Cumulative Link Models for Ordinal Regression with the R Package ordinal

Rune Haubo B Christensen
Technical University of Denmark
&
Christensen Statistics

Abstract

This paper introduces the R-package **ordinal** for the analysis of ordinal data using cumulative link models. The model framework implemented in **ordinal** includes partial proportional odds, structured thresholds, scale effects and flexible link functions. The package also support cumulative link models with random effects which are covered in a future paper. A speedy and reliable regularized Newton estimation scheme using analytical derivatives provides maximum likelihood estimation of the model class. The paper describes the implementation in the package as well as how to use the functionality in the package for analysis of ordinal data including topics on model identifiability and customized modelling. The package implements methods for profile likelihood confidence intervals, analysis of deviance tables with type I, II and III tests, predictions of various kinds as well as methods for checking the convergence of the fitted models.

Keywords: ordinal, cumulative link models, proportional odds, scale effects, R.

This is a copy of an article that is no longer submitted for publication in Journal of Statistical Software (`https://www.jstatsoft.org/`).

1. Introduction

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

This paper introduces the **ordinal** package (Christensen 2019) for R (R Core Team 2020) for the analysis of ordinal data with cumulative link models. The paper describes how **ordinal** supports the fitting of CLMs with various models structures, model assessment and inferential options including tests of partial proportional odds, scale effects, threshold structures and flexible link functions. The implementation, its flexibility in allowing for costumizable models and an effective fitting algorithm is also described. The **ordinal** package also supports cumulative link *mixed* models (CLMM); CLMs with normally distributed random effects. The support of this model class will not be given further treatment here but remain a topic for a future paper.

The name, *cumulative link models* is adopted from Agresti (2002), but the model class has been referred to by several other names in the literature, such as *ordered logit models* and *ordered probit models* (Greene and Hensher 2010) for the logit and probit link functions. The cumulative link model with a logit link is widely known as the *proportional odds model* due to McCullagh (1980) and with a complementary log-log link, the model is sometimes referred to as the *proportional hazards model* for grouped survival times.

CLMs is one of several types of models specifically developed for ordinal data. Alternatives to CLMs include continuation ratio models, adjacent category models, and stereotype models (Ananth and Kleinbaum 1997) but only models in the CLM framework will be considered in this paper.

1.1. Software review

Cumulative link models can be fitted by all the major software packages and while some software packages support scale effects, partial proportional odds (also referred to as unequal slopes, partial effects, and nominal effects), different link functions and structured thresholds all model structures are not available in any one package or implementation. The following brief software review is based on the publicly available documentation at software package websites retrieved in May 2020.

IBM SPSS (IBM Corp. 2017) implements McCullagh's **PLUM** (McCullagh 1980) procedure, allows for the five standard link functions (cf. Table 3) and scale effects. Estimation is via Fisher-Scoring and a test for equal slopes is available for the location-only model while it is not possible to estimate a partial proportional odds model.

Stata (StataCorp 2017) includes the `ologit` and `oprobit` procedures for CLMs with logistic and probit links but without support for scale effects, partial effect or structured thresholds. The add-on package **oglm** (Williams 2010) allows for all five standard link functions and scale effects. The **GLLAMM** package (Rabe-Hesketh, Skrondal, and Pickles 2004) also has some support for CLMs in addition to some support for random effects.

SAS (SAS Institute Inc. 2010) implements CLMs with logit links in `proc logistic` and CLMs with the 5 standard links in `prog genmod`.

Matlab (Matlab 2020) fits CLMs with the `mnrfit` function allowing for logit, probit, complementary log-log and log-log links.

Python has a package **mord** (Pedregosa-Izquierdo 2015) for ordinal classification and prediction focused at machine learning applications.

In R, several packages on the Comprehensive R Archive Network (CRAN) implements CLMs. `polr` from **MASS** (Venables and Ripley 2002) implements standard CLMs allowing for the 5 standard link functions but no further extensions; the **VGAM** package (Yee 2010) includes CLMs via the `vglm` function using the `cumulative` link. `vglm` allows for several link functions as well as partial effects. The `lrm` and `orm` functions from the **rms** package (Harrell Jr 2018) also implements CLMs with the 5 standard link functions but without scale effects, partial or structured thresholds. A Bayesian alternative is implemented in the **brms** package (Bürkner 2017; **?**) which includes structured thresholds in addition to random-effects.

In addition, several other R packages include methods for analyses of ordinal data including

Fitting	Miscellaneous	Former impl.	Distributions
`clm`	`convergence`	`clm2`	`[pdqrg]gumbel`^{c}
`clmm`^{c}	`slice`	`clmm2`^{c}	`[pdg]lgamma`^{c}
`clm.fit`	`drop.coef`	`clm2.control`	`gnorm`^{c}
`clm.control`		`clmm2.control`	`glogis`^{c}
`clmm.control`			`gcauchy`^{c}

Table 1: Key functions in **ordinal**. Superscript "c" indicates (partial or full) implementation in C.

oglmx (Carroll 2018), **MCMCpack** (Martin, Quinn, and Park 2011), **mvord** (Hirk, Hornik, and Vana 2020), **CUB** (Iannario, Piccolo, and Simone 2020), and **ordinalgmifs** (Archer, Hou, Zhou, Ferber, Layne, and Gentry 2014).

1.2. ordinal package overview

The **ordinal** package implements CLMs and CLMMs along with functions and methods to support these model classes as summarized in Table 1. The two key functions in **ordinal** are `clm` and `clmm` which fits CLMs and CLMMs respectively; `clm2` and `clmm2`[1] provide legacy implementations primarily retained for backwards compatibility. This paper introduces `clm` and its associated functionality covering CLMs with location, scale and nominal effects, structured thresholds and flexible link functions. `clm.fit` is the main work horse behind `clm` and an analogue to `lm.fit` for linear models. The package includes methods for assessment of convergence with `convergence` and `slice`, an auxiliary method for removing linearly dependent columns from a design matrix in `drop.coef`. Distributional support functions in **ordinal** provide support for Gumbel and log-gamma distributions as well as gradients[2] of normal, logistic and Cauchy probability density functions which are used in the iterative methods implemented in `clm` and `clmm`.

As summarized in Table 2, **ordinal** provides the familiar suite of extractor and print methods for `clm` objects known from `lm` and `glm`. These methods all behave in ways similar to those for `glm`-objects with the exception of `model.matrix` which returns a list of model matrices and `terms` which can return the `terms` object for each of three formulae. The inference methods facilitate profile likelihood confidence intervals via `profile` and `confint`, likelihood ratio tests for model comparison via `anova`, model assessment by tests of removal of model terms via `drop1` and addition of new terms via `add1` or AIC-based model selection via `step`. Calling `anova` on a single `clm`-object provides an analysis of deviance table with type I, II or III Wald-based χ^2 tests following the SAS-definitions of such tests (SAS Institute Inc. 2008). In addition to standard use of `clm`, the implementation facilitates extraction a model environment containing a complete representation of the model allowing the user to fit costumized models containing, for instance, special structures on the threshold parameters, restrictions on regression parameters or other case-specific model requirements. As CLMMs are not covered by this paper methods for `clmm` objects will not be discussed.

---

[1] A brief tutorial on `clmm2` is currently available at the package website on CRAN: `https://CRAN.R-project.org/package=ordinal`

[2] gradients with respect to x, the quantile; not the parameters of the distributions

Extractor and Print		Inference	Checking
`coef`	`print`	`anova`	`slice`
`fitted`	`summary`	`drop1`	`convergence`
`logLik`	`model.frame`	`add1`	
`nobs`	`model.matrix`	`confint`	
`vcov`	`update`	`profile`	
`AIC`, `BIC`		`predict`	
`extractAIC`		`step`, `stepAIC`	

Table 2: Key methods for `clm` objects.

Other packages including **emmeans** (Lenth 2020), **margins** (Leeper 2018), **ggeffects** (Lüdecke 2018), **generalhoslem** (Jay 2019) and **effects** (Fox and Weisberg 2018; Fox and Hong 2009) extend the **ordinal** package by providing methods marginal means, tests of functions of the coefficients, goodness-of-fit tests and methods for illustration of fitted models.

The **ordinal** package is therefore unique in providing a comprehensive framework for cumulative link models exceeding that of other software packages with its functionality extended by a series of additional R packages.

1.3. Organization of the paper

The remainder of the paper is organized as follows. The next section establishes notation by defining CLMs and associated log-likelihood functions, then describes the extended class of CLMs that is implemented in **ordinal** including details about scale effects, structured thresholds, partial proportional odds and flexible link functions. The third section describes how maximum likelihood (ML) estimation of CLMs is implemented in **ordinal**. The fourth section describes how CLMs are fitted and ordinal data are analysed with **ordinal** including sections on nominal effects, scale effects, structured thresholds, flexible link functions, profile likelihoods, assessment of model convergence, fitted values and predictions. The final parts of section four is on a more advanced level and include issues around model identifiability and customizable fitting of models not otherwise covered by the **ordinal** API. We end in section 5 with Conclusions.

2. Cumulative link models

A cumulative link model is a model for ordinal-scale observations, i.e., observations that fall in an ordered finite set of categories. Ordinal observations can be represented by a random variable Y_i that takes a value j if the ith ordinal observations falls in the j'th category where $j = 1, \ldots, J$ and $J \geq 2$.[3]A basic cumulative link model is

$$\gamma_{ij} = F(\eta_{ij}) \ , \quad \eta_{ij} = \theta_j - \boldsymbol{x}_i^\top \boldsymbol{\beta} \ , \quad i = 1, \ldots, n \ , \quad j = 1, \ldots, J-1 \ , \tag{1}$$

---

[3] binomial models ($J = 2$) are also included.

where

$$\gamma_{ij} = \mathsf{P}(Y_i \leq j) = \pi_{i1} + \ldots + \pi_{ij} \quad \text{with} \quad \sum_{j=1}^{J} \pi_{ij} = 1$$

are cumulative probabilities[4], π_{ij} is the probability that the ith observation falls in the jth category, η_{ij} is the linear predictor and $\boldsymbol{x}_i^\top$ is a p-vector of regression variables for the parameters, $\boldsymbol{\beta}$ without a leading column for an intercept and F is the inverse link function. The thresholds (also known as cut-points or intercepts) are strictly ordered:

$$-\infty \equiv \theta_0 \leq \theta_1 \leq \ldots \leq \theta_{J-1} \leq \theta_J \equiv \infty.$$

2.1. The multinomial distribution and the log-likelihood function

The ordinal observation Y_i which assumes the value j can be represented by a multinomially distributed variable $\boldsymbol{Y}_i^* \sim \text{multinom}(\boldsymbol{\pi}_i, 1)$, where $\boldsymbol{Y}_i^*$ is a J-vector with a 1 at the j'th entry and 0 otherwise, and with probability mass function

$$\mathsf{P}(\boldsymbol{Y}_i^* = \boldsymbol{y}_i^*) = \prod_j \pi_{ij}^{y_{ij}^*} \ . \tag{2}$$

The log-likelihood function can therefore be written as

$$\ell(\boldsymbol{\theta}, \boldsymbol{\beta}; \boldsymbol{y}^*) = \sum_i \sum_j y_{ij}^* \log \pi_{ij}$$

or equivalently

$$\begin{aligned} \ell(\boldsymbol{\theta}, \boldsymbol{\beta}; \boldsymbol{y}) &= \sum_i \sum_j \mathrm{I}(y_i = j) \log \pi_{ij} \\ &= \sum_i \log \tilde{\pi}_i \end{aligned}$$

where $\tilde{\pi}_i$ is the j'th entry in J-vector $\boldsymbol{\pi}_i$ with elements π_{ij} and $\mathrm{I}(\cdot)$ is the indicator function. Allowing for observation-level weights (case weights), w_i leads finally to

$$\ell(\boldsymbol{\theta}, \boldsymbol{\beta}; \boldsymbol{y}) = \sum_i w_i \log \tilde{\pi}_i \ . \tag{3}$$

Likelihood based inference

Confidence intervals for model parameters are obtained by appealing to the asymptotic normal distribution of a statistic $s(\cdot)$ for a scalar parameter of interest β_a and defined as

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

where $z_{1-\alpha/2}$ is the $(1-\alpha/2)$ quantile of the standard normal cumulative distribution function. Taking $s(\cdot)$ to be the Wald statistic $s(\beta_a): \ w(\beta_a) = (\hat{\beta}_a - \beta_a)/\hat{\mathrm{se}}(\hat{\beta}_a)$ leads to the classical

---

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

symmetric intervals. Better confidence intervals can be obtained by choosing instead the likelihood root statistic (see e.g., Pawitan 2001; Brazzale, Davison, and Reid 2007):

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

where

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

is the profile likelihood for the scalar parameter β_a and $\boldsymbol{\beta}_{-a}$ is the vector of regression parameters without the a'th one.

While the profile likelihood has to be optimized over all parameters except β_a we define a *log-likelihood slice* as

$$\ell_{\text{slice}}(\beta_a; \boldsymbol{y}) = \ell(\beta_a; \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\beta}}_{-a}, \boldsymbol{y})\ , \tag{4}$$

which is the log-likelihood function evaluated at β_a while keeping the remaining parameters fixed at their ML estimates.

A quadratic approximation to the log-likelihood slice is $(\hat{\beta}_a - \beta_a)^2/2\tau_a^2$ where the *curvature unit* τ_a is the square root of a'th diagonal element of the Hessian of $-\ell(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\beta}}; \boldsymbol{y})$.

