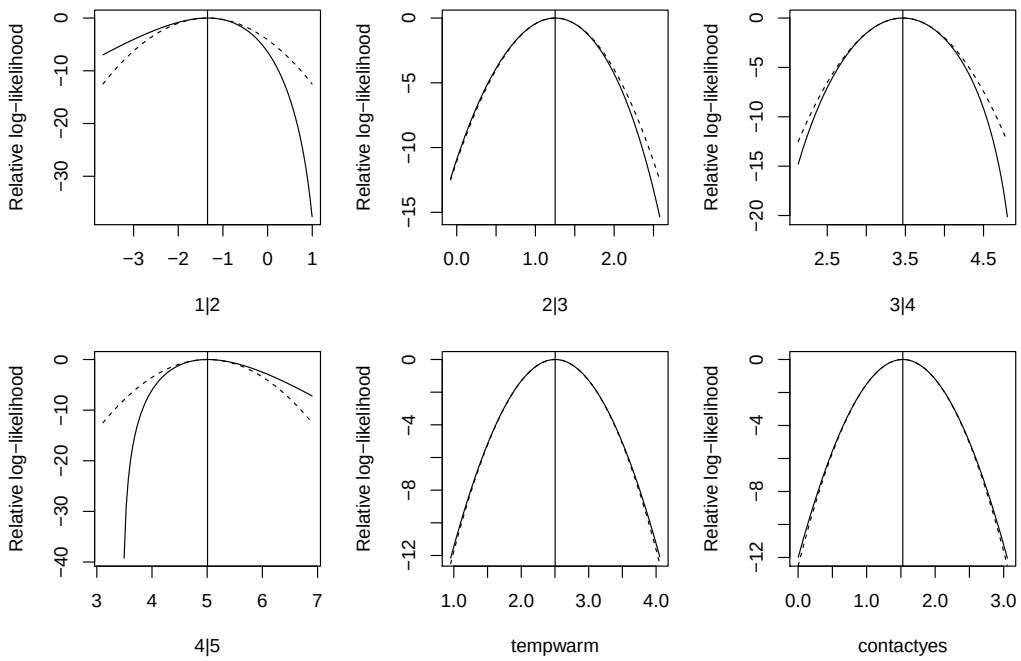

Figure 5: Slices of the (negative) log-likelihood function for parameters in a model for the bitterness-of-wine data. Dashed lines indicate quadratic approximations to the log-likelihood function and vertical bars the indicate maximum likelihood estimates.

models, the threshold parameters are even restricted to be ordered and therefore naturally bounded. It can therefore be interesting to visualize the likelihood function in the neighborhood of the optimum. To do this we use the `slice` function. As the name implies, it extracts a (one-dimensional) slice of the likelihood function:

```
> fm1 <- clm(rating ~ temp + contact, data = wine)
> slice.fm1 <- slice(fm1, lambda = 5)
> par(mfrow = c(2, 3))
> plot(slice.fm1)
```

The result is shown in Fig. 5. `lambda` controls the range of parameter values for which the slice is computed. The square root of the diagonal elements of the Hessian at the optimum are measures of curvature in the likelihood functions for each of the parameters, and `lambda` indicates how many of these curvature units away from the optimum the slice is to be computed. Working in curvature units is a way to standardize the parameter range even if some parameters are much better determined (much less curvature in the log-likelihood) than others. Here we wanted to assess the shape of the log-likelihood function, so we chose a relatively large value for `lambda`. By default the quadratic approximation is included for reference in the plot.

For this model we see that the log-likelihood function is nicely quadratic for the regression parameters while it is less so for the threshold parameters and particularly bad for the end thresholds. There also appears to be only one optimum.