2.2. Link functions

A commonly used link function is the logit link which leads to

$$\mathrm{logit}(\gamma_{ij}) = \log \frac{\mathsf{P}(Y_i \leq j)}{1 - \mathsf{P}(Y_i \leq j)} \tag{5}$$

The odds ratio (OR) of the event $Y_i \leq j$ at $\boldsymbol{x}_1$ relative to the same event at $\boldsymbol{x}_2$ is then

$$\mathrm{OR} = \frac{\gamma_j(\boldsymbol{x}_1)/[1 - \gamma_j(\boldsymbol{x}_1)]}{\gamma_j(\boldsymbol{x}_2)/[1 - \gamma_j(\boldsymbol{x}_2)]} = \frac{\exp(\theta_j - \boldsymbol{x}_1^\top \boldsymbol{\beta})}{\exp(\theta_j - \boldsymbol{x}_2^\top \boldsymbol{\beta})} = \exp[(\boldsymbol{x}_2^\top - \boldsymbol{x}_1^\top)\boldsymbol{\beta}] \tag{6}$$

which is independent of j. Thus the cumulative odds ratio is proportional to the distance between $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ which motivated McCullagh (1980) to denote 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 $\exp(\beta_{\text{treatment}})$.

The probit link has its own interpretation through a normal linear model for a latent variable which is considered in section 2.4.

The complementary log-log (clog-log) link is also sometimes used because of its interpretation as a proportional hazards model for grouped survival times:

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

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

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

thus the ratio of hazards at $\boldsymbol{x}_1$ relative to $\boldsymbol{x}_2$ are proportional. If the log-log link is used on the response categories in the reverse order, this is equivalent to using the clog-log link on

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

Table 3: Summary of the five standard link functions. ^{a}: the *complementary log-log* link; ^{b}: the Gumbel distribution is also known as the extreme value (type I) distribution for extreme minima or maxima. It is also sometimes referred to as the Weibull (or log-Weibull) distribution; ^{c}: the Cauchy distribution is a t-distribution with one degree of freedom.

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.

Details of the most common link functions are described in Table 3.

The **ordinal** package allows for the estimation of an extended class of cumulative link models in which the basic model (1) is extended in a number of ways including structured thresholds, partial proportional odds, scale effects and flexible link functions. The following sections will describe these extensions of the basic CLM.

2.3. Extensions of cumulative link models

A general formulation of the class of models (excluding random effects) that is implemented in **ordinal** can be written

$$\gamma_{ij} = F_\lambda(\eta_{ij}), \quad \eta_{ij} = \frac{g_{\boldsymbol{\alpha}}(\theta_j) - \boldsymbol{x}_i^\top \boldsymbol{\beta} - \boldsymbol{w}_i^\top \tilde{\boldsymbol{\beta}}_j}{\exp(\boldsymbol{z}_i \boldsymbol{\zeta})} \tag{7}$$

where

F_λ is the inverse link function. It may be parameterized by the scalar parameter λ in which case we refer to F_λ^{-1} as a *flexible link function*,

$g_{\boldsymbol{\alpha}}(\theta_j)$ parameterises thresholds $\{\theta_j\}$ by the vector $\boldsymbol{\alpha}$ such that g restricts $\{\theta_j\}$ to be for example symmetric or equidistant. We denote this *structured thresholds*.

$\boldsymbol{x}_i^\top \boldsymbol{\beta}$ are the ordinary regression effects,

$\boldsymbol{w}_i^\top \tilde{\boldsymbol{\beta}}_j$ are regression effects which are allowed to depend on the response category j and they are denoted *partial* or *non-proportional odds* (Peterson and Harrell Jr. 1990) when the logit link is applied. To include other link functions in the terminology we denote these effects *nominal effects* (in text and code) because these effects are not integral to the ordinal nature of the data.

$\exp(\boldsymbol{z}_i\boldsymbol{\zeta})$ are *scale effects* since in a latent variable view these effects model the scale of the underlying location-scale distribution.

With the exception of the structured thresholds, these extensions of the basic CLM have been considered individually in a number of sources but to the author's best knowledge not previously in a unified framework. For example partial proportional odds have been considered by Peterson and Harrell Jr. (1990) and scale effect have been considered by McCullagh (1980) and Cox (1995).

2.4. Latent variable motivation of CLMs

It is natural to motivate the CLM from a linear model for a categorized version of a latent variable. Assume the following linear model for an unobserved latent variable:

$$S_i = \alpha^* + \boldsymbol{x}_i^\top \boldsymbol{\beta}^* + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^{*2}) \tag{8}$$

If S_i falls between two thresholds, $\theta^*_{j-1} < S_i \leq \theta^*_j$ where

$$-\infty \equiv \theta^*_0 < \theta^*_1 < \ldots < \theta^*_{J-1} < \theta^*_J \equiv \infty \tag{9}$$

then $Y_i = j$ is observed and the cumulative probabilities are:

$$\gamma_{ij} = \mathsf{P}(Y_i \leq j) = \mathsf{P}(S_i \leq \theta^*_j) = \mathsf{P}\left(Z \leq \frac{\theta^*_j - \alpha^* - \boldsymbol{x}_i^\top \boldsymbol{\beta}^*}{\sigma^*}\right) = \Phi(\theta_j - \boldsymbol{x}_i^\top \boldsymbol{\beta})$$

where Z follows a standard normal distribution, Φ denotes the standard normal cumulative distribution function, parameters with an "*" exist on the latent scale, $\theta_j = (\theta^*_j - \alpha^*)/\sigma^*$ and $\boldsymbol{\beta} = \boldsymbol{\beta}^*/\sigma^*$. Note that α^*, $\boldsymbol{\beta}^*$ and σ^* would have been identifiable if the latent variable S was directly observed, but they are not identifiable with ordinal observations.

If we allow a log-linear model for the scale such that

$$\varepsilon_i \sim N(0, \sigma_i^{*2}), \quad \sigma^*_i = \exp(\mu + \boldsymbol{z}_i^\top \boldsymbol{\zeta}) = \sigma^* \exp(\boldsymbol{z}_i^\top \boldsymbol{\zeta})$$

where $\boldsymbol{z}_i$ is the i'th row of a design matrix $\boldsymbol{Z}$ without a leading column for an intercept and $\sigma^* = \exp(\mu)$, then

$$\gamma_{ij} = \mathsf{P}\left(Z \leq \frac{\theta^*_j - \alpha^* - \boldsymbol{x}_i^\top \boldsymbol{\beta}^*}{\sigma^*_i}\right) = \Phi\left(\frac{\theta_j - \boldsymbol{x}_i^T \boldsymbol{\beta}}{\sigma_i}\right)$$

where $\sigma_i = \sigma^*_i/\sigma^* = \exp(\boldsymbol{z}_i^\top \boldsymbol{\zeta})$ is the *relative* scale.

The common link functions: probit, logit, log-log, c-log-log and cauchit correspond to inverse cumulative distribution functions of the normal, logistic, Gumbel(max), Gumbel(min) and Cauchy distributions respectively. These distributions are all members of the location-scale family with common form $F(\mu, \sigma)$, with location μ and non-negative scale σ, for example, the logistic distribution has mean μ and standard deviation $\sigma\pi/\sqrt{3}$. Choosing a link function therefore corresponds to assuming a particular distribution for the latent variable S in which $\boldsymbol{x}_i^\top \boldsymbol{\beta}$ and $\exp(\boldsymbol{z}_i^\top \boldsymbol{\zeta})$ models location *differences* and scale *ratios* respectively of that distribution.

2.5. Structured thresholds

Structured thresholds, $\{g(\boldsymbol{\alpha})_j\}$ makes it possible to impose restrictions on the thresholds $\boldsymbol{\theta} = g(\boldsymbol{\alpha})$. For instance restricting the thresholds to be equidistant means that only the location of, say, the first threshold and the spacing between adjacent thresholds has to be estimated, thus only two parameters are used to parameterize the thresholds irrespective of the number of response categories.

ordinal takes $g(\boldsymbol{\alpha})$ to be a linear function and operates with

$$g(\boldsymbol{\alpha}) = \mathcal{J}^\top \boldsymbol{\alpha} = \boldsymbol{\theta}$$

where the Jacobian $\mathcal{J}$ defines the mapping from the parameters $\boldsymbol{\alpha}$ to the thresholds $\boldsymbol{\theta}$. The traditional ordered but otherwise unrestricted thresholds are denoted *flexible thresholds* and obtained by taking $\mathcal{J}$ to be an identity matrix.

Assuming $J = 6$ ordered categories, the Jacobians for equidistant and symmetric thresholds (denoted `equidistant` and `symmetric` in the `clm`-argument `threshold`) are

$$\mathcal{J}_{\text{equidistant}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 \end{bmatrix}, \quad \mathcal{J}_{\text{symmetric}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Another version of symmetric thresholds (denoted `symmetric2`) is sometimes relevant with an unequal number of response categories here illustrated with $J = 5$ together with the `symmetric` thresholds:

$$\mathcal{J}_{\text{symmetric2}} = \begin{bmatrix} 0 & -1 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix}, \quad \mathcal{J}_{\text{symmetric}} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ -1 & 0 & 0 & 1 \end{bmatrix}$$

The nature of $\mathcal{J}$ for a particular model can always be inspected by printing the `tJac` component of the `clm` fit.

2.6. Partial proportional odds and nominal effects

The nominal effects $\boldsymbol{w}_i^\top \tilde{\boldsymbol{\beta}}_j$ can be considered an extension of the regression part of the model $\boldsymbol{x}_i^\top \boldsymbol{\beta}$ in which the regression effects are allowed to vary with j. The nominal effects can also be considered an extension of the thresholds θ_j which allows them to depend on variables $\boldsymbol{w}_i^\top$: $\tilde{\theta}_{ij}(\boldsymbol{w}_i^\top) = \theta_j - \boldsymbol{w}_i^\top \tilde{\boldsymbol{\beta}}_j$ is the j'th threshold for the i'th observation. The following treatment assumes for latter view.

In general let $\boldsymbol{W}$ denote the design matrix for the nominal effects without a leading column for an intercept; the nominal-effects parameter vector $\tilde{\boldsymbol{\beta}}_j$ is then $\text{ncol}(\boldsymbol{W})$ long and $\tilde{\boldsymbol{\beta}}$ is $\text{ncol}(\boldsymbol{W}) \cdot (J - 1)$ long.

If $\boldsymbol{W}$ is the design matrix for the nominal effects containing a single column for a continuous variable then $\tilde{\beta}_j$ is the slope parameter corresponding to the j'th threshold and θ_j is the j'th intercept, i.e., the threshold when the covariate is zero. Looking at $\tilde{\theta}_{ij}(\boldsymbol{w}_i^\top) = \theta_j - \boldsymbol{w}_i^\top \tilde{\boldsymbol{\beta}}_j$ as a linear model for the thresholds facilitates the interpretation.

If, on the other hand, $\boldsymbol{W}$ is the design matrix for a categorical variable (a `factor` in R) then the interpretation of $\tilde{\boldsymbol{\beta}}_j$ depends on the contrast-coding of $\boldsymbol{W}$. If we assume that the categorical variable has 3 levels, then $\tilde{\boldsymbol{\beta}}_j$ is a 2-vector. In the default treatment contrast-coding (`"contr.treatment"`) θ_j is the j'th threshold for the first (base) level of the factor, $\tilde{\beta}_{1j}$ is the differences between thresholds for the first and second level and $\tilde{\beta}_{2j}$ is the difference between the thresholds for the first and third level.

In general we define $\boldsymbol{\Theta}$ as a matrix with $J-1$ columns and with 1 row for each combination of the levels of factors in $\boldsymbol{W}$. This matrix is available in the `Theta` component of the model fit.

Note that variables in $\boldsymbol{X}$ cannot also be part of $\boldsymbol{W}$ if the model is to remain identifiable. **ordinal** detects this and automatically removes the offending variables from $\boldsymbol{X}$.

2.7. Flexible link functions

The **ordinal** package allows for two kinds of flexible link functions due to Aranda-Ordaz (1983) and Genter and Farewell (1985).

The link function proposed by Aranda-Ordaz (1983) reads

$$F_\lambda^{-1}(\gamma_{ij}) = \log\left\{\frac{(1-\gamma_{ij})^{-\lambda} - 1}{\lambda}\right\} ,$$

which depends on the auxiliary parameter $\lambda \in]0, \infty[$. When $\lambda = 1$, the logistic link function arise, and when $\lambda \to 0$,

$$\{(1-\gamma_{ij})^{-\lambda} - 1\}/\lambda \to \log(1-\gamma_{ij})^{-1} ,$$

so the log-log link arise.

The inverse link function and its derivative are given by

$$\begin{aligned} F(\eta) &= 1 - (\lambda \exp(\eta) + 1)^{-\lambda^{-1}} \\ f(\eta) &= \exp(\eta)(\lambda \exp(\eta) + 1)^{-\lambda^{-1}-1} \end{aligned}$$

The density implied by the inverse link function is left-skewed if $0 < \lambda < 1$, symmetric if $\lambda = 1$ and right-skewed if $\lambda > 1$, so the link function can be used to assess the evidence about possible skewness of the latent distribution.

The log-gamma link function proposed by Genter and Farewell (1985) is based on the log-gamma density by Farewell and Prentice (1977). The cumulative distribution function and hence inverse link function reads

$$F_\lambda(\eta) = \begin{cases} 1 - G(q; \lambda^{-2}) & \lambda < 0 \\ \Phi(\eta) & \lambda = 0 \\ G(q; \lambda^{-2}) & \lambda > 0 \end{cases}$$

where $q = \lambda^{-2}\exp(\lambda\eta)$ and $G(\cdot; \alpha)$ denotes the Gamma distribution with shape parameter α and unit rate parameter, and Φ denotes the standard normal cumulative distribution function.

The corresponding density function reads

$$f_\lambda(\eta) = \begin{cases} |\lambda| k^k \Gamma(k)^{-1} \exp\{k(\lambda\eta - \exp(\lambda\eta))\} & \lambda \neq 0 \\ \phi(\eta) & \lambda = 0 \end{cases}$$

where $k = \lambda^{-2}$, $\Gamma(\cdot)$ is the gamma function and ϕ is the standard normal density function.

By attaining the Gumbel(max) distribution at $\lambda = -1$, the standard normal distribution at $\lambda = 0$ and the Gumbel(min) distribution at $\lambda = 1$ the log-gamma link bridges the log-log, probit and complementary log-log links providing right-skew, symmetric and left-skewed latent distributions in a single family of link functions.

Note that choice and parameterization of the predictor, η_{ij}, e.g., the use of scale effects, can affect the evidence about the shape of the latent distribution. There are usually several link functions which provide essentially the same fit to the data and choosing among the good candidates is often better done by appealing to arguments such as ease of interpretation rather than arguments related to fit.

3. Implementation of ML Estimation of CLMs in ordinal

In the **ordinal** package cumulative link models are (by default) estimated with a regularized Newton-Raphson (NR) algorithm with step-halving (line search) using analytical expressions for the gradient and Hessian of the negative log-likelihood function.

This NR algorithm with analytical derivatives is used irrespective of whether the model contains structured thresholds, nominal effects or scale effects; the only exception being models with flexible link functions for which a general-purpose quasi-Newton optimizer is used.

Due to computationally cheap and efficient evaluation of the analytical derivatives, the relative well-behaved log-likelihood function (with exceptions described below) and the speedy convergence of the Newton-Raphson algorithm, the estimation of CLMs is virtually instant on a modern computer even with complicated models on large datasets. This also facilitates simulation studies. More important than speed is perhaps that the algorithm is reliable and accurate.

Technical aspects of the regularized NR algorithm with step-halving (line search) are described in appendix A and analytical gradients are described in detail in Christensen (2012).

Properties of the log-likelihood function for extended CLMs

Pratt (1981) and Burridge (1981) showed (seemingly independent of each other) that the log-likelihood function of the basic cumulative link model (1) is concave. This means that there is a unique global optimum of the log-likelihood function and therefore no risk of convergence to a local optimum.

It also means that the Hessian matrix for the negative log-likelihood is strictly positive definite and therefore also that the Newton step is always in direction of higher likelihood. The genuine Newton step may be too long to actually cause an increase in likelihood from one iteration to the next (this is called "overshoot"). This is easily overcome by successively halving the length of the Newton step until an increase in likelihood is achieved.

Exceptions to the strict concavity of the log-likelihood function include models using the cauchit link, flexible link functions as well as models with scale effects. Notably models with structured thresholds as well as nominal effects do not affect the linearity of the predictor, η_{ij} and so are also guaranteed to have concave log-likelihoods.

The restriction of the threshold parameters $\{\theta_j\}$ being non-decreasing is dealt with by defining $\ell(\boldsymbol{\theta}, \boldsymbol{\beta}; y) = \infty$ when $\{\theta_j\}$ are not in a non-decreasing sequence. If the algorithm attempts evaluation at such illegal values step-halving effectively brings the algorithm back on track.

Other implementations of CLMs re-parameterize $\{\theta_j\}$ such that the non-decreasing nature of $\{\theta_j\}$ is enforced by the parameterization, for example, `MASS::polr` (package version 7.3.49) optimize the likelihood using

$$\tilde{\theta}_1 = \theta_1, \; \tilde{\theta}_2 = \exp(\theta_2 - \theta_1), \; \ldots, \; \tilde{\theta}_{J-1} = \exp(\theta_{J-2} - \theta_{J-1})$$

This is deliberately not used in **ordinal** because the log-likelihood function is generally closer to quadratic in the original parameterization in our experience which facilitates faster convergence.

Starting values

For the basic CLMs (1) the threshold parameters are initialized to an increasing sequence such that the cumulative density of a logistic distribution between consecutive thresholds (and below the lowest or above the highest threshold) is constant. The regression parameters $\boldsymbol{\beta}$, scale parameters $\boldsymbol{\zeta}$ as well as nominal effect $\boldsymbol{\beta}^*$ are initialized to 0.

If the model specifies a cauchit link or includes scale parameters estimation starts at the parameter estimates of a model using the probit link and/or without the scale-part of the model.

Estimation problems

With many nominal effects it may be difficult to find a model in which the threshold parameters are strictly increasing for all combinations of the parameters. Upon convergence of the NR algorithm the model evaluates the $\boldsymbol{\Theta}$-matrix and checks that each row of threshold estimates are increasing.

When a continuous variable is included among the nominal effects it is often helpful if the continuous variable is centered at an appropriate value (at least within the observed range of the data). This is because $\{\theta_j\}$ represent the thresholds when the continuous variable is zero and $\{\theta_j\}$ are enforced to be a non-decreasing sequence. Since the nominal effects represent different slopes for the continuous variable the thresholds will necessarily be ordered differently at some other value of the continuous variable.

Convergence codes

Irrespective of the fitting algorithm, **ordinal** reports the following convergence codes for CLMs in which negative values indicate convergence failure:

-3 Not all thresholds are increasing. This is only possible with nominal effects and the resulting fit is invalid.

-2 The Hessian has at least one negative eigenvalue. This means that the point at which the algorithm terminated does not represent an optimum.

-1 Absolute convergence criterion (maximum absolute gradient) was not satisfied. This means that the algorithm couldn't get close enough to a stationary point of the log-likelihood function.

0 Successful convergence.

1 The Hessian is singular (i.e., at least one eigenvalue is zero). This means that some parameters are not uniquely determined.

Note that with convergence code **1** the optimum of the log-likelihood function has been found although it is not a single point but a line (or in general a (hyper) plane), so while some parameters are not uniquely determined the value of the likelihood is valid enough and can be compared to that of other models.

In addition to these convergence codes, the NR algorithm in **ordinal** reports the following messages:

0 Absolute and relative convergence criteria were met

1 Absolute convergence criterion was met, but relative criterion was not met

2 iteration limit reached

3 step factor reduced below minimum

4 maximum number of consecutive Newton modifications reached

Note that convergence is assessed irrespective of potential messages from the fitting algorithm and irrespective of whether the tailored NR algorithm or a general-purpose quasi-Newton optimizer is used.

4. Fitting cumulative link models in ordinal with `clm`

The `clm` function takes the following arguments:

```
clm(formula, scale, nominal, data, weights, start, subset,
     doFit = TRUE, na.action, contrasts, model = TRUE, control = list(),
     link = c("logit", "probit", "cloglog", "loglog", "cauchit",
         "Aranda-Ordaz", "log-gamma"), threshold = c("flexible",
         "symmetric", "symmetric2", "equidistant"), ...)
```

Several arguments are standard and well-known from `lm` and `glm` and will not be described in detail; `formula`, `data`, `weights`, `subset` and `na.action` are all parts of the standard model specification in R.

`scale` and `nominal` are interpreted as R-formulae with no left hand sides and specifies the scale and nominal effects of the model respectively, see sections 4.3 and 4.2 for details; `start` is an optional vector of starting values; `doFit` can be set to `FALSE` to prompt `clm` to return a model *environment*, for details see section 4.10; `model` controls whether the `model.frame` should be included in the returned model fit; `link` specifies the link function and `threshold` specifies an optional threshold structure, for details see section 4.4.

Note the absence of a separate `offset` argument. Since `clm` allows for different offsets in `formula` and `scale`, offsets have to be specified within a each formulae, e.g., `scale = ~ x1 + offset(x2)`.

Methods for `clm` model fits are summarized in Table 2 and introduced in the following sections.

Control parameters can either be specified as a named list, among the optional `...` arguments, or directly as a call to `clm.control` — in the first two cases the arguments are passed on to `clm.control`. `clm.control` takes the following arguments:

```
clm.control(method = c("Newton", "model.frame", "design", "ucminf",
     "nlminb", "optim"), sign.location = c("negative", "positive"),
     sign.nominal = c("positive", "negative"), ..., trace = 0L,
     maxIter = 100L, gradTol = 1e-06, maxLineIter = 15L, relTol = 1e-06,
     tol = sqrt(.Machine$double.eps), maxModIter = 5L, convergence = c("warn",
         "silent", "stop", "message"))
```

The `method` argument specifies the optimization and/or return method. The default estimation method (`Newton`) is the regularized Newton-Raphson estimation scheme described in section A; options `model.frame` and `design` prompts `clm` to return respectively the `model.frame` and a list of objects that represent the internal representation instead of fitting the model; options `ucminf`, `nlminb` and `optim` represent different general-purpose optimizers which may be used to fit the model (the former from package **ucminf** (Nielsen and Mortensen 2016), the latter two from package **stats**). The `sign.location` and `sign.nominal` options allow the user to flip the signs on the location and nominal model terms. The `convergence` argument instructs `clm` how to alert the user of potential convergence problems; `...` are optional arguments passed on to the general purpose optimizers; `trace` applies across all optimizers and positive values lead to printing of progress during iterations; the remaining arguments (`maxIter, gradTol, maxLineIter, relTol, tol`) control the behavior of the regularized NR algorithm described in appendix A.

4.1. Fitting a basic cumulative link model with `clm`

In the following examples we will use the wine data from Randall (1989) available in the object `wine` in package **ordinal**, cf., Table 4. The data represent a factorial experiment on factors determining the bitterness of wine with 1 = “least bitter” and 5 = “most bitter”. Two treatment factors (temperature and contact) each have two levels. Temperature and contact between juice and skins can be controlled when crushing grapes during wine production. Nine judges each assessed wine from two bottles from each of the four treatment conditions, hence there are 72 observations in all. The main objective is to examine the effect of contact and temperature on the perceived bitterness of wine.

Initially we consider the following cumulative link model for the wine data:

$$\begin{gathered}\operatorname{logit}(P(Y_i \leq j)) = \theta_j - \beta_1(\texttt{temp}_i) - \beta_2(\texttt{contact}_i) \\ i = 1, \ldots, n, \quad j = 1, \ldots, J-1\end{gathered} \tag{10}$$

where $\beta_1(\texttt{temp}_i)$ attains the values $\beta_1(\texttt{cold})$ and $\beta_1(\texttt{warm})$, and $\beta_2(\texttt{contact}_i)$ attains the values $\beta_2(\texttt{no})$ and $\beta_2(\texttt{yes})$. The effect of temperature in this model is illustrated in Figure 1.

This is a model for the cumulative probability of the ith rating falling in the jth category or below, where i index all observations ($n = 72$), $j = 1, \ldots, J$ index the response categories ($J = 5$) and θ_j is the intercept or threshold for the jth cumulative logit: $\operatorname{logit}(P(Y_i \leq j))$.

		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 4: The number of ratings from nine judges in bitterness categories 1 — 5. Wine data from Randall (1989) aggregated over bottles and judges.

Fitting the model with `clm` we obtain:

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

formula: rating ~ temp + contact
data:    wine

 link  threshold nobs logLik AIC    niter max.grad cond.H
 logit flexible  72   -86.49 184.98 6(0)  4.02e-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 `summary` method prints basic information about the fitted model. The primary result is the coefficient table with parameter estimates, standard errors and Wald based p values for tests of the parameters being zero. If one of the flexible link functions (`link = "log-gamma"` or `link = "Aranda-Ordaz"`) is used a coefficient table for the link parameter, λ is also included. The maximum likelihood estimates of the model coefficients are:

$$\begin{gathered}\hat{\beta}_1(\texttt{warm} - \texttt{cold}) = 2.50, \quad \hat{\beta}_2(\texttt{yes} - \texttt{no}) = 1.53, \\ \{\hat{\theta}_j\} = \{-1.34,\ 1.25,\ 3.47,\ 5.01\}.\end{gathered} \tag{11}$$

The coefficients for `temp` and `contact` are positive indicating that higher temperature and contact increase the bitterness of wine, i.e., rating in higher categories is more likely. Because

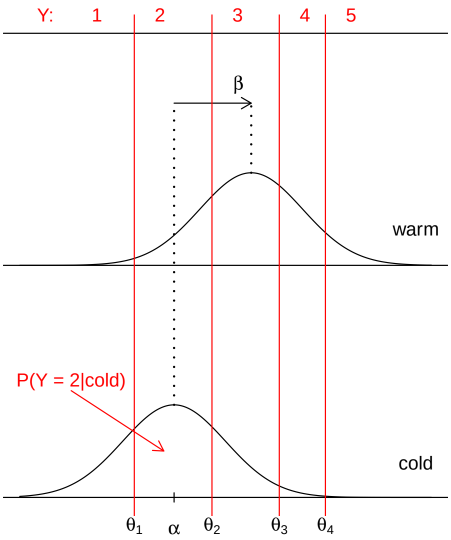

Figure 1: Illustration of the effect of temperature in the standard cumulative link model in Equation 10 for the wine data in Table 4 through a latent variable interpretation.

the treatment contrast coding which is the default in R was used, $\{\hat{\theta}_j\}$ refers to the thresholds at the setting with $\texttt{temp}_i = \texttt{cold}$ and $\texttt{contact}_i = \texttt{no}$. Three natural and complementing interpretations of this model are

1. The thresholds $\{\hat{\theta}_j\}$ at $\texttt{contact}_i = \texttt{yes}$ conditions have been shifted a constant amount 1.53 relative to the thresholds $\{\hat{\theta}_j\}$ at $\texttt{contact}_i = \texttt{no}$ conditions.