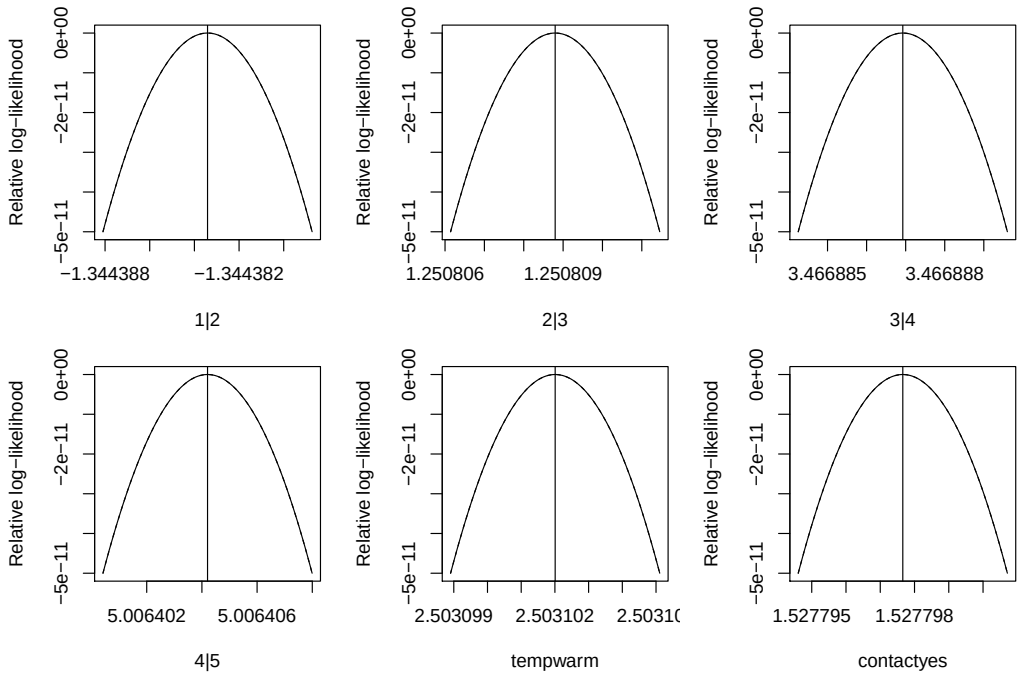

Figure 6: Slices of the (negative) log-likelihood function for parameters in a model for the bitterness-of-wine data very close to the MLEs. Dashed lines indicate quadratic approximations to the log-likelihood function and vertical bars the indicate maximum likelihood estimates.

From Fig. 5 it seems that the parameter estimates as indicated by the vertical bars are close to the optimum indicating successful model convergence. To investigate more closely we slice the likelihood at a much smaller scale:

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

The resulting figure is shown in Fig. 6. Observe that 1) the model has converged and all parameters estimates are correct to at least six decimals and 2) the quadratic approximation is indistinguishable from the log-likelihood at this scale.

Unfortunately there is no general way to infer confidence intervals from the likelihood slices—for that we have to use the computationally more intensive profile likelihoods. Compared to the profile likelihoods discussed in section 5, the slice is much less computationally demanding since the likelihood function is only evaluated—not optimized, at a range of parameter values.

5 Confidence intervals and profile likelihood

write section with examples

Confidence intervals are convenient for summarizing the uncertainty about estimated parameters. The classical symmetric estimates given by $\hat{\beta} \pm z_{1-\alpha/2}\hat{\text{se}}(\hat{\beta})$ are based on the

Wald statistic[14], $w(\beta) = (\hat{\beta} - \beta)/\hat{\text{se}}(\hat{\beta})$ and available by:

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

```
               2.5 %   97.5 %
tempwarm   1.4669081 3.539296
contactyes 0.5936345 2.461961
```

A similar result could be obtained by `confint.default(fm1)`. However, outside linear models asymmetric confidence intervals often better reflect the uncertainty in the parameter estimates. More accurate, and generally asymmetric, confidence intervals can be obtained by using the likelihood root statistic instead; this relies on the so-called profile likelihood written here for an arbitrary scalar parameter β_a:

$$\ell_p(\beta_a; \boldsymbol{y}) = \arg\max_{\boldsymbol{\theta}, \boldsymbol{\beta}_{-a}} \ell(\boldsymbol{\theta}, \boldsymbol{\beta}; \boldsymbol{y}) \ ,$$

where $\boldsymbol{\beta}_{-a}$ is the vector of regression parameters without the ath one. In words, the profile log-likelihood for β_a is given as the full log-likelihood optimized over all parameters but β_a. To obtain a smooth function, the likelihood is optimized over a range of values of β_a around the ML estimate, $\hat{\beta}_a$, further, these points are interpolated by a spline to provide an even smoother function.