2. The location of the latent distribution has been shifted $+1.53\sigma^*$ (scale units) at $\texttt{contact}_i = \texttt{yes}$ relative to $\texttt{contact}_i = \texttt{no}$.

3. The odds ratio of bitterness being rated in category j or above ($\text{OR}(Y \geq j)$) is $\exp(\hat{\beta}_2(\texttt{yes} - \texttt{no})) = 4.61$.

Note that there are no p values displayed for the threshold coefficients because it usually does not make sense to test the hypothesis that they equal zero.

The number of Newton-Raphson iterations is given below `niter` with the number of step-halvings in parenthesis. `max.grad` is the maximum absolute gradient of the log-likelihood function with respect to the parameters. The condition number of the Hessian (`cond.H`) is well below 10^4 and so does not indicate a problem with the model.

The `anova` method produces an analysis of deviance (ANODE) table also based on Wald χ^2-tests and provides tables with type I, II and III hypothesis tests using the SAS definitions. A type I table, the R default for linear models fitted with `lm`, sequentially tests terms from first to last, type II tests attempt to respect the principle of marginality and test each term after all others while ignoring higher order interactions, and type III tables are based on orthogonalized contrasts and tests of main effects or lower order terms can often be interpreted as averaged over higher order terms. Note that in this implementation any type of contrasts (e.g., `contr.treatment` or `contr.SAS` as well as `contr.sum`) can be used to produce type III tests. For further details on the interpretation and definition of type I, II and III tests, please see (Kuznetsova, Brockhoff, and Christensen 2017) and (SAS Institute Inc. 2008).

Here we illustrate with a type III ANODE table, which in this case is equivalent to type I and II tables since the variables are balanced:

```
R> anova(fm1, type = "III")

Type III Analysis of Deviance Table with Wald chi-square tests

        Df  Chisq Pr(>Chisq)
temp     1 22.417  2.195e-06 ***
contact  1 10.275   0.001348 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Likelihood ratio tests, though asymptotically equivalent to the Wald tests usually better reflect the evidence in the data. These tests can be obtained by comparing nested models with the `anova` method, for example, the likelihood ratio test of `contact` is

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

Likelihood ratio tests of cumulative link models:

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

    no.par    AIC  logLik LR.stat df Pr(>Chisq)
fm2      5 194.03 -92.013
fm1      6 184.98 -86.492  11.043  1  0.0008902 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

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

Single term deletions

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

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

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

Single term additions

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

Confidence intervals of the parameter estimates are provided by the `confint` method which by default compute the so-called profile likelihood confidence intervals:

```
R> confint(fm1)

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

The cumulative link model in Equation 10 assumes that the thresholds, $\{\theta_j\}$ are constant for all values of the remaining explanatory variables, here `temp` and `contact`. This is generally referred to as the *proportional odds assumption* or *equal slopes assumption.* We can relax this assumption in two general ways: with nominal effects and scale effects examples of which will now be presented in turn.

4.2. Partial and non-proportional odds: nominal effects

The CLM in Equation 10 specifies a structure in which the regression parameters, $\boldsymbol{\beta}$ are not allowed to vary with j or equivalently that the threshold parameters $\{\theta_j\}$ are not allowed to depend on regression variables. In the following model this assumption is relaxed and the threshold parameters are allowed to depend on `contact`. This leads to the so-called partial proportional odds for `contact`:

$$\begin{gathered}\text{logit}(P(Y_i \leq j)) = \theta_j + \tilde{\beta}_j(\texttt{contact}_i) - \beta(\texttt{temp}_i) \\ i = 1, \ldots, n, \quad j = 1, \ldots, J-1\end{gathered} \qquad (12)$$

One way to view this model is to think of two sets of thresholds being applied at conditions with and without contact as illustrated in Figure 2. The model is specified as follows with `clm`:

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

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

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

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

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

As can be seen from the output of `summary` there are no regression coefficient estimated for `contact`, but there are additional threshold coefficients estimated instead. The naming and meaning of the threshold coefficients depend on the contrast coding applied to `contact`. Here the R default treatment contrasts (`"contr.treatment"`) are used.

Here coefficients translate to the following parameter functions:

$$
\begin{aligned}
\hat{\beta}(\texttt{warm} - \texttt{cold}) &= 2.52, \\
\{\hat{\theta}_j\} &= \{-1.32,\ 1.25,\ 3.55,\ 4.66\}, \\
\{\hat{\tilde{\beta}}_j(\texttt{yes} - \texttt{no})\} &= \{-1.62,\ -1.51,\ -1.67,\ -1.05\}.
\end{aligned} \tag{13}
$$

Again $\{\theta_j\}$ refer to the thresholds at $\texttt{temp}_i = \texttt{cold}$ and $\texttt{contact}_i = \texttt{no}$ settings while the thresholds at $\texttt{temp}_i = \texttt{cold}$ and $\texttt{contact}_i = \texttt{yes}$ are $\{\hat{\theta}_j + \hat{\tilde{\beta}}_j(\texttt{yes} - \texttt{no})\}$. The odds ratio of bitterness being rated in category j or above ($\mathrm{OR}(Y \geq j)$) now depend on j: $\{\exp(-\hat{\tilde{\beta}}_j(\texttt{yes} - \texttt{no}))\} = \{5.03,\ 4.53,\ 5.34,\ 2.86\}$.

The resulting thresholds for each level of `contact`, i.e., the estimated $\mathbf{\Theta}$-matrix can be extracted with:

```
R> fm.nom$Theta

  contact       1|2        2|3      3|4      4|5
1      no -1.323043  1.2464435 3.550044 4.660247
2     yes -2.938103 -0.2651238 1.875288 3.609624
```

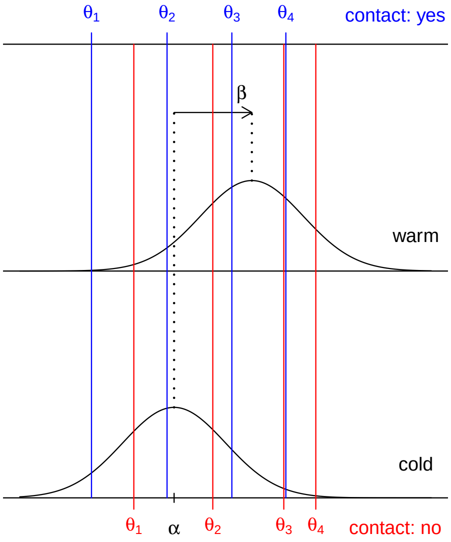

Figure 2: Illustration of nominal effects leading to different sets of thresholds being applied for each level of `contact` in a latent variable interpretation, cf., Equation 12.

As part of the convergence checks, `clm` checks the validity of $\boldsymbol{\Theta}$, i.e., that each row of the threshold matrix is non-decreasing.

We can perform a likelihood ratio test of the proportional odds assumption for `contact` by comparing the likelihoods of models (10) and (12) as follows:

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

```
Likelihood ratio tests of cumulative link models:

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

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

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

It is not possible to estimate both $\beta_2(\texttt{contact}_i)$ and $\tilde{\beta}_j(\texttt{contact}_i)$ in the same model. Consequently variables that appear in `nominal` cannot enter in `formula` as well. For instance, not all parameters are identifiable in the following model:

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

We are made aware of this when summarizing or printing the model in which the coefficient for `contactyes` is `NA`:

```
R> fm.nom2

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

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

Coefficients: (1 not defined because of singularities)
  tempwarm contactyes
     2.519         NA

Threshold coefficients:
            1|2    2|3    3|4    4|5
(Intercept) -1.323  1.246  3.550  4.660
contactyes  -1.615 -1.512 -1.675 -1.051
```

To test the proportional odds assumption for all variables, we can use

```
R> nominal_test(fm1)

Tests of nominal effects

formula: rating ~ temp + contact
        Df  logLik    AIC    LRT Pr(>Chi)
<none>      -86.492 184.98
temp     3 -84.904 187.81 3.1750   0.3654
contact  3 -86.209 190.42 0.5667   0.9040
```

This function *moves* all terms in `formula` to `nominal` and *copies* all terms in `scale` to `nominal` one by one and produces an `add1`-like table with likelihood ratio tests of each term.

4.3. Modelling scale effects

To allow the scale of the latent variable distribution to depend on explanatory variables we could for instance consider the following model where the scale is allowed to differ between cold and warm conditions. The location of the latent distribution is allowed to depend on both temperature and contact:

$$\begin{gathered}\operatorname{logit}(P(Y_i \leq j)) = \frac{\theta_j - \beta_1(\texttt{temp}_i) - \beta_2(\texttt{contact}_i)}{\exp(\zeta(\texttt{temp}_i))} \\ i = 1, \ldots, n, \quad j = 1, \ldots, J-1\end{gathered} \tag{14}$$

This model structure is illustrated in Figure 3 and can be estimated with:

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

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

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

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

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

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

In a latent variable interpretation the location of the latent distribution is shifted $2.63\sigma^*$ (scale units) from cold to warm conditions and $1.59\sigma^*$ from absence to presence of contact. The scale of the latent distribution is σ^* at cold conditions but $\sigma^* \exp(\zeta(\texttt{warm}-\texttt{cold})) = \sigma^* \exp(0.095) = 1.10\sigma^*$, i.e., 10% higher, at warm conditions. However, observe that the p value for the scale effect in the summary output shows that the ratio of scales is not significantly different from 1 (or equivalently that the difference on the log-scale is not different from 0).

Scale effects offer an alternative to nominal effects (partial proportional odds) when non-proportional odds structures are encountered in the data. Using scale effects is often a better approach because the model is well-defined for all values of the explanatory variables irrespective of translocation and scaling of covariates. Scale effects also use fewer parameters which often lead to more sensitive tests than nominal effects. Potential scale effects of variables already included in `formula` can be discovered using `scale_test`. This function adds each model term in `formula` to `scale` in turn and reports the likelihood ratio statistic in an `add1`-like fashion:

```
R> scale_test(fm1)

Tests of scale effects

formula: rating ~ temp + contact
        Df  logLik    AIC     LRT Pr(>Chi)
<none>     -86.492 184.98
```

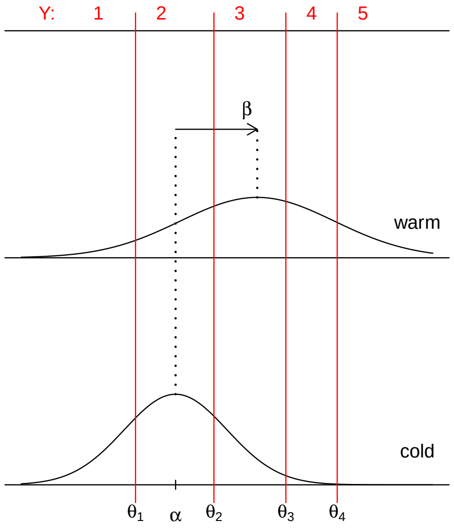

Figure 3: Illustration of scale effects leading to different scales of the latent variable, cf., Equation 14.

```
temp     1 -86.439 186.88 0.10492   0.7460
contact  1 -86.355 186.71 0.27330   0.6011
```

`confint` and `anova` methods apply with no change to models with scale and nominal parts, but `drop1`, `add1` and `step` methods will only drop or add terms to the (location) `formula`.

4.4. Structured thresholds

In section 4.2 nominal effects were described where the assumption that regression parameters have the same effect across all thresholds was relaxed. In this section additional restrictions on the thresholds will be imposed instead. The following model requires that the thresholds, $\{\theta_j\}$ are equidistant or equally spaced. This allows us to assess an assumption that judges are using the response scale in such a way that there is the same distance between adjacent response categories, i.e., that $\theta_j - \theta_{j-1} = \text{constant}$ for $j = 2, ..., J-1$. The effect of equidistant thresholds is illustrated in Figure 4 and can be fitted with:

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

formula: rating ~ temp + contact
data:    wine

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

Coefficients:
```

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

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

The parameters determining the thresholds are now the first threshold (`threshold.1`) and the spacing among consecutive thresholds (`spacing`). The mapping to this parameterization is stored in the transpose of the Jacobian matrix (`tJac`) component of the model fit. This makes it possible to extract the thresholds imposed by the equidistance structure with

```
R> drop(fm.equi$tJac %*% coef(fm.equi)[c("threshold.1", "spacing")])
```

```
      1|2       2|3       3|4       4|5
-1.001044  1.121892  3.244828  5.367764
```

These thresholds are in fact already stored in the `Theta` component of the model fit. The following shows that the average distance between consecutive thresholds in `fm1` which did not restrict the thresholds is very close to the `spacing` parameter from `fm.equi`:

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

```
[1] 2.116929
```

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

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

```
Likelihood ratio tests of cumulative link models:

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

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

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

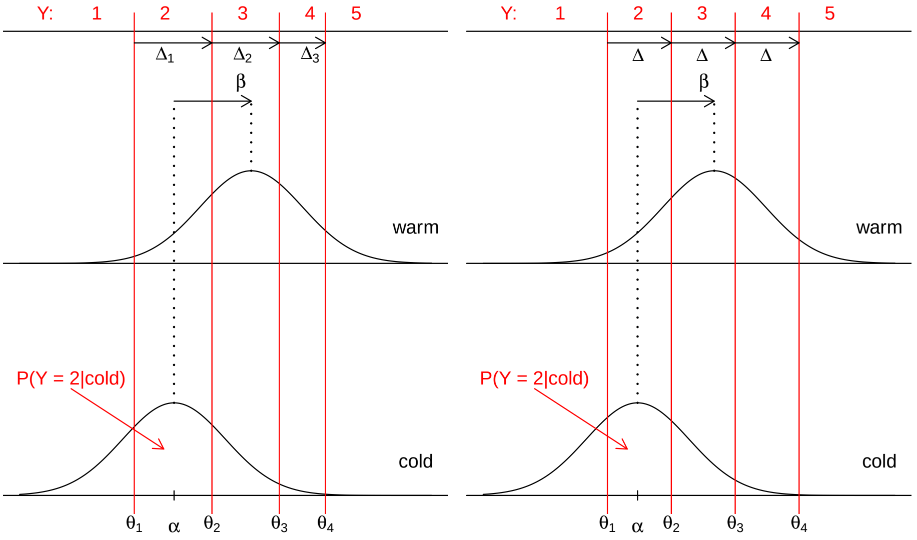

Figure 4: Illustration of flexible (left) and equidistant (right) thresholds being applied in a cumulative link model in a latent variable interpretation.

Note that the shape of the latent distribution (determined by the choice of link function) also affects the distances between the thresholds. If thresholds are equidistant under a normal distribution (i.e., with the logit link) they will in general[5] not be equidistant under a differently shaped latent distribution such as a skew latent distribution (e.g., with the log-log or clog-log link).

4.5. Scale effects, nominal effects and link functions

This section presents an example that connects aspects of scale effects, nominal effects and link functions. The example is based on the `soup` data available in the **ordinal** package. This dataset represents a sensory discrimination study of packet soup in which 185 respondents assessed a reference product and one of 5 test products on an ordinal sureness-scale with 6 levels from "reference, sure" to "test, sure".

The two key explanatory variables in this example are `PRODID` and `PROD`. `PRODID` identifies all 6 products while `PROD` distinguishes test and reference products:

```
R> with(soup, table(PROD, PRODID))
```

```
      PRODID
PROD     1   2   3   4   5   6
  Ref  739   0   0   0   0   0
  Test   0 369 184 185 185 185
```

The so-called bi-normal model plays a special role in the field of signal detection theory (DeCarlo 1998; Macmillan and Creelman 2005) and in sensometrics (Christensen, Cleaver,

---

[5] The exception is perfect fits such as CLMs with flexible thresholds and no predictors where models have the same likelihood irrespective of link function.

and Brockhoff 2011) and assumes the existence of normal latent distributions potentially with different variances. The bi-normal model can be fitted to ordinal data by identifying it as a CLM with a probit link. The following bi-normal model assumes that the location of the normal latent distribution depends on `PRODID` while the scale only varies with `PROD`:

```
R> fm_binorm <- clm(SURENESS ~ PRODID, scale = ~ PROD,
+                  data = soup, link="probit")
R> summary(fm_binorm)

formula: SURENESS ~ PRODID
scale:   ~PROD
data:    soup

 link   threshold nobs logLik   AIC     niter max.grad cond.H 
 probit flexible  1847 -2677.01 5376.02 9(1)  5.38e-13 3.4e+02

Coefficients:
        Estimate Std. Error z value Pr(>|z|)    
PRODID2  0.64203    0.09107   7.050 1.79e-12 ***
PRODID3  1.03043    0.13049   7.896 2.87e-15 ***
PRODID4  0.60131    0.11511   5.224 1.75e-07 ***
PRODID5  0.91243    0.12582   7.252 4.11e-13 ***
PRODID6  1.13821    0.13451   8.462  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

log-scale coefficients:
         Estimate Std. Error z value Pr(>|z|)    
PRODTest  0.20206    0.06129   3.297 0.000979 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Threshold coefficients:
    Estimate Std. Error z value
1|2 -0.89853    0.05245 -17.131
2|3 -0.29373    0.04385  -6.698
3|4 -0.08049    0.04309  -1.868
4|5  0.08979    0.04335   2.071
5|6  0.54868    0.04830  11.360
```

Here we observe significant differences in scale for reference and test products and this is an example of what would have been denoted non-proportional odds had the link function been the logit function. In this context differences in scale are interpreted to mean that a location shift of the latent normal distribution is not enough to represent the data. Another test of such non-location effects is provided by the nominal effects:

```
R> fm_nom <- clm(SURENESS ~ PRODID, nominal = ~ PROD,
+                data = soup, link="probit")
```

A comparison of these models shows that the scale effects increase the likelihood substantially using only one extra parameter. The addition of nominal effects provides a smaller increase in likelihood using three extra parameters:

```
R> fm_location <- update(fm_binorm, scale = ~ 1)
R> anova(fm_location, fm_binorm, fm_nom)

Likelihood ratio tests of cumulative link models:

            formula:          nominal: scale: link:  threshold:
fm_location SURENESS ~ PRODID ~1       ~1     probit flexible
fm_binorm   SURENESS ~ PRODID ~1       ~PROD  probit flexible
fm_nom      SURENESS ~ PRODID ~PROD    ~1     probit flexible

            no.par    AIC  logLik LR.stat df Pr(>Chisq)
fm_location     10 5384.9 -2682.5
fm_binorm       11 5376.0 -2677.0 10.8911  1  0.0009663 ***
fm_nom          14 5374.3 -2673.2  7.7018  3  0.0525946 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note that both the location-only and bi-normal models are nested under the model with nominal effects making these models comparable in likelihood ratio tests. This example illustrates an often seen aspect: that models allowing for scale differences frequently capture the majority of deviations from location-only effects that could otherwise be captured by nominal effects using fewer parameters.

The role of link functions in relation to the evidence of non-location effects is also illustrated by this example. If we consider the complementary log-log link it is apparent that there is no evidence of scale differences. Furthermore, the likelihood of a complementary log-log model with constant scale is almost the same as that of the bi-normal model:

```
R> fm_cll_scale <- clm(SURENESS ~ PRODID, scale = ~ PROD,
+                  data = soup, link="cloglog")
R> fm_cll <- clm(SURENESS ~ PRODID,
+                   data = soup, link="cloglog")
R> anova(fm_cll, fm_cll_scale, fm_binorm)

Likelihood ratio tests of cumulative link models:

             formula:          scale: link:   threshold:
fm_cll       SURENESS ~ PRODID ~1     cloglog flexible
fm_cll_scale SURENESS ~ PRODID ~PROD  cloglog flexible
fm_binorm    SURENESS ~ PRODID ~PROD  probit  flexible

             no.par    AIC  logLik LR.stat df Pr(>Chisq)
fm_cll           10 5374.8 -2677.4
fm_cll_scale     11 5376.5 -2677.2  0.3736  1      0.541
fm_binorm        11 5376.0 -2677.0  0.4461  0
```

Using the log-gamma link we can also confirm that a left-skewed latent distribution ($\lambda > 0$) is best supported by the data and that the estimate of λ is close to 1 at which the complementary log-log link is obtained:

```
R> fm_loggamma <- clm(SURENESS ~ PRODID, data = soup, link="log-gamma")
R> summary(fm_loggamma)

formula: SURENESS ~ PRODID
data:    soup

 link      threshold nobs logLik   AIC     niter   max.grad cond.H
 log-gamma flexible  1847 -2676.98 5375.96 80(816) 1.41e-03 1.3e+03

Coefficients:
        Estimate Std. Error z value Pr(>|z|)
PRODID2  0.61385    0.08814   6.964 3.30e-12 ***
PRODID3  1.01960    0.13830   7.373 1.67e-13 ***
PRODID4  0.57953    0.11203   5.173 2.30e-07 ***
PRODID5  0.91642    0.13259   6.912 4.79e-12 ***
PRODID6  1.12139    0.14315   7.834 4.74e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Link coefficients:
       Estimate Std. Error z value Pr(>|z|)
lambda   0.7799     0.2295   3.398  0.00068 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Threshold coefficients:
    Estimate Std. Error z value
1|2 -1.37115    0.22819  -6.009
2|3 -0.60685    0.13456  -4.510
3|4 -0.36553    0.11384  -3.211
4|5 -0.18211    0.10083  -1.806
5|6  0.27569    0.07823   3.524
```

The analysis of link functions shown here can be thought of as providing a framework analogous to that of Box-Cox transformations for linear models.

4.6. Profile likelihood

In addition to facilitating the generally quite accurate profile likelihood confidence intervals which were illustrated in section 4.1, the profile likelihood function can also be used to illustrate the relative importance of parameter values.

As an example, the profile likelihood of model coefficients for `temp` and `contact` in `fm1` can be obtained with

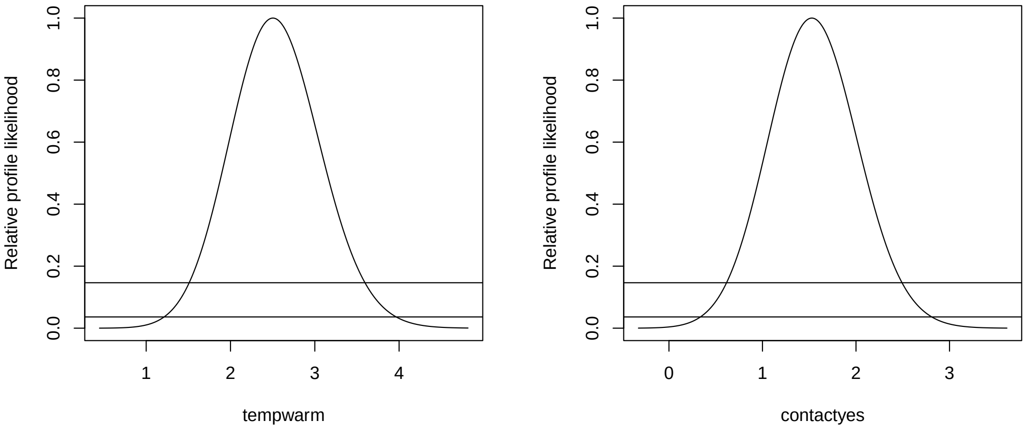

Figure 5: Relative profile likelihoods for the regression parameters in `fm1` for the wine data. Horizontal lines indicate 95% and 99% confidence bounds.

```
R> pr1 <- profile(fm1, alpha = 1e-4)
R> plot(pr1)
```

The resulting plots are provided in Figure 5. The `alpha` argument controls how far from the maximum likelihood estimate the likelihood function should be profiled: the profile strays no further from the MLE when values outside an (`1 - alpha`)-level profile likelihood confidence interval.

From the relative profile likelihood in Figure 5 for `tempwarm` we see that parameter values between 1 and 4 are reasonably well supported by the data, and values outside this range has little likelihood. Values between 2 and 3 are very well supported by the data and have high likelihood.

Profiling is implemented for regression (β) and scale (ζ) parameters but not available for threshold, nominal and flexible link parameters.

4.7. Assessment of model convergence

Likelihood slices

The maximum likelihood estimates of the parameters in cumulative link models do not have closed form expressions, so iterative methods have to be applied to fit the models. Further, CLMs are non-linear models and in general the likelihood function is not guaranteed to be well-behaved or even uni-model. In addition, the special role of the threshold parameters and the restriction on them being ordered can affect the appearance of the likelihood function.

To confirm that an unequivocal optimum has been reached and that the likelihood function is reasonably well-behaved around the reported optimum we can inspect the likelihood function in a neighborhood around the reported optimum. For these purposes we can display slices of the likelihood function.

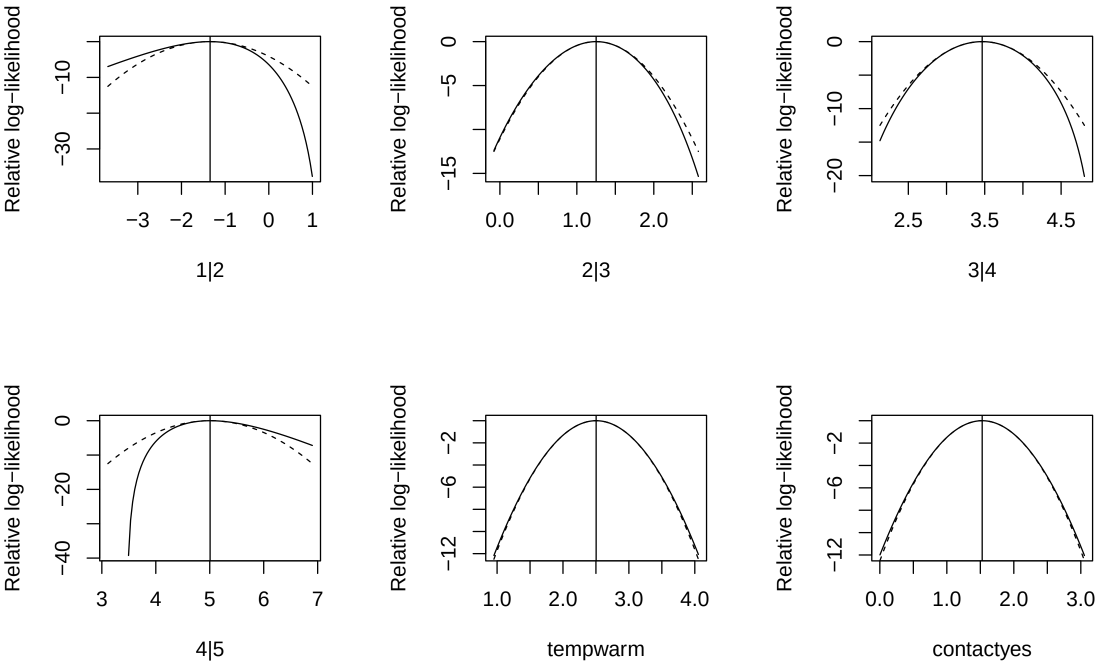

Figure 6: Slices of the (negative) log-likelihood function (solid) for parameters in `fm1` for the wine data. Dashed lines indicate quadratic approximations to the log-likelihood function and vertical bars indicate maximum likelihood estimates.