The likelihood root statistic [refs] is defined as:

should the minus be there?

$$r(\beta_a) = \text{sign}(\hat{\beta}_a - \beta_a)\sqrt{-2[\ell(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\beta}}; \boldsymbol{y}) - \ell_p(\beta_a; \boldsymbol{y})]}$$

and just like the Wald statistic its reference distribution is the standard normal. Confidence intervals based on the likelihood root statistic are defined as those values of β_a for which $r(\beta_a)$ is in between, say, -1.96 and 1.96 for 95% confidence intervals. Formally the confidence intervals are defined as

$$CI: \ \left\{\beta_a; |r(\beta_a)| < z_{1-\alpha/2}\right\} \ .$$

To fix ideas the likelihood root statistic[15] is shown for the `tempwarm` parameter in the model for the wine data given above in Fig. 7. The figure is made with the following code:

```
> pr1 <- profile(fm1, which.beta = "tempwarm")
> plot(pr1, root = TRUE)
```

The horizotal lines indicate 95% and 99% confidence intervals and the dashed line is the Wald statistic. Incidently, and as is clear from the figure, the Wald statistic is also the tangient line to the likelihood root statistic in the ML estimate. As indicated by the figure, the profile likelihood confidence limits for `tempwarm` are larger than the Wald counterparts and the discrepancy increases with the confidence level.

The profile likelihood confidence intervals are provided by default application of `confint`:

```
> confint(fm1)
```

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

The visualization of the likelihood root statistic can be helpful in diagnozing non-linearity in the parameterization of the model. The linear scale is particularly suited for this rather than

[14] where $z_{1-\alpha/2}$ is the $(1-\alpha/2)$-quantile of the standard normal CDF.

[15] actually we reversed the sign of the statistic in the display since a line from lower-left to upper-right looks better than a line from upper-left to lower-right.

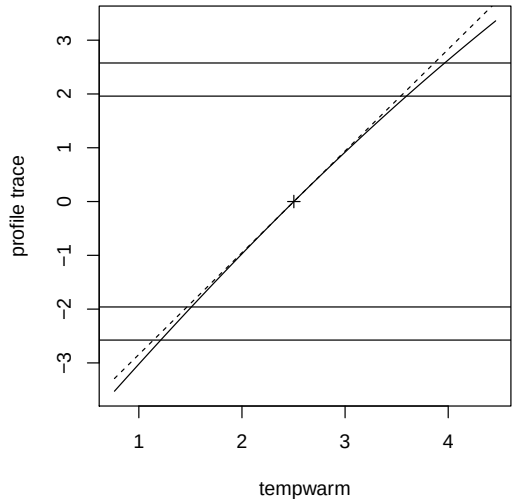

Figure 7: Likelihood root statistic (solid) and Wald statistic (dashed) for the `tempwarm` parameter in the model for the wine data.

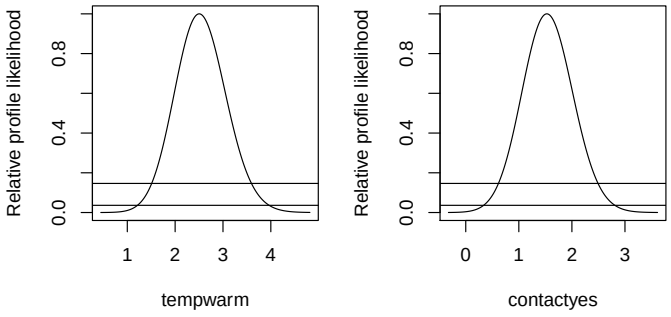

Figure 8: Relative profile likelihoods for the regression parameters in the Wine study.

other scales, such as the quadratic scale at which the log-likelihood lives. In summarizing the results of a models fit, I find the relative likelihood scale, $\exp(-r(\beta_a)^2/2)$ informative. These relative profile likelihoods are obtained with

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

and provided in Fig. 8. The evidence about the parameters is directly visible; the ML estimate has maximum support, 1 and values away from here are less supported by the data, 95% and 99% confidence intervals are readily read of the plots as intersections with the horizontal lines. Most importantly the plots emphasize that a range of parameter values are actually quite well supported by the data—something which is easy to forget when focus is on the precise numbers of the ML estimates.