The following code produces the slices shown in Figure 6 which displays the shape of the log-likelihood function in a fairly wide neighborhood around the reported MLE; here we use $\lambda = 5$ curvature units, as well as it's quadratic approximation.

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

Figure 6 shows that log-likelihood function is fairly well behaved and relatively closely quadratic for most parameters.

Looking at the log-likelihood function much closer to the reported optimum (using $\lambda = 10^{-5}$) we can probe how accurately the parameter estimates are determined. The likelihood slices in Figure 7 which are produced with the following code shows that the parameters are determined accurately with at least 5 correct decimals. Slices are shown for two parameters and the slices for the remaining 4 parameters are very similar.

```
R> slice2.fm1 <- slice(fm1, parm = 4:5, lambda = 1e-5)
R> par(mfrow = c(1, 2))
R> plot(slice2.fm1)
```

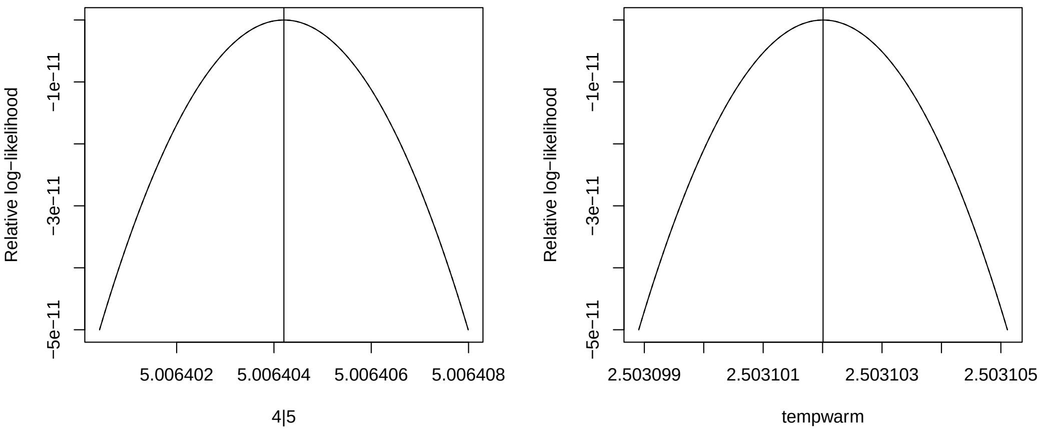

Figure 7: Slices of the (negative) log-likelihood function (solid) for parameters in `fm1` for the wine data very close to the MLEs. Dashed lines (indistinguishable from the solid lines) indicate quadratic approximations to the log-likelihood function and vertical bars the indicate maximum likelihood estimates.

Parameter accuracy

As discussed in section A the method independent error estimate provides an assessment of the accuracy with which the ML estimates of the parameters have been determined by the fitting algorithm. This error estimate is implemented in the `convergence` method which we now illustrate on a model fit:

```
R> convergence(fm1)
```

```
 nobs logLik niter max.grad cond.H  logLik.Error
 72   -86.49 6(0)  4.02e-12 2.7e+01 <1e-10

           Estimate Std.Err  Gradient     Error Cor.Dec Sig.Dig
1|2          -1.344  0.5171  2.06e-12  3.09e-13      12      13
2|3           1.251  0.4379  2.11e-12 -2.43e-13      12      13
3|4           3.467  0.5978 -4.02e-12 -9.33e-13      11      12
4|5           5.006  0.7309 -7.04e-14 -9.21e-13      11      12
tempwarm      2.503  0.5287 -4.54e-13 -6.33e-13      11      12
contactyes    1.528  0.4766  5.38e-14 -2.96e-13      12      13

Eigen values of Hessian:
21.7090 18.5615 10.3914  5.2093  4.0955  0.8163

Convergence message from clm:
(0) successful convergence
In addition: Absolute and relative convergence criteria were met
```

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

Note that the assessment of the number of correctly determined decimals and significant digits is only reliable sufficiently close to the optimum so in practice we caution against this assessment if the algorithm did not converge successfully.

4.8. Fitted values and predictions

Several types of fitted values and predictions can be extracted from a CLM depending on how it is viewed.

By *fitted values* we denote the values ($i = 1, \ldots, n$)

$$\hat{\tilde{\pi}}_i = \tilde{\pi}_i(\hat{\boldsymbol{\psi}})$$

that is, the value of $\tilde{\pi}_i$, cf., Equation 3 evaluated at the ML estimates $\hat{\boldsymbol{\psi}}$. These are the values returned by the `fitted` and `fitted.values` extractor methods and stored in the `fitted.values` component of the model fit.

The values of π_{ij} (cf., Equation 2) evaluated at the ML estimates of the parameters (i.e., $\hat{\pi}_{ij}$) can also be thought of as fitted values for the multinomially distributed variable $\boldsymbol{Y}_i^*$. These values can be obtained from the model fit by use of the `predict` method:

```
R> head(pred <- predict(fm1, newdata = subset(wine, select = -rating))$fit)

           1         2         3          4          5
1 0.20679013 0.5706497 0.1922909 0.02361882 0.00665041
2 0.20679013 0.5706497 0.1922909 0.02361882 0.00665041
3 0.05354601 0.3776461 0.4430599 0.09582084 0.02992711
4 0.05354601 0.3776461 0.4430599 0.09582084 0.02992711
5 0.02088771 0.2014157 0.5015755 0.20049402 0.07562701
6 0.02088771 0.2014157 0.5015755 0.20049402 0.07562701
```

Note that the original data set should be supplied in the `newdata` argument *without* the response variable (here `rating`). If the response variable is *present* in `newdata` predictions are produced for only those rating categories which were observed and we get back the fitted values:

```
R> stopifnot(isTRUE(all.equal(fitted(fm1), t(pred)[
+    t(col(pred) == wine$rating)])),
+    isTRUE(all.equal(fitted(fm1), predict(fm1, newdata = wine)$fit)))
```

Class predictions are also available and defined here as the response class with the highest probability, that is, for the i'th observation the class prediction is the mode of $\boldsymbol{\pi}_i$. To obtain class predictions use `type = "class"` as illustrated in the following small table:

```
R> newData <- expand.grid(temp    = levels(wine$temp),
+                         contact = levels(wine$contact))
R> cbind(newData, round(predict(fm1, newdata = newData)$fit, 3),
+       "class" = predict(fm1, newdata = newData, type = "class")$fit)
```

```
  temp contact     1     2     3     4     5 class
1 cold      no 0.207 0.571 0.192 0.024 0.007     2
2 warm      no 0.021 0.201 0.502 0.200 0.076     3
3 cold     yes 0.054 0.378 0.443 0.096 0.030     3
4 warm     yes 0.005 0.054 0.304 0.364 0.274     4
```

Other definitions of class predictions can be applied, e.g., nearest mean predictions:

```
R> head(apply(pred, 1, function(x) round(weighted.mean(1:5, x))))
```

```
1 2 3 4 5 6
2 2 3 3 3 3
```

which in this case happens to be identical to the default class predictions.

Standard errors and confidence intervals of predictions are also available, for example:

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

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

where the default 95% confidence level can be changed with the `level` argument.

Here the standard errors of fitted values or predictions, $\hat{\tilde{\pi}} = \tilde{\pi}(\hat{\boldsymbol{\psi}})$ are obtained by application of the delta method:

$$\mathsf{Var}(\hat{\tilde{\boldsymbol{\pi}}}) = \boldsymbol{C}\mathsf{Var}(\hat{\boldsymbol{\psi}})\boldsymbol{C}^\top, \quad \boldsymbol{C} = \frac{\partial \tilde{\boldsymbol{\pi}}(\boldsymbol{\psi})}{\partial \boldsymbol{\psi}}\Big|_{\boldsymbol{\psi}=\hat{\boldsymbol{\psi}}}$$

where $\mathsf{Var}(\hat{\boldsymbol{\psi}})$ is the estimated variance-covariance matrix of the parameters $\boldsymbol{\psi}$ evaluated at the ML estimates $\hat{\boldsymbol{\psi}}$ as given by the observed Fisher Information matrix and finally the standard errors are extracted as the square root of the diagonal elements of $\mathsf{Var}(\hat{\tilde{\boldsymbol{\pi}}})$.

Since symmetric confidence intervals for probabilities are not appropriate unless perhaps if they are close to one half a more generally applicable approach is to form symmetric Wald intervals on the logit scale and then subsequently transform the confidence bounds to the probability scale. `predict.clm` takes this approach and computes the standard error of $\hat{\kappa}_i = \text{logit}(\hat{\hat{\pi}}_i)$ by yet an application of the delta method:

$$\text{se}(\hat{\kappa}_i) = \frac{\partial g(\hat{\hat{\pi}}_i)}{\partial \hat{\hat{\pi}}_i} \text{se}(\hat{\hat{\pi}}_i) = \frac{\text{se}(\hat{\hat{\pi}}_i)}{\hat{\hat{\pi}}_i(1 - \hat{\hat{\pi}}_i)}, \quad g(\hat{\hat{\pi}}_i) = \log \frac{\hat{\hat{\pi}}_i}{1 - \hat{\hat{\pi}}_i}.$$

4.9. Model identifiability

Unidentifiable models or unidentifiable parameters may happen in CLMs for several reasons some of which are special to the model class. In this section we describe issues around model identifiability and how this is handled by `ordinal::clm`.

Material in the remainder of this section is generally on a more advanced level than up to now.

Complete separation

In binary logistic regression the issue of *complete separation* is well known. This may happen, for example if only "success" or only "failure" is observed for a level of a treatment factor. In CLMs the issue may appear even when outcomes are observed in more than one response category. This can be illustrated using the `wine` data set if we combine the three central categories:

```
R> wine <- within(wine, {
+    rating_comb3 <- factor(rating, labels = c("1", "2-4", "2-4", "2-4", "5"))
+  })
R> ftable(rating_comb3 ~ temp, data = wine)

     rating_comb3  1 2-4  5
temp
cold               5  31  0
warm               0  29  7

R> fm.comb3 <- clm(rating_comb3 ~ temp, data = wine)
R> summary(fm.comb3)

formula: rating_comb3 ~ temp
data:    wine

 link  threshold nobs logLik AIC   niter max.grad cond.H
 logit flexible  72   -32.24 70.48 22(0) 3.06e-09 5.0e+09

Coefficients:
         Estimate Std. Error z value Pr(>|z|)
tempwarm    21.89         NA      NA       NA

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

```
1|2-4   -1.825          NA      NA
2-4|5   23.310          NA      NA
```

Here the true ML estimates of the coefficients for `temp` and the second threshold are at infinity but the algorithm in `clm` terminates when the likelihood function is sufficiently flat. This means that the reported values of the coefficients for `temp` and the second threshold are arbitrary and will change if the convergence criteria are changed or a different optimization method is used. The standard errors of the coefficients are not available because the Hessian is effectively singular and so cannot be inverted to produce the variance-covariance matrix of the parameters. The ill-determined nature of the Hessian is seen from the very large condition number of the Hessian, `cond.H`.

Note, however, that while the model parameters cannot be uniquely determined, the likelihood of the model is well defined and as such it can be compared to the likelihood of other models. For example, we could compare it to a model that excludes `temp`

```
R> fm.comb3_b <- clm(rating_comb3 ~ 1, data = wine)
R> anova(fm.comb3, fm.comb3_b)

Likelihood ratio tests of cumulative link models:

           formula:             link: threshold:
fm.comb3_b rating_comb3 ~ 1    logit flexible
fm.comb3   rating_comb3 ~ temp logit flexible

           no.par    AIC  logLik LR.stat df Pr(>Chisq)
fm.comb3_b      2 85.181 -40.591
fm.comb3        3 70.479 -32.240  16.702  1  4.373e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The difference in log-likelihood is substantial, however, the criteria for the validity of the likelihood ratio test are not fulfilled, so the p value should not be taken at face value.