6 Cumulative Link Mixed Models

The effects of explanatory variables in the cumulative link models we have considered until now are said to be *fixed effects*. In this chapter we will introduce so-called *random effects* which combined with fixed effects leads to the so-called *mixed effects models*, or simply *mixed models*. The motivations for using random effects are many and diverse. Often some kind of grouping structure is present in the data and random effects are introduced to model this grouping structure. Grouped data can be understood as a *samples of samples*, where the samples as well as the samples-of-samples are considered random.

As an example consider a study of consumer preference of two products, A and B, say. Consumers can be considered a random sample from the population of consumers to which our results are to generalize, while we are not intested in generalizing the effect of products to the population; only these two products are of interest. This is a general theme: when we wish our results or inference for a particular explanatory variable to generalize to a population, we consider the effects of this variable random; otherwise the effects are usually considered fixed. Also observe that we obtained a *sample* of consumers which each provided a *sample* of preference ratings, thus we have a sample of samples.

Another characteristic is that samples from the same consumer are more similar than samples from different consumers (on average) since samples from the same consumer share a random consumer effect. Observations on the same consumer are said to be correlated.

A basic cumulative link mixed model reads

$$\gamma_{ijk} = F(\theta_j - \boldsymbol{x}_{ik}^T\boldsymbol{\beta} - u_k), \tag{19}$$

where u_k is the random effect for the kth group or cluster, x_{ik} are the explanatory variables for the ith sample on the kth cluster and $\boldsymbol{\beta}$ are the parameters for the fixed effects. Observe the minus before the random effects, so they have the same direction of effect as $\boldsymbol{\beta}$, cf. section 2.6.

We will assume throughout that the random effects are independently and identically normally distributed:

$$U_k \sim N(0, \sigma_u^2), \tag{20}$$

where u_k are the observed values of U_k. Though other distributions are possible, this choice is conventional and convenient and often a reasonable choice, at least as a first approximation. The ordinal observations are assumed to follow a multinomial distribution conditional on $\boldsymbol{X}$ and the observed value of the random effects $U_k = u_k$ for all k.

The random effects in (19) are said to work on the intercepts, $\{\theta_j\}$ since $\theta_j - u_k$ is a k-specific shift of the thresholds or intercepts. In a latent distribution interpretation, the u_k induce a cluster-specific shift of the latent distributions relative to the thresholds.

In (19) the $\{u_k\}$ are not parameters as are $\boldsymbol{\beta}$, rather, they are realized values of the random variable, U_k. The values, $\{u_k\}$ are unobservable though it is possible to construct predictions of them based on the fitted model. The parameter that determines the distribution of u_k is σ_u, so in (19) random effects are introduced at the expense of a single parameter, though there can be thousands or more random effects. Sometimes an alternative to the mixed model is to allow the grouping variable to have fixed effects—this is possible, in particular, when the there are many cluster-specific samples relative to the number of clusters. One problem by doing so is that the number of parameters increase with the number of clusters

and this cause the ML estimators of the model parameters to be inconsistent. On the other hand the estimators are consistent in the mixed model since the number of parameters is constant with the increase of the number of clusters.

Example 16: In this example we will revisit the wine data presented in example 1 and Table 1. In this example we will extend previous analyses of these data by considering the effect of judges. Nine judges made each their ratings of the wines and it is possible that judges perceived and used the response scale differently and therefore induced judge-specific structural differences in the wine ratings.

Judges are effectively used as the measurement instrument with which wine bitterness is measured and a general challenge is that as measurement instruments, humans are very hard to calibrate. Further, the concepts measured in this way are often not possible to quantify; in addition to perceptions, attitude, agreement and preference are other concepts without unambiguous and meaningful continuous measurement scales. A certain degree of bitternes is not uniquely related to any category on the rating scale, further *bitterness* is a personal perception that vary between individuals. Possibly the simplest type of judge-specific differences arize if judges perception of bitterness are shifted relative to each other on the response scale. Thus if one judge rates two wines as 2 and 4, another judge may rate them as 3 and 5 because that judge perceives the bitterness as more intense than the first judge or because the judge perceives the response scale different from the first judge.