The complete-separation issue may also appear in less obvious situations. If, for example, the following model is considered allowing for nominal effects of `temp` the issue shows up:

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

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

 link  threshold nobs logLik AIC    niter max.grad cond.H
 logit flexible  72   -84.90 187.81 20(0) 4.35e-09 4.2e+10

Coefficients:
```

```
           Estimate Std. Error z value Pr(>|z|)
contactyes    1.465         NA      NA       NA

Threshold coefficients:
                Estimate Std. Error z value
1|2.(Intercept)   -1.266         NA      NA
2|3.(Intercept)    1.104         NA      NA
3|4.(Intercept)    3.766         NA      NA
4|5.(Intercept)   24.896         NA      NA
1|2.tempwarm     -21.095         NA      NA
2|3.tempwarm      -2.153         NA      NA
3|4.tempwarm      -2.873         NA      NA
4|5.tempwarm     -22.550         NA      NA
```

Analytical detection of which coefficients suffer from unidentifiability due to *complete separation* is a topic for future research and therefore unavailable in current versions of **ordinal**.

Aliased coefficients

Aliased coefficients can occur in all kinds of models that build on a design matrix including linear models as well as generalized linear models. `lm` and `glm` determine the rank deficiency of the design matrix using the rank-revealing implementation of the QR-decomposition in `LINPACK` and displays the aliased coefficients as `NA`s[6]. Though the QR decomposition is not used during iterations in `clm`, it is used initially to determine aliased coefficients. An example is provided using the `soup` data available in the **ordinal** package:

```
R> fm.soup <- clm(SURENESS ~ PRODID * DAY, data = soup)
R> summary(fm.soup)

formula: SURENESS ~ PRODID * DAY
data:    soup

 link  threshold nobs logLik   AIC     niter max.grad cond.H
 logit flexible  1847 -2672.08 5374.16 6(1)  1.41e-13 9.4e+02

Coefficients: (1 not defined because of singularities)
             Estimate Std. Error z value Pr(>|z|)
PRODID2        0.6665     0.2146   3.106  0.00189 **
PRODID3        1.2418     0.1784   6.959 3.42e-12 ***
PRODID4        0.6678     0.2197   3.040  0.00237 **
PRODID5        1.1194     0.2400   4.663 3.11e-06 ***
PRODID6        1.3503     0.2337   5.779 7.53e-09 ***
DAY2          -0.4134     0.1298  -3.186  0.00144 **
PRODID2:DAY2   0.4390     0.2590   1.695  0.09006 .
PRODID3:DAY2       NA         NA      NA       NA
PRODID4:DAY2   0.3308     0.3056   1.083  0.27892
```

---

[6] if the `singular.ok = TRUE` which is the default.

```
PRODID5:DAY2   0.3871     0.3248   1.192  0.23329
PRODID6:DAY2   0.5067     0.3350   1.513  0.13030
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

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

The source of the singularity is revealed in the following table:

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

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

which shows that the third `PRODID` was not presented at the second day.

The issue of aliased coefficients extends in CLMs to nominal effects since the joint design matrix for location and nominal effects will be singular if the same variables are included in both location and nominal formulae. `clm` handles this by not estimating the offending coefficients in the location formula as illustrated with the `fm.nom2` model fit in section 4.2.

Over parameterization

The scope of model structures allowed in `clm` makes it possible to specify models which are over parameterized in ways that do not lead to rank deficient design matrices and as such are not easily detected before fitting the model. An example is given here which includes both additive (location) and multiplicative (scale) effects of `contact` for a binomial response variable but the issue can also occur with more than two response categories:

```
R> wine <- within(wine, {
+    rating_comb2 <- factor(rating, labels = c("1-2", "1-2", "3-5", "3-5", "3-5"))
+  })
R> ftable(rating_comb2 ~ contact, data = wine)
```

```
        rating_comb2 1-2 3-5
contact
no                    18  18
yes                    9  27
```

```
R> fm.comb2 <- clm(rating_comb2 ~ contact, scale = ~ contact, data = wine)
R> summary(fm.comb2)
```

```
formula: rating_comb2 ~ contact
scale:   ~contact
data:    wine

 link  threshold nobs logLik AIC   niter max.grad cond.H 
 logit flexible  72   -45.20 96.39 4(0)  1.38e-11 -4.1e+12

Coefficients:
           Estimate Std. Error z value Pr(>|z|)
contactyes    1.099         NA      NA       NA

log-scale coefficients:
             Estimate Std. Error z value Pr(>|z|)
contactyes -1.174e-06         NA      NA       NA

Threshold coefficients:
          Estimate Std. Error z value
1-2|3-5 -7.171e-17         NA      NA
```

4.10. Customized modelling

Using the `doFit` argument `clm` can be instructed to return a *model environment* that we denote `rho`:

```
R> rho <- update(fm1, doFit=FALSE)
R> names(rho)
```

```
 [1] "par"        "nlambda"   "link"      "gfun"      "dfun"     
 [6] "pfun"       "clm.hess"  "clm.grad"  "clm.nll"   "wts"      
[11] "fitted"     "has.scale" "sigma"     "k"         "Soff"     
[16] "S"          "n.psi"     "o2"        "o1"        "B2"       
[21] "B1"
```

This environment holds a complete specification of the cumulative link models including design matrices `B1`, `B2`, `S` and other components. The environment also contains the cumulative distribution function that defines the inverse link function `pfun` and its first and second derivatives, i.e., the corresponding density function `dfun` and gradient `gfun`. Of direct interest here is the parameter vector `par` and functions that readily evaluate the negative log-likelihood (`clm.nll`), its gradient with respect to the parameters (`clm.grad`) and the Hessian (`clm.hess`). The negative log-likelihood and the gradient at the starting values is therefore

```
R> rho$clm.nll(rho)
```

```
[1] 115.8795
```

```
R> c(rho$clm.grad(rho))
```

```
[1]  13.6   4.8 -16.8  -4.0  -7.2  -3.6
```

Similarly at the MLE they are:

```
R> rho$clm.nll(rho, par = coef(fm1))
```

```
[1] 86.49192
```

```
R> print(c(rho$clm.grad(rho)), digits = 3)
```

```
[1]  2.06e-12  2.11e-12 -4.02e-12 -7.04e-14 -4.54e-13  5.38e-14
```

Note that the gradient function `clm.grad` assumes that `clm.nll` has been evaluated at the current parameter values; similarly, `clm.hess` assumes that `clm.grad` has been evaluated at the current parameter values. The NR algorithm in **ordinal** takes advantage of this so as to minimize the computational load.

If interest is in fitting a *custom* CLM with, say, restrictions on the parameter space, this can be achieved by a combination of a general purpose optimizer and the functions `clm.nll` and optionally `clm.grad`. Assume for instance we know that the regression parameters can be no larger than 2, then the model can be fitted with the following code:

```
R> nll <- function(par, envir) {
+    envir$par <- par
+    envir$clm.nll(envir)
+  }
R> grad <- function(par, envir) {
+    envir$par <- par
+    envir$clm.nll(envir)
+    envir$clm.grad(envir)
+  }
R> nlminb(rho$par, nll, grad, upper = c(rep(Inf, 4), 2, 2), envir = rho)$par
```

```
       1|2        2|3        3|4        4|5   tempwarm contactyes
 -1.470404   1.014029   3.073575   4.548003   2.000000   1.444449
```

Constrained partial proportional odds

A type of models which are not implemented in full generality in **ordinal** are the so-called *constrained* partial proportional odds models proposed by Peterson and Harrell Jr. (1990). These models impose restrictions on the nominal effects considered in section 4.2 and are well suited to illustrate the customisable modelling options available in the **ordinal** package. We consider an example from Peterson and Harrell Jr. (1990) in which disease status is tabulated by smoking status:

```
R> artery <- data.frame(disease = factor(rep(0:4, 2), ordered = TRUE),
+                       smoker  = factor(rep(c("no", "yes"), each = 5)),
+                       freq    = c(334, 99, 117, 159, 30, 350, 307,
+                                   345, 481, 67))
R> addmargins(xtabs(freq ~ smoker + disease, data = artery), margin = 2)
```

```
      disease
smoker    0    1    2    3    4  Sum
   no   334   99  117  159   30  739
   yes  350  307  345  481   67 1550
```

The overall odds-ratio of smoking is

```
R> fm <- clm(disease ~ smoker, weights = freq, data = artery)
R> exp(fm$beta)
```

```
smokeryes
 2.090131
```

showing that overall the odds of worse disease rating is twice as high for smokers compared to non-smokers.

Allowing for nominal effects we see that the log odds-ratio for smoking clearly changes with disease status, and that it does so in an almost linearly decreasing manor:

```
R> fm.nom <- clm(disease ~ 1, nominal = ~ smoker, weights = freq,
+                 data = artery, sign.nominal = "negative")
R> coef(fm.nom)[5:8]
```

```
0|1.smokeryes 1|2.smokeryes 2|3.smokeryes 3|4.smokeryes
   1.03939761    0.65405519    0.46469327    0.06552842
```

Peterson and Harrell Jr. (1990) suggested a model which restricts the log odds-ratios to be linearly decreasing with disease status modelling only the intercept (first threshold) and slope of the log odds-ratios:

```
R> coef(fm.lm <- lm(I(coef(fm.nom)[5:8]) ~ I(0:3)))
```

```
(Intercept)      I(0:3)
  1.0225640  -0.3110969
```

We can implement the log-likelihood of this model as follows. As starting values we combine parameter estimates from `fm.nom` and the linear model `fm.lm`, and finally optimize the log-likelihood utilizing the `fm.nom` model environment:

```
R> nll2 <- function(par, envir) {
+    envir$par <- c(par[1:4], par[5] + par[6] * (0:3))
+    envir$clm.nll(envir)
+  }
R> start <- unname(c(coef(fm.nom)[1:4], coef(fm.lm)))
R> fit <- nlminb(start, nll2, envir = update(fm.nom, doFit = FALSE))
R> round(fit$par[5:6], 2)
```

```
[1]  1.02 -0.30
```

Thus the log-odds decrease linearly from 1.02 for the first two disease categories by 0.3 per disease category.

5. Conclusions

This paper has described the class of cumulative link models for the analysis of ordinal data and the implementation of such models in the R package **ordinal**. It is shown how the package supports model building and assessment of CLMs with scale effects, partial proportional odds, structured thresholds, flexible link functions and how models can be costumized to specific needs. A number of examples have been given illustrating analyses of ordinal data using `clm` in practice.

The significant flexibility of model structures available in **ordinal** is in one respect a clear advantage but it can also be a challenge when particular model variants turn out to be unidentifiable. Analytical detection of unidentifiable models could prove very useful in the analysis of ordinal data, but it is, unfortunately, a difficult question that remains a topic of future research.

In a wider data analysis perspective, cumulative link models have been described as a very rich model class—a class that sits in between, in a sense, the perhaps the two most important model classes in statistics; linear models and logistic regression models. The greater flexibility of CLMs relative to binary logistic regression models facilitates the ability to check assumptions such as the partial proportional odds assumption. A latent variable interpretation connects cumulative link models to linear models in a natural way and also motivates non-linear structures such as scale effects. In addition to nominal effects and the non-linear scale effects, the ordered nature of the thresholds gives rise to computational challenges that we have described here and addressed in the **ordinal** package.

In addition to computational challenges, practical data analysis with CLMs can also be challenging. In our experience a top-down approach in which a "full" model is fitted and gradually simplified is often problematic, not only because this easily leads to unidentifiable models but also because there are many different ways in which models can be reduced or expanded. A more pragmatic approach is often preferred; understanding the data through plots, tables, and even linear models can aid in finding a suitable intermediate ordinal starting model.

Attempts to identify a "correct" model will also often lead to frustrations; the greater the model framework, the greater the risk that there are multiple models which fit the data (almost) equally well. It is well known statistical wisdom that with enough data many goodness of fit tests become sensitive to even minor deviations of little practical relevance. This is particularly true for tests of partial proportional odds; in the author's experience almost all CLMs on real data show some evidence of non-proportional odds for one or more variables but it is not always the case that models with partial or non-proportional odds are the most useful. Such effects complicate the interpretation and often generalize poorly outside the observed data and models assuming proportional odds or including scale effects are often more appropriate.

Computational details

The results in this paper were obtained using R 4.4.1 with **ordinal**, version 2023.12.4.1. R itself and all packages used are available from CRAN at `https://CRAN.R-project.org/`.

References

Agresti A (2002). *Categorical Data Analysis*. 3rd edition. John Wiley & Sons. ISBN 978-0470463635.

Ananth CV, Kleinbaum DG (1997). "Regression Models for Ordinal Responses: A Review of Methods and Applications." *International Journal of Epidemiology*, **26**(6), 1323–1333. ISSN 0300-5771. `doi:10.1093/ije/26.6.1323`. `https://academic.oup.com/ije/article-pdf/26/6/1323/18477637/261323.pdf`, URL `https://doi.org/10.1093/ije/26.6.1323`.

Aranda-Ordaz FJ (1983). "An Extension of the Proportional-Hazards Model for Grouped Data." *Biometrics*, **39**, 109–117. `doi:10.2307/2530811`.

Archer KJ, Hou J, Zhou Q, Ferber K, Layne JG, Gentry AE (2014). "**ordinalgmifs**: An R Package for Ordinal Regression in High-dimensional Data Settings." *Cancer Informatics*, **13**, 187–195. `doi:10.4137/CIN.S20806`. URL `http://www.la-press.com/article.php?article_id=4569`.

Brazzale AR, Davison AC, Reid N (2007). *Applied Asymptotics—Case Studies in Small-Sample Statistics*. Cambridge University Press. ISBN 9780521847032.

Burridge J (1981). "A Note on Maximum Likelihood Estimation for Regression Models Using Grouped Data." *Journal of the Royal Statistical Society B*, **43**(1), 41–45. ISSN 00359246.

Bürkner PC (2017). "**brms**: An R Package for Bayesian Multilevel Models Using **Stan**." *Journal of Statistical Software*, **80**(1), 1–28. `doi:10.18637/jss.v080.i01`.

Carroll N (2018). ***oglmx**: Estimation of Ordered Generalized Linear Models*. R package version 3.0.0.0, URL `https://CRAN.R-project.org/package=oglmx`.

Christensen RHB (2012). *Sensometrics: Thurstonian and Statistical Models*. Ph.D. thesis, Technical University of Denmark (DTU). URL `http://orbit.dtu.dk/files/12270008/phd271_Rune_Haubo_net.pdf`.

Christensen RHB (2019). "**ordinal**—Regression Models for Ordinal Data." R package version 2019.12-10, URL `http://www.cran.r-project.org/package=ordinal/`.

Christensen RHB, Cleaver G, Brockhoff PB (2011). "Statistical and Thurstonian Models for the A-not A Protocol with and without Sureness." *Food Quality and Preference*, **22**, 542–549. `doi:10.1016/j.foodqual.2011.03.003`.