We could model the judge-specific shifts of bitterness perception by including a fixed effect of judges in a cumulative link model. However, we are not interested in the judge-specific effects per say, rather we want to control for these effects in the interpretation of the other effects.

The judge-specific shifts of bitterness perception can be modeled with cumulative link mixed model with the following structure in the linear predictor:

$$\eta_{ij} = \theta_j + \beta_1(\texttt{temp}_i) + \beta_2(\texttt{contact}_i) + u(\texttt{judge}_\texttt{i}) \ . \tag{21}$$

where $u(\texttt{judge}_\texttt{i}) \sim N(0, \sigma_u^2)$ This model can be fitted and summarized with

```
> mm1 <- clmm(rating ~ temp + contact + (1 | judge), data = wine)
> summary(mm1)

Cumulative Link Mixed Model fitted with the Laplace approximation

formula: rating ~ temp + contact + (1 | judge)
data:    wine

 link  threshold nobs logLik AIC    niter    max.grad cond.H
 logit flexible  72   -81.57 177.13 500(934) 4.11e-06 2.8e+01

Random effects:
        Var Std.Dev
judge 1.279   1.131

Coefficients:
           Estimate Std. Error z value Pr(>|z|)
tempwarm     3.0630     0.5954   5.145 2.68e-07 ***
contactyes   1.8349     0.5126   3.580 0.000344 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Threshold coefficients:
    Estimate Std. Error z value
```

```
1|2  -1.6237     0.6824  -2.379
2|3   1.5134     0.6037   2.507
3|4   4.2285     0.8090   5.227
4|5   6.0888     0.9725   6.261
```

The computational method used is the Laplace approximation and the parameter estimates are

$$\hat{\beta}_1 = 3.06 \quad \hat{\beta}_2 = 1.83 \quad \sigma_u^2 = 1.13^2$$

Observe that these estimates are larger in magnitude than the estimates from the model that ignored the judges effects in example 3. This effect is known as the *attenuation* effect which reflects that effects in models that average over (or marginalizes over) the distribution of judges (which corresponds to ignoring judges in the model) are attenuated, i.e., smaller in absolute magnitude than the effects that conditional on the judge effects. Correspondingly the models are sometimes referred to as *marginal models* and *conditional models*. □

References

Agresti, A. (2002). *Categorical Data Analysis* (Second ed.). Wiley.

Bauer, D. (2009). A note on comparing the estimates of models forăcluster-correlated or longitudinal data with binary orăordinalăoutcomes. *Psychometrika 74*, pp. 97–105.

Burridge, J. (1981). A note on maximum likelihood estimation for regression models using grouped data. *Journal of the Royal Statistical Society. Series B (Methodological) 43*(1), pp. 41–45.

Collett, D. (2002). *Modelling binary data* (2nd ed.). London: Chapman & Hall/CRC.

Efron, B. and D. V. Hinkley (1978). Assessing the accuracy of the maximum likelihood estimator: Observed versus expected Fisher information. *Biometrika 65*(3), pp. 457–487.

Fahrmeir, L. and G. Tutz (2001). *Multivariate Statistical Modelling Based on Generalized Linear Models* (2nd ed.). Springer series in statistics. Springer-Verlag New York, Inc.

Fielding, A. (2004). Scaling for residual variance components of ordered category responses in generalised linear mixed multilevel models. *Quality & Quantity 38*, pp. 425–433.

Greene, W. H. and D. A. Hensher (2010). *Modeling Ordered Choices: A Primer*. Cambridge University Press.

McCullagh, P. (1980). Regression models for ordinal data. *Journal of the Royal Statistical Society, Series B 42*, pp. 109–142.

McCullagh, P. and J. Nelder (1989). *Generalized Linear Models* (Second ed.). Chapman & Hall/CRC.

Pratt, J. W. (1981). Concavity of the log likelihood. *Journal of the American Statistical Association 76*(373), pp. 103–106.

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

Winship, C. and R. D. Mare (1984). Regression models with ordinal variables. *American Sociological Review 49*(4), pp. 512–525.