Cox C (1995). "Location-Scale Cumulative Odds Models for Ordinal Data: A Generalized Non-Linear Model Approach." *Statistics in Medicine*, **14**, 1191–1203. `doi:10.1002/sim.4780141105`.

DeCarlo LT (1998). "Signal Detection Theory and Generalized Linear Models." *Psychological Methods*, **3**(2), 185–205. `doi:10.1037/1082-989X.3.2.186`.

Eldén L, Wittmeyer-Koch L, Nielsen HB (2004). *Introduction to Numerical Computation — Analysis and MATLAB Illustrations*. Studentlitteratur. ISBN 978-9144037271.

Farewell VT, Prentice RL (1977). "A Study of Distributional Shape in Life Testing." *Technometrics*, **19**, 69–77. `doi:10.2307/1268257`.

Fox J, Hong J (2009). "Effect Displays in R for Multinomial and Proportional-Odds Logit Models: Extensions to the **effects** Package." *Journal of Statistical Software*, **32**(1), 1–24. URL `http://www.jstatsoft.org/v32/i01/`.

Fox J, Weisberg S (2018). "Visualizing Fit and Lack of Fit in Complex Regression Models with Predictor Effect Plots and Partial Residuals." *Journal of Statistical Software*, **87**(9), 1–27. `doi:10.18637/jss.v087.i09`. URL `https://www.jstatsoft.org/v087/i09`.

Genter FC, Farewell VT (1985). "Goodness-of-Link Testing in Ordinal Regression Models." *The Canadian Journal of Statistics*, **13**(1), 37–44. `doi:10.2307/3315165`.

Greene WH, Hensher DA (2010). *Modeling Ordered Choices: A Primer*. Cambridge University Press.

Harrell Jr FE (2018). ***rms**: Regression Modeling Strategies*. R package version 5.1-2, URL `https://CRAN.R-project.org/package=rms`.

Hirk R, Hornik K, Vana L (2020). "**mvord**: An R Package for Fitting Multivariate Ordinal Regression Models." *Journal of Statistical Software*, **93**(4), 1–41. `doi:10.18637/jss.v093.i04`.

Iannario M, Piccolo D, Simone R (2020). ***CUB**: A Class of Mixture Models for Ordinal Data*. R package version 1.1.4, URL `https://CRAN.R-project.org/package=CUB`.

IBM Corp (2017). *IBM SPSS Statistics for Windows, Version 25.0*. IBM Corp., Armonk, NY.

Jay M (2019). ***generalhoslem**: Goodness of Fit Tests for Logistic Regression Models*. R package version 1.3.4, URL `https://CRAN.R-project.org/package=generalhoslem`.

Kuznetsova A, Brockhoff P, Christensen R (2017). "**lmerTest** Package: Tests in Linear Mixed Effects Models." *Journal of Statistical Software, Articles*, **82**(13), 1–26. ISSN 1548-7660. `doi:10.18637/jss.v082.i13`. URL `https://www.jstatsoft.org/v082/i13`.

Leeper TJ (2018). ***margins**: Marginal Effects for Model Objects*. R package version 0.3.23.

Lenth R (2020). ***emmeans**: Estimated Marginal Means, aka Least-Squares Means*. R package version 1.4.6, URL `https://CRAN.R-project.org/package=emmeans`.

Lüdecke D (2018). "**ggeffects**: Tidy Data Frames of Marginal Effects from Regression Models." *Journal of Open Source Software*, **3**(26), 772. `doi:10.21105/joss.00772`.

Macmillan NA, Creelman CD (2005). *Detection Theory, A User's Guide*. 2nd edition. Lawrence Elbaum Associates, Publishers. ISBN 978-0805842319.

Martin AD, Quinn KM, Park JH (2011). "**MCMCpack**: Markov Chain Monte Carlo in R." *Journal of Statistical Software*, **42**(9), 22. `doi:10.18637/jss.v042.i09`. URL `http://www.jstatsoft.org/v42/i09/`.

McCullagh P (1980). "Regression Models for Ordinal Data." *Journal of the Royal Statistical Society B*, **42**, 109–142.

Nielsen HB, Mortensen SB (2016). ***ucminf**: General-Purpose Unconstrained Non-Linear Optimization*. R package version 1.1-4, URL `https://CRAN.R-project.org/package=ucminf`.

Pawitan Y (2001). *In All Likelihood—Statistical Modelling and Inference Using Likelihood*. Oxford University Press. ISBN 978-0198507659.

Pedregosa-Izquierdo F (2015). *Feature Extraction and Supervised Learning on fMRI: From Practice to Theory*. Ph.D. thesis, Université Pierre et Marie Curie, Paris VI. URL `https://pythonhosted.org/mord/`.

Peterson B, Harrell Jr FE (1990). "Partial Proportional Odds Models for Ordinal Response Variables." *Applied Statistics*, **39**, 205–217. `doi:10.2307/2347760`.

Pratt JW (1981). "Concavity of the Log Likelihood." *Journal of the American Statistical Association*, **76**(373), 103–106. ISSN 01621459. `doi:10.2307/2287052`.

Matlab (2020). *Matlab version 9.8 (R2020a)*. The Mathworks, Inc., Natick, Massachusetts.

SAS Institute Inc (2008). *The Four Types of Estimable Functions – SAS/STAT ®9.22 User's Guide*. SAS Institute Inc., Cary, NC. URL `https://support.sas.com/documentation/cdl/en/statugestimable/61763/PDF/default/statugestimable.pdf`.

SAS Institute Inc (2010). *SAS/STAT ®9.22 User's Guide*. SAS Institute Inc., Cary, NC. URL `https://support.sas.com/documentation/`.

Rabe-Hesketh S, Skrondal A, Pickles A (2004). "Generalized Multilevel Structural Equation Modeling." *Psychometrika*, **69**(2), 167–190. ISSN 1860-0980. `doi:10.1007/BF02295939`. URL `https://doi.org/10.1007/BF02295939`.

Randall J (1989). "The Analysis of Sensory Data by Generalised Linear Model." *Biometrical journal*, **7**, 781–793. `doi:10.1002/bimj.4710310703`.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL `https://www.R-project.org/`.

StataCorp (2017). *Stata 15 Base Reference Manual*. College Station, TX. URL `https://www.stata.com/`.

Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. 4th edition. Springer-Verlag, New York. `doi:10.1007/978-0-387-21706-2`.

Williams R (2010). "Fitting Heterogeneous Choice Models with **oglm**." *Stata Journal*, **10**(4), 540–567(28). URL `http://www.stata-journal.com/article.html?article=st0208`.

Yee TW (2010). "The **VGAM** Package for Categorical Data Analysis." *Journal of Statistical Software*, **32**(10), 1–34. `doi:10.18637/jss.v032.i10`.

A. A regularized Newton-Raphson algorithm with step halving

The regularized NR algorithm is an iterative method that produce a sequence of estimates $\boldsymbol{\psi}^{(0)}, \ldots, \boldsymbol{\psi}^{(i)}, \ldots$, where parenthesized superscripts denote iterations. From the ith estimate, the $(i+1)$'th estimate is given by

$$\boldsymbol{\psi}^{(i+1)} = \boldsymbol{\psi}^{(i)} - c_1 \boldsymbol{h}^{(i)}, \quad \boldsymbol{h}^{(i)} = \tilde{\boldsymbol{H}}(\boldsymbol{\psi}^{(i)}; \boldsymbol{y})^{-1} \boldsymbol{g}(\boldsymbol{\psi}^{(i)}; \boldsymbol{y})$$

where

$$\tilde{\boldsymbol{H}}(\boldsymbol{\psi}^{(i)}; \boldsymbol{y}) = \boldsymbol{H}(\boldsymbol{\psi}^{(i)}; \boldsymbol{y}) + c_2(c_3 + \min(\boldsymbol{e}^{(i)}))\boldsymbol{I},$$

$\boldsymbol{H}(\boldsymbol{\psi}^{(i)}; \boldsymbol{y})$ and $\boldsymbol{g}(\boldsymbol{\psi}^{(i)}; \boldsymbol{y})$ are the Hessian and gradient of the negative log-likelihood function with respect to the parameters evaluated at the current estimates; $\boldsymbol{e}^{(i)}$ is a vector of eigenvalues of $\boldsymbol{H}(\boldsymbol{\psi}^{(i)}; \boldsymbol{y})$, $\boldsymbol{h}^{(i)}$ is the i'th step, c_1 is a scalar parameter which controls the step halving, and c_2, c_3 are scalar parameters which control the regularization of the Hessian.

Regularization is only enforced when the Hessian is not positive definite, so $c_2 = 1$ when $\min(\boldsymbol{e}^{(i)}) < \tau$ and zero otherwise, were τ is an appropriate tolerance. The choice of c_3 is to some extent arbitrary (though required positive) and the algorithm in **ordinal** sets $c_3 = 1$.

Step-halving is enforced when the full step $\boldsymbol{h}^{(i)}$ causes a decrease in the likelihood function in which case c_1 is consecutively halved, $c_1 = \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots$ until the step $c_1 \boldsymbol{h}^{(i)}$ is small enough to cause an increase in the likelihood or until the maximum allowed number of consecutive step-halvings has been reached.

The algorithm in **ordinal** also deals with a couple of numerical issues that may occur. For example, the likelihood function may be sufficiently flat that the change in log-likelihood is smaller than what can be represented in double precision, and so, while the new parameters may be closer to the true ML estimates and be associated with a smaller gradient, it is not possible to measure progress by the change in log-likelihood.

The NR algorithm in **ordinal** has two convergence criteria: (1) an absolute criterion requesting that $\max |\boldsymbol{g}(\boldsymbol{\psi}^{(i)}; \boldsymbol{y})| < \tau_1$ and (2) a relative criterion requesting that $\max |\boldsymbol{h}^{(i)}| < \tau_2$ where the default thresholds are $\tau_1 = \tau_2 = 10^{-6}$.

Here the first criterion attempts to establish closeness of $\boldsymbol{\psi}^{(i)}$ to the true ML estimates in absolute terms; the second criterion is an estimate of relative closeness of to the true ML estimates. Both convergence criteria are needed if both small (e.g., ≈ 0.0001) and large (e.g., ≈ 1000) parameter estimates are to be determined accurately with an appropriate number of correct decimals as well as significant digits.

The NR algorithm in **ordinal** attempts to satisfy the absolute criterion first and will then only attempt to satisfy the relative criterion if it can take the full un-regularized NR step and then only for a maximum of 5 steps.

A.1. Convergence properties and parameter accuracy

Convergence to a well-defined optimum is achieved when the gradient of the negative log-likelihood function with respect to the parameters is small and the Hessian is positive definite i.e., having only positive eigenvalues away from zero. Identifiability problems occur when the likelihood function is flat in directions of one or more parameters (or linear functions of the parameters) while well-defined, i.e., pointy in other directions. It may happen that a parameter is exactly unidentifiable and `clm` is in some cases (including rank-deficient design matrices) able to detect this and exclude the parameter from the optimization procedure. In other cases the likelihood is almost flat in one or more directions. These cases are not uncommon in practice and it is not possible to reduce the parameter space before optimizing the model. To measure the degree of empirical identifiability `clm` reports the condition number of the Hessian which is the ratio of the largest to the smallest eigenvalue. A large condition number of the Hessian does not necessarily mean there is a problem with the model, but it can be. A small condition number of the Hessian, say smaller than about 10^4 or 10^6, on the other hand is a good assurance that a well-defined optimum has been reached.

A key problem for optimization methods is when to stop iterating: when have the parameters that determine the optimum of the function been found with sufficient accuracy? The *method independent error estimate* (Eldén, Wittmeyer-Koch, and Nielsen 2004) provides a way to approximate the error in the parameter estimates. Sufficiently close to the optimum the Newton-Raphson step provides this estimate:

$$|\hat{\boldsymbol{\alpha}}^{(i)} - \boldsymbol{\alpha}^*| \lesssim \boldsymbol{h}^{(i)}, \quad \boldsymbol{h}^{(i)} = \boldsymbol{H}(\boldsymbol{\psi}^{(i)}; \boldsymbol{y})^{-1} \boldsymbol{g}(\boldsymbol{\psi}^{(i)}; \boldsymbol{y})$$

where $\boldsymbol{\alpha}^*$ is the exact (but unknown) value of the ML estimate, $\hat{\boldsymbol{\alpha}}^{(i)}$ is the ML estimator of $\boldsymbol{\alpha}$ at the i'th iteration and $\boldsymbol{h}^{(i)}$ is the full unregularized NR step at the i'th iteration. Since the gradient and Hessian of the negative log-likelihood function with respect to the parameters is already evaluated and part of the model fit at convergence, it is essentially computationally cost-free to approximate the error in the parameter estimates. Based on the error estimate the number of correctly determined decimals and significant digits is determined for each parameter.

The assessment of the number of correctly determined decimals and significant digits is only reliable sufficiently close to the optimum and when the NR algorithm converges without regularization and step-halving. In practice we caution against this assessment if the algorithm did not converge successfully.

Affiliation:

Rune Haubo Bojesen Christensen
Section for Statistics and Data Analysis
Department of Applied Mathematics and Computer Science
DTU Compute
Technical University of Denmark
Richard Petersens Plads
Building 324
DK-2800 Kgs. Lyngby, Denmark
and
Christensen Statistics
Bringetoften 7
DK-3500 Værløse, Denmark
E-mail: `Rune.Haubo@gmail.com`; `Rune@ChristensenStatistics.dk